UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS

UNIVERSIDADE DO PORTO
FACULDADE DE CIÊNCIAS

Genomics of ecotype formation: interplay between demographic processes and natural selection

"Documento Definitivo"

Doutoramento em Biodiversidade, Genética e Evolução

João Gonçalo Monteiro Carvalho

Tese orientada por:
Professor Vítor Conde Sousa
Doutor Rui Faria
Professor Roger K. Butlin

Documento especialmente elaborado para a obtenção do grau de doutor

2023

UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

UNIVERSIDADE DO PORTO

FACULDADE DE CIÊNCIAS

Genomics of ecotype formation: interplay between demographic processes and natural selection

Doutoramento em Biodiversidade, Genética e Evolução

João Gonçalo Monteiro Carvalho

Tese orientada por:
Professor Vítor Conde Sousa
Doutor Rui Faria
Professor Roger K. Butlin

Júri:
Presidente:

- Sólveig Thorsteinsdóttir, Professora Catedrática e Presidente do Departamento de Biologia Animal da Faculdade de Ciências da Universidade de Lisboa

Vogais:

- Doutor Pierre-Alexandre Gagnaire, *Directeur de Recherche*, *Institute des Sciences de L' Évolution de Montpellier* (ISEM), França
- Doutora Joana Isabel Meier, *Group Leader*, *Tree of Life Programme do Wellcome Sanger Institute*, Reino Unido
- Doutor José Fernando Melo Ferreira, Investigador Principal, Centro de Investigação em Biodiversidade e Recursos Genéticos (CIBIO-InBIO) da Universidade do Porto
- Doutora Inês Regina Lopes Mendonça Fragata Almeida, Professora Auxiliar, Faculdade de Ciências da Universidade de Lisboa
- Doutor Vítor Martins Conde e Sousa, Professor Auxiliar, Faculdade de Ciências da Universidade de Lisboa

Documento especialmente elaborado para a obtenção do grau de doutor

Fundação para a Ciência e a Tecnologia
(Bolsa PD/BD/128350/2017)

2023

Nota prévia: A presente tese apresenta artigos científicos já publicados (capítulos 2, 3 e 4), de acordo com o previsto no nº2 do artigo 25º do Regulamento de Estudos de Pós-Graduação da Universidade de Lisboa, publicado no Diário da República, 2.ª série – N.º 60 – 26 de março de 2018. Uma vez que estes trabalhos foram realizados em colaboração o candidato esclarece que participou integralmente na conceção dos trabalhos, obtenção e análise dos dados, discussão dos resultados, bem como na redação dos manuscritos.

Lisboa, setembro de 2023

João Gonçalo Monteiro Carvalho

“Não posso querer ser nada.
À parte isso, tenho em mim todos os sonhos do mundo”
Álvaro de Campos

Table of Contents

ACKNOWLEDGEMENTS

Well, I guess this is it. It is finally done. These years have been full of challenges, both scientific and personal. And oh, let's not forget that whole global pandemic mess that got tossed into the mix. But, amidst all that, there have been a bunch of new things to experience, loads of learning, and some exciting results. As Charles Dickens said, *it was the best of times, it was the worst of times*.

Luckily, I did not have to go through this all alone because science is truly a team sport. You can't piece together meaningful knowledge without leaning on others. Throughout this thesis, I had the good fortune and privilege of working with amazing mentors, co-authors, colleagues, and institutions. Their big contributions played a massive role in making this research a success and, honestly, kept the whole journey fun. I am eternally thankful to all those who both directly and indirectly accompanied me on this long journey.

First, I need to acknowledge and express my gratitude to *Fundação para a Ciência e a Tecnologia* for funding my PhD. I am also thankful to the *Biodiv PhD programme* for accepting me and introducing me to brilliant young researchers in both Lisbon and Porto. It was a privilege to be part of the organizing committee of the first *BIODIV* meeting. I am particularly grateful to *cE3c* for hosting and receiving me in a centre with some of the most ingenious and hard-working people I am fortunate to call colleagues. I would be remiss if I did not mention the people working in the *cE3c* office, which were always helpful and available for any question.

I owe a massive debt of gratitude to my supervisors for their unwavering support, invaluable guidance, and infinite patience. It feels a bit strange to express this in such a distant way, as if they're strangers. Honestly, I'd much rather speak to each of them directly. First and foremost, my infinite gratitude to you, *Vítor Sousa*, because this work would not have been possible without you. I consider myself incredibly fortunate that you came back to Portugal when you did because I cannot imagine this thesis being supervised by anyone else. This work would be much worse without your guidance through all technical aspects of the methodologies developed here. Our frequent discussions, your profound knowledge of every topic I struggled with (or was just curious about), and your kindness and wholehearted desire to help will forever stay with me and inspire me. And *Rui Faria*, you have been putting up with me since my Master's and you still have so much to teach me even after all these years. Your passion for science and all things about *Littorina* is infectious. You have truly shaped my scientific career. Last but definitely not least, *Roger Butlin*, for agreeing to supervise a random Portuguese guy you hardly knew. Your reputation for kindness and expertise is known by anyone working in the field and I will always cherish the opportunity to proudly mention that *Roger* was one of my PhD supervisors.

The three of you have left an indelible mark on my research and work ethic. Your insights, recommendations, and encouragement were absolutely pivotal, and I will forever hold immense gratitude for that. Honestly, I could not have wished for a better team to guide me.

I would also like to thank *Hernán Morales* for his immense help with the genomic data. He was always available to send me any data I requested, answer any questions and comment on many drafts that went on to become chapters in this thesis. I send my best wishes and sincerely hope we may collaborate again in the future. Additionally, I would like to convey my gratitude to the anonymous reviewers who have generously contributed to enhancing the clarity of our manuscripts.

They say that work is more enjoyable when you work with people you like. I am happy to report that, during my PhD, I conducted a parallel study that confirmed this idea. On those days when work felt like a slog and you lack motivation, I was lucky to be surrounded by kind and funny people. I am thankful to the *evolutionary genetics group*, for including me in the best group of the entire centre and accompanying my work with so much enthusiasm. This group works on so many interesting theoretical questions and systems that our meetings were frequently a hot-spot of new concepts and ideas. Before starting my PhD, I never imagined I would know so much about background selection, associative overdominance, Dobzhansky-Muller incompatibilities, proteins, fishes and ants, just to name a few topics. And even though I have not totally embraced the fish fan club within the group, I will always owe a big thank you to everyone for taking a moment to learn a little about *Littorina*.

Every single person in the group was always up for tossing around new ideas and throwing out questions that pushed my research to the next level. They have acted as test subjects for my untrained presentations more times than I can count, and their insights – both on the science and the design – seriously leveled up my work and how I talk about it. *Vítor* put together a great group and I thrilled to have been there from day one. I wish you all the best and I can not wait to see all the exciting research you will do.

To all my friends, whether I stumbled upon you in the world of academia or outside its walls: to PH, ML, LR, DG, JB, TS, JA and also to CA, DO, BA, FL and MM. I am not always the easiest person to get to know and I am even worse at expressing how much you mean to me. But seriously, from the bottom of my heart, thank you for accepting me as I am. You have been there to cheer me on when I have snagged those small victories, and more importantly, when I have stumbled and hit a rough patch. Hanging out with you all is when I am the happiest and I feel a burst of pride knowing I can call you my friends. Lastly, a special mention to Artur, so he can say he had his first acknowledgement in a PhD thesis even before celebrating his first birthday.

To my family, especially my grandparents, my parents and my sister and brother. This thesis is dedicated to all of you. It stands as a tribute to the sacrifices you have made on my behalf and as a testament to the values and teachings you have instilled in me. Your unwavering support has propelled me through the challenges and triumphs that have led me to this point. We go through life together and that will never change if I can help it. I am beyond grateful for your unconditional love and support. That's all folks.

ABSTRACT

Determining whether phenotypic divergence linked to different ecotypes resulted from a single event or multiple parallel events is a crucial question in evolutionary biology. This thesis aimed to investigate ecotype formation at distinct stages of the speciation continuum and determine whether the evolution of phenotypic divergence in two Littorina species occurred in parallel across multiple geographical locations or if it originated in a single location followed by dispersal and colonization. Genomic data from multiple populations allows distinguishing between these scenarios. One viable approach to assess genome-wide diversity patterns in multiple populations involves next-generation sequencing of pooled samples (Pool-seq). However, Pool-seq introduces specific sources of noise, such as unequal individual contributions, affecting allele frequency estimates. Consequently, the use of Pool-seq data to test alternative scenarios using model-based inference has been hindered by the absence of methods designed to infer demographic history that explicitly model the errors intrinsic to Pool-seq. In this thesis, we compared genetic structure patterns between nonoutlier and outlier loci of *Littorina fabalis* ecotypes, finding evidence compatible with parallel evolution of phenotypic divergence and providing the first characterization of the genetic and morphological divergence in this system. Subsequently, we developed model-based inference methods that leverage Pool-seq data to differentiate between scenarios of ecotype formation and applied those methods to *Littorina saxatilis* data. Overall, this thesis demonstrates that the observed phenotypic divergence in these marine gastropods likely emerged as a consequence of parallel evolution across multiple geographical locations. Moreover, it highlights that this divergence was probably accompanied by gene flow between the diverging ecotypes. The methodologies developed here can be applicable to any taxonomic groups characterized by the presence of ecotypes across a broad geographical range. Therefore, they can streamline investigations into ecotype formation, enhancing our comprehension of the intricate processes that shape biodiversity and the origin of new species.

Keywords: Pool-seq, demographic inference, Approximate Bayesian Computation, *Littorina*, ecotype formation

RESUMO

Compreender se a divergência fenotípica ligada a diferentes ecótipos foi resultado de um único evento ou uma série de eventos paralelos é uma questão fundamental na biologia evolutiva. O principal objetivo desta tese foi estudar o processo de formação de ecótipos em diferentes estágios do contínuo de especiação e avaliar se a evolução da divergência fenotípica observada em duas espécies de Littorina ocorreu em paralelo ou se teve origem em um único local seguido de dispersão e colonização. Dados genómicos de múltiplas populações permitem distinguir entre esses cenários. Uma abordagem viável para avaliar padrões genómicos de diversidade em múltiplas populações envolve o sequenciamento de amostras agrupadas (Pool-seq). No entanto, o Pool-seq introduz fontes específicas de incerteza, como contribuições desiguais de indivíduos, que podem afetar as estimativas de frequência alélica. Consequentemente, o uso de dados de Pool-seq para testar cenários alternativos tem sido dificultado pela ausência de métodos desenhados para inferir a história demográfica que modelem explicitamente os erros inerentes ao Pool-seq. Nesta tese, caracterizámos pela primeira vez a divergência genética e morfológica entre ecótipos de *Littorina fabalis*, revelando evidências compatíveis com evolução paralela de divergência fenotípica. Posteriormente, desenvolvemos métodos de inferência que utilizam dados de Pool-seq para diferenciar entre cenários de formação de ecótipos e aplicamos esses métodos a dados de *Littorina saxatilis*. Esta tese demonstra que a divergência fenotípica observada nestes gastrópodes marinhos provavelmente emergiu como consequência da evolução paralela em várias localidades geográficas. Além disso, destaca que essa divergência foi acompanhada por fluxo génico entre os ecótipos divergentes. As metodologias aqui desenvolvidas são aplicáveis a qualquer grupo taxonómico caracterizado pela presença de ecótipos em uma ampla área geográfica. Portanto, estes métodos podem facilitar investigações sobre a formação de ecótipos, aprimorando a nossa compreensão dos processos complexos que moldam a biodiversidade e a origem de novas espécies.

Palavras-chave: Pool-seq, inferência demográfica, *Littorina*, formação de ecótipos, evolução paralela

RESUMO ALARGADO

A capacidade de distinguir entre verdadeiros exemplos de evolução paralela e aqueles que resultam de outras trajetórias evolutivas é crucial para percebermos como a seleção natural impulsiona a diversificação e, em última instância, leva à especiação. O principal objetivo desta tese foi estudar o processo de formação de ecótipos em duas etapas distintas do contínuo de especiação e avaliar se a evolução da divergência fenotípica observada em duas espécies de *Littorina* resulta de múltiplos eventos paralelos ou de um único evento. Distinguir entre esses dois cenários requer o uso de dados genómicos de várias populações, que são normalmente dispendiosos. O sequenciamento de amostras agrupadas (Pool-seq) é uma alternativa viável para avaliar padrões de diversidade em múltiplas populações. No entanto, fontes específicas de incerteza, tais como a contribuições individuais desiguais, podem afetar as estimativas de frequências alélicas. Consequentemente, o uso do Pool-seq como fonte de polimorfismos genéticos para contrastar modelos demográficos tem sido limitado pela ausência de métodos capazes de simultaneamente inferir a história demográfica e modelar explicitamente os erros inerentes ao Pool-seq.

Nesta dissertação estabelece-se uma primeira caracterização da divergência genética e morfológica entre os ecótipos de *Littorina fabalis*. Posteriormente, são utilizados métodos de inferência desenvolvidos neste trabalho para diferenciar entre cenários de formação de ecótipos em *Littorina saxatilis*, fazendo uso de dados obtidos através de Pool-seq. Este trabalho demonstra que a divergência fenotípica observada nestas duas espécies de gastrópodes marinhos é compatível com evolução paralela em múltiplas localizações geográficas. Além disso, esta tese destaca que essa divergência provavelmente foi acompanhada por fluxo génico entre ecótipos. Após a introdução do tema e objetivos da tese no **capítulo 1**, o trabalho prático desenvolvido nesta dissertação inicia-se com o estudo dos ecótipos de *L. fabalis*.

Capítulo 2 - Divergência genética e morfológica entre os ecótipos de *Littorina fabalis* no Norte da Europa.

Este capítulo da tese concentra-se nas diferenças genéticas e morfológicas entre os ecótipos de *L. fabalis*, contribuindo para a compreensão da adaptação local em espécies intertidais. O **capítulo 2** representa o primeiro esforço para identificar variações genéticas associadas a ecótipos divergentes nesta espécie. As análises da morfologia da concha revelam diferenças consistentes entre os ecótipos, principalmente ao nível do tamanho e com efeitos diretamente relacionados com o habitat. A análise genética revela diferentes padrões de diferenciação entre os ecótipos, com um subconjunto de loci não-neutrais que resistem ao efeito do fluxo génico que erode a diferenciação nas restantes partes do genoma. Assim, os resultados do **capítulo 2** sugerem que a seleção natu-

ral desempenha um papel significativo na divergência dos ecótipos de *L. fabalis*. Neste capítulo, examina-se também a extensão de partilha de loci não-neutrais entre populações de *L. fabalis* do norte da Europa. O nível de partilha sustenta a hipótese de evolução em conjunto, ou seja, da partilha de alelos adaptativos entre populações através de fluxo génico. No entanto, não foi possível descartar a hipótese de que este nível de partilha seja resultado de polimorfismo ancestral. Os resultados obtidos sugerem que a formação de ecótipos em *L. fabalis* é compatível com evolução paralela. No entanto, é necessário modelar diferentes cenários de formação de ecótipos e usar dados genómicos mais completos para confirmar esta hipótese. Este trabalho abre as portas a estudos comparativos entre *L. fabalis* e *L. saxatilis*, os quais podem ser uma importante fonte de conhecimentos sobre divergência em transições ambientais e evolução paralela.

Capítulo 3 - poolHelper: um pacote R para auxiliar o planeamento de estudos de Pool-Seq.
Este capítulo apresenta uma abordagem inovadora para simular dados de Pool-seq e integra-a num pacote de software desenvolvido na linguagem de programação R chamado *poolHelper*. O **capítulo 3** demonstra que a seleção do desenho experimental mais adequado para experiências que envolvam a sequenciação de amostras em grupo (*pool*) é um processo complexo, visto que diferentes combinações de cobertura média e número de indivíduos incluídos no *pool* podem levar a erros semelhantes nas frequências alélicas. Tal realça a necessidade de realizar estudos preliminares baseados em simulações, de modo a determinar esquemas de amostragem eficientes que produzam frequências alélicas precisas com o menor esforço de sequenciação. O pacote *poolHelper* aborda essa necessidade ao oferecer funções para simular dados de Pool-seq de um modo personalizado. O software desenvolvido no **capítulo 3** encontra-se disponível para qualquer investigador interessado em simular dados de Pool-seq, permitindo ainda calcular o erro nas frequências alélicas das amostras e na heterozigotia esperada, utilizando diferentes desenhos experimentais e aplicando diversos filtros. É importante destacar que o trabalho apresentado neste capítulo tem em consideração fontes de erro associadas a Pool-seq que foram negligenciadas por trabalhos anteriores, o que leva a uma simulação mais abrangente e precisa de dados de Pool-seq. Para além disso, o **capítulo 3** contemplou também a elaboração de um manual e de um guia de utilização detalhado.

Capítulo 4 - Integração de incertezas associadas ao Pool-seq no processo de inferência demográfica.
O **capítulo 4** desta tese integra o trabalho desenvolvido no capítulo anterior numa estrutura de *Approximate Bayesian Computation* (ABC). Esta abordagem é utilizada para analisar dados de sequenciação em grupo (Pool-seq), permitindo a inferência conjunta de fontes de erro associadas a Pool-seq e parâmetros demográficos. O método apresentado neste capítulo é computacionalmente eficiente, pois recorre a simulações de subconjuntos de loci, estima parâmetros relativos e usa estatísticas sumário relativas. O método está disponível na forma de um pacote desenvolvido em R

que permite contrastar modelos e inferir parâmetros da história demográfica usando dados de Pool-seq. Os resultados do **capítulo 4** demonstram que a precisão das estimativas obtidas é comparável com outros métodos e que ignorar os erros associados ao Pool-seq leva a uma maior imprecisão das estimativas obtidas. Aplicado aos ecótipos de *L. saxatilis* amostrados em dois locais na Suécia, este método revelou que os ecótipos tiveram uma origem única, há cerca de 15.000 anos, seguida por uma divisão populacional nos dois locais amostrados que ocorreu aproximadamente 1.000 gerações atrás. Embora os modelos examinados sejam relativamente simples, os resultados do **capítulo 4** demonstram que é possível combinar eficazmente dados obtidos através de Pool-seq com ABC para investigar cenários de formação de ecótipos. O pacote resultante do trabalho desenvolvido neste capítulo facilita a exploração de processos demográficos complexos e inclui funções para calcular a precisão das estimativas, avaliar qual o modelo que mais se ajusta aos dados e realizar estudos de simulação. Tal como anteriormente, este capítulo englobou a elaboração de um manual e de um guia de utilização abrangente.

Capítulo 5 - Evolução paralela de adaptação local em *Littorina saxatilis* inferida atráves de dados de Pool-seq.

Neste capítulo examinou-se a formação de ecótipos em *L. saxatilis* utilizando dados obtidos por meio de sequenciação em grupo (Pool-seq). Usando a abordagem desenvolvida nos capítulos anteriores, no **capítulo 5** foram explorados modelos demográficos complexos e avaliou-se a precisão da inferência de parâmetros e da seleção de modelos. Os resultados indicaram que as estimativas obtidas são precisas para a maioria dos parâmetros demográficos, embora com maiores incertezas em parâmetros relacionados com populações ancestrais. Os resultados do **capítulo 5** confirmaram a utilidade de combinar dados obtidos através de Pool-seq e *Approximate Bayesian Computation* para distinguir cenários de formação de ecótipos, mesmo usando modelos mais complexos. Utilizando dados de Pool-seq de ecótipos de *L. saxatilis* amostrados na Suécia e na Espanha, este capítulo suporta um cenário de origem paralela dos ecótipos, sem um período de isolamento e após a colonização das duas regiões distintas. As estimativas de divergência obtidas apontam para uma separação recente dos ecótipos: cerca de 15.000 e 57.000 anos na Suécia e Espanha, respetivamente. O processo de divergência ocorreu apesar da existência de fluxo génico entre os ecótipos, já que os resultados apontam para uma alta taxa de migração entre os ecótipos. Estes resultados reforçam o papel que a seleção divergente teve na origem dos ecótipos de *L. saxatilis*, potencialmente promovendo a origem e manutenção de rearranjos cromossómicos. No entanto, os modelos aqui considerados não contemplaram a existência de períodos de isolamento (alopatria), não sendo por isso possível descartar que tais períodos tenham tido influência na história evolutiva destes ecótipos. A abordagem apresentada neste capítulo deve, no futuro, ser aplicada a mais populações de *L. saxatilis*, incluindo locais adicionais de amostragem na Suécia, Espanha e Reino Unido.

Em resumo, esta tese apresenta evidências que sustentam a ideia de que as diferenças fenotípicas observadas nestas duas espécies de gastrópodes marinhos do género *Littorina* são, provavelmente, o resultado de eventos de evolução paralela. O trabalho aqui desenvolvido resultou ainda em dois pacotes R que permitem a simulação de dados de Pool-seq e a sua integração em processos de inferência demográfica. É de esperar que este trabalho incentive o uso de dados de Pool-seq em estudos focados em questões relacionadas com evolução paralela e a formação de ecótipos. Além disso, esta tese representa um exemplo de investigação em diferentes estágios do contínuo de especiação e em espécies diferentes, enfatizando o valor de estudos comparativos para aprimorar a nossa compreensão dos processos evolutivos. Estes estudos permitem-nos identificar padrões intra- e interespecíficos, contribuindo assim para o nosso conhecimento sobre biodiversidade e origem das espécies.

CHAPTER 1

General Introduction

GENERAL INTRODUCTION

1.1 THE SPECIATION PROCESS

Speciation, which ultimately underpins the abundance of species and the intricate biodiversity that exists today, has long been regarded as a central process in evolutionary biology. This process usually involves the gradual divergence of one ancestral population into two or more distinct populations that eventually become reproductively isolated from one another, giving rise to new species. Speciation begins with the accumulation of genetic differences between populations. These differences can arise and accumulate due to various mechanisms such as new mutations, genetic drift and natural selection. Over time, as these genetic differences accumulate, populations may start to develop reproductive barriers that prevent them from interbreeding or producing viable offspring. These barriers can be broadly categorized into two groups: pre-zygotic barriers, which come into play prior to mating, and post-zygotic barriers, which manifest after mating (Mayr, 1963). The gradual buildup of these barriers results in increasing reproductive isolation, which can be defined as a quantitative measure of the effect that genetic differences between populations have on gene flow (Westram, Stankowski, Surendranadh, & Barton, 2022).

Research in this field has primarily been focused on the causes and the consequences of speciation (Butlin et al., 2012; Seehausen et al., 2014). Although the ultimate consequence is reproductive isolation, the causes that contribute to this outcome are diverse. This distinction between causes and consequences reflects historical contingencies. Charles Darwin's groundbreaking perspective on speciation, outlined in his seminal work *"On the origin of species by means of natural selection"* published in 1859, emphasized the gradual evolution of biological distinctions and the driving forces that guide their divergence, postulating that new species could be the product of natural selection occuring over very long periods of time. In contrast, Ernst Mayr's influential viewpoint presented in 1942, highlighted the significance of the evolution of reproductive barriers between different species. For many years, the primary research focus in the field of speciation has been the perspective put forth by Mayr, which places significant emphasis on geographic separation and isolation and argues that reproductive isolation is based on pre- or post-zygotic barriers to gene flow, without necessarily invoking the action of natural selection (Coyne & Orr, 2004). However, in recent years, there has been a notable shift towards investigating the causes of divergence that propel speciation. Consequently, contemporary research often endeavors to comprehend how divergent evolution transforms populations into reproductively and genetically isolated groups.

This recent research trend has shown that divergent natural selection, driving local adaptation in disparate environments, may frequently trigger the development of reproductive isolation and ultimately lead to speciation (Nosil, 2012; Schluter, 2009). However, disentangling and evaluating the response to selection in habitats linked by dispersal presents a challenge, as gene flow and recombination may hinder both local adaptation and the subsequent reinforcement of reproductive isolation (Felsenstein, 1981; Smadja & Butlin, 2011). Although the conventional categorization of speciation processes into allopatric, parapatric, and sympatric types has become somewhat outdated (Butlin, Galindo, & Grahame, 2008), it is still important to consider the spatial context of speciation and the degree of gene flow during different stages of the process. These factors play a crucial role in determining whether and how quickly reproductive isolation will develop (Butlin et al., 2012) and may also influence the genomic architecture of speciation. Therefore, to understand local adaptation and speciation, it is necessary to make inferences about the demographic history of populations and their biogeographical context, which may have changed significantly during the process of speciation (Abbott et al., 2013; Hewitt, 2011). For instance, alternating cycles of isolation by geographical barriers and secondary contact may facilitate speciation (Bierne, Welch, Loire, Bonhomme, & David, 2011) and/or certain spatial arrangements of habitats may allow for local adaptation more readily than others (Gavrilets, Vose, Barluenga, Salzburger, & Meyer, 2007). These complexities underscore the interconnectedness of ecological factors and geographical context in shaping the speciation process.

1.2 SPECIATION CONTINUUM

Despite being the focus of decades of research, the speciation process is still puzzling and marked by a disconnect between the causes and consequences of the speciation process. This is primarily because the process usually occurs over an evolutionary time frame that offers multiple temporal perspectives from which to observe it, making it challenging to understand. An effective way of combining both the causes and consequences of the speciation process in the same framework is to visualize the process as a "speciation continuum," representing a sequence of genetically-driven alterations that occur as two lineages diverge on the path towards complete reproductive isolation (Figure 1.1). The premise of the speciation continuum is straightforward, suggesting that we can imagine a continuum between a panmitic population and two completely reproductively isolated species. Thus, differences in the degree of reproductive isolation reflect the position of a given pair of populations/species along the speciation continuum. We can consider each pair of populations as a snapshot of a specific point along the continuum of reproductive isolation and arrange several pairs in a sequence, following the principle that "the present is the key to the past" (Lyell, 1830). By

comparing different pairs of populations along the speciation continuum the aim is to comprehend how the speciation process unfolds from a panmitic population to two species. Yet, it should be noted that the evolutionary path of a pair of populations is not unidirectional towards complete isolation, and that this path can move or jump back and forth along the speciation continuum. Nevertheless, the usefulness of the speciation continuum concept has prompted many researchers to structure their research questions within the framework it provides (e.g. Hendry, Bolnick, Berner, & Peichel, 2009; Ravinet et al., 2018).

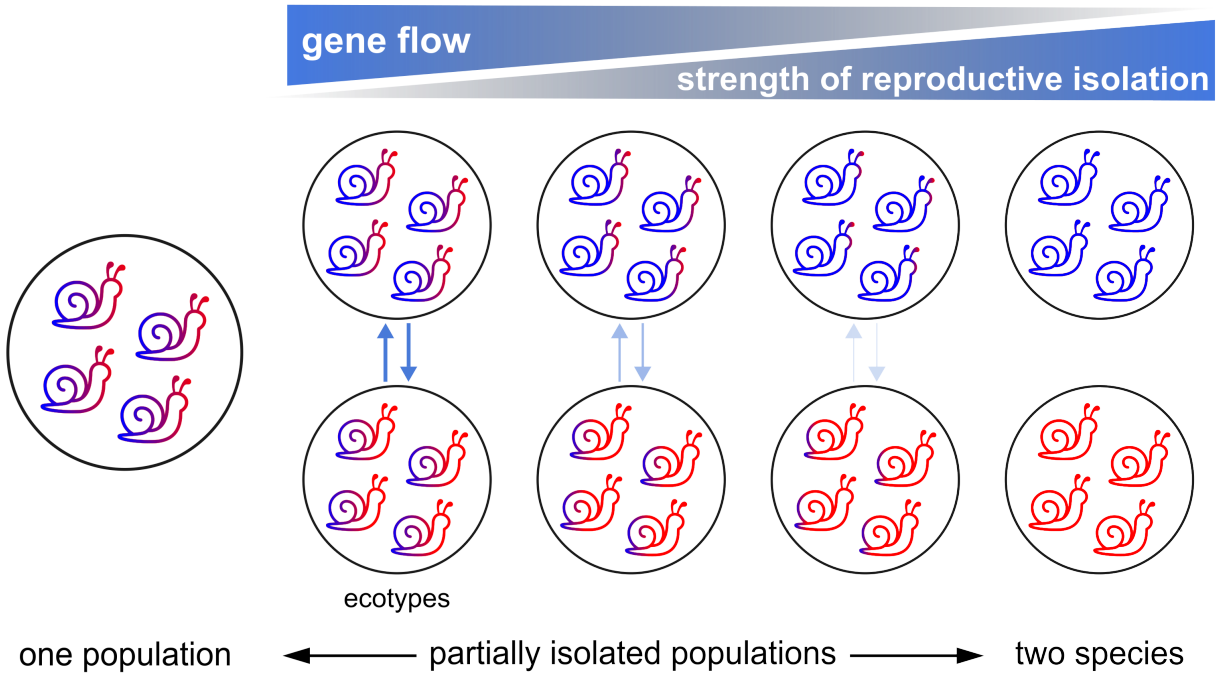


Figure 1.1: Graphical illustration of the speciation continuum. The speciation continuum can be understood as a gradient where one end is represented by a single population and the other end by two completely isolated species. In-between these two defined stages, all the other states in the gradient represent partially reproductively isolated populations. Ecotype formation is normally considered to be an incipient stage of the speciation process, closer to the end represented by a single population.

However, there is a rough division in terms of where research is placed along this continuum. On one hand, studies that investigate speciation with gene flow tend to focus on populations with low levels of reproductive isolation, which are in the early stages of the process. These populations are often referred to as ”incipient species” and the studies typically investigate the role of divergent selection on adaptive traits (e.g., Marques et al., 2016; Turner, Hahn, & Nuzhdin, 2005). On the other hand, there is an extensive body of research that focuses more on the end stages of the speciation process, particularly the genetic mechanisms behind hybrid sterility and inviability (e.g., Phadnis et al., 2015; Presgraves, 2002). These studies require the evolution of intrinsic postzy-

gotic isolation, and therefore, speciation is often viewed as a past event or a process that led to the development of irreversible reproductive isolation.

Despite these differences, the idea of a speciation continuum has allowed these two seemingly contrasting types of research to coexist within the same conceptual framework (Stankowski & Ravinet, 2021). As research on speciation continues to move forward, it is becoming increasingly evident that conventional methods and concepts are being replaced by new questions and research areas. This is exemplified by the shift from studying a limited number of model systems to a more comprehensive approach, which allows researchers to gain novel insights into the evolutionary processes that shape diversity patterns across various levels of biological hierarchy. Thus, data from multiple points along the speciation continuum helps us predict how genomic changes underlying reproductive isolation accumulate and interact with genomic changes related with phenotypic evolution (Roux et al., 2016; Stankowski & Ravinet, 2021). More precisely, comparisons between closely related species that may potentially represent different stages of the speciation continuum, particularly when those species are faced with similar selective pressures within comparable environmental gradients, have the potential to be particularly informative.

1.3 PARALLEL EVOLUTION OF REPRODUCTIVE ISOLATION

One of the potential initial stages in the process of speciation is the formation of ecotypes (Lowry, 2012; Turesson, 1922; Via, 2009, Figure 1.1). Ecotypes can be defined as distinct populations that exhibit adaptive divergence associated to contrasting environments (Funk, 2012). The process of divergent natural selection, which results in the formation of ecotypes, can often continue to play a significant role in enhancing reproductive isolation after ecotype formation (Nosil, 2012; Schluter, 2009). Moreover, ecotypes are frequently found in multiple geographical locations. These repeated instances of reproductive isolation resulting from adaptation to distinct pairs of habitats provide strong evidence for the role of natural selection in speciation (Lenormand, Roze, & Rousset, 2009; Schluter & Nagel, 1995) and allow powerful tests of the underlying evolutionary processes (Jones et al., 2012). Instances of parallel evolution have been reported in animals, such as sticklebacks (e.g. Colosimo et al., 2005), stick insects (e.g. Soria-Carrasco et al., 2014), cichlid fishes (e.g. Weber, Rajkov, Smailus, Egger, & Salzburger, 2021) and *Heliconius* butterflies (e.g. Lewis et al., 2019), as well as in plants such as the wildflower *Senecio lautus* (e.g. Roda, Walter, Nipper, & Ortiz-Barrientos, 2017), *Arabidopsis* (e.g. Bohutínská et al., 2021) and rice (e.g. Cai et al., 2019).

It is important to note that this pattern of parallel local adaptation, in the context of ongoing gene flow, could result from very different historical sequences of events (Johannesson et al., 2010). Indeed, several scenarios can explain this pattern of parallel local adaptation. One important distinction is whether the initial adaptive differentiation, leading to the divergence of ecotypes, occurred only once, possibly in isolation, with subsequent colonization of the respective habitats by the disparate adapted forms in a scenario known as single origin. Alternatively, evolutionary divergence could have occurred repeatedly in multiple locations, with or without spatial isolation, a scenario termed parallel origin (Figure 1.2).

Although a scenario of parallel origin for locally adapted ecotypes is often invoked (e.g. Arciniega et al., 2016; James et al., 2021), explicit tests between alternative hypotheses are rare. An important factor to consider is whether these repeated events of phenotypic divergence involve the same or different *de novo* mutations in the same or different genomic regions, or a shared polymorphism due to standing genetic variation or gene flow. The extent to which we can differentiate between these alternative hypotheses depends on our ability to effectively characterize the genetic variation that underlies local adaptation and ecotype divergence in order to evaluate which scenario is more probable (Elmer & Meyer, 2011; Faria et al., 2014; Johannesson et al., 2010). This, in turn, requires information from putatively neutral genetic markers to establish the demographic history of the populations and the analysis of loci underlying adaptation whose history may be significantly different from the neutral markers (Colosimo et al., 2005; Jones et al., 2012; Sousa, Carneiro, Ferrand, & Hey, 2013).

The study of parallel phenotypic divergence across microhabitats can help us understand if adaptation usually follows the same genetic pathways, as well as the relative contributions of ancestral polymorphism and/or gene flow. By employing genetic data and coalescent-modeling techniques, we can reconstruct the demographic history of populations, discerning whether divergence or colonization occurred first, thereby distinguishing between these two different scenarios. In fact, it is now possible to use individual-based whole-genome data in combination with several modelling approaches (e.g. Excoffier et al., 2021) to reconstruct the demographic history of species. These methods could be extended to readily differentiate between the single and parallel origin scenarios, especially in situations where recent gene flow adds complexity to this distinction.

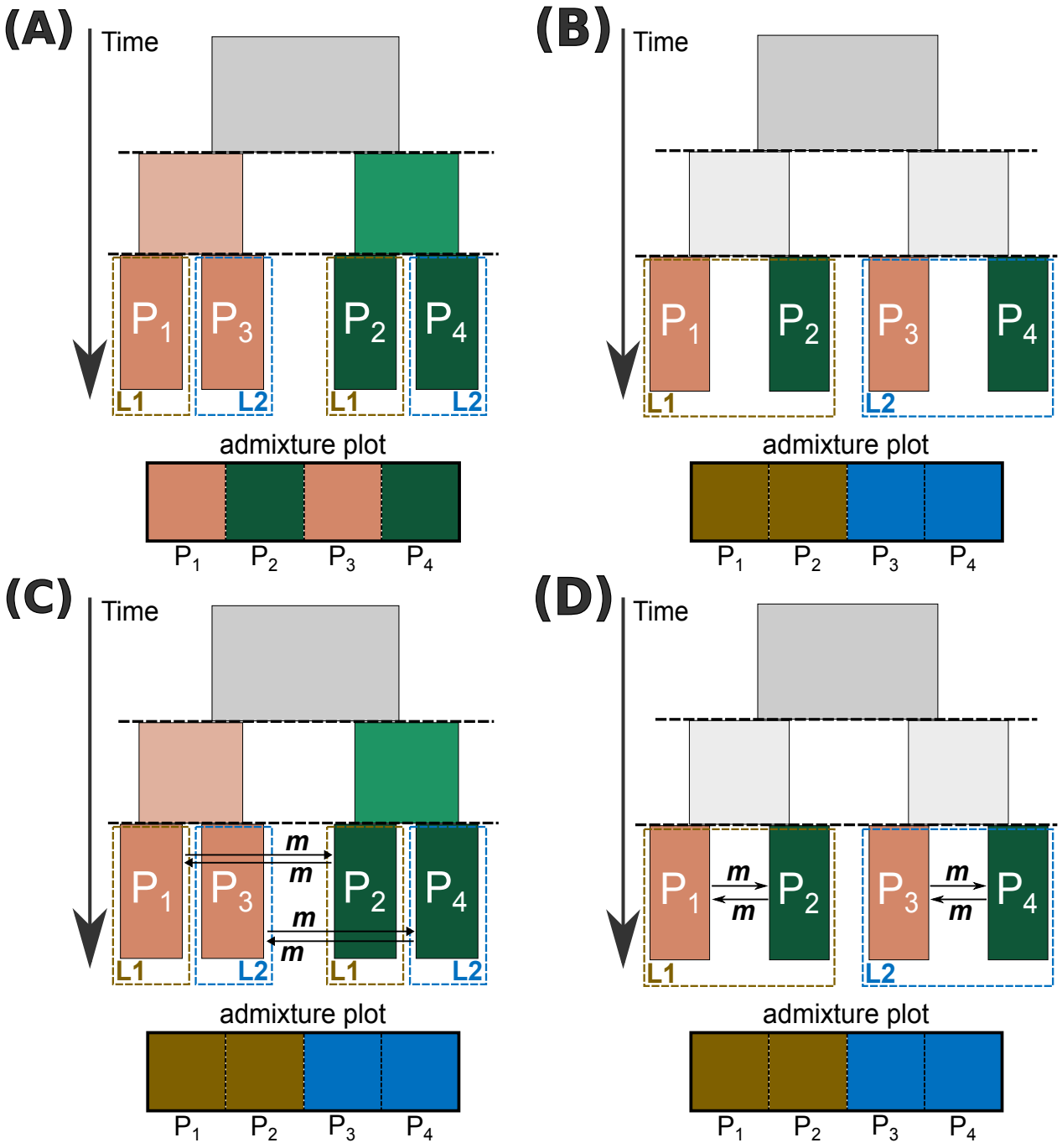


Figure 1.2: Demographic models for the single (A) and parallel (B) scenarios of ecotype formation. Green and orange colours indicate the two different ecotypes. Different locations are represented by the boxes labeled L1 and L2. The schematic of an admixture plot on the bottom of each population tree indicates the expected population structure for that scenario. Present-day populations are indicated by P_1 to P_4, both in the population trees and in the admixture plots, and m indicates the probability per generation that an individual migrates from one population to the population of the other ecotype. Note that without migration, the ecotypes should cluster by ecotype in the single origin scenario (A) and by location in the parallel origin scenario (B). When gene flow between the divergent ecotypes exists (indicated by the black arrows and m), both the single (C) and the parallel (D) origin scenario can led to clustering of the individuals by location.

1.4 DEMOGRAPHIC INFERENCE

To understand whether and how rapidly reproductive isolation may evolve, it is crucial to assess the demographic history of populations (Butlin et al., 2012). One method of inferring past demographic events is to use a combination of population genomic data and coalescent-modelling approaches. This enables the comparison of alternative models and the estimation of relevant demographic parameters. In recent years, advances in high performance computing have promoted the spread of methods that are capable of simulating genealogies under complex demographic models (Fu & Li, 1997; Tavaré, Balding, Griffiths, & Donnelly, 1997). Model-based inference methods, such as Approximate Bayesian Computation (ABC; Beaumont, Zhang, & Balding, 2002), allow for explicit and joint consideration of evolutionary processes and sampling effects (Bertorelle, Benazzo, & Mona, 2010; Csilléry, Blum, Gaggiotti, & François, 2010).

Approximate Bayesian computation is a flexible and widely-used method for model selection and parameter inference because it circumvents the calculation of an exact probability function, allowing the evaluation of complex models (Bertorelle et al., 2010; Li et al., 2012). ABC uses summary statistics (such as dxy, and F_{ST}, etc) to replace the original data, and then employs simulations to select models and estimate parameters. The basic process requires the simulation of thousands or millions of data sets under the demographic models of interest and using different parameter values, randomly sampled from a prior distribution. Summary statistics are calculated from the simulated genealogies and compared with the statistics obtained from real data. The simplest ABC algorithm follows a rejection approach (Tavaré et al., 1997). In this method, parameter values and/or models that are randomly sampled from the prior distribution are accepted if the distance between the simulated summary statistics and the observed summary statistics falls below a specified threshold (i.e. tolerance), and rejected otherwise. The acceptance of parameter values based on this distance threshold provides a sample of independent points from the posterior distribution.

Due to its flexibility, ABC has been widely used in several fields, including ecology (Pontarp, Brännström, & Petchey, 2019; Zhang, Dennis, Landers, Bell, & Perry, 2017), systems biology (Liepe et al., 2014), and population genetics (Cooke & Nakagome, 2018; Rougemont & Bernatchez, 2018). Various software implementations of ABC are available for use (Boitard, Rodríguez, Jay, Mona, & Austerlitz, 2016; Cornuet et al., 2014; Huang, Takebayashi, Qi, & Hickerson, 2011; Wegmann, Leuenberger, Neuenschwander, & Excoffier, 2010). However, implementing ABC for whole-genome data is challenging (Smith & Flaxman, 2020), due to the significant computational burden and difficulties in simulating recombination and mutation rate variation along the genome (Jay, Boitard, & Austerlitz, 2019).

The development of advanced methods like ABC, combined with the rise of new sequencing techniques has resulted in the simultaneous occurrence of algorithms capable of simulating genomic data under complex evolutionary scenarios, and an abundance of genomic data to compare against these simulations. Thus, researchers can now estimate posterior distributions for relevant parameters, probabilistically compare alternative models, and assess the accuracy of their results. Indeed, numerous studies have shown that genetic data and coalescent-modelling can be used to reconstruct the demographic history of populations (Beichman, Huerta-Sanchez, & Lohmueller, 2018; Sousa et al., 2013). Nevertheless, it is important to acknowledge that the outcome of the demographic modelling approach is dependent on the composition of the models being compared. Furthermore, even the best model is unlikely to be a full representation of the real demographic history, as models must strike a delicate balance between incorporating enough real complexity to draw meaningful inferences, while remaining simple enough to be manageable (Hickerson, 2014).

Additionally, the reconstruction of complex gene-flow histories is expected to be difficult (Strasburg & Rieseberg, 2013) due to a myriad of factors. In real-world scenarios, gene flow is rarely a linear process between two populations. Instead, it often involves multiple interactions among various populations, leading to intricate networks of genetic exchange. Disentangling the extent of gene flow between any two given populations in this network is challenging. Moreover, random events such as population bottlenecks, founder effects, and other environmental fluctuations, can affect signals of gene flow (Hey & Machado, 2003; Momigliano, Florin, & Merilä, 2021; Sousa & Hey, 2013). It is also difficult to distinguish between secondary contact and regular gene flow, particularly when hybrid offspring can backcross with parental populations (Payseur & Rieseberg, 2016). Finally, the gradual accumulation of genetic changes in populations over time adds another layer of complexity, making it tough to determine the timing and direction of historical gene flow events (Galtier, 2023).

1.5 NEXT GENERATION SEQUENCING OF POOLED SAMPLES

Population genomics data can be used to infer and establish the demographic history that has shaped natural populations. The advent of Next Generation Sequencing (NGS) has revolutionized the field of population genomics, enabling researchers to reconstruct evolutionary histories using thousands of Single Nucleotide Polymorphisms (SNPs) across the genome (Ellegren, 2014).

However, for some species (e.g., small organisms), generating and sequencing individual libraries can be problematic. Additionally, studying parallel phenotypic divergence across several locations requires population-level genomic data from multiple populations, which might quickly become expensive. In such cases, pooling DNA from multiple individuals to create a single library that can be sequenced (Pool-seq) offers an effective alternative. This reduces the amount of DNA required per individual, which minimizes the laboratory work by reducing the number of library preparations needed. As a result, costs are lowered while still allowing the comparison of populations on a genomic scale (Schlötterer, Tobler, Kofler, & Nolte, 2014). Nonetheless, non-equimolar DNA concentrations and stochastic variations in amplification or sequencing efficiency between individuals in the pool can result in loss of accuracy of allele frequency estimates (Figure 1.3; Anderson, Skaug, & Barshis, 2014; Cutler & Jensen, 2010; Ellegren, 2014).

Furthermore, DNA from multiple individuals can be extracted in batches, and combining these batches into a single pool for library preparation and sequencing (Morales et al., 2019; Ross, Endersby-Harshman, & Hoffmann, 2019) can lead to unequal representation due to variation in extraction efficiency and/or non-equimolar concentrations of DNA between batches. Despite this, theoretical and empirical research has shown that, for an equal sequencing effort, Pool-seq allows the analysis of a greater number of individuals, resulting in similar or even more precise allele frequency estimates (Futschik & Schlötterer, 2010; Gautier et al., 2013).

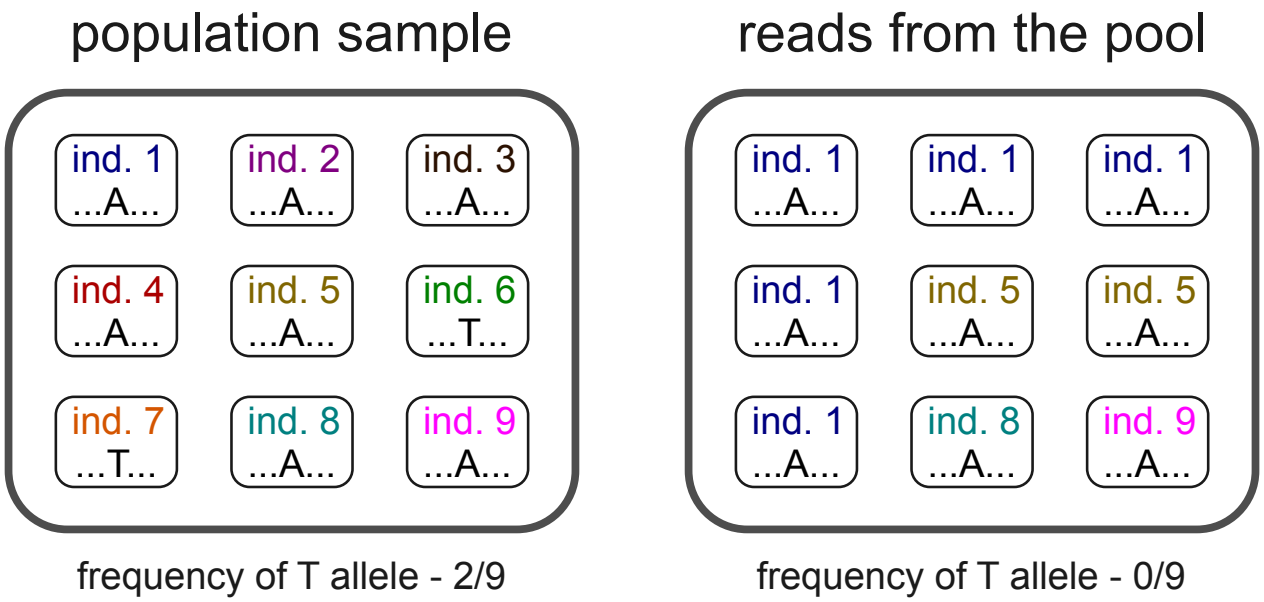


Figure 1.3: Graphical illustration of unequal individual representation associated with Pool-seq data. In this example, a population sample consisting of 9 individuals is pooled and sequenced. Due to non-equimolar DNA concentrations and stochastic variations in amplification or sequencing efficiency, some individuals (e.g. individual 1) can contribute more than one read, while other individuals (e.g. individual 7) might not contribute any reads at all. Note that this example leads to the loss of the T allele present in the sample.

Although empirical studies have demonstrated that individual-based sequencing is more suited than Pool-seq to detect fine-scale population substructure (e.g., hybrids and migrants), both methods are suitable for inferring population genetic structure (Chen et al., 2022; Dorant et al., 2019). In fact, when a large number of samples are available, Pool-seq data produces more accurate estimates of effective population sizes and divergence or admixture time events (Collin et al., 2021). As a result, Pool-seq has been utilized in various studies, ranging from population genomic analysis (Begun et al., 2007; Ferretti, Ramos-Onsins, & Pérez-Enciso, 2013; Rubin et al., 2012) to experimental evolution (Parts et al., 2011; Turner, Stewart, Fields, Rice, & Tarone, 2011; Zhou et al., 2011) and human genetics applications (Calvo et al., 2010; Lieberman et al., 2014; Prescott et al., 2015). Nonetheless, the use of Pool-seq to conduct demographic history inference has been impeded by a lack of tools that explicitly model this type of data (but see Collin et al., 2021; Taus, Futschik, & Schlötterer, 2017). Thus, although Pool-seq data could be a valuable source of neutral genetic polymorphism for establishing the demographic history of populations undergoing parallel phenotypic divergence, currently there is a lack of methods to model and simulate this type of data.

1.6 INTERTIDAL ZONE AS A NATURAL LABORATORY TO STUDY LOCAL ADAPTATION

One striking example of the new wave of speciation research and the shift towards new systems is the work performed on marine intertidal species. These organisms inhabit the transition zone between land and sea, which constitutes an abrupt environmental transition (Little & Kitching, 1996; Raffaelli & Hawkins, 1996; Tomanek & Helmuth, 2002). Inhabiting the intertidal zone requires organisms to cope with the physical stresses exerted by wave action (Le Pennec et al., 2017), which is a critical selective factor that influences the biodiversity of habitats exposed to or shielded from waves (Helmuth & Denny, 2003). Exposure to thermal extremes during low tide periods is another significant challenge (Denny & Wethey, 2001; Helmuth & Hofmann, 2001). Moreover, intertidal organisms must deal with selective pressures posed by predation (Paine & Fenchel, 1994) and competition for space (Connell, 1961). The patchwork combination of diverse environmental conditions and pressures across the tidal range leads to vertical and horizontal zonation patterns in species diversity and intraspecific phenotypic variation (Connell, 1972; Little & Kitching, 1996; Raffaelli & Hawkins, 1996). Due to the highly spatially condensed and physically challenging nature of the intertidal zone, it has long been recognized as an ideal natural laboratory for studying the interactions between physical and biological factors in determining the distribution and abundance of species in nature (Paine, 1966; Paine & Fenchel, 1994).

More recently, there has been a shift towards understanding the mechanisms and processes that affect species abundance and distribution, including local adaptation. This has led to the intertidal zone gaining recognition as an excellent setting to address several pertinent questions in the field of evolutionary biology (Sanford & Kelly, 2011). Intertidal taxa with limited dispersal, where divergent selection can be strong enough to oppose connectivity between populations and increase divergence, are particularly well-suited for studying local adaptation and the evolution of reproductive isolation (Sanford & Kelly, 2011; Smadja & Butlin, 2011). Among these taxa, those exhibiting ecotypic differentiation - where different phenotypes are associated with distinct microhabitats - are particularly informative for understanding how natural selection drives biological diversification (Faria, Johannesson, & Stankowski, 2021; Kess, Galindo, & Boulding, 2018). Furthermore, instances of parallel evolution of ecotypes across similar environmental gradients in multiple locations along a species' distribution are considered strong evidence for the role of natural selection in driving phenotypic divergence, as it is unlikely that stochastic processes alone would lead to the same evolutionary outcome (Johannesson, 2001; Nosil, 2012; Schluter, 2000).

As mentioned before, the intertidal zone provides a natural laboratory for investigating local adaptation, and many species living in this habitat exhibit phenotypic divergence (Kess & Boulding, 2019; Maltseva et al., 2021; Thia et al., 2021). Studying such cases of phenotypic divergence across a broad geographical range (comparing multiple intertidal zones) contributes to improve our understanding of how natural selection drives the process of divergence (Schluter & Nagel, 1995). Therefore, by combining tools for modelling and simulating Pool-seq data with data obtained from multiple intertidal populations, we can gain insights into the mechanisms underlying local adaptation and parallel phenotypic divergence. Among the intertidal species, gastropods of the genus *Littorina* constitute a unique system for studying parallel phenotypic divergence resulting from adaptation to divergent habitats in the intertidal.

1.7 THE *LITTORINA* GENUS

Parallel ecotype divergence across multiple intertidal zones is well documented in the marine gastropod, *Littorina saxatilis*, commonly known as rough periwinkle (Galindo & Grahame, 2014; Reid, 1996). This rocky-shore gastropod from the North Atlantic is notable for having a low lifetime dispersal and bearing live young (Reid, 1996). Two main ecotypes have been described: a large, thick-shelled ecotype that lives in sheltered microhabitats and faces significant crab predation (Crab ecotype); and a small, thin-shelled one that inhabits exposed microhabitats and faces heavy surf (Wave ecotype) (Figure 1.4; Johannesson, 2016; reviewed in Johannesson et al., 2010).

These ecotypes, which are separated by only a few hundred meters in Spain, the United Kingdom (UK), and Sweden, have been extensively studied, albeit in a largely independent fashion, in these three European regions. Research has uncovered evidence for assortative mating, so that each ecotype mates preferentially with similar individuals (Conde-Padín, Cruz, Hollander, & Rolan-Alvarez, 2008), as well as a genome-wide partial barrier to gene flow, with evidence for divergent selection on some loci (Grahame, Wilding, & Butlin, 2006).

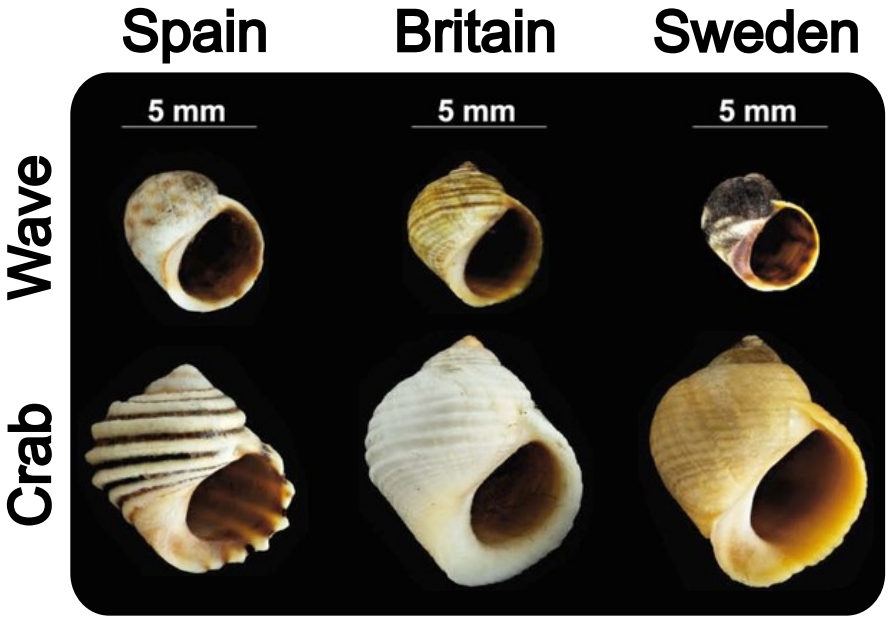


Figure 1.4: Typical shells of the Crab and Wave ecotype of *L. saxatilis* from three European regions: Spain, Britain and Sweden. Figure adapted from Butlin et al. (2014).

Genetic studies using amplified fragment length polymorphisms (AFLPs) allowed the initial identification of highly differentiated loci that could be associated with phenotypic divergence between ecotypes (i.e. outlier loci), as well as the first estimates of the proportion of outliers shared at various geographic scales (Galindo, Martínez-Fernández, Rodríguez-Ramilo, & Rolán-Alvarez, 2013; Galindo, Morán, & Rolán-Alvarez, 2009; Hollander, Galindo, & Butlin, 2015; Wilding, Butlin, & Grahame, 2001). However, those outliers were defined in terms of unusually high differentiation (measured by F_{ST}), which might be explained by processes other than divergent selection (Cruickshank & Hahn, 2014; Ravinet et al., 2017). It has been proposed that ecotype differentiation occurred in parallel on different shores within Sweden and Spain (Johannesson, 2001; Johannesson, Johannesson, & Rolán-Alvarez, 1993; Panova, Hollander, & Johannesson, 2006; Quesada, Posada, Caballero, Morán, & Rolán-Alvarez, 2007). This hypothesis has been widely accepted (Ostevik, Moyers, Owens, & Rieseberg, 2012), although the evidence has been questioned (Butlin et al., 2008) and alternative hypotheses have only once been investigated (Butlin et al., 2014).

Furthermore, explicit tests of parallel divergence using a model-based approach were conducted only once on a regional level (i.e., using data from the United Kingdom, Sweden, and Spain) and with AFLP loci (Butlin et al., 2014). Thus, it is still plausible to speculate that the ecotypes evolved independently in different parts of Europe, but that some regions (e.g., Sweden) were colonized by individuals originating from a single location. In fact, phylogeographic analyses (Doellman, Trussell, Grahame, & Vollmer, 2011; Panova et al., 2011) imply that Iberian populations have been genetically separated from northern European populations for a longer time than Swedish and British populations, which are likely to have shared a postglacial colonization history.

The recent assembly of a reference genome and the construction of a genetic map for *L. saxatilis* (Westram et al., 2018), allowed the confirmation of a pattern of heterogeneous genomic differentiation between ecotypes and the identification of some genomic regions, which tend to coincide with polymorphic inversions, enriched for the presence of outliers (Faria et al., 2019; Westram, Faria, Johannesson, & Butlin, 2021; Westram et al., 2018). Moreover, whole-genome sequencing of pools of individuals from several populations of both ecotypes revealed that outlier sharing is common even among distant populations, although it decreases as the distance between populations increases (Morales et al., 2019).

Other less-well studied species inhabiting the intertidal zone also exhibit phenotypic variation associated with an environmental cline. One of these is the flat periwinkle, *Littorina fabalis*, a closely related species of *L. saxatilis* (Kemppainen, Nes, Ceder, & Johannesson, 2005; Reimchen, 1981; Tatarenkov & Johannesson, 1994, 1998, 1999). In northern Europe, phenotypic variation in this species is associated with different levels of wave exposure. On relatively exposed coastlines, we find individuals with large, thick shells, known as "large ecotype", whereas on sheltered habitats, individuals with smaller and thinner shells, known as "dwarf ecotype" are more common. Unlike *L. saxatilis*, which lives on rocks, these *L. fabalis* ecotypes dwell on brown macroalgae (*Fucus spp.* and *Ascophyllum spp.*), grazing on the epiphytes that grow on the fronds of the algae (Williams, 1990). It is also possible that the fronds serve as refuge against one of their main predators, the green crab (*Carcinus maenas*), which is prevalent in both microhabitats (Kemppainen et al., 2005).

Previous work in Swedish populations of *L. fabalis* revealed habitat-related variation in one allozyme locus (arginine kinase, Ark), suggesting that this locus is under the effect of natural selection related to wave exposure and/or other connected factors (Tatarenkov & Johannesson, 1994). These differences in Ark allele frequencies between individuals found at sheltered and moderately exposed habitats were also linked with variation at a random amplification of polymorphic DNA (RAPD) locus and the aforementioned size differences. This pattern persisted even in sites char-

acterized by moderate wave exposure. This suggests that despite the presence of gene flow, as indicated by the absence of differentiation in the majority of analysed allozyme loci, there is some reproductive isolation between the ecotypes (Johannesson & Mikhailova, 2004; Tatarenkov & Johannesson, 1998).

Analogous habitat-related phenotypic divergence has also been reported in the United Kingdom (Wales), France, and Norway, where, unlike Sweden, *L. fabalis* is also subjected to high tidal amplitudes (Kemppainen, Panova, Hollander, & Johannesson, 2009; Reimchen, 1981; Tatarenkov & Johannesson, 1999). While it is possible that a northern *L. fabalis* refugium existed during the last glacial maximum (LGM) (Sotelo et al., 2020), it is more likely that the majority of these coastlines were colonized after the LGM (Charbit, Ritz, Philippon, Peyaud, & Kageyama, 2007). This, together with the absence of appreciable genetic differentiation at neutral markers between the ecotypes across this region (Sotelo et al., 2020), implies a relatively recent local establishment of habitat-related phenotypic divergence.

In clear contrast with the pattern observed for neutral markers, Ark intron sequencing revealed significantly substantial divergence between individuals from sheltered and moderately exposed habitats (Kemppainen, Lindskog, Butlin, & Johannesson, 2011). Results from populations in these different countries showed that one haplotype was nearly fixed and shared throughout sheltered habitats, while wave-exposed habitats maintained higher variation (Johannesson & Mikhailova, 2004; Kemppainen et al., 2011). This raised the possibility of "evolution in concert," in which locally adapted genes emerge once and then spread to distant populations, inhabiting the same habitat, via ecotype-specific selective sweeps (Johannesson et al., 2010; Kemppainen et al., 2011; Schluter & Conte, 2009). Despite this, and except for Ark and the putatively associated RAPD locus, it is still unknown whether the same genetic variation is involved in the evolution of these ecotypes. Additionally, a more systematic analysis of shell morphology is required to test if the observed differences between ecotyopes are consistent across multiple locations.

1.8 THESIS AIMS AND OUTLINE

In this thesis, I studied two different species of the *Littorina* genus that potentially represent distinct stages of the speciation continuum. The fundamental question throughout was whether the evolution of phenotypically divergent ecotypes observed in *L. fabalis* and *L. saxatilis* occured through parallel evolution independently in different locations or whether the ecotypes had a single origin. Parallel evolution provides indirect evidence for the action of natural selection in driving diver-

gence, hence it is important to disentangle these two scenarios. Another key question was quantifying gene flow levels during ecotype formation, and assessing whether the remarkable phenotypic differences between ecotypes found in close proximity could evolve despite gene flow. To answer these questions I combined modelling with population genetics data, analysing patterns of genetic structure and sharing of outlier loci and leveraging whole-genome data obtained with Pool-seq to explicitly contrast models representing scenarios of parallel or single origin. To accomplish the main goals of this thesis, I defined the following specific objectives:

- Detect nonoutlier and outlier loci of *Littorina fabalis* ecotypes and compare their patterns to assess whether population structure is consistent with parallel evolution.
- Develop new methods to model and simulate Pool-seq data, accounting for differences in coverage depth, variations in individual contributions, and sequencing errors.
- Streamline the use of Pool-seq data in demographic inference by developing and validating a method to distinguish between various scenarios of ecotype formation.
- Contrast scenarios of ecotype formation in *Littorina saxatilis* at different geographic scales, using Pool-seq data from populations from Sweden and Spain.

The thesis is divided into four distinct chapters, preceded by a general introduction and followed by a general discussion that integrates the content of these chapters. Each of these four chapters tackles one of aforementioned objectives:

- **Chapter 2 -** Genetic and morphological divergence between *Littorina fabalis* ecotypes in Northern Europe
- **Chapter 3 -** poolHelper: an R package to help in designing Pool-Seq studies
- **Chapter 4 -** Integrating Pool-seq uncertainties into demographic inference
- **Chapter 5 -** Parallel evolution of local adaptation in *Littorina saxatilis* inferred with whole-genome pool-seq data

In the first chapter, I focused on two *Littorina fabalis* ecotypes from Northern European shores. The main objective of this chapter was to provide an initial understanding of the patterns of habitat-related phenotypic and genetic divergence across three different geographic levels in *Littorina fabalis*, which remains less explored in scientific literature compared to *Littorina saxatilis*. To achieve

this, I conducted a comprehensive analysis, comparing the patterns of phenotypic and genetic divergence across three distinct geographic scales: local, regional, and global. My aim was to evaluate the extent to which the patterns of phenotypic and genetic divergence differ between locations and if the genetic basis for the observed phenotypic divergence is the same. Through characterizing the phenotypic and genetic differentiation of these *L. fabalis* ecotypes, this chapter contributed to advancing our knowledge of adaptation and its role in promoting diversification within the marine ecosystem. Ultimately, the findings from this chapter provide a preliminary understanding of the speciation dynamics in *Littorina fabalis*, offering a foundation for future investigations in this species.

In the next chapter, I sought to address the previously identified lack of methods to model and simulate Pool-seq data. The goal of this chapter was to develop a convenient and easily accessible method to simulate Pool-seq data, effectively accounting for various factors such as differences in coverage depth, variations in individual contributions, and sequencing errors. My aim was first, to develop an effective tool capable of simulating Pool-seq data, and secondly, to utilize this tool to evaluate the precision of allele frequency estimates obtained through Pool-seq. Thus, my aim was not only to enable the simulation of Pool-seq data but also to offer a means to assess the reliability and accuracy of allele frequency estimates obtained from Pool-seq experiments. The work developed here led to the release of a software package, implemented in the R environment, that allows researchers to simulate Pool-seq data under different combinations of parameters (e.g. pool size, depth of coverage, unequal individual contribution) before sampling and generating data. Thus, this package provides a valuable tool for researchers, allowing them to identify the best sampling scheme to answer their research questions.

The subsequent chapter of my thesis built upon the previous chapter by incorporating the method for simulating Pool-seq data into an Approximate Bayesian Computation (ABC) framework. The main objective of this chapter was twofold: firstly, to bridge the gap between Pool-seq data and demographic inference and secondly, to provide an easy-to-use method capable of leveraging whole-genome data to differentiate between scenarios of ecotype formation. In this chapter, I developed a novel method capable of jointly modelling Pool-seq data, demographic history and the effects of selection due to barrier loci. Through this approach I obtained accurate estimates of demographic history parameters while accounting for technical errors associated with Pool-seq. The resulting method has been made accessible as an R package, allowing researchers to differentiate between general scenarios of ecotype formation (single versus parallel origin) using genomic data obtained from Pool-seq across multiple populations. To validate the efficacy of the method, I conducted a comprehensive simulation study and applied it to real *Littorina saxatilis* Pool-seq data. The results showed that demographic modelling and inference can be successfully achieved

using Pool-seq data within an ABC framework. Overall, this chapter aims to fill an important gap in the field, offering an efficient tool for researchers interested in using Pool-seq data to perform demographic inference.

In the final chapter, I applied the methodologies developed in the previous two chapters to analyse a large Pool-seq dataset of the rough periwinkle, *Littorina saxatilis*. The goal of this chapter was to contrast scenarios of ecotype formation and assess whether the origin of the ecotypes was the result of parallel evolution. Understanding and assessing the neutral demographic history of species exhibiting the recurrence of similar phenotypes across a wide geographical range is essential in enhancing our comprehension of how natural selection influences the process of divergence. Furthermore, it is important to pinpoint the specific taxa where this re-occurrence of phenotypes occurred in parallel. Identifying such taxa is crucial because only populations from those species can be regarded as natural replicates, providing a strong foundation for studying local adaptation and reproductive isolation. Ultimately, the work developed in this thesis may contribute to the study of speciation and adaptation in other taxa with similar patterns of phenotypic divergence across diverse geographical regions.

1.9 REFERENCES

Abbott, R., Albach, D., Ansell, S., Arntzen, J. W., Baird, S. J., Bierne, N., ... others (2013). Hybridization and speciation. *Journal of evolutionary biology*, *26*(2), 229-246. doi: 10.1111/j.1420-9101.2012.02599.x

Anderson, E. C., Skaug, H. J., & Barshis, D. J. (2014). Next-generation sequencing for molecular ecology: a caveat regarding pooled samples. *Molecular Ecology*, *23*(3), 502-512. doi: 10.1111/mec.12609

Arciniega, M., Clemento, A. J., Miller, M. R., Peterson, M., Garza, J. C., & Pearse, D. E. (2016). Parallel evolution of the summer steelhead ecotype in multiple populations from oregon and northern california. *Conservation Genetics*, *17*, 165-175. doi: 10.1007/s10592-015-0769-2

Beaumont, M. A., Zhang, W., & Balding, D. J. (2002). Approximate bayesian computation in population genetics. *Genetics*, *162*(4), 2025-2035. doi: 10.1111/j.1937-2817.2010.tb01236.x

Begun, D. J., Holloway, A. K., Stevens, K., Hillier, L. W., Poh, Y.-P., Hahn, M. W., ... others (2007). Population genomics: whole-genome analysis of polymorphism and divergence in *Drosophila simulans*. *PLoS biology*, *5*(11), e310. doi: 10.1371/journal.pbio.0050310

Beichman, A. C., Huerta-Sanchez, E., & Lohmueller, K. E. (2018). Using genomic data to infer historic population dynamics of nonmodel organisms. *Annual Review of Ecology, Evolution, and Systematics*, *49*, 433-456. doi: 10.1146/annurev-ecolsys-110617-062431

Bertorelle, G., Benazzo, A., & Mona, S. (2010). Abc as a flexible framework to estimate demog-

raphy over space and time: Some cons, many pros. *Molecular Ecology*, *19*(13), 2609-2625. doi: 10.1111/j.1365-294X.2010.04690.x

Bierne, N., Welch, J., Loire, E., Bonhomme, F., & David, P. (2011). The coupling hypothesis: why genome scans may fail to map local adaptation genes. *Molecular ecology*, *20*(10), 2044-2072. doi: 10.1111/j.1365-294X.2011.05080.x

Bohutínská, M., Vlček, J., Yair, S., Laenen, B., Konečná, V., Fracassetti, M., ... Kolář, F. (2021). Genomic basis of parallel adaptation varies with divergence in arabidopsis and its relatives. *Proceedings of the National Academy of Sciences*, *118*(21), e2022713118. doi: 10.1073/pnas.2022713118

Boitard, S., Rodríguez, W., Jay, F., Mona, S., & Austerlitz, F. (2016). Inferring population size history from large samples of genome-wide molecular data - an approximate bayesian computation approach. *PLOS Genetics*, *12*(3), 1-36. doi: 10.1371/journal.pgen.1005877

Butlin, R. K., Debelle, A., Kerth, C., Snook, R. R., Beukeboom, L. W., Cajas, R. C., ... Schilthuizen, M. (2012). What do we need to know about speciation? *Trends in Ecology & Evolution*, *27*(1), 27-39. doi: 10.1016/j.tree.2011.09.002

Butlin, R. K., Galindo, J., & Grahame, J. W. (2008). Sympatric, parapatric or allopatric: the most important way to classify speciation? *Philosophical Transactions of the Royal Society B: Biological Sciences*, *363*(1506), 2997-3007. doi: 10.1098/rstb.2008.0076

Butlin, R. K., Saura, M., Charrier, G., Jackson, B., André, C., Caballero, A., ... Rolán-Alvarez, E. (2014). Parallel evolution of local adaptation and reproductive isolation in the face of gene flow. *Evolution*, *68*(4), 935-949. doi: 10.1111/evo.12329

Cai, Z., Zhou, L., Ren, N.-N., Xu, X., Liu, R., Huang, L., ... Ge, S. (2019). Parallel speciation of wild rice associated with habitat shifts. *Molecular Biology and Evolution*, *36*(57), 875-889. doi: 10.1093/molbev/msz029

Calvo, S. E., Tucker, E. J., Compton, A. G., Kirby, D. M., Crawford, G., Burtt, N. P., ... Mootha, V. K. (2010). High-throughput, pooled sequencing identifies mutations in nubpl and foxred1 in human complex i deficiency. *Nature Genetics*, *42*(10), 851-858. doi: 10.1038/ng.659

Charbit, S., Ritz, C., Philippon, G., Peyaud, V., & Kageyama, M. (2007). Numerical reconstructions of the northern hemisphere ice sheets through the last glacial-interglacial cycle. *Climate of the Past*, *3*(1), 15-37. doi: 10.5194/cp-3-15-2007

Chen, C., Parejo, M., Momeni, J., Langa, J., Nielsen, R. O., Shi, W., ... others (2022). Population structure and diversity in european honey bees (*Apis mellifera L.*)—an empirical comparison of pool and individual whole-genome sequencing. *Genes*, *13*(2), 182. doi: 10.3390/genes13020182

Collin, F.-d., Durif, G., Raynal, L., Lombaert, E., Gautier, M., Vitalis, R., ... Estoup, A. (2021). Extending approximate bayesian computation with supervised machine learning to infer demographic history from genetic polymorphisms using diyabc random forest. *Molecular Ecology Resources*, *21*(8), 2598-2613. doi: 10.1111/1755-0998.13413

Colosimo, P. F., Hosemann, K. E., Balabhadra, S., Villarreal, G., Dickson, M., Grimwood, J., ...

Kingsley, D. M. (2005). Widespread parallel evolution in sticklebacks by repeated fixation of ectodysplasin alleles. *Science*, *307*(5717), 1928-1933. doi: 10.1126/science.1107239

Conde-Padín, P., Cruz, R., Hollander, J., & Rolan-Alvarez, E. (2008). Revealing the mechanisms of sexual isolation in a case of sympatric and parallel ecological divergence. *Biological Journal of the Linnean Society*, *94*(3), 513-526. doi: 10.1111/j.1095-8312.2008.00998.x

Connell, J. H. (1961). The influence of interspecific competition and other factors on the distribution of the barnacle *Chthamalus stellatus*. *Ecology*, *42*(4), 710-723. doi: 10.2307/1933500

Connell, J. H. (1972). Community interactions on marine rocky intertidal shores. *Annual review of ecology and systematics*, *3*(1), 169-192. doi: 10.1146/annurev.es.03.110172.001125

Cooke, N. P., & Nakagome, S. (2018). Fine-tuning of approximate bayesian computation for human population genomics. *Current Opinion in Genetics and Development*, *53*, 60-69. doi: 10.1016/j.gde.2018.06.016

Cornuet, J. M., Pudlo, P., Veyssier, J., Dehne-Garcia, A., Gautier, M., Leblois, R., ... Estoup, A. (2014). Diyabc v2.0: A software to make approximate bayesian computation inferences about population history using single nucleotide polymorphism, dna sequence and microsatellite data. *Bioinformatics*, *30*(8), 1187-1189. doi: 10.1093/bioinformatics/btt763

Coyne, J., & Orr, H. (2004). *Speciation*. Sunderland, MA. Sinauer Associates.

Cruickshank, T. E., & Hahn, M. W. (2014). Reanalysis suggests that genomic islands of speciation are due to reduced diversity, not reduced gene flow. *Molecular ecology*, *23*(13), 3133-3157. doi: 10.1111/mec.12796

Csilléry, K., Blum, M. G., Gaggiotti, O. E., & François, O. (2010). Approximate bayesian computation (abc) in practice. *Trends in Ecology & Evolution*, *25*(7), 410-418. doi: 10.1016/j.tree.2010.04.001

Cutler, D. J., & Jensen, J. D. (2010). To pool, or not to pool? *Genetics*, *186*(1), 41-43. doi: 10.1534/genetics.110.121012

Darwin, C. (1859). *On the origin of species by means of natural selection*. London: Murray.

Denny, M., & Wethey, D. (2001). Physical processes that generate patterns in marine communities. *Marine community ecology*, 3-37.

Doellman, M. M., Trussell, G. C., Grahame, J. W., & Vollmer, S. V. (2011). Phylogeographic analysis reveals a deep lineage split within north atlantic *Littorina saxatilis*. *Proceedings of the Royal Society B: Biological Sciences*, *278*(1722), 3175-3183. doi: 10.1098/rspb.2011.0346

Dorant, Y., Benestan, L., Rougemont, Q., Normandeau, E., Boyle, B., Rochette, R., & Bernatchez, L. (2019). Comparing pool-seq, rapture, and gbs genotyping for inferring weak population structure: The american lobster (*Homarus americanus*) as a case study. *Ecology and evolution*, *9*(11), 6606-6623. doi: 10.1002/ece3.5240

Ellegren, H. (2014). Genome sequencing and population genomics in non-model organisms. *Trends in Ecology & Evolution*, *29*(1), 51-63. doi: 10.1016/j.tree.2013.09.008

Elmer, K. R., & Meyer, A. (2011). Adaptation in the age of ecological genomics: insights from parallelism and convergence. *Trends in ecology & evolution*, *26*(6), 298-306. doi: 10.1016/j.tree.2011.02.008

Excoffier, L., Marchi, N., Marques, D. A., Matthey-Doret, R., Gouy, A., & Sousa, V. C. (2021). fastsimcoal2: demographic inference under complex evolutionary scenarios. *Bioinformatics*, *37*(24), 4882-4885. doi: 10.1093/bioinformatics/btab468

Faria, R., Chaube, P., Morales, H. E., Larsson, T., Lemmon, A. R., Lemmon, E. M., ... Butlin, R. K. (2019). Multiple chromosomal rearrangements in a hybrid zone between *Littorina saxatilis* ecotypes. *Molecular Ecology*, *28*(6), 1375-1393. doi: 10.1111/mec.14972

Faria, R., Johannesson, K., & Stankowski, S. (2021). Speciation in marine environments: Diving under the surface. *Journal of Evolutionary Biology*, *34*(1), 4-15. doi: 10.1111/jeb.13756

Faria, R., Renaut, S., Galindo, J., Pinho, C., Melo-Ferreira, J., Melo, M., ... Butlin, R. K. (2014). Advances in ecological speciation: an integrative approach. *Molecular Ecology*, *23*(3), 513-521. doi: 10.1111/mec.12616

Felsenstein, J. (1981). Skepticism towards santa rosalia, or why are there so few kinds of animals? *Evolution*, *35*(1), 124-138. doi: 10.2307/2407946

Ferretti, L., Ramos-Onsins, S. E., & Pérez-Enciso, M. (2013). Population genomics from pool sequencing. *Molecular Ecology*, *22*(22), 5561-5576. doi: 10.1111/mec.12522

Fu, Y.-X., & Li, W.-H. (1997). Estimating the age of the common ancestor of a sample of dna sequences. *Molecular biology and evolution*, *14*(2), 195-199. doi: 10.1093/oxfordjournals.molbev.a025753

Funk, D. J. (2012). Of "host forms" and host races: Terminological issues in ecological speciation. *International Journal of Ecology*. doi: 10.1155/2012/506957

Futschik, A., & Schlötterer, C. (2010). The next generation of molecular markers from massively parallel sequencing of pooled dna samples. *Genetics*, *186*(1), 207-218. doi: 10.1534/genetics.110.114397

Galindo, J., & Grahame, J. W. (2014). Ecological speciation and the intertidal snail *Littorina saxatilis*. *Advances in Ecology*.

Galindo, J., Martínez-Fernández, M., Rodríguez-Ramilo, S. T., & Rolán-Alvarez, E. (2013). The role of local ecology during hybridization at the initial stages of ecological speciation in a marine snail. *Journal of Evolutionary Biology*, *26*(7), 1472-1487. doi: 10.1111/jeb.12152

Galindo, J., Morán, P., & Rolán-Alvarez, E. (2009). Comparing geographical genetic differentiation between candidate and noncandidate loci for adaptation strengthens support for parallel ecological divergence in the marine snail *Littorina saxatilis*. *Molecular Ecology*, *18*(5), 919-930. doi: 10.1111/j.1365-294X.2008.04076.x

Galtier, N. (2023). Phylogenetic conflicts: distinguishing gene flow from incomplete lineage sorting. *bioRxiv*. doi: 10.1101/2023.07.06.547897

Gautier, M., Foucaud, J., Gharbi, K., Cézard, T., Galan, M., Loiseau, A., ... Estoup, A.

(2013). Estimation of population allele frequencies from next-generation sequencing data: pool-versus individual-based genotyping. *Molecular Ecology*, *22*(14), 3766-3779. doi: 10.1111/mec.12360

Gavrilets, S., Vose, A., Barluenga, M., Salzburger, W., & Meyer, A. (2007). Case studies and mathematical models of ecological speciation. 1. cichlids in a crater lake. *Molecular Ecology*, *16*(14), 2893-2909. doi: 10.1111/j.1365-294X.2007.03305.x

Grahame, J. W., Wilding, C. S., & Butlin, R. K. (2006). Adaptation to a steep environmental gradient and an associated barrier to gene exchange in *Littorina saxatilis*. *Evolution*, *60*(2), 268-278. doi: 10.1111/j.0014-3820.2006.tb01105.x

Helmuth, B., & Denny, M. W. (2003). Predicting wave exposure in the rocky intertidal zone: do bigger waves always lead to larger forces? *Limnology and Oceanography*, *48*(3), 1338-1345. doi: 10.4319/lo.2003.48.3.1338

Helmuth, B., & Hofmann, G. E. (2001). Microhabitats, thermal heterogeneity, and patterns of physiological stress in the rocky intertidal zone. *The Biological Bulletin*, *201*(3), 374-384. doi: 10.2307/1543615

Hendry, A. P., Bolnick, D. I., Berner, D., & Peichel, C. L. (2009). Along the speciation continuum in sticklebacks. *Journal of fish biology*, *75*(8), 2000-2036. doi: 10.1111/j.1095-8649.2009.02419.x

Hewitt, G. M. (2011). Quaternary phylogeography: the roots of hybrid zones. *Genetica*, *139*, 617-638. doi: 10.1007/s10709-011-9547-3

Hey, J., & Machado, C. A. (2003). The study of structured populations — new hope for a difficult and divided science. *Nature Reviews Genetics*, *4*(7), 535-543. doi: doi.org/10.1038/nrg1112

Hickerson, M. J. (2014). All models are wrong. *Molecular Ecology*, *23*(12), 2887-2889. doi: 10.1111/mec.12794

Hollander, J., Galindo, J., & Butlin, R. K. (2015). Selection on outlier loci and their association with adaptive phenotypes in *Littorina saxatilis* contact zones. *Journal of Evolutionary Biology*, *28*(2), 328-337. doi: 10.1111/jeb.12564

Huang, W., Takebayashi, N., Qi, Y., & Hickerson, M. J. (2011). Mtml-msbayes: Approximate bayesian comparative phylogeographic inference from multiple taxa and multiple loci with rate heterogeneity. *BMC Bioinformatics*, *12*(1), 1-14. doi: 10.1186/1471-2105-12-1

James, M. E., Arenas-Castro, H., Groh, J. S., Allen, S. L., Engelstädter, J., & Ortiz-Barrientos, D. (2021). Highly replicated evolution of parapatric ecotypes. *Molecular Biology and Evolution*, *38*(11), 4805-4821. doi: 10.1093/molbev/msab207

Jay, F., Boitard, S., & Austerlitz, F. (2019). An abc method for whole-genome sequence data: inferring paleolithic and neolithic human expansions. *Molecular biology and evolution*, *36*(7), 1565-1579. doi: 10.1093/molbev/msz038

Johannesson, K. (2001). Parallel speciation: a key to sympatric divergence. *Trends in Ecology & Evolution*, *16*(3), 148-153. doi: 10.1016/S0169-5347(00)02078-4

Johannesson, K. (2016). What can be learnt from a snail? *Evolutionary applications*, *9*(1), 153-165. doi: 10.1111/eva.12277

Johannesson, K., Johannesson, B., & Rolán-Alvarez, E. (1993). Morphological differentiation and genetic cohesiveness over a microenvironmental gradient in the marine snail *Littorina saxatilis*. *Evolution*, *47*(6), 1770-1787. doi: 10.1111/j.1558-5646.1993.tb01268.x

Johannesson, K., & Mikhailova, N. (2004). Habitat-related genetic substructuring in a marine snail (*Littorina fabalis*) involving a tight link between an allozyme and a dna locus. *Biological Journal of the Linnean Society*, *81*(2), 301-306. doi: 10.1111/j.1095-8312.2003.00288.x

Johannesson, K., Panova, M., Kemppainen, P., André, C., Rolán-Alvarez, E., & Butlin, R. K. (2010). Repeated evolution of reproductive isolation in a marine snail: unveiling mechanisms of speciation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *365*(1547), 1735-1747. doi: 10.1098/rstb.2009.0256

Jones, F. C., Grabherr, M. G., Chan, Y. F., Russell, P., Mauceli, E., Johnson, J., ... Team, B. I. G. S. P. . W. G. A. (2012). The genomic basis of adaptive evolution in threespine sticklebacks. *Nature*, *484*(7392), 55-61. doi: 10.1038/nature10944

Kemppainen, P., Lindskog, T., Butlin, R. K., & Johannesson, K. (2011). Intron sequences of arginine kinase in an intertidal snail suggest an ecotype-specific selective sweep and a gene duplication. *Heredity*, *106*(5), 808-816. doi: 10.1038/hdy.2010.123

Kemppainen, P., Nes, S. v., Ceder, C., & Johannesson, K. (2005). Refuge function of marine algae complicates selection in an intertidal snail. *Oecologia*, *143*, 402-411. doi: 10.1007/s00442-004-1819-5

Kemppainen, P., Panova, M., Hollander, J., & Johannesson, K. (2009). Complete lack of mitochondrial divergence between two species of ne atlantic marine intertidal gastropods. *Journal of evolutionary biology*, *22*(10), 2000-2011. doi: 10.1111/j.1420-9101.2009.01810.x

Kess, T., & Boulding, E. G. (2019). Genome-wide association analyses reveal polygenic genomic architecture underlying divergent shell morphology in spanish *Littorina saxatilis* ecotypes. *Ecology and evolution*, *9*(17), 9427-9441. doi: 10.1002/ece3.5378

Kess, T., Galindo, J., & Boulding, E. G. (2018). Genomic divergence between spanish *Littorina saxatilis* ecotypes unravels limited admixture and extensive parallelism associated with population history. *Ecology and Evolution*, *8*(16), 8311-8327. doi: 10.1002/ece3.4304

Lenormand, T., Roze, D., & Rousset, F. (2009). Stochasticity in evolution. *Trends in ecology & evolution*, *24*(3), 157-165. doi: 10.1016/j.tree.2008.09.014

Le Pennec, G., Butlin, R. K., Jonsson, P. R., Larsson, A. I., Lindborg, J., Bergström, E., ... Johannesson, K. (2017). Adaptation to dislodgement risk on wave-swept rocky shores in the snail *Littorina saxatilis*. *PloS one*, *12*(10), e0186901. doi: 10.1371/journal.pone.0186901

Lewis, J. J., Geltman, R. C., Pollak, P. C., Rondem, K. E., Van Belleghem, S. M., Hubisz, M. J., ... Reed, R. D. (2019). Parallel evolution of ancient, pleiotropic enhancers underlies butterfly wing pattern mimicry. *Proceedings of the National Academy of Sciences*, *116*(48), 24174-24183. doi: 10.1073/pnas.1907068116

Li, J., Li, H., Jakobsson, M., Li, S., SjÖDin, P., & Lascoux, M. (2012). Joint analysis of demography and selection in population genetics: where do we stand and where could we go? *Molecular ecology*, *21*(1), 28-44. doi: 10.1111/j.1365-294X.2011.05308.x

Lieberman, T. D., Flett, K. B., Yelin, I., Martin, T. R., McAdam, A. J., Priebe, G. P., & Kishony, R. (2014). Genetic variation of a bacterial pathogen within individuals with cystic fibrosis provides a record of selective pressures. *Nature Genetics*, *46*(1), 82-87. doi: 10.1038/ng.2848

Liepe, J., Kirk, P., Filippi, S., Toni, T., Barnes, C. P., & Stumpf, M. P. (2014). A framework for parameter estimation and model selection from experimental data in systems biology using approximate bayesian computation. *Nature Protocols*, *9*(2), 439-456. doi: 10.1038/nprot.2014.025

Little, C., & Kitching, J. A. (1996). *The biology of rocky shores*. Oxford University Press, USA.

Lowry, D. B. (2012). Ecotypes and the controversy over stages in the formation of new species. *Biological Journal of the Linnean Society*, *106*(2), 241-257. doi: 10.1111/j.1095-8312.2012.01867.x

Lyell, C. (1830). *Principles of geology* (Vol. 1). Univ. of Chicago Press, Chicago.

Maltseva, A. L., Varfolomeeva, M. A., Ayanka, R. V., Gafarova, E. R., Repkin, E. A., Pavlova, P. A., ... Granovitch, A. I. (2021). Linking ecology, morphology, and metabolism: Niche differentiation in sympatric populations of closely related species of the genus littorina (neritrema). *Ecology and evolution*, *11*(16), 11134-11154. doi: 10.1002/ece3.7901

Marques, D. A., Lucek, K., Meier, J. I., Mwaiko, S., Wagner, C. E., Excoffier, L., & Seehausen, O. (2016). Genomics of rapid incipient speciation in sympatric threespine stickleback. *PLoS genetics*, *12*(2), e1005887. doi: 10.1371/journal.pgen.1005887

Mayr, E. (1942). *Systematics and the origin of species from the viewpoint of zoologist*. London: Harvard University Press.

Mayr, E. (1963). *Animal species and evolution*. London: Harvard University Press.

Momigliano, P., Florin, A.-B., & Merilä, J. (2021). Biases in demographic modeling affect our understanding of recent divergence. *Molecular Biology and Evolution*, *38*(7), 2967-2985. doi: 10.1093/molbev/msab047

Morales, H. E., Faria, R., Johannesson, K., Larsson, T., Panova, M., Westram, A. M., & Butlin, R. K. (2019). Genomic architecture of parallel ecological divergence: beyond a single environmental contrast. *Science advances*, *5*(12), eaav9963. doi: 10.1126/sciadv.aav9963

Nosil, P. (2012). *Ecological speciation*. Oxford University Press.

Ostevik, K. L., Moyers, B. T., Owens, G. L., & Rieseberg, L. H. (2012). Parallel ecological speciation in plants? *International Journal of Ecology*. doi: 10.1155/2012/939862

Paine, R. T. (1966). Food web complexity and species diversity. *The American Naturalist*, *100*(910), 65-75. doi: 10.1086/282400

Paine, R. T., & Fenchel, T. (1994). *Marine rocky shores and community ecology: an experimental-*

ist's perspective. Ecology Institute Oldendorf/Luhe, Germany.

Panova, M., Blakeslee, A. M., Miller, A. W., Mäkinen, T., Ruiz, G. M., Johannesson, K., & André, C. (2011). Glacial history of the north atlantic marine snail, *Littorina saxatilis*, inferred from distribution of mitochondrial dna lineages. *PLoS One*, *6*(3), e17511. doi: 10.1371/journal.pone.0017511

Panova, M., Hollander, J., & Johannesson, K. (2006). Site-specific genetic divergence in parallel hybrid zones suggests nonallopatric evolution of reproductive barriers. *Molecular Ecology*, *15*(13), 4021-4031. doi: 10.1111/j.1365-294X.2006.03067.x

Parts, L., Cubillos, F. A., Warringer, J., Jain, K., Salinas, F., Bumpstead, S. J., ... Liti, G. (2011). Revealing the genetic structure of a trait by sequencing a population under selection. *Genome Research*, *21*(7), 1131-1138. doi: 10.1101/gr.116731.110

Payseur, B. A., & Rieseberg, L. H. (2016). A genomic perspective on hybridization and speciation. *Molecular ecology*, *25*(11), 2337-2360. doi: 10.1111/mec.13557

Phadnis, N., Baker, E. P., Cooper, J. C., Frizzell, K. A., Hsieh, E., de la Cruz, A. F. A., ... Malik, H. S. (2015). An essential cell cycle regulation gene causes hybrid inviability in *Drosophila*. *Science*, *350*(6267), 1552-1555. doi: 10.1126/science.aac7504

Pontarp, M., Brännström, Å., & Petchey, O. L. (2019). Inferring community assembly processes from macroscopic patterns using dynamic eco-evolutionary models and approximate bayesian computation (abc). *Methods in Ecology and Evolution*, *10*(4), 450-460. doi: 10.1111/2041-210X.13129

Prescott, N. J., Lehne, B., Stone, K., Lee, J. C., Taylor, K., Knight, J., ... Consortium, U. I. G. (2015). Pooled sequencing of 531 genes in inflammatory bowel disease identifies an associated rare variant in btnl2 and implicates other immune related genes. *PLOS Genetics*, *11*(2), 1-19. doi: 10.1371/journal.pgen.1004955

Presgraves, D. C. (2002). Patterns of postzygotic isolation in *Lepidoptera*. *Evolution*, *56*(6), 1168-1183. doi: 10.1111/j.0014-3820.2002.tb01430.x

Quesada, H., Posada, D., Caballero, A., Morán, P., & Rolán-Alvarez, E. (2007). Phylogenetic evidence for multiple sympatric ecological diversification in a marine snail. *Evolution: International Journal of Organic Evolution*, *61*(7), 1600-1612. doi: 10.1111/j.1558-5646.2007.00135.x

Raffaelli, D., & Hawkins, S. J. (1996). *Intertidal ecology*. Springer Science & Business Media.

Ravinet, M., Faria, R., Butlin, R., Galindo, J., Bierne, N., Rafajlović, M., ... Westram, A. (2017). Interpreting the genomic landscape of speciation: a road map for finding barriers to gene flow. *Journal of evolutionary biology*, *30*(8), 1450-1477. doi: 10.1111/jeb.13047

Ravinet, M., Yoshida, K., Shigenobu, S., Toyoda, A., Fujiyama, A., & Kitano, J. (2018). The genomic landscape at a late stage of stickleback speciation: high genomic divergence interspersed by small localized regions of introgression. *PLoS genetics*, *14*(5), e1007358. doi: 10.1371/journal.pgen.1007358

Reid, D. G. (1996). *Systematics and evolution of Littorina*. London: Ray Society.

Reimchen, T. (1981). Microgeographical variation in *Littorina mariae Sacchi & Rastelli* and a taxonomic consideration. *Journal of conchology*, *30*, 341-350.

Roda, F., Walter, G. M., Nipper, R., & Ortiz-Barrientos, D. (2017). Genomic clustering of adaptive loci during parallel evolution of an australian wildflower. *Molecular Ecology*, *26*(14), 3687-3699. doi: 10.1111/mec.14150

Ross, P. A., Endersby-Harshman, N. M., & Hoffmann, A. A. (2019). Shedding light on the grey zone of speciation along a continuum of genomic divergence. *Evolutionary applications*, *12*(3), 572-586. doi: doi.org/10.1111/eva.12740

Rougemont, Q., & Bernatchez, L. (2018). The demographic history of atlantic salmon (*Salmo salar*) across its distribution range reconstructed from approximate bayesian computations. *Evolution*, *72*(6), 1261-1277. doi: 10.1111/evo.13486

Roux, C., Fraisse, C., Romiguier, J., Anciaux, Y., Galtier, N., & Bierne, N. (2016). A comprehensive assessment of inbreeding and laboratory adaptation in *Aedes aegypti* mosquitoes. *PLoS biology*, *14*(12), e2000234. doi: 10.1371/journal.pbio.2000234

Rubin, C.-J., Megens, H.-J., Barrio, A. M., Maqbool, K., Sayyab, S., Schwochow, D., ... Andersson, L. (2012). Strong signatures of selection in the domestic pig genome. *Proceedings of the National Academy of Sciences*, *109*(48), 19529-19536. doi: 10.1073/pnas.1217149109

Sanford, E., & Kelly, M. W. (2011). Local adaptation in marine invertebrates. *Annual review of marine science*, *3*, 509-535. doi: 10.1146/annurev-marine-120709-142756

Schlötterer, C., Tobler, R., Kofler, R., & Nolte, V. (2014). Sequencing pools of individuals — mining genome-wide polymorphism data without big funding. *Nature Reviews Genetics*, *15*(11), 749-763. doi: 10.1038/nrg3803

Schluter, D. (2000). *The ecology of adaptive radiation*. Oxford University Press.

Schluter, D. (2009). Evidence for ecological speciation and its alternative. *Science*, *323*(5915), 737-741. doi: 10.1126/science.1160006

Schluter, D., & Conte, G. L. (2009). Genetics and ecological speciation. *Proceedings of the National Academy of Sciences*, *106*, 9955-9962. doi: 10.1073/pnas.0901264106

Schluter, D., & Nagel, L. M. (1995). Parallel speciation by natural selection. *The American Naturalist*, *146*(2), 292-301. doi: 10.1086/285799

Seehausen, O., Butlin, R. K., Keller, I., Wagner, C. E., Boughman, J. W., Hohenlohe, P. A., ... Widmer, A. (2014). Genomics and the origin of species. *Nature Reviews Genetics*, *15*(3), 176-192. doi: 10.1038/nrg3644

Smadja, C. M., & Butlin, R. K. (2011). A framework for comparing processes of speciation in the presence of gene flow. *Molecular Ecology*, *20*(24), 5123-5140. doi: 10.1111/j.1365-294X.2011.05350.x

Smith, C. C., & Flaxman, S. M. (2020). Leveraging whole genome sequencing data for demo-

graphic inference with approximate bayesian computation. *Molecular ecology resources*, *20*(1), 125-139. doi: 10.1111/1755-0998.13092

Soria-Carrasco, V., Gompert, Z., Comeault, A. A., Farkas, T. E., Parchman, T. L., Johnston, J. S., ... others (2014). Stick insect genomes reveal natural selection's role in parallel speciation. *Science*, *344*(6185), 738-742. doi: 10.1126/science.1252136

Sotelo, G., Duvetorp, M., Costa, D., Panova, M., Johannesson, K., & Faria, R. (2020). Phylogeographic history of flat periwinkles, *Littorina fabalis* and *L. obtusata*. *BMC Evolutionary Biology*, *20*(23), 1-18. doi: 10.1186/s12862-019-1561-6

Sousa, V. C., Carneiro, M., Ferrand, N., & Hey, J. (2013). Identifying loci under selection against gene flow in isolation-with-migration models. *Genetics*, *194*(1), 211-233. doi: 10.1534/genetics.113.149211

Sousa, V. C., & Hey, J. (2013). Understanding the origin of species with genome-scale data: modelling gene flow. *Nature Reviews Genetics*, *14*(6), 404-414. doi: 10.1038/nrg3446

Stankowski, S., & Ravinet, M. (2021). Defining the speciation continuum. *Evolution*, *75*(6), 1256-1273. doi: 10.1111/evo.14215

Strasburg, J., & Rieseberg, L. (2013). Methodological challenges to realizing the potential of hybridization research. *Journal of evolutionary biology*, *26*(2), 259-260. doi: 10.1111/jeb.12006

Tatarenkov, A., & Johannesson, K. (1994). Habitat related allozyme variation on a microgeographic scale in the marine snail *Littorina mariae* (prosobranchia: Littorinacea). *Biological Journal of the Linnean Society*, *53*(2), 105-125. doi: 10.1111/j.1095-8312.1994.tb01004.x

Tatarenkov, A., & Johannesson, K. (1998). Evidence of a reproductive barrier between two forms of the marine periwinkle *Littorina fabalis* (gastropoda). *Biological Journal of the Linnean Society*, *63*(3), 349-365. doi: 10.1111/j.1095-8312.1998.tb01522.x

Tatarenkov, A., & Johannesson, K. (1999). Micro-and macrogeographic allozyme variation in *Littorina fabalis*; do sheltered and exposed forms hybridize? *Biological Journal of the Linnean Society*, *67*(2), 199-212. doi: 10.1111/j.1095-8312.1999.tb01861.x

Taus, T., Futschik, A., & Schlötterer, C. (2017). Quantifying selection with pool-seq time series data. *Molecular biology and evolution*, *34*(11), 3023-3034. doi: 10.1093/molbev/msx225

Tavaré, S., Balding, D. J., Griffiths, R. C., & Donnelly, P. (1997). Inferring coalescence times from dna sequence data. *Genetics*, *145*(2), 505-518.

Thia, J. A., McGuigan, K., Liggins, L., Figueira, W. F., Bird, C. E., Mather, A., ... Riginos, C. (2021). Genetic and phenotypic variation exhibit both predictable and stochastic patterns across an intertidal fish metapopulation. *Molecular Ecology*, *30*(18), 4392-4414. doi: 10.1111/mec.15829

Tomanek, L., & Helmuth, B. (2002). Physiological ecology of rocky intertidal organisms: a synergy of concepts. *Integrative and Comparative Biology*, *42*(4), 771-775. doi: 10.1093/icb/42.4.771

Turesson, G. (1922). The genotypical response of the plant species to the habitat. *Hereditas*, *3*(3), 211-350.

Turner, T. L., Hahn, M. W., & Nuzhdin, S. V. (2005). Genomic islands of speciation in *Anopheles gambiae*. *PLoS biology*, *3*(9), e285. doi: 10.1371/journal.pbio.0030285

Turner, T. L., Stewart, A. D., Fields, A. T., Rice, W. R., & Tarone, A. M. (2011). Population-based resequencing of experimentally evolved populations reveals the genetic basis of body size variation in *Drosophila melanogaster*. *PLOS Genetics*, *7*(3), 1-10. doi: 10.1371/journal.pgen.1001336

Via, S. (2009). Natural selection in action during speciation. *Proceedings of the National Academy of Sciences*, *106*, 9939-9946. doi: 10.1073/pnas.0901397106

Weber, A., Rajkov, J., Smailus, K., Egger, B., & Salzburger, W. (2021). Speciation dynamics and extent of parallel evolution along a lake-stream environmental contrast in african cichlid fishes. *Science Advances*, *7*(45), eabg5391. doi: 10.1126/sciadv.abg5391

Wegmann, D., Leuenberger, C., Neuenschwander, S., & Excoffier, L. (2010). Abctoolbox: a versatile toolkit for approximate bayesian computations. *BMC Bioinformatics*, *11*(1), 1-7. doi: 10.1186/1471-2105-11-116

Westram, A. M., Faria, R., Johannesson, K., & Butlin, R. K. (2021). Using replicate hybrid zones to understand the genomic basis of adaptive divergence. *Molecular ecology*, *30*(15), 3797-3814. doi: 10.1111/mec.15861

Westram, A. M., Rafajlović, M., Chaube, P., Faria, R., Larsson, T., Panova, M., ... Butlin, R. K. (2018). Clines on the seashore: The genomic architecture underlying rapid divergence in the face of gene flow. *Evolution letters*, *2*(4), 297-309. doi: 10.1002/evl3.74

Westram, A. M., Stankowski, S., Surendranadh, P., & Barton, N. (2022). What is reproductive isolation? *Journal of evolutionary biology*, *35*(9), 1143-1164. doi: 10.1111/jeb.14005

Wilding, C., Butlin, R. K., & Grahame, J. (2001). Differential gene exchange between parapatric morphs of *Littorina saxatilis* detected using aflp markers. *Journal of Evolutionary Biology*, *14*(4), 611-619. doi: 10.1046/j.1420-9101.2001.00304.x

Williams, G. A. (1990). Periwinkles, *Littorina obtusata* (l.) and *L. mariae Sacchi et Rastelli*. *Field Studies*, *7*, 469-482.

Zhang, J., Dennis, T. E., Landers, T. J., Bell, E., & Perry, G. L. (2017). Linking individual-based and statistical inferential models in movement ecology: A case study with black petrels (*Procellaria parkinsoni*). *Ecological Modelling*, *360*, 425-436. doi: 10.1016/j.ecolmodel.2017.07.017

Zhou, D., Udpa, N., Gersten, M., Visk, D. W., Bashir, A., Xue, J., ... Haddad, G. G. (2011). Experimental selection of hypoxia-tolerant *Drosophila melanogaster*. *Proceedings of the National Academy of Sciences*, *108*(6), 2349-2354. doi: 10.1073/pnas.1010643108

CHAPTER 2

Genetic and morphological divergence between *Littorina fabalis* ecotypes in Northern Europe

Juan Galindo, **João Carvalho**, Graciela Sotelo, Mårten Duvetorp, Diana Costa, Petri Kemppainen, Marina Panova, Antigoni Kaliontzopoulou, Kerstin Johannesson & Rui Faria

Published in *Journal of Evolutionary Biology* (2021) DOI: 10.1111/jeb.13705

2.1 ABSTRACT

Low dispersal marine intertidal species facing strong divergent selective pressures associated with steep environmental gradients have a great potential to inform us about local adaptation and reproductive isolation. Among these, gastropods of the genus *Littorina* offer a unique system to study parallel phenotypic divergence resulting from adaptation to different habitats related with wave exposure. In this study, we focused on two *Littorina fabalis* ecotypes from Northern European shores and compared patterns of habitat-related phenotypic and genetic divergence across three different geographic levels (local, regional and global). Geometric morphometric analyses revealed that individuals from habitats moderately exposed to waves usually present a larger shell size with a wider aperture than those from sheltered habitats. The phenotypic clustering of *L. fabalis* by habitat across most locations (mainly in terms of shell size) support an important role of ecology in morphological divergence. A genome scan based on amplified fragment length polymorphisms (AFLPs) revealed a heterogeneous pattern of differentiation across the genome between populations from the two different habitats, suggesting ecotype divergence in the presence of gene flow. The contrasting patterns of genetic structure between nonoutlier and outlier loci, and the decreased sharing of outlier loci with geographic distance among locations are compatible with parallel evolution of phenotypic divergence, with an important contribution of gene flow and/or ancestral variation. In the future, model-based inference studies based on sequence data across the entire genome will help unravelling these evolutionary hypotheses, improving our knowledge about adaptation and its influence on diversification within the marine realm.

Keywords: adaptation, AFLPs, divergent natural selection, ecological speciation, flat periwinkles, gene flow, geometrics–morphometrics, parallel evolution, shell morphology

2.2 INTRODUCTION

The marine rocky intertidal represents one of the most abrupt environmental gradients on Earth (Little & Kitching, 1996; Raffaelli & Hawkins, 1996; Tomanek & Helmuth, 2002). Different environmental conditions across the tidal range result in patterns of vertical and horizontal (along the shore) zonation both in terms of species diversity and in terms of intraspecific phenotypic variation (Connell, 1972; Little & Kitching, 1996; Raffaelli & Hawkins, 1996; Southward, 1957). Wave action is one of the most important physical selective agents across intertidal environments worldwide, shaping both axes of zonation (Le Pennec et al., 2017). Habitats exposed to, or sheltered from waves differ consistently in their biodiversity (Denny & Wethey, 2001; Helmuth & Denny,

2003). Biotic factors are also known to affect the intertidal community (e.g. presence of predators; Paine and Fenchel 1994, which adds to the abiotic selective pressures (Seeley, 1986). Altogether, these environmental gradients make the intertidal a natural laboratory to study local adaptation and ecological speciation.

Taxa with low dispersal, where divergent selection can be strong enough to counteract gene flow among populations and promote divergence with gene flow, are particularly well suited for studies about local adaptation and ecological speciation (Sanford & Kelly, 2011; Smadja & Butlin, 2011). Among these, species with distinctive phenotypes associated with different microhabitats (i.e. ecotypes) in the intertidal can provide important information about how natural selection influences biological diversification (e.g. Coyer et al. 2011; Kess, Galindo, and Boulding 2018; Sanford and Kelly 2011; Wilding, Butlin, and Grahame 2001). Instances of parallel evolution of ecotypes across similar environmental gradients in multiple locations across a species' distribution, that is parallel evolution, are viewed as support for a role of natural selection in driving divergence. This is because it is unlikely that ecotypes have repeatedly evolved in the same phenotypic direction if only purely stochastic processes were involved (K. Johannesson, 2001; Nosil, 2012; Schluter, 2000).

The characterization of the genetic variation underlying parallel evolution allows to distinguish if repeated events of phenotypic divergence tend to involve the same de novo mutations, different de novo mutations in the same or different genomic regions, ancestrally shared standing polymorphisms, the same alleles due to migration between populations or a combination of all the above (Elmer & Meyer, 2011; Faria et al., 2014; K. Johannesson et al., 2010; Nosil, 2012). Thus, the study of parallel phenotypic divergence across intertidal microhabitats can help us understand if adaptation usually involves the same genetic paths, as well as the relative contributions of ancestral polymorphism and/or gene flow.

Parallel ecotypic divergence in the intertidal zone has been particularly well documented in the rough periwinkle *Littorina saxatilis* (Gastropoda; Galindo and Grahame 2014; Reid 1996; Rolán-Alvarez, Austin, and Boulding 2015). Two main ecotypes have been described: a large, thick-shelled ecotype inhabiting sheltered microhabitats that faces intense crab predation (Crab ecotype); and a small, thin-shelled one facing heavy surf in exposed microhabitats (Wave ecotype) (reviewed in K. Johannesson et al. 2010). These ecotypes, found only tens of metres apart in Spain, the United Kingdom (UK) and Sweden, have diverged in parallel within each of these countries (Butlin et al., 2014). Amplified fragment length polymorphisms (AFLPs) allowed the initial identification of loci under divergent selection (hereafter "outliers") between ecotypes and provided the first insights on the proportion of outliers shared at different geographic scales, within and among countries

(Galindo, Martínez-Fernández, Rodríguez-Ramilo, & Rolán-Alvarez, 2013; Galindo, Morán, & Rolán-Alvarez, 2009; Grahame, Wilding, & Butlin, 2006; Hollander, Galindo, & Butlin, 2015; Wilding et al., 2001). Benefiting from the assembly of a reference genome and the construction of a genetic map for *L. saxatilis*, the heterogeneous genomic differentiation between ecotypes was recently confirmed with the identification of some genomic regions enriched for the presence of outliers, which tend to coincide with polymorphic inversions (Faria et al., 2019; Westram et al., 2018). Moreover, whole-genome sequencing of pools of individuals from multiple populations of both ecotypes across the species' distribution range revealed that outlier sharing tends to be high even among distant populations, although it decreases with the geographic distance between populations (Morales et al., 2019).

A closely related species for which phenotypic variation is also found associated with an environmental cline in wave exposure is *Littorina fabalis* (Kemppainen, Nes, Ceder, & Johannesson, 2005; Reimchen, 1981; Tatarenkov & Johannesson, 1994, 1998, 1999). Individuals with large and thick shells (hereafter "large ecotype") are commonly found in moderately exposed shores, whereas individuals with smaller and thinner shells (hereafter "dwarf ecotype") predominate in sheltered habitats. Contrary to *L. saxatilis* that live on rocks, these two *L. fabalis* ecotypes dwell on brown macroalgae (*Fucus* spp. and *Ascophyllum* spp.), grazing on the epiphytes that grow on the algae fronds (Williams, 1990). The fronds are also thought to provide refuge against one of their main predators, the green crabs (*Carcinus maenas*), that are found in both microhabitats (Kemppainen et al., 2005).

Habitat-related variation in one allozyme locus (arginine kinase, *Ark*) was initially found in Swedish populations of *L. fabalis*, suggesting that this locus was under the influence of natural selection related to wave exposure and/or other associated factors (Tatarenkov & Johannesson, 1994). The differences in *Ark* allele frequencies between sheltered and moderately exposed habitats were also associated with variation at a random amplification of polymorphic DNA (RAPD) locus and with the size differences described above. This is true even for sites with intermediate exposure, suggesting some reproductive isolation between the ecotypes despite gene flow (K. Johannesson & Mikhailova, 2004; Tatarenkov & Johannesson, 1998).

Similar habitat-related phenotypic divergence has also been observed at least in the UK (Wales), France and Norway, where contrary to Sweden, *L. fabalis* also has to face high tidal amplitudes (Kemppainen, Lindskog, Butlin, & Johannesson, 2011; Kemppainen, Panova, Hollander, & Johannesson, 2009; Reimchen, 1981; Tatarenkov & Johannesson, 1999). Even if a northern refugium could have existed during the last glacial maximum (LGM) for *L. fabalis* (Sotelo et al., 2020), most of these shores were likely colonized after the LGM (Charbit, Ritz, Philippon, Peyaud, &

Kageyama, 2007). This, together with the absence of significant genetic differentiation between the ecotypes across this region using neutral markers (Sotelo et al., 2020), suggests a relatively recent (after the LGM) local establishment of habitat-related phenotypic divergence.

Contrary to neutral markers, *Ark* intron sequencing revealed highly significant divergence between individuals from sheltered and moderately exposed habitats (Kemppainen et al., 2011). Results show that one haplotype was almost fixed and shared across sheltered habitats of these different countries whereas wave-exposed habitats maintained similarly higher variation. This increases the possibility of "evolution in concert", where locally adapted alleles could arise once and subsequently spread to geographically distant populations inhabiting the same habitat by means of ecotype-specific selective sweeps (K. Johannesson et al., 2010; Kemppainen et al., 2011; Schluter, 2009). However, except for *Ark* and the putatively linked RAPD locus, whether the same genetic variation is involved in the evolution of these ecotypes is currently unknown. Thus, studies that integrate both morphological data and a high number of nuclear markers from different locations are needed to evaluate the parallelism of phenotypic divergence in *L. fabalis* across its geographic range, as well as the genetic variation and processes involved.

In this study, we used shell geometric morphometrics and AFLPs to perform the first characterization of *L. fabalis* ecotypes across multiple Northern European populations (Norway, Sweden and the UK) with four main goals: (a) to understand if shell shape and size divergence evolved in the same direction among locations within the same country and across countries; (b) to identify loci involved in the differentiation between sheltered and exposed sites (i.e. outlier loci); (c) to quantify the degree of outlier sharing at two different geographic scales: within and among countries; and (d) to contrast population structure and relatedness using outlier (putatively adaptive) versus nonoutlier loci (putatively neutral) in order to gain insights about the evolutionary history of phenotypic divergence. Although not fully conclusive, our results are compatible with phenotypic parallel divergence within *L. fabalis* associated with habitat and reveal a pattern of outlier sharing that suggests a relevant role of gene flow and/or retention of ancestral polymorphism in the evolution of ecotypes.

2.3 MATERIAL AND METHODS

2.3.1 SAMPLING

Littorina fabalis individuals from both sheltered and moderately exposed habitats were randomly collected with respect to their shell morphology between August and October 2012, the sample size in each population varied between 21 and 72 (average $N = 39$) (Table A2.1). A nested sampling design was implemented: samples were collected from each habitat within each location with replicates at two geographical scales: regional (two to three locations within each country, < 50 km) and global (different countries, > 1000 km) (Figure 2.1, Figure A2.1, Table A2.1). This allows the investigation of parallel evolution at these two contrasting scales.

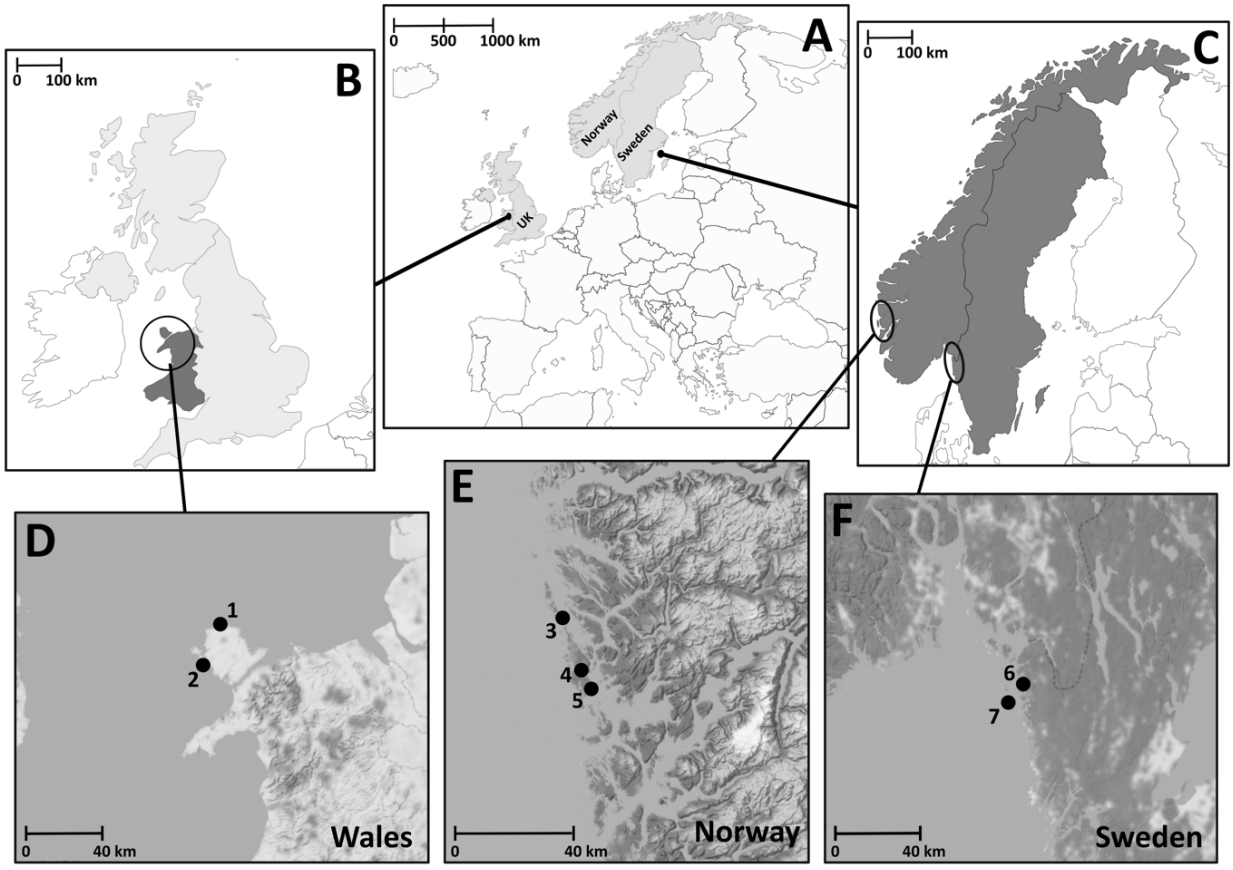


Figure 2.1: Sampling locations. (A) Map of Europe highlighting the three countries where populations of Littorina fabalis were sampled: (B) Wales, United Kingdom (UK), (C) Norway and Sweden. Zoom in showing the sampling locations in Wales (D), Norway (E) and Sweden (F): 1 - Anglesey North and 2 - Anglesey South, in Wales; 3 - Seløyna, 4 - Syltøyna and 5 - Hummelsund, in Norway; 6 - Lökholmen and 7 - Ursholmen, in Sweden.

Within each location, sites representing each type of habitat were preselected based on information from previous studies (Kemppainen et al., 2009; Tatarenkov & Johannesson, 1994), or on

their orientation and topography as retrieved from the Google Earth Engine (Gorelick et al., 2017). This preclassification was further confirmed in situ, where sheltered habitats were distinguished from moderately exposed ones by the high abundance of *Ascophyllum* spp., a macroalga that is commonly used as an indicator of sheltered shores (Tatarenkov & Johannesson, 1998). However, two putatively exposed sites did not adjust to these patterns: those in Anglesey South (UK) and in Ursholmen (Sweden) (Table A2.1). In Anglesey South, *Ascophyllum* spp. was only abundant in the upper part of the shore but individuals were collected from the lower part, where wave action is likely rather high. In Ursholmen, the moderately exposed site was chosen based on its orientation towards relatively open sea but *Ascophyllum* spp. was highly abundant there. The site was still included in the study but re-classified as sheltered (Table A2.1). Since we were not able to sample a proper moderately exposed site in Ursholmen, this location was excluded from statistical tests on shell morphology. It is also important to emphasize that the distance between habitats within each location differs among countries. In the Scandinavian locations, snails are continuously distributed along the shore from sheltered to moderately exposed sites (i.e. parapatric), which are < 500 m apart. In the UK, the two habitats within each sampled location are geographically isolated from each other, from 400 m to 8 km apart (Figure 2.1, Figure A2.1). After collection, individuals were brought alive to the laboratory where they were frozen at -20°C and then stored in 95% ethanol.

2.3.2 SAMPLE PROCESSING AND CLASSIFICATION INTO SPECIES

Shells of adult individuals were photographed over graph paper (used for scale) in a standardized position following Carvajal-Rodríguez, Conde-Padín, and Rolán-Alvarez (2005) and using a digital ICA video module fitted on a MZ12 dissection microscope (Leica) for subsequent geometric morphometric analyses. Shells were then broken, and individuals dissected and sexed under the same microscope. Since shell morphology is not completely diagnostic between the two closely related species of flat periwinkles (*L. fabalis* and *L. obtusata*), we could not guarantee that all collected individuals were indeed *L. fabalis* just based on their shells. Thus, we focused our analysis on males, which were classified into *L. fabalis* or *L. obtusata* based on the morphology of their genitalia, one of the most distinctive traits between the two species (Costa et al., 2020; Reid, 1996). In sites where the number of males was too low, females and immature individuals were also included. Nevertheless, all samples (including males) were later classified into species using the AFLP genotypes. To do so, 17–19 individuals from *L. obtusata* populations within each country were deliberately included in the genetic analysis as references.

2.3.3 SHELL GEOMETRIC MORPHOMETRICS

Geometric morphometric (GM) analyses were carried out following the methodology previously developed for flat periwinkles by Costa et al. (2020). This consisted in digitizing a total of 28 landmarks for each shell of adult individuals (males and females classified as *L. fabalis*), including 4 fixed and 24 sliding semilandmarks, using TPSDIG v1.40 (Rohlf, 2006).

Superimposition of landmark coordinates was performed using generalized Procrustes analysis (GPA; Rohlf and Slice 1990), retrieving shape variables (i.e. Procrustes residuals) and centroid size (CS), the latter used as an estimate of size for each individual. The final dataset consisted of 318 adult individuals (confirmed to be *L. fabalis* based on genitalia morphology and/or genetics), representing all seven locations.

A principal component analysis (PCA) of shape variables was conducted to assess overall patterns of variation. In addition, UPGMA dendrograms for size and shape were generated to evaluate clustering patterns between samples from each habitat and location, based on Euclidean distances (for size) and Procrustes distances (for shape) of population means. General linear models (GLMs) were then used to assess if shell shape and size (logCS) differed significantly between the two sampled habitats and to evaluate whether these differences varied among locations and countries. As such, GLMs included country, habitat and sampling location (nested within country) as main effects, as well as all interaction terms. Allometric effects were also investigated by performing a GLM on shape with logCS as a covariate, using the same factorial design. These analyses were carried out in the geomorph R package (Adams, Collyer, & Kaliontzopoulou, 2019), using randomized residual permutation procedures (RRPP) of 1,000 permutations and Z-scores for significance assessment (M. Collyer & Adams, 2019; M. L. Collyer & Adams, 2018). Deformation grids were used to visualize differences in shape between *L. fabalis* individuals from the two habitats within each location.

Finally, a discriminant function analysis (DFA) based on shape was implemented using the R package MASS v7.3.50 to infer the probability of morphological assignment of individuals into moderately exposed or sheltered habitats. The DFA was constructed based on the 287 individuals from all locations under study except for Ursholmen, using a leave-one-out cross-validation procedure. The resulting morphological posterior probability (PP) assignments were then compared with the genetic membership coefficients obtained in STRUCTURE (see below) for the 83 individuals for which both genetic and phenotypic data were available. All morphological data analyses where carried out in the R language for statistical computing (Team, 2019).

2.3.4 GENETIC ANALYSES

Genomic DNA was extracted using the CTAB–chloroform protocol described in Galindo et al. (2009). DNA quantity and purity were assessed with a BioDrop μlite (BioDrop) spectrophotometer and adjusted to a final concentration of 20 $ng/\mu l$ for each sample.

The AFLP analysis was performed using a modified version of the protocol developed for *L. saxatilis* (Galindo et al., 2009). Briefly, 100 ng of DNA was digested in a final volume of 12 µl with 4U EcoRI (New England Biolabs, NEB) and 2U MseI (NEB) in 1X Buffer EcoRI (NEB) supplemented with 0.03 µg of BSA. After a 3.5-hr incubation at 37°C, ligation of Eco and Mse adaptors was done by adding 3 µl of a solution with 5 pmol of each adaptor and 0.25 U T4 DNA ligase (Roche) in 1X ligase buffer. Samples were incubated for 16 hr at 16°C. In digestion and subsequent steps, all samples were randomly distributed across 96-well plates, and 15% of replicate samples were included. The preselective PCR was performed in 10 µl final volume containing 2 µl of a 1:4 dilution of the ligation product, 0.3 mM of dNTP mix (Applied Biosystems), 2 mM of MgCl2, 5 pmol of each preselective primer (Eco + A, Mse + C) and 0.3 U of Taq polymerase (Bioline) in 1X PCR Buffer. The first selective PCR was performed on 1 µl of 1:4 dilution of the preselective PCR using the same reaction mixture but with the addition of 4 pmol Eco + ACT (FAM labelled), 2.5 pmol Eco + AAG (NED) and 5 pmol Mse + CAA primers. A second selective PCR was performed with Eco + ACT, Eco + AAG and Mse + CAC. PCR conditions, adaptor and primer sequences are described in Galindo et al. (2009). Selective PCR products were analysed on an ABI 3130 sequencer (Applied Biosystems) at CACTI (Centro de Apoyo Científico y Tecnológico a la Investigación, University of Vigo) along with GeneScan 500ROX (Applied Biosystems). Electropherograms were initially analysed with PeakScanner v.2.0. Loci were manually assigned by defining bins from the overlapping electropherograms of all the samples in RawGeno v.2.0 (Arrigo, Holderegger, & Alvarez, 2012). The error rate (9% on average overall primer combinations) was estimated with the R package AFLPtools (`https://github.com/genevalab/AFLPTools`) that follows AFLPScore methodology (Whitlock, Hipperson, Mannarelli, Butlin, & Burke, 2008). The final genotypes were obtained using the same package.

In order to identify *L. obtusata* individuals that could have been erroneously classified as *L. fabalis* based on shell appearance (prior GM analyses), AFLP-SURV v.1.0 (Vekemans, Beauwens, Lemaire, & Roldán-Ruiz, 2002) was used to calculate 1-relatedness coefficient (Lynch & Milligan, 1994) matrix between pairs of individuals, and then, a multidimensional scaling analysis (MDS) was implemented in R (Team, 2019). The individuals that clustered with the *L. obtusata* individuals were removed from all morphological and genetic analyses, resulting in a final dataset formed by 503 *L. fabalis* individuals.

2.3.5 DETECTION OF OUTLIER LOCI

Two different methodologies were used to identify outlier loci between *L. fabalis* exposed and sheltered habitats, BAYESCAN v.2.1 (Foll & Gaggiotti, 2008) and DFDIST (Beaumont & Nichols, 1996). In both cases, the analyses were carried out independently within each location. As previously mentioned, both Ursholmen sites were likely sheltered. Thus, we do not expect to detect outliers related to the level of wave-exposure or associated factors. Nevertheless, outliers between the two sites were still estimated as a control.

BAYESCAN calculates population-specific and locus-specific F_{ST} coefficients and then estimates the posterior probabilities of two alternative models (including or excluding the effect of selection) for each locus using a reversible-jump Markov chain Monte Carlo (MCMC) approach. Ten pilot runs (10,000 iterations) were performed to tune the model parameters, followed by 400,000 iterations (100,000 as burn-in, 20 as thinning interval and 20,000 as sample size). Loci were identified as outliers when the posterior probability was higher than 0.75 (equivalent to a Bayes factor of 3 or greater) (see Foll and Gaggiotti 2008). DFDIST simulates loci (200,000) under a neutral model of two symmetrical islands with a mean F_{ST} adjusted to the empirical F_{ST} between the ecotypes calculated from the AFLP loci. Those loci with F_{ST} values significantly greater than the simulated neutral distribution (F_{ST} conditional on heterozygosity) were considered as outliers using two different stringency criteria ($p > .95$ and $p > .99$). This analysis was conducted using the same parameters as Galindo et al. (2009).

Simulation studies that compare BAYESCAN to other methodologies for outlier detection, including DFDIST, have shown that BAYESCAN is more conservative concerning the number of outliers detected (Pérez-Figueroa, García-Pereira, Saura, Rolán-Alvarez, & Caballero, 2010). In this study, all BAYESCAN outliers were consistently included within the set of DFDIST outliers ($p > .99$). Because the goal of this study was to focus on the genetic structure of outlier versus nonoutlier loci (Galindo et al., 2009) rather than follow-up specific outlier loci (e.g. Wood, Grahame, Humphray, Rogers, and Butlin 2008), we carried out subsequent genetic analyses using the set of DFDIST outliers. Since the number of outliers (using the $p > .99$ cut-off) within some locations was relatively low (see results), substructure analyses were based on a more lenient criterion ($p > .95$). However, a more stringent criterion ($p > .99$) was used to compare the amount of shared outliers among locations (conservative). On the other hand, in order to be conservative, we considered nonoutlier loci only those with $p < .80$ in the DFDIST analysis.

2.3.6 GENETICS SUBSTRUCTURE ANALYSIS

Genetic substructure analyses were performed with these two different sets of loci, all outliers (i.e. putatively under divergent selection) and nonoutliers (i.e. putatively neutral). In both cases, they represent the combination of loci detected in each locality. AFLP-SURV v.1.0 (Vekemans et al., 2002) was used to calculate population pairwise genetic differentiation (F_{ST}) and Nei's genetic distance (10,000 bootstrap) using Zhivotovsky (1999) Bayesian approach. Neighbour-joining (NJ) trees were constructed based on Nei's genetic distance using the NEIGHBOR routine implemented in the PHYLIP package (Felsenstein, 2013). The CONSENSE routine in PHYLIP was used to determine the bootstrap percentage supporting each branch of the tree. Trees were visualized using FigTree v.1.4.4 (`http://tree.bio.ed.ac.uk/software/figtree/`). A Bayesian clustering analysis was carried out in STRUCTURE v.2.3.4 (Pritchard, Stephens, & Donnelly, 2000) under three different hierarchical levels: (a) global, including all sites; (b) regional, including all sites within a country; and (c) local, including both the sheltered and exposed site within a location. All analyses were performed with the outlier and nonoutlier data sets, considering only as outliers those detected within the respective hierarchical levels (i.e. for the global analysis using all detected outliers in all pairwise comparisons, for the regional analysis only those detected in the pairwise comparisons within the corresponding region/country, whereas for the local analysis using only the outliers detected in that specific location). The consistency of individuals' assignment based on outliers across the different hierarchical levels was assessed for each location using correlation tests (Pearson correlation coefficient (r) - Sokal and Rohlf 1995), implemented in the R package Stats v.3.6.1. For each K-value, we analysed five replicate runs of 500,000 iterations (100,000 as burn-in), assuming an admixture model, correlated allele frequencies and without prior population information. All analyses were carried out from K = 1 up to K = number of sites plus one, depending on the hierarchical level analysed. The method developed by Evanno, Regnaut, and Goudet (2005) implemented in STRUCTURE HARVESTER was employed to determine the best K.

2.4 RESULTS

2.4.1 CHARACTERIZATION OF SHELL MORPHOLOGY AND PHENOTYPIC DIVERGENCE

Adult individuals from each habitat differed significantly in size, with individuals from moderately exposed habitats being consistently larger than those from sheltered habitats (Figure 2.2, Table

A2.2). Accordingly, the UPGMA dendrogram based on centroid size (CS) revealed two main groups (Figure 2.2b): individuals from sheltered habitats from all locations on one hand, and all moderately exposed ones on the other except Syltøyna (Norway), which clustered within the sheltered group. Within each group, there was no particular geographic trend except for the clustering of all Norwegian sheltered habitats together. Other significant factors affecting size revealed by the general linear model (GLM) analysis included location and the interaction between habitat and location, suggesting that size differences between habitats varied across locations. However, size differences were not significant between countries (Table A2.2).

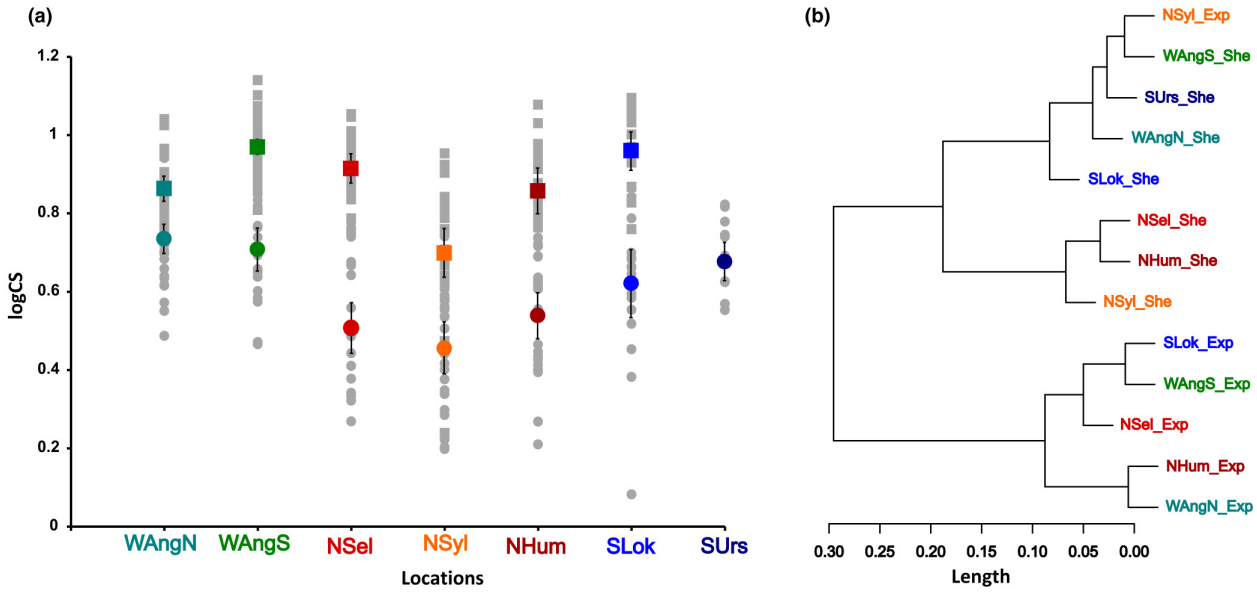


Figure 2.2: Results of the geometric morphometric (GM) analyses of shell size. (a) Mean size (logCS) of individuals for each habitat across locations. Mean values are indicated by coloured symbols (vertical bars denote 95% confidence intervals), whereas individuals are represented in grey; square and circle symbols represent the moderately exposed and sheltered habitats, respectively. (b) UPGMA dendrogram based on Euclidean distances of mean shell size. For Ursholmen, only samples of the sheltered habitat were included (see main text). Population codes are the same as in Table A2.1 and Figure A2.1

The same factors influencing size also influence shape variation, as reflected by significant effects of habitat, location and their interaction, but not of country (Table A2.2). When considering allometric effects, we found a significant influence of size on overall shape. In addition, size seems to affect shape differently across locations (significant interaction between both effects) (Table A2.2). Taking this effect of size into account, the same effects of habitat, location and their interaction remained significant, suggesting that although size accounts for some of the observed shape variation, it is not the only factor influencing it.

The distribution of individuals in the morphospace of the two principal components of shape variation, which explained a total of 69.4% of variation (PC1: 42.1%; PC2: 27.3%), revealed that

average shapes of individuals from the moderately exposed habitats tend to cluster in the lower-left quadrant, whereas the average shape of individuals from the sheltered habitats is found in the upper-right part, with the exception of Anglesey South (Figure 2.3a). This agrees with the GLM results that show habitat as the factor with the strongest effect (based on Z-scores) on shape variation (as well as on size), followed by location (Table A2.2). The UPGMA dendrogram based on shape revealed two main groups, one consisting on the moderately exposed habitat from Scandinavian locations except Syltøyna (as for size); and the second comprising the sheltered habitats from Scandinavian locations and Syltøyna as well (Figure 2.3b).

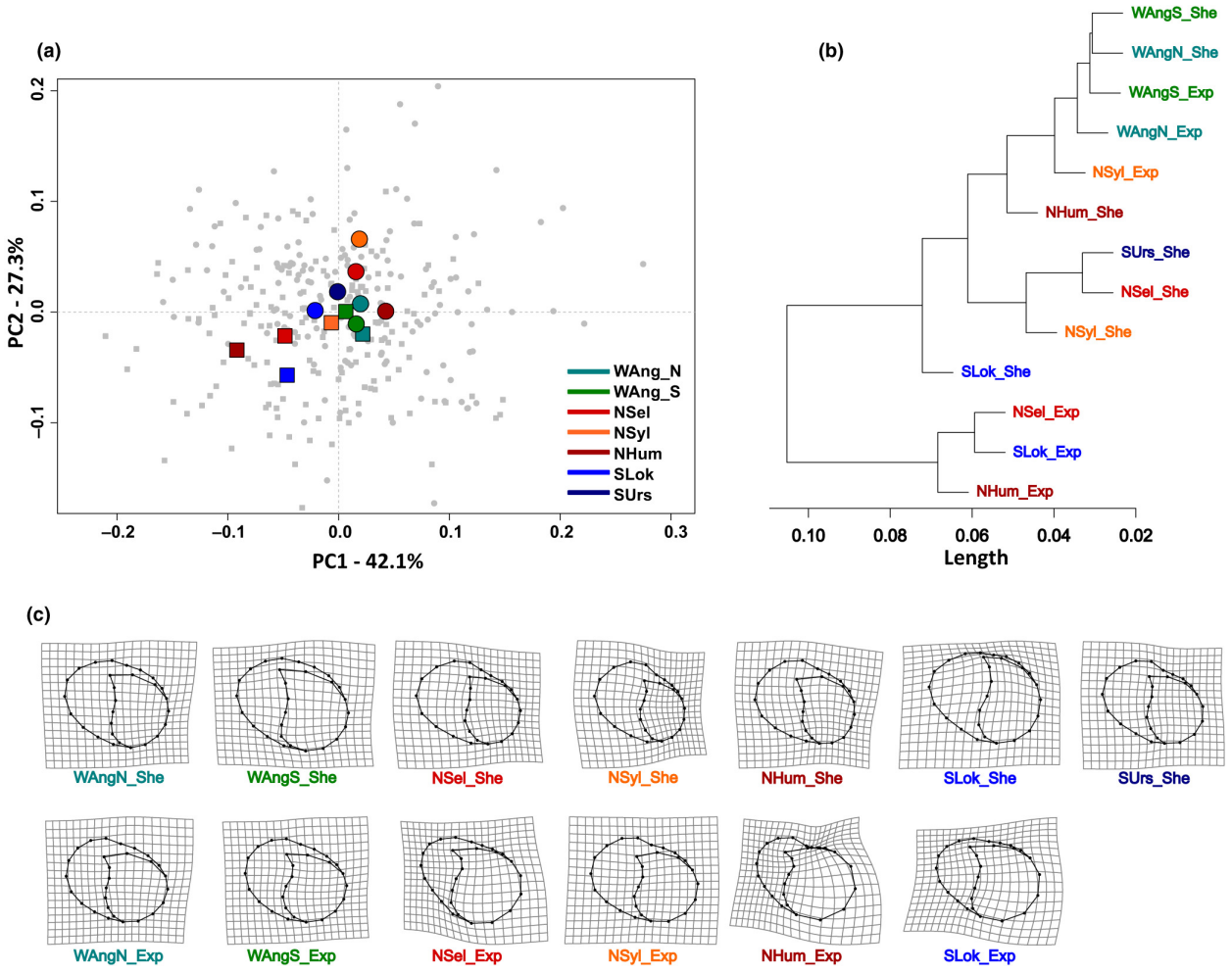


Figure 2.3: Results of the geometric morphometric (GM) analyses of shell shape. (a) Mean shape of individuals of each habitat and location based on the two first principal components of the PCA, where mean values are indicated by coloured symbols, whereas individuals are represented in grey; square and circle symbols represent the moderately exposed and sheltered habitats, respectively. (b) UPGMA dendrogram based on the Procrustes shape distances. (c) Deformation grids depicting the average shape of individuals of each habitat and location compared to the global mean. Mean shapes were magnified x2 to improve visualization. For Ursholmen, only samples of the sheltered habitat were included (see main text). Population codes are the same as in Table A2.1 and Figure A2.1

Contrary to what was observed in terms of size, although populations within the UK clustered by habitat rather than by location, they were nested within the clade composed mostly by sheltered sites from other countries, suggesting also important geographic influence on shape, in accordance with GLM results (Table A2.2). The inspection of deformation grids depicting the mean shape for individuals from each habitat (Figure 2.3c) shows that individuals inhabiting the moderately exposed habitat generally exhibit a larger width of the aperture (relative to overall shell size), whereas individuals of the sheltered habitat tend to exhibit a smaller aperture.

2.4.2 DETECTION OF OUTLIER LOCI AND COMPARISON AMONG LOCATIONS

A total of 746 AFLP loci were analysed in 299 individuals (average sample size per population of 21) (Table A2.1). The lowest proportion of outliers was detected in the Swedish site Ursholmen (1.2%–5.3% for $p > .99$ and $p > .95$, respectively); and none of these outliers were shared with other locations (for $p > .99$), not even with the other Swedish location (Lökholmen, $\sim$ 10 km apart) (Table 2.1). This suggests that indeed the populations from the two Ursholmen sites did not (or poorly) diverge according to the same axis of divergence (sheltered vs. moderately exposed) as in other locations, in agreement with the abundance of *Ascophyllum* spp. in both sampled habitats.

Table 2.1: Results of the outlier detection analysis performed with BAYESCAN and DFDIST at different significance levels ($p > .99$, $p > .95$) for each location. The number of outlier loci and percentage (between brackets) over the total number of loci analysed is shown. The number and percentage of nonoutlier (putatively neutral) loci (see Methods) is also shown. Between ecotype F_{ST} values were calculated in AFLP-SURV for overall loci, with all outlier 99 ($p > .99$), all outlier 95 ($p > .95$), and all nonoutlier ($p < .80$).

Location	FST overall	BAYESCAN	DFDIST N outlier 99	FST outlier 99	DFDIST N outlier 95	FST outlier 95	DFDIST N nonoutlier	FST nonoutlier
WAngN	0.0469	2 (0.4)	26 (4.7)	0.1886	53 (9.5)	0.1488	437 (78.6)	0
WAngS	0.0369	5 (0.9)	16 (2.8)	0.1699	36 (6.4)	0.1114	454 (80.2)	0
NSel	0.0557	8 (1.5)	23 (4.2)	0.2304	41 (7.5)	0.1581	434 (79.2)	0
NSyl	0.0363	2 (0.4)	23 (4.1)	0.1425	42 (7.5)	0.1063	442 (79.2)	0
NHum	0.0805	9 (1.7)	29 (5.6)	0.2922	48 (9.2)	0.2079	422 (81.0)	0
SLok	0.0208	0	8 (1.6)	0.0895	31 (6.3)	0.0682	420 (84.7)	0
SUrs	0.0109	0	6 (1.2)	0.0364	27 (5.3)	0.0410	414 (80.9)	0

Excluding Ursholmen, a total of 1.6%–5.6% of outliers ($p > .99$) and 6.3% – 9.5% of outliers ($p > .95$) were detected across all locations using DFDIST (Table 2.1, Figure A2.2). Using the most conservative threshold ($p > .99$), the number of outliers per location ranged from 6 in Lökholmen (Sweden) to 29 in Hummelsund (Norway) with the F_{ST} between habitats for these loci ranging from 0.0895 to 0.2922, respectively. This contrasts with lack of differentiation for nonoutlier loci (Table 2.1). In total, only three outliers (3.8%) are shared among all countries whereas the highest number

of outliers are shared between locations within each country (Figure 2.4), with Norway presenting the highest values (between 8 and 14 outliers, 21% and 36.8%, respectively) followed by the UK (7 outliers, 20%) (Table 2.2). The number and proportion of shared outliers between locations of different countries were generally lower, ranging from 2 (6.9%) between Lökholmen (Sweden) and Syltøyna (Norway) to 7 (14.6%) between Anglesey North (UK) and Hummelsund (Norway; Table 2.2).

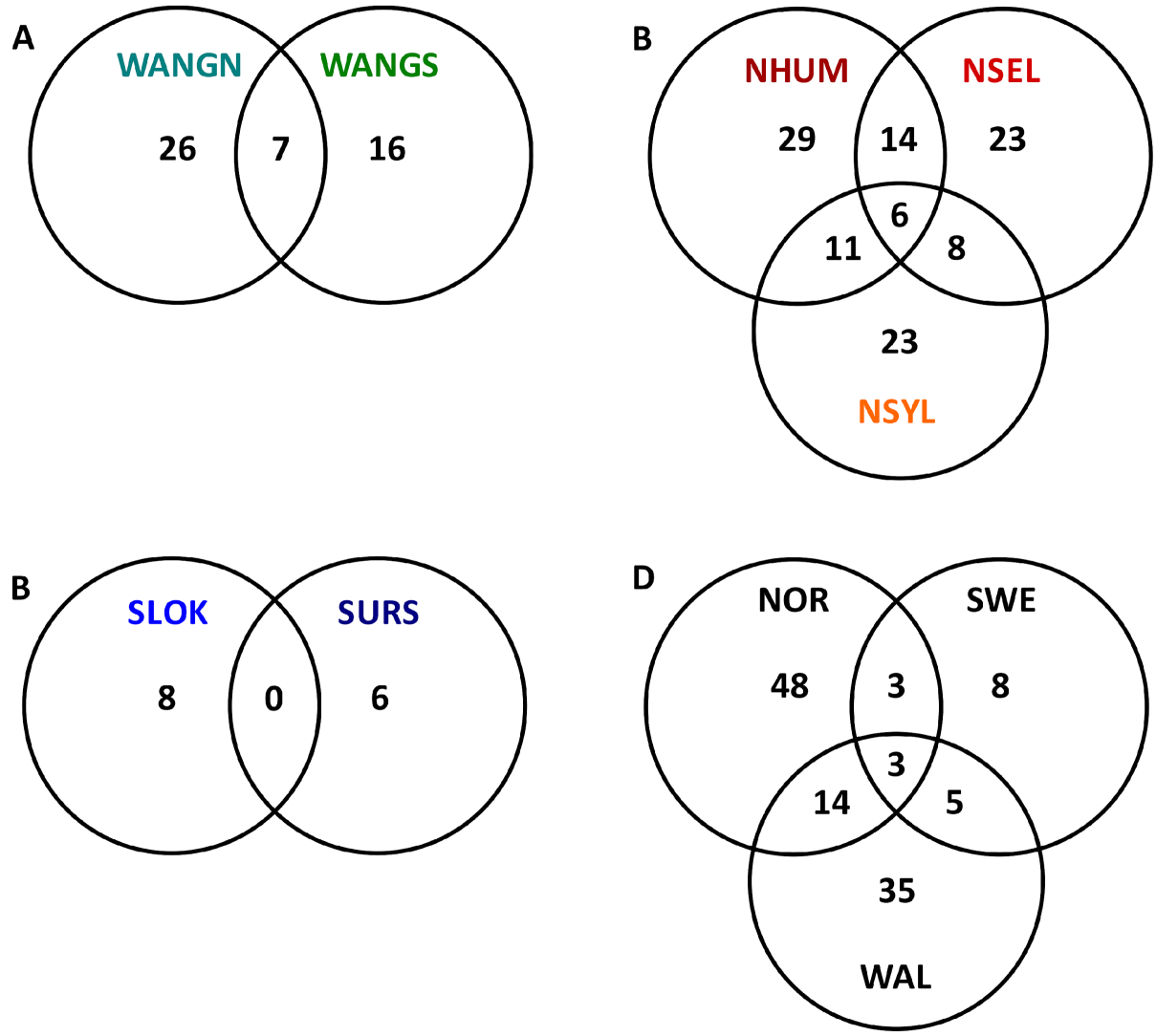


Figure 2.4: Number of shared DFDIST outliers ($p > .99$) between locations for Wales, United Kingdom (A), Norway (B), Sweden (C); and between countries (taking into account all the outliers across locations within a country) (D), except Sweden for which only the outliers detected in Lökholmen were considered (see main text). The total number of outliers within locations is represented inside the circles (including the shared ones)

In total, 147 outliers ($p > .95$) were detected across locations; this is combining the outlier loci that were detected between habitats within each location. The NJ tree based on these 147 outliers

shows that the populations group first by habitat and only then by country within each habitat (Figure 2.5a). The two Ursholmen populations cluster together within the "sheltered" clade confirming the absence of a moderately exposed site within this location in our dataset. Consequently, this site was excluded from the discussion on the main patterns revealed by this study, except when otherwise specified.

The STRUCTURE analyses for outlier loci rendered $K = 2$ as the most likely number of genetic clusters regardless of the geographic scale (global, regional and local; see Methods and Figure 2.6a). The results are similar across all three levels (correlation between the individuals' membership (r) $> 0.92, p < 2.2e^{-16}$), showing that the two genetic clusters roughly coincide with the two habitats (with the exception of Ursholmen, (r) $< -0.02, p > .76$ between local memberships and the two other levels). However, some individuals present a genetic composition that is typical of the opposite habitat where they were sampled, whereas others are genetically admixed between the two clusters, suggesting migration and interbreeding between individuals from the two habitats.

Table 2.2: Number of shared DFDIST outliers ($p > .99$, below diagonal; $p > .95$, above diagonal) between locations. In brackets, the percentage of shared outliers from the total number of outliers in the pairwise comparison.

Location	**WAngN**	**WAngS**	**NSel**	**NSyl**	**NHum**	**SLok**	**SUrs**
WAngN		16 (22.0)	11 (13.2)	11 (13.1)	13 (14.8)	10 (13.5)	9 (12.7)
WAngS	7 (20.0)		14 (22.2)	11 (16.4)	16 (23.5)	10 (17.5)	2 (3.3)
NSel	6 (13.9)	5 (14.7)		17 (25.7)	26 (41.3)	11 (18.0)	3 (4.6)
NSyl	4 (8.9)	3 (8.3)	8 (21.0)		20 (30.0)	9 (14.1)	3 (4.5)
NHum	7 (14.6)	5 (12.5)	14 (36.8)	11 (26.8)		10 (14.5)	6 (8.7)
SLok	3 (9.7)	2 (9.1)	2 (6.9)	3 (10.7)	3 (8.8)		5 (9.4)
SUrs	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	

2.4.3 GENETIC SUBSTRUCTURE BASED ON NONOUTLIER LOCI

In contrast, the NJ tree for nonoutlier loci (380 loci, combining the nonoutlier loci detected individually in each pairwise comparison) (Figure 2.5b) shows that the populations primarily group by country and only then by location. The STRUCTURE results for $K = 2$ based on nonoutliers also contrasted with those obtained based on outliers (Figure 2.6b). Instead of the two main clusters representing the habitats where the individuals were found, the two main clusters separate UK from Scandinavian populations (for $K = 3$) and only for $K = 4$ a split between Norwegian and Swedish populations is observed. A similar substructure was observed for $K = 7$ (Figure A2.3), the most probable number of clusters according to the Evanno method (not shown).

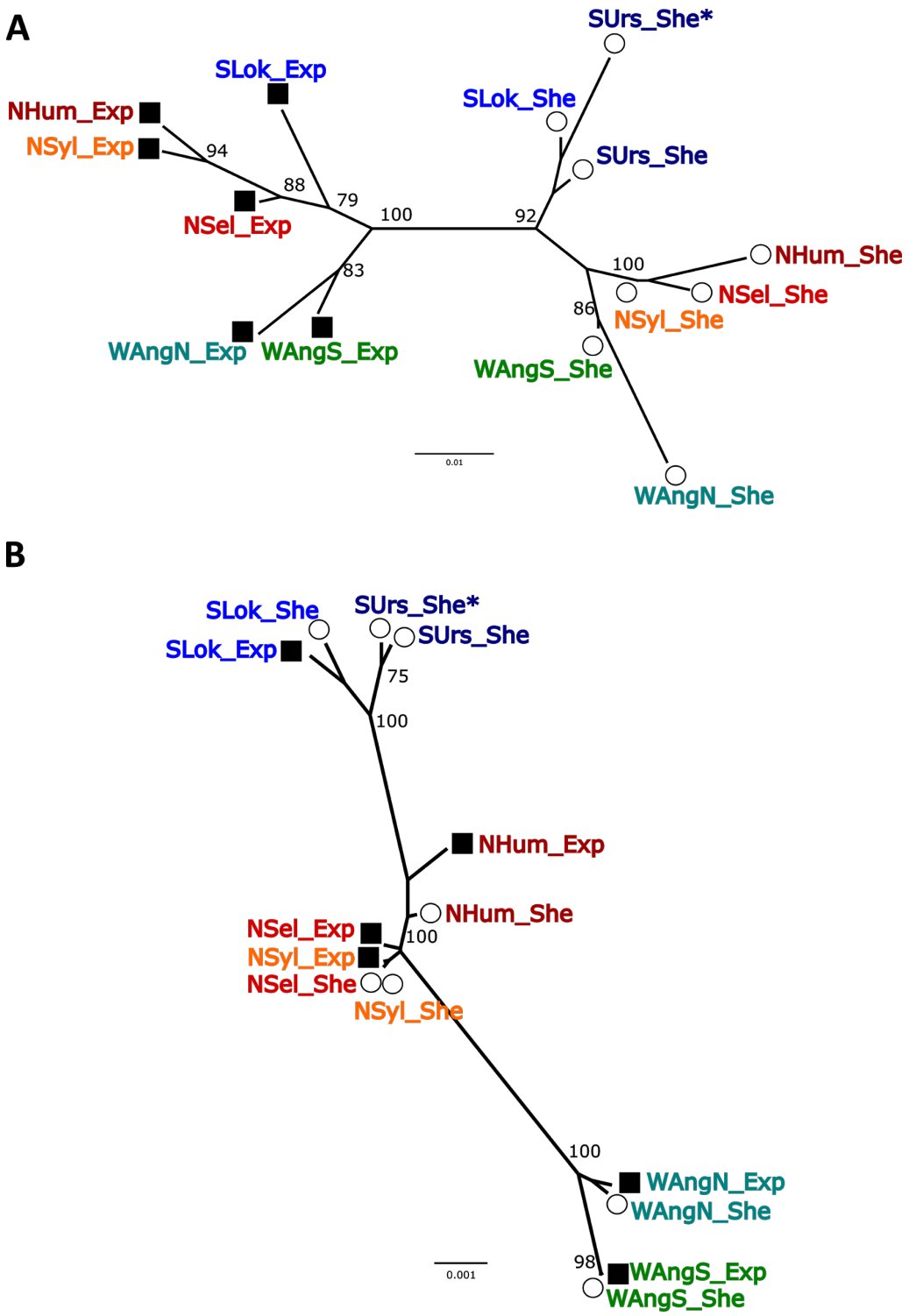


Figure 2.5: Neighbour-Joining trees based on Nei's genetic distance calculated with AFLP-SURV in two sets of loci: (A) outliers 95 ($p > .95$, DFDIST) (147 loci) and (B) nonoutliers ($p < .80$) (380 loci), in both cases combining the loci from all comparisons (see main text). Population codes are the same as in Table A2.1 and Figure A2.1. Dark squares represent populations from exposed sites, whereas white circles represent populations from sheltered sites. Only bootstrap support values above 70 are shown

Mean pairwise F_{ST} values show the lowest differentiation between populations from Norway and Sweden, followed by populations from Norway and UK and then by populations from Sweden and UK, in agreement with the geographic distance between countries (Figure A2.4A). The genetic differentiation between locations is relatively higher in UK (Figure A2.4B), followed by Sweden and Norway. Within each country, the populations from sheltered habitats showed higher differentiation than the exposed ones, except in Norway where F_{ST} between Syltøyna and Seløyna is close to zero in both sheltered and moderately exposed habitats (Figure A2.4D). Pairwise F_{ST} between populations living in different habitats within each location was zero, whereas this was generally not the case between populations from the same habitat among locations, apart from moderately exposed locations from Norway (Figure A2.4 and Table A2.3).

Using a PP ≥ 0.90 criterion to classify individuals as one of two ecotypes, 49 individuals over a total of 83 analysed for both genetics and shape show concordant classification between the two types of information, with the majority (96%, $N = 47$) of these samples assigned to the cluster/form typical of the habitat where they were sampled (Table A2.4). The proportion of concordance was similar between sheltered and moderately exposed habitats (59% and 55%, respectively), and although this varies substantially between sites, we refrain to draw any conclusions given the small sample sizes per site (from 1 to 11, Table A2.4). Some individuals that are genetically classified as pure for the typical sheltered cluster show a typical shape of the exposed habitat ($N = 9$), and vice versa ($N = 5$). However, most of these individuals ($N = 9$) belong to the genetic cluster typical of the habitat where they were sampled (Figure A2.5, top left and bottom right). Finally, genetically admixed individuals ($N = 14$) present a wide range of shell shape, with the majority ($N = 9$) showing a morphology typical of moderately exposed habitats.

2.5 DISCUSSION

Studies of systems like *L. fabalis*, where different ecotypes coexist across a species' distribution range, offer important insights about the genetic variation underlying traits involved in local adaptation and ecological speciation. Combining an AFLP genome scan for outlier loci with geometric morphometric analysis, we made the first attempt to detect genetic variation associated with divergent ecotypes in terms of wave-exposure and/or related ecological factors (environmental or biological) and to evaluate the level of sharing of these outlier loci between ecotypes across populations from the UK, Sweden and Norway.

2.5.1 THE ROLE OF NATURAL SELECTION ON PHENOTYPIC DIVERGENCE

In agreement with previous studies (Kemppainen et al., 2011, 2005; Tatarenkov & Johannesson, 1998), mean shell size was larger in moderately exposed than in sheltered sites (Figure 2.2a). The remarkable influence of habitat type in moulding size variation is evident in the size-based UPGMA dendrograms where moderately exposed sites (with a single exception) consistently cluster together, irrespective of their geographic origin (Figure 2.2b). Although shell shape variation was also influenced by geography, divergence between populations from the two contrasting habitats was significant across locations (Figure 2.3a). Thus, despite local specific phenotypic effects, we found consistent divergence in shell morphology between *L. fabalis* populations at the extremes of similar environmental transitions across localities from the same or different European regions. Although this could not be assessed in Ursholmen, observations from a moderately exposed site recently visited show that this pattern holds for this island too, at least for size (R. Faria, personal observation).

The patterns of phenotypic and genetic variation can be explained by both genetic and environmental factors. Although a plastic component cannot be excluded, the association between size/growth and genetic variation found by Tatarenkov and Johannesson (1998) in individuals originally from the same intermediately exposed habitat suggests an important heritable component. This is in agreement with variation in shell morphology being largely genetically inherited in other species within the genus, such as *L. saxatilis* (Conde-Padín, Caballero, & Rolán-Alvarez, 2009; Conde-Padín, Grahame, & Rolán-Alvarez, 2007; Galindo, Cacheda, Caballero, & Rolán-Alvarez, 2019; Hollander & Butlin, 2010; B. Johannesson & Johannesson, 1996; K. Johannesson, Rolán-Alvarez, & Erlandsson, 1997) and *L. subrotundata* (Boulding & Hay, 1993; Kyle & Boulding, 1998). Thus, shell divergence between habitats across multiple populations of *L. fabalis* strongly suggests local adaptation to different levels of wave exposure and/or related ecological factors. For example, a larger shell aperture is observed in exposed sites when compared with sheltered sites in *L. fabalis* (Figure 2.3c), which agrees with earlier findings in *L. saxatilis* (K. Johannesson et al., 2010). In contrast, *L. fabalis* from central and northern Europe are larger in moderately exposed habitats than in sheltered habitats (Reimchen, 1981; Tatarenkov & Johannesson, 1998), whereas the opposite is true in *L. saxatilis* (K. Johannesson et al., 2010). However, both the sheltered and moderately exposed habitats of *L. fabalis* correspond to the ”crab” sites of *L. saxatilis*, whereas the ”wave” sites of *L. saxatilis* are far more exposed and crab free.

A previous ecological study of the two *L. fabalis* ecotypes suggested that morphological differences are due to a complex interaction between wave exposure and the algae canopy inhabited by the snails, which provides shelter against crabs living in the boulders below (Kemppainen et al., 2005). According to this hypothesis, individuals with a larger and thicker shell would be favoured against predators in moderately exposed habitats where dislodgment is more likely. The shape patterns observed here are consistent with this hypothesis as the large shell aperture characteristic of individuals from moderately exposed habitats can accommodate a larger foot and protect them against dislodgement. In this sense, the moderately exposed ecotype seems to be adjusted for both higher predation (large size) and avoiding dislodgement (large aperture relative to body size), in contrast to *L. saxatilis* where the exposed ecotype does not need protection against crab predation and is only adapted to fit into crevices and cling tightly to the rocks to avoid wave dislodgement (reviewed in K. Johannesson et al. 2010).

Strikingly, the moderately exposed population of Syltøyna (Norway) groups with the sheltered populations when considering both size and shape (Figures 2.2b and 2.3b). This suggests that selection related with wave action is weaker in this site, which is supported by our observations in the field. Independently of the causes, this highlights that besides general habitat-related differences there are also important location-specific effects on shell morphology (Table A2.2).

2.5.2 HABITAT-RELATED GENETIC DIFFERENTIATION

As commonly found in other studies (reviewed by Ravinet et al. 2017), our results reveal heterogeneous differentiation between ecotypes, with a relatively small proportion of loci showing high levels of differentiation likely resisting the substantial gene flow that seems to be eroding differentiation to very low levels in the rest of the genome here assessed. However, an important caveat of genome scans is that the number (and proportion) of discovered outliers depends on how stringent is the cut-off used to detect them (Faria et al., 2014). Here, we used different methods, including BAYESCAN that is known to be conservative (Pérez-Figueroa et al., 2010), as well as different stringency levels to detect outliers. For instance, when considering the less stringent criterion and method (DFDIST, $p > .95$), we detected $\sim 7\%$ of outliers across locations, which is similar to that detected between other pairs of divergent ecotypes in a wide range of taxa (reviewed in Nosil, Harmon, and Seehausen 2009), including the Crab and Wave ecotypes of *L. saxatilis* (Butlin et al., 2014). Nonetheless, as with any genome scan, these outliers must be seen as a list of potential candidate loci influenced by divergent selection that need further confirmation by alternative approaches, as suggested by Ravinet et al. 2017.

Outlier loci identified by means of genome scans could also result from factors not related with local adaptation, such as background selection (Cruickshank & Hahn, 2014; Ravinet et al., 2017). However, given that F_{ST} between habitats based on nonoutliers is zero in all locations, it is very unlikely that background selection would be strong enough to originate peaks of F_{ST} under such substantial levels of gene flow. An important limitation of using AFLPs is that we cannot directly assess the degree of physical linkage between loci (Stapley et al., 2010). Thus, we do not know if these outliers are clustered within a single genomic region or are widespread across the genome, limiting our power to further interpret the differentiation heterogeneity. However, the consistent patterns of differentiation between habitats at different geographical scales (global, regional and local) suggest that the outliers we detected are indeed influenced by the same axis of selection across locations. The proportion of outliers and associated genetic differentiation vary across locations, with the highest F_{ST} and number of outliers in Hummelsund (Norway), followed by intermediate values in UK and lower differentiation in Sweden. Given the absence of significant genetic differentiation based on nonoutlier loci in all locations (Sotelo et al. 2020; this study), these patterns are compatible with differences in the strength of selection, which can result in geographic heterogeneity under migration–selection equilibria. Nevertheless, future studies of hybrid zones based on cline analysis of loci across the genome, like the one implemented by Westram et al. (2018), will be important to confirm the role of divergent selection in this system.

2.5.3 OVERLAP BETWEEN OUTLIERS ACROSS LOCATIONS

A small number of outliers (3, $p > .99$) were shared among all countries but these estimates of outlier sharing need to be interpreted with caution. Although the genome size of *L. fabalis* is currently unknown, data from closely related species (*L. obtusata* and *L. saxatilis*) suggest a genome size of 1.2 – 1.35 Gb (Animal Genome Size Database 2.0, `http://www.genomesize.com/`). Thus, the 746 AFLP loci we genotyped here confer relatively low resolution to assess both the number of outliers and the amount of sharing across locations with high precision when compared, for instance, with whole-genome sequencing. However, the number of shared outliers across the three countries using the most stringent criterion is higher than what would be expected just by chance (0). Furthermore, the trend for higher outlier sharing at smaller geographic scales, as we observe here when we compare locations within and among countries, is consistent with the hypothesis of evolution in concert, where some adaptive alleles spread over populations through gene flow whereas others originated locally (K. Johannesson et al., 2010; Kemppainen et al., 2011).

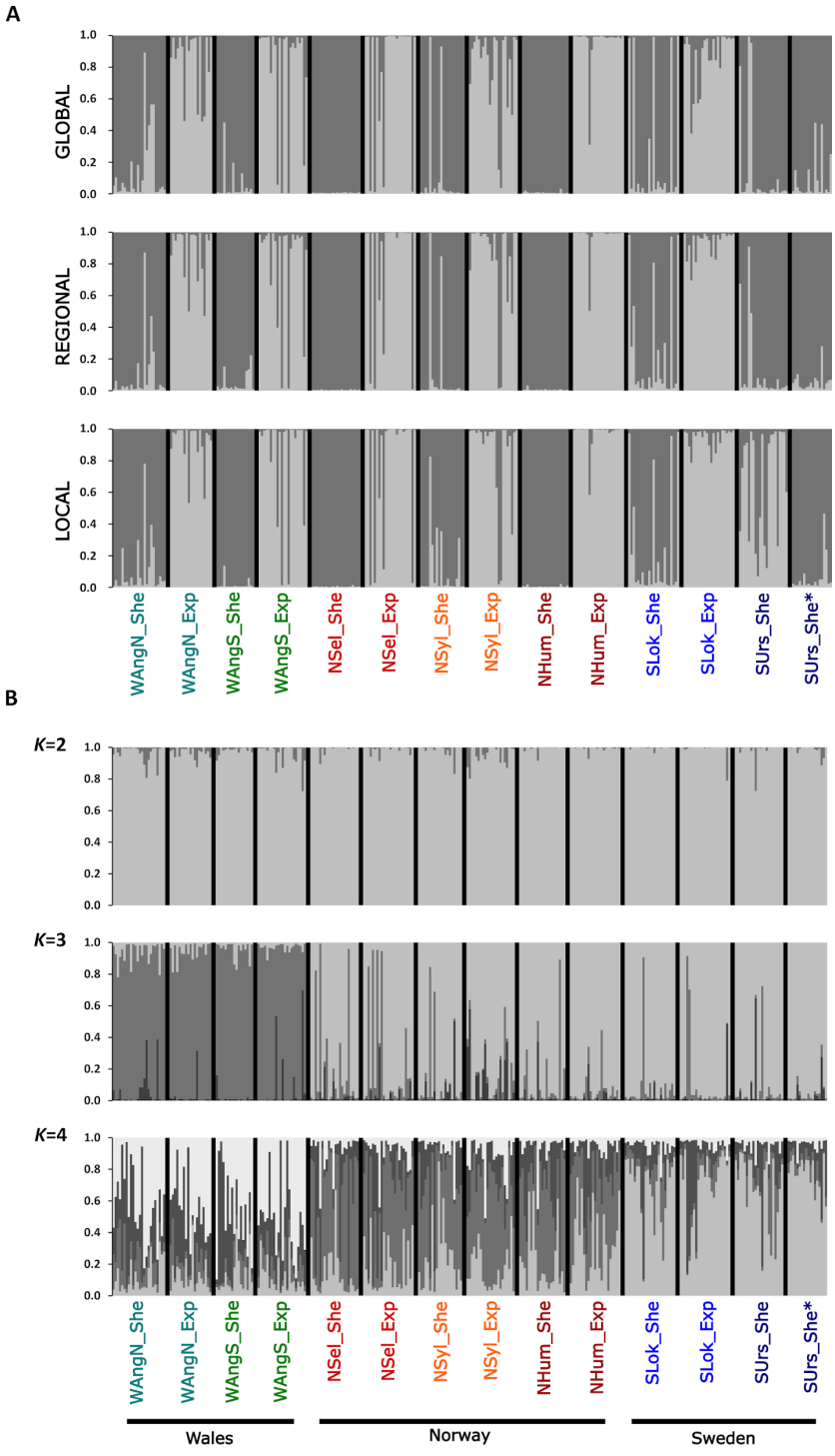


Figure 2.6: STRUCTURE plots ($2 \leq K \leq 4$) for: (a) three hierarchical levels for $K = 2$ (global, regional, and local; see Methods) using outlier loci ($p > .95$, DFDIST) (147 loci) and (b) nonoutliers ($p < .80$) (380 loci), in both cases combining the loci from all comparisons (see main text). Population codes are the same as in Table A2.1 and Figure A2.1. Each cluster is represented by a different grey shade. For the global analysis, shades are comparable across all countries and locations, whereas for the regional analysis shades are comparable only within countries. Shades are not comparable among locations in the local analysis.

Alternatively, similar patterns could also have resulted from shared standing genetic variation inherited from a common ancestral population (Westram, Panova, Galindo, & Butlin, 2016). In fact, accumulated evidence from multiple systems suggests that an important component of the variation involved in repeated episodes of adaptation is relatively old and maintained as standing genetic variation (e.g. Marques, Lucek, Sousa, Excoffier, and Seehausen 2019; Roesti, Kueng, Moser, and Berner 2015.

Ultimately, the level of outlier sharing also depends on the genomic architecture of the traits under selection, including whether they have a polygenic basis versus just a few genes of large effect involved, the genomic distribution of the underlying genes and on the recombination landscape across the genome. Indeed, genes involved in adaptation and speciation tend to cluster in regions of low recombination (Ravinet et al. 2017 and references therein). The high linkage disequilibrium between loci within these regions can lead to an increase in the amount of outliers shared across locations, even if just one locus is influenced by selection or if the selective pressures actually differ among locations (Berner & Roesti, 2017; Haenel, Laurentino, Roesti, & Berner, 2018; Roesti, Hendry, Salzburger, & Berner, 2012). This is particularly true within inverted regions, where recombination is heavily reduced in heterokaryotypes, as shown in multiple systems (Wellenreuther & Bernatchez, 2018), including *L. saxatilis* (Faria et al., 2019; Morales et al., 2019). The presence of inversions in *L. fabalis* has been previously hypothesized based on strong association between snail size, *Ark* genotype, and a RAPD locus genotype (K. Johannesson & Mikhailova, 2004; Tatarenkov & Johannesson, 1999). Since the location of the AFLP loci (anonymous markers) in the *L. fabalis* genome is unknown, a contiguous reference genome together with a high-resolution genome scan and linkage maps will be needed to understand the genomic architecture of adaptation in this system and how it relates with the recombination landscape. In particular, the characterization of transects across the environmental gradient using targeted-capture or whole-genome sequencing, as those performed in *L. saxatilis* (Faria et al., 2019; Westram et al., 2018), will be important to inform us about the role of inversions in *L. fabalis* diversification.

The contrast between the patterns of genetic structure based on outlier loci and on putatively neutral variation is noteworthy. The clustering of populations from both ecotypes by geography, when considering nonoutlier loci, suggests parallel evolution of *L. fabalis* ecotypes (Figures 2.4 and 2.5). However, gene flow between populations from different habitats is likely high enough to generate a pattern of parallel evolution, even if they had a single origin and came into secondary contact across multiple locations (Faria et al., 2014). Thus, a modelling approach, possibly based on approximate Bayesian computation and using sequence data across the genome needs to be implemented in future studies to infer the demographic history of ecotypes and specifically test whether parallel evolution, as observed in *L. saxatilis* (Butlin et al., 2014), is more likely than a

single origin of ecotypes. Nevertheless, the fact that almost half of all outliers (79 of 147 for $p >$ .95) are private to one location suggests that at least some components of divergent evolution are site-specific. Consequently, even if some of the genetic variation involved in ecotype differentiation had a single origin and was shared after secondary contact, each ecotype is likely to follow its own evolutionary trajectory in each location.

2.5.4 IMPLICATIONS FOR THE STUDY OF SPECIATION

Studies of local adaptation in intertidal habitats of rocky shores are key to quantify the contribution of ecological speciation to marine biodiversity (Sanford & Kelly, 2011). Overall, our results are consistent with the role of natural selection in driving divergence between *L. fabalis* ecotypes in the presence of gene flow. Previous mate choice experiments have shown that, although both males and females from the large ecotype tend to mate assortatively, males of the small ecotype can mate with both small and large ecotype females (Saltin, Schade, & Johannesson, 2013). However, while mate choice is known to play an important role in reproductive isolation in *L. saxatilis* (Perini, Rafajlović, Westram, Johannesson, & Butlin, 2020), whether this partial reproductive barrier is likely to result in significant reproductive isolation remains to be evaluated in natural populations for *L. fabalis*. Nevertheless, the maintenance of habitat-related phenotypic differentiation in *L. fabalis* despite high gene flow likely involves multiple loci influencing different traits, including shell thickness, growth and shape, suggesting some degree of reproductive isolation generated by extrinsic (ecological) factors. Although more detailed studies are needed to confirm and extend some of the reported results, the heterogeneous patterns of differentiation here identified, where a relatively small proportion of loci resist the homogenizing effects of gene flow in comparison with lack of differentiation at most studied loci, is compatible with initial stages of ecological speciation (Nosil, 2012; Seehausen et al., 2014). The *L. fabalis* system thus comprises an interesting system from the marine environment where information from multiple instances of divergence across an environmental transition opens doors for further studies on how populations cope with environmental changes (Sgrò, Lowe, & Hoffmann, 2011) and how different reproductive barriers accumulate during speciation.

2.6 ACKNOWLEDGEMENTS

We would like to thank Carolina Pereira for her help in processing and extracting DNA of some samples. This study was supported by: European Regional Development Fund (FCOMP-01-0124-

FEDER-014272), FCT – Foundation for Science and Technology (PTDC/BIA-EVF/113805/2009) and ASSEMBLE (grant agreement number 227799). RF was financed by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement number 706376 and by FCT (SFRH/BPD/89313/2012). JC was funded by a PhD scholarship from FCT (PD/BD/128350/2017). GS was financed by Volkswagen Stiftung (50500776). AK was supported by an IF contract (IF/00641/2014/CP1256/CT0008) from Fundação para a Ciência e a Tecnologia (FCT, Portugal). JG was funded by a JIN project (Jóvenes Investigadores sin vinculación o con vinculación temporal, Ministerio de Ciencia, Innovación y Universidades, Spain, RTI2018-101274-J-I00).

2.7 REFERENCES

Adams, D. C., Collyer, M., & Kaliontzopoulou, A. (2019). *Geomorph: Software for geometric morphometric analyses. r package version 3.1.0.*

Arrigo, N., Holderegger, R., & Alvarez, N. (2012). Automated scoring of aflps using rawgeno v 2.0, a free r cran library. *Data production and analysis in population genomics: methods and protocols*, 155-175.

Beaumont, M. A., & Nichols, R. A. (1996). Evaluating loci for use in the genetic analysis of population structure. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, *263*(1377), 1619-1626. doi: 10.1098/rspb.1996.0237

Berner, D., & Roesti, M. (2017). Genomics of adaptive divergence with chromosome-scale heterogeneity in crossover rate. *Molecular Ecology*, *26*(22), 6351-6369. doi: 10.1111/mec.14373

Boulding, E. G., & Hay, T. K. (1993). Quantitative genetics of shell form of an intertidal snail: constraints on short-term response to selection. *Evolution*, *47*(2), 576-592. doi: 10.1111/j.1558-5646.1993.tb02114.x

Butlin, R. K., Saura, M., Charrier, G., Jackson, B., André, C., Caballero, A., ... Rolán-Alvarez, E. (2014). Parallel evolution of local adaptation and reproductive isolation in the face of gene flow. *Evolution*, *68*(4), 935-949. doi: 10.1111/evo.12329

Carvajal-Rodríguez, A., Conde-Padín, P., & Rolán-Alvarez, E. (2005). Decomposing shell form into size and shape by geometric morphometric methods in two sympatric ecotypes of littorina saxatilis. *Journal of Molluscan studies*, *71*(4), 313-318. doi: 10.1093/mollus/eyi037

Charbit, S., Ritz, C., Philippon, G., Peyaud, V., & Kageyama, M. (2007). Numerical reconstructions of the northern hemisphere ice sheets through the last glacial-interglacial cycle. *Climate of the Past*, *3*(1), 15-37. doi: 10.5194/cp-3-15-2007

Collyer, M., & Adams, D. C. (2019). *Rrpp: Linear model evaluation with randomized residuals in a permutation procedure. r package version 0.4.0.*

Collyer, M. L., & Adams, D. C. (2018). Rrpp: An r package for fitting linear models to high-

dimensional data using residual randomization. *Methods in Ecology and Evolution*, *9*(7), 1772-1779. doi: 10.1111/2041-210X.13029

Conde-Padín, P., Caballero, A., & Rolán-Alvarez, E. (2009). Relative role of genetic determination and plastic response during ontogeny for shell-shape traits subjected to diversifying selection. *Evolution*, *63*(5), 1356-1363. doi: 10.1111/j.1558-5646.2009.00636.x

Conde-Padín, P., Grahame, J., & Rolán-Alvarez, E. (2007). Detecting shape differences in species of the littorina saxatilis complex by morphometric analysis. *Journal of Molluscan Studies*, *73*(2), 147-154. doi: 10.1093/mollus/eym009

Connell, J. H. (1972). Community interactions on marine rocky intertidal shores. *Annual review of ecology and systematics*, *3*(1), 169-192. doi: 10.1146/annurev.es.03.110172.001125

Costa, D., Sotelo, G., Kaliontzopoulou, A., Carvalho, J., Butlin, R., Hollander, J., & Faria, R. (2020). Hybridization patterns between two marine snails, littorina fabalis and l. obtusata. *Ecology and evolution*, *10*(3), 1158-1179. doi: 10.1002/ece3.5943

Coyer, J., Hoarau, G., Costa, J., Hogerdijk, B., Serrão, E., Billard, E., ... Olsen, J. (2011). Evolution and diversification within the intertidal brown macroalgae fucus spiralis/f. vesiculosus species complex in the north atlantic. *Molecular phylogenetics and evolution*, *58*(2), 283-296. doi: 10.1016/j.ympev.2010.11.015

Cruickshank, T. E., & Hahn, M. W. (2014). Reanalysis suggests that genomic islands of speciation are due to reduced diversity, not reduced gene flow. *Molecular ecology*, *23*(13), 3133-3157. doi: 10.1111/mec.12796

Denny, M., & Wethey, D. (2001). Physical processes that generate patterns in marine communities. *Marine community ecology*, 3-37.

Elmer, K. R., & Meyer, A. (2011). Adaptation in the age of ecological genomics: insights from parallelism and convergence. *Trends in ecology & evolution*, *26*(6), 298-306. doi: 10.1016/j.tree.2011.02.008

Evanno, G., Regnaut, S., & Goudet, J. (2005). Detecting the number of clusters of individuals using the software structure: a simulation study. *Molecular ecology*, *14*(8), 2611-2620. doi: 10.1111/j.1365-294X.2005.02553.x

Faria, R., Chaube, P., Morales, H. E., Larsson, T., Lemmon, A. R., Lemmon, E. M., ... Butlin, R. K. (2019). Multiple chromosomal rearrangements in a hybrid zone between *Littorina saxatilis* ecotypes. *Molecular Ecology*, *28*(6), 1375-1393. doi: 10.1111/mec.14972

Faria, R., Renaut, S., Galindo, J., Pinho, C., Melo-Ferreira, J., Melo, M., ... Butlin, R. K. (2014). Advances in ecological speciation: an integrative approach. *Molecular Ecology*, *23*(3), 513-521. doi: 10.1111/mec.12616

Felsenstein, J. (2013). *Phylip (phylogeny inference package).*

Foll, M., & Gaggiotti, O. (2008). A genome-scan method to identify selected loci appropriate for both dominant and codominant markers: a bayesian perspective. *Genetics*, *180*(2), 977-993. doi: 10.1534/genetics.108.092221

Galindo, J., Cacheda, D., Caballero, A., & Rolán-Alvarez, E. (2019). Untangling the contribution of genetic and environmental effects to shell differentiation across an environmental cline in a marine snail. *Journal of Experimental Marine Biology and Ecology*, *513*, 27-34. doi: 10.1016/j.jembe.2019.02.004

Galindo, J., & Grahame, J. W. (2014). Ecological speciation and the intertidal snail *Littorina saxatilis*. *Advances in Ecology*.

Galindo, J., Martínez-Fernández, M., Rodríguez-Ramilo, S. T., & Rolán-Alvarez, E. (2013). The role of local ecology during hybridization at the initial stages of ecological speciation in a marine snail. *Journal of Evolutionary Biology*, *26*(7), 1472-1487. doi: 10.1111/jeb.12152

Galindo, J., Morán, P., & Rolán-Alvarez, E. (2009). Comparing geographical genetic differentiation between candidate and noncandidate loci for adaptation strengthens support for parallel ecological divergence in the marine snail *Littorina saxatilis*. *Molecular Ecology*, *18*(5), 919-930. doi: 10.1111/j.1365-294X.2008.04076.x

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote sensing of Environment*, *202*, 18-27. doi: 10.1016/j.rse.2017.06.031

Grahame, J. W., Wilding, C. S., & Butlin, R. K. (2006). Adaptation to a steep environmental gradient and an associated barrier to gene exchange in *Littorina saxatilis*. *Evolution*, *60*(2), 268-278. doi: 10.1111/j.0014-3820.2006.tb01105.x

Haenel, Q., Laurentino, T. G., Roesti, M., & Berner, D. (2018). Meta-analysis of chromosome-scale crossover rate variation in eukaryotes and its significance to evolutionary genomics. *Molecular Ecology*, *27*(11), 2477-2497. doi: 10.1111/mec.14699

Helmuth, B., & Denny, M. W. (2003). Predicting wave exposure in the rocky intertidal zone: do bigger waves always lead to larger forces? *Limnology and Oceanography*, *48*(3), 1338-1345. doi: 10.4319/lo.2003.48.3.1338

Hollander, J., & Butlin, R. K. (2010). The adaptive value of phenotypic plasticity in two ecotypes of a marine gastropod. *BMC evolutionary biology*, *10*, 1-7. doi: 10.1186/1471-2148-10-333

Hollander, J., Galindo, J., & Butlin, R. K. (2015). Selection on outlier loci and their association with adaptive phenotypes in *Littorina saxatilis* contact zones. *Journal of Evolutionary Biology*, *28*(2), 328-337. doi: 10.1111/jeb.12564

Johannesson, B., & Johannesson, K. (1996). Population differences in behaviour and morphology in the snail littorina saxatilis: phenotypic plasticity or genetic differentiation? *Journal of Zoology*, *240*(3), 475-493. doi: 10.1111/j.1469-7998.1996.tb05299.x

Johannesson, K. (2001). Parallel speciation: a key to sympatric divergence. *Trends in Ecology & Evolution*, *16*(3), 148-153. doi: 10.1016/S0169-5347(00)02078-4

Johannesson, K., & Mikhailova, N. (2004). Habitat-related genetic substructuring in a marine snail (*Littorina fabalis*) involving a tight link between an allozyme and a dna locus. *Biological Journal of the Linnean Society*, *81*(2), 301-306. doi: 10.1111/j.1095-8312.2003.00288.x

Johannesson, K., Panova, M., Kemppainen, P., André, C., Rolán-Alvarez, E., & Butlin, R. K. (2010). Repeated evolution of reproductive isolation in a marine snail: unveiling mechanisms of speciation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *365*(1547), 1735-1747. doi: 10.1098/rstb.2009.0256

Johannesson, K., Rolán-Alvarez, E., & Erlandsson, J. (1997). Growth rate differences between upper and lower shore ecotypes of the marine snail littorina saxatilis (olivi)(gastropoda). *Biological Journal of the Linnean Society*, *61*(2), 267-279. doi: 10.1111/j.1095-8312.1997.tb01790.x

Kemppainen, P., Lindskog, T., Butlin, R. K., & Johannesson, K. (2011). Intron sequences of arginine kinase in an intertidal snail suggest an ecotype-specific selective sweep and a gene duplication. *Heredity*, *106*(5), 808-816. doi: 10.1038/hdy.2010.123

Kemppainen, P., Nes, S. v., Ceder, C., & Johannesson, K. (2005). Refuge function of marine algae complicates selection in an intertidal snail. *Oecologia*, *143*, 402-411. doi: 10.1007/s00442-004-1819-5

Kemppainen, P., Panova, M., Hollander, J., & Johannesson, K. (2009). Complete lack of mitochondrial divergence between two species of ne atlantic marine intertidal gastropods. *Journal of evolutionary biology*, *22*(10), 2000-2011. doi: 10.1111/j.1420-9101.2009.01810.x

Kess, T., Galindo, J., & Boulding, E. G. (2018). Genomic divergence between spanish *Littorina saxatilis* ecotypes unravels limited admixture and extensive parallelism associated with population history. *Ecology and Evolution*, *8*(16), 8311-8327. doi: 10.1002/ece3.4304

Kyle, C. J., & Boulding, E. G. (1998). Molecular genetic evidence for parallel evolution in a marine gastropod, littorina subrotundata. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, *265*(1393), 303-308. doi: 10.1098/rspb.1998.0296

Le Pennec, G., Butlin, R. K., Jonsson, P. R., Larsson, A. I., Lindborg, J., Bergström, E., ... Johannesson, K. (2017). Adaptation to dislodgement risk on wave-swept rocky shores in the snail *Littorina saxatilis*. *PloS one*, *12*(10), e0186901. doi: 10.1371/journal.pone.0186901

Little, C., & Kitching, J. A. (1996). *The biology of rocky shores*. Oxford University Press, USA.

Lynch, M., & Milligan, B. G. (1994). Analysis of population genetic structure with rapd markers. *Molecular Ecology*, *3*(2), 91-99. doi: 10.1111/j.1365-294X.1994.tb00109.x

Marques, D. A., Lucek, K., Sousa, V. C., Excoffier, L., & Seehausen, O. (2019). Admixture between old lineages facilitated contemporary ecological speciation in lake constance stickleback. *Nature communications*, *10*(1), 1-14. doi: 10.1038/s41467-019-12182-w

Morales, H. E., Faria, R., Johannesson, K., Larsson, T., Panova, M., Westram, A. M., & Butlin, R. K. (2019). Genomic architecture of parallel ecological divergence: beyond a single environmental contrast. *Science advances*, *5*(12), eaav9963. doi: 10.1126/sciadv.aav9963

Nosil, P. (2012). *Ecological speciation*. Oxford University Press.

Nosil, P., Harmon, L. J., & Seehausen, O. (2009). Ecological explanations for (incomplete) speciation. *Trends in ecology & evolution*, *24*(3), 145-156. doi: 10.1016/j.tree.2008.10.011

Paine, R. T., & Fenchel, T. (1994). *Marine rocky shores and community ecology: an experimentalist's perspective*. Ecology Institute Oldendorf/Luhe, Germany.

Pérez-Figueroa, A., García-Pereira, M., Saura, M., Rolán-Alvarez, E., & Caballero, A. (2010). Comparing three different methods to detect selective loci using dominant markers. *Journal of evolutionary biology*, *23*(10), 2267-2276. doi: 10.1111/j.1420-9101.2010.02093.x

Perini, S., Rafajlović, M., Westram, A. M., Johannesson, K., & Butlin, R. K. (2020). Assortative mating, sexual selection, and their consequences for gene flow in littorina. *Evolution*, *74*(7), 1482-1497. doi: 10.1111/evo.14027

Pritchard, J. K., Stephens, M., & Donnelly, P. (2000). Inference of population structure using multilocus genotype data. *Genetics*, *155*(2), 945-959. doi: 10.1093/genetics/155.2.945

Raffaelli, D., & Hawkins, S. J. (1996). *Intertidal ecology*. Springer Science & Business Media.

Ravinet, M., Faria, R., Butlin, R., Galindo, J., Bierne, N., Rafajlović, M., ... Westram, A. (2017). Interpreting the genomic landscape of speciation: a road map for finding barriers to gene flow. *Journal of evolutionary biology*, *30*(8), 1450-1477. doi: 10.1111/jeb.13047

Reid, D. G. (1996). *Systematics and evolution of Littorina*. London: Ray Society.

Reimchen, T. (1981). Microgeographical variation in *Littorina mariae Sacchi & Rastelli* and a taxonomic consideration. *Journal of conchology*, *30*, 341-350.

Roesti, M., Hendry, A. P., Salzburger, W., & Berner, D. (2012). Genome divergence during evolutionary diversification as revealed in replicate lake-stream stickleback population pairs. *Molecular Ecology*, *21*(12), 2852-2862. doi: 10.1111/j.1365-294X.2012.05509.x

Roesti, M., Kueng, B., Moser, D., & Berner, D. (2015). The genomics of ecological vicariance in threespine stickleback fish. *Nature communications*, *6*(1), 8767. doi: 10.1038/ncomms9767

Rohlf, F. J. (2006). *Tps series*. Department of Ecology and Evolution, State University of New York at Stony Brook.

Rohlf, F. J., & Slice, D. (1990). Extensions of the procrustes method for the optimal superimposition of landmarks. *Systematic zoology*, *39*(1), 40-59. doi: 10.2307/2992207

Rolán-Alvarez, E., Austin, C. J., & Boulding, E. G. (2015). The contribution of the genus littorina to the field of evolutionary ecology. *Oceanography and Marine Biology: an annual review*, *53*, 157-214.

Saltin, S. H., Schade, H., & Johannesson, K. (2013). Preference of males for large females causes a partial mating barrier between a large and a small ecotype of littorina fabalis (w. turton, 1825). *Journal of Molluscan Studies*, *79*(2), 128-132. doi: 10.1093/mollus/eyt003

Sanford, E., & Kelly, M. W. (2011). Local adaptation in marine invertebrates. *Annual review of marine science*, *3*, 509-535. doi: 10.1146/annurev-marine-120709-142756

Schluter, D. (2000). *The ecology of adaptive radiation*. Oxford University Press.

Schluter, D. (2009). Evidence for ecological speciation and its alternative. *Science*, *323*(5915), 737-741. doi: 10.1126/science.1160006

Seehausen, O., Butlin, R. K., Keller, I., Wagner, C. E., Boughman, J. W., Hohenlohe, P. A., ... Widmer, A. (2014). Genomics and the origin of species. *Nature Reviews Genetics*, *15*(3), 176-192. doi: 10.1038/nrg3644

Seeley, R. H. (1986). Intense natural selection caused a rapid morphological transition in a living marine snail. *Proceedings of the National Academy of Sciences*, *83*(18), 6897-6901. doi: 10.1073/pnas.83.18.6897

Sgrò, C. M., Lowe, A. J., & Hoffmann, A. A. (2011). Building evolutionary resilience for conserving biodiversity under climate change. *Evolutionary applications*, *4*(2), 326-337. doi: 10.1111/j.1752-4571.2010.00157.x

Smadja, C. M., & Butlin, R. K. (2011). A framework for comparing processes of speciation in the presence of gene flow. *Molecular Ecology*, *20*(24), 5123-5140. doi: 10.1111/j.1365-294X.2011.05350.x

Sokal, R., & Rohlf, F. (1995). *Biometry. the principles and practice of statistics in biological research*. W. H. Freeman and Co, New York.

Sotelo, G., Duvetorp, M., Costa, D., Panova, M., Johannesson, K., & Faria, R. (2020). Phylogeographic history of flat periwinkles, *Littorina fabalis* and *L. obtusata*. *BMC Evolutionary Biology*, *20*(23), 1-18. doi: 10.1186/s12862-019-1561-6

Southward, A. (1957). On the behaviour of barnacles iii. further observations on the influence of temperature and age on cirral activity. *Journal of the Marine Biological Association of the United Kingdom*, *36*(2), 323-334. doi: 10.1017/S0025315400016830

Stapley, J., Reger, J., Feulner, P. G., Smadja, C., Galindo, J., Ekblom, R., ... Slate, J. (2010). Adaptation genomics: the next generation. *Trends in ecology & evolution*, *25*(12), 705-712. doi: 10.1016/j.tree.2010.09.002

Tatarenkov, A., & Johannesson, K. (1994). Habitat related allozyme variation on a microgeographic scale in the marine snail *Littorina mariae* (prosobranchia: Littorinacea). *Biological Journal of the Linnean Society*, *53*(2), 105-125. doi: 10.1111/j.1095-8312.1994.tb01004.x

Tatarenkov, A., & Johannesson, K. (1998). Evidence of a reproductive barrier between two forms of the marine periwinkle *Littorina fabalis* (gastropoda). *Biological Journal of the Linnean Society*, *63*(3), 349-365. doi: 10.1111/j.1095-8312.1998.tb01522.x

Tatarenkov, A., & Johannesson, K. (1999). Micro-and macrogeographic allozyme variation in *Littorina fabalis*; do sheltered and exposed forms hybridize? *Biological Journal of the Linnean Society*, *67*(2), 199-212. doi: 10.1111/j.1095-8312.1999.tb01861.x

Team, R. C. (2019). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `https://www.R-project.org/`

Tomanek, L., & Helmuth, B. (2002). Physiological ecology of rocky intertidal organisms: a synergy of concepts. *Integrative and Comparative Biology*, *42*(4), 771-775. doi: 10.1093/icb/42.4.771

Vekemans, X., Beauwens, T., Lemaire, M., & Roldán-Ruiz, I. (2002). Data from amplified frag-

ment length polymorphism (aflp) markers show indication of size homoplasy and of a relationship between degree of homoplasy and fragment size. *Molecular ecology*, *11*(1), 139-151. doi: 10.1046/j.0962-1083.2001.01415.x

Wellenreuther, M., & Bernatchez, L. (2018). Eco-evolutionary genomics of chromosomal inversions. *Trends in ecology & evolution*, *33*(6), 427-440. doi: 10.1016/j.tree.2018.04.002

Westram, A. M., Panova, M., Galindo, J., & Butlin, R. K. (2016). Targeted resequencing reveals geographical patterns of differentiation for loci implicated in parallel evolution. *Molecular ecology*, *25*(13), 3169-3186. doi: 10.1111/mec.13640

Westram, A. M., Rafajlović, M., Chaube, P., Faria, R., Larsson, T., Panova, M., ... Butlin, R. K. (2018). Clines on the seashore: The genomic architecture underlying rapid divergence in the face of gene flow. *Evolution letters*, *2*(4), 297-309. doi: 10.1002/evl3.74

Whitlock, R., Hipperson, H., Mannarelli, M., Butlin, R., & Burke, T. (2008). An objective, rapid and reproducible method for scoring aflp peak-height data that minimizes genotyping error. *Molecular Ecology Resources*, *8*(4), 725-735. doi: 10.1111/j.1755-0998.2007.02073.x

Wilding, C., Butlin, R. K., & Grahame, J. (2001). Differential gene exchange between parapatric morphs of *Littorina saxatilis* detected using aflp markers. *Journal of Evolutionary Biology*, *14*(4), 611-619. doi: 10.1046/j.1420-9101.2001.00304.x

Williams, G. A. (1990). Periwinkles, *Littorina obtusata* (l.) and *L. mariae Sacchi et Rastelli*. *Field Studies*, *7*, 469-482.

Wood, H. M., Grahame, J. W., Humphray, S., Rogers, J., & Butlin, R. K. (2008). Sequence differentiation in regions identified by a genome scan for local adaptation. *Molecular Ecology*, *17*(13), 3123-3135. doi: 10.1111/j.1365-294X.2008.03755.x

Zhivotovsky, L. A. (1999). Estimating population structure in diploids with multilocus dominant dna markers. *Molecular ecology*, *8*(6), 907-913. doi: 10.1046/j.1365-294x.1999.00620.x

2.8 APPENDIX

Table A2.1: Sampling information. Country, location, habitat (sheltered, moderately-exposed), code, geographic coordinates, date, number of individuals sampled (*N*) and number of individuals analysed for AFLPs (N_{AFLP}) for each location. Habitat was inferred based on the abundance of *Ascophyllum* spp. Location numbers follow those present in Figure 2.1. Sheltered sites are named with the abbreviation "She" and moderately-exposed sites with "Exp".

Country	**Location**	**Habitat**	**Code**	**Coordinates**	**Date**	***N***	***NAFLP***
Wales	Anglesey – North (1)	Mod-Exp	WAngN_Exp	53º25'07"N, 4º27'06"W	September 2012	21	19
Wales	Anglesey – North (1)	Sheltered	WAngN_She	53º25'20"N, 4º22'09"W	September 2012	50	24
Wales	Anglesey – South (2)	Mod-Exp	WAngS_Exp	53º14'26"N, 4º35'48"W	September 2012	72	22
Wales	Anglesey – South (2)	Sheltered	WAngS_She	53º14'28"N, 4º35'35"W	September 2012	56	17
Norway	Seløyna (3)	Mod-Exp	NSel_Exp	60º38'05"N, 4º47'30"E	August 2012	30	23
Norway	Seløyna (3)	Sheltered	NSel_She	60º38'13"N, 4º47'34"E	August 2012	26	22
Norway	Syltøyna (4)	Mod-Exp	NSyl_Exp	60º18'40"N, 4º58'23"E	August 2012	30	22
Norway	Syltøyna (4)	Sheltered	NSyl_She	60º18'44"N, 4º58'41"E	August 2012	22	20
Norway	Hummelsund (5)	Mod-Exp	NHum_Exp	60º10'19"N, 5º04'58"E	August 2012	33	23
Norway	Hummelsund (5)	Sheltered	NHum_She	60º10'17"N, 5º04'45"E	August 2012	38	21
Sweden	Lökholmen (6)	Mod-Exp	SLok_Exp	58º53'20"N, 11º06'32"E	Sept./Oct. 2012	43	23
Sweden	Lökholmen (6)	Sheltered	SLok_She	58º53'21"N, 11º06'38"E	Sept./Oct. 2012	41	23
Sweden	Ursholmen (7)	Sheltered	SUrs_She	58º50'06"N, 10º59'26"E	Sept./Oct. 2012	59	22
Sweden	Ursholmen (7)	Sheltered*	SUrs_She*	58º49'55"N, 10º59'39"E	Sept./Oct. 2012	35	18

*Despite previous information suggesting this site was moderately-exposed, the high density of *Ascophyllum* spp. observed *in situ* rather suggests that it is sheltered (see main text).

Table A2.2: General Linear Model (GLM) statistics for the analysis of variation in shell morphology across countries, habitats and locations, regarding size (logCS) and shape separately, as well as shape accounting for the influence of size. df, degrees of freedom; SS, sums of squares; Z, Z-scores; P, p-value (* indicates significant values).

		df	***SS***	***Z***	***P***
Size	country	2	2.6054	0.88121	0.172
	habitat	1	5.1304	3.00222	0.001*
	location	3	0.4899	2.74808	0.001*
	country:habitat	2	0.2555	0.30797	0.379
	habitat:location	3	0.2733	2.10947	0.003*
	Residuals	275	4.7219		
	Total	286	13.4764		
		df	***SS***	***Z***	***P***
Shape	country	2	0.1569	0.9182	0.188
	habitat	1	0.2281	4.7894	0.001*
	location	3	0.1196	3.8528	0.001*
	country:habitat	2	0.1484	1.0216	0.152
	habitat:location	3	0.1159	3.5351	0.001*
	Residuals	275	2.511		
	Total	286	3.28		
		df	***SS***	***Z***	***P***
Shape Accounting for size	CS	1	0.0884	3.7042	0.001*
	country	2	0.1707	0.9007	0.186
	habitat	1	0.1614	4.3813	0.001*
	location	3	0.134	4.2491	0.001*
	CS:country	2	0.134	1.3922	0.074
	CS:habitat	1	0.008	0.2556	0.400
	country:habitat	2	0.0829	0.7487	0.217
	CS:location	3	0.0772	2.7671	0.003*
	habitat:location	3	0.0774	2.754	0.002*
	CS:country:habitat	2	0.0346	0.2294	0.406
	CS:habitat:location	3	0.0451	1.4693	0.072*
	Residuals	263	2.2663		
	Total	286	3.28		

Table A2.3: Pairwise F_{ST} values estimated on AFLP-SURV. The lower matrix is calculated from the 147 outlier loci ($p > 0.95$), the upper matrix is calculated from the 380 non-outlier loci ($p < 0.80$), both sets obtained by combining results from all pairwise comparisons. Population codes are the same as in Figure A2.1 and Table A2.1.

	WAngN_ She	**WAngN_ Exp**	**WAngS_ She**	**WAngS_ Exp**	**NSel_ She**	**NSel_ Exp**	**NSyl_ She**	**NSyl_ Exp**	**NHum_ She**	**NHum_ Exp**	**SLok_ She**	**SLok_ Exp**	**SUrs_ She**	**SUrs_ She***
WAngN_She		0	0.0215	0.0168	0.0412	0.0422	0.033	0.0411	0.0501	0.0528	0.0796	0.0583	0.0755	0.0605
WAngN_Exp	0.1488		0.0074	0.0116	0.0413	0.0344	0.0334	0.0389	0.048	0.0431	0.0659	0.0562	0.0705	0.0535
WAngS_She	0.0547	0.1356		0	0.0533	0.0473	0.053	0.064	0.0657	0.0626	0.0791	0.0695	0.0616	0.0558
WAngS_Exp	0.118	0.0522	0.1114		0.0411	0.0423	0.0454	0.0543	0.0566	0.0557	0.0783	0.0705	0.076	0.0637
NSel_She	0.0748	0.1628	0.0448	0.1283		0	0.0012	0	0.0101	0.01	0.07	0.0476	0.0618	0.0499
NSel_Exp	0.1541	0.0808	0.1354	0.0914	0.1581		0.0005	0	0.0062	0	0.0423	0.0312	0.0317	0.0282
NSyl_She	0.0804	0.1431	0.045	0.1161	0.0222	0.1245		0	0.0056	0	0.0529	0.0419	0.0492	0.0342
NSyl_Exp	0.1113	0.1217	0.1455	0.0919	0.1273	0.0676	0.1063		0.0068	0.0008	0.0513	0.0346	0.0492	0.0442
NHum_She	0.1467	0.1708	0.0767	0.161	0.0511	0.1526	0.0383	0.1639		0	0.043	0.029	0.0444	0.0353
NHum_Exp	0.176	0.1028	0.186	0.0881	0.1913	0.0398	0.1517	0.0353	0.2079		0.029	0.0234	0.027	0.0183
SLok_She	0.1146	0.0984	0.0682	0.0881	0.0866	0.0982	0.0612	0.1209	0.0845	0.1231		0	0.0156	0.011
SLok_Exp	0.194	0.0993	0.1616	0.1039	0.1821	0.0649	0.1451	0.1089	0.1646	0.0834	0.0682		0.0153	0.0144
SUrs_She	0.0854	0.1157	0.0564	0.1156	0.0627	0.1183	0.0381	0.1215	0.0728	0.1456	0.0037	0.1005		0
SUrs_She*	0.172	0.1664	0.1074	0.1743	0.1308	0.1727	0.1131	0.1945	0.1077	0.2138	0.0363	0.1421	0.041	

Table A2.4: Number and percentage of individuals analysed for both genetics and morphology (shape) for each site. GEN & GM, individuals with concordant genetics and morphological classification ($PP \geq 0.90$); GEN & GM & Habitat, individuals with concordant genetics and morphological classification ($PP \geq 0.90$) typical of their habitat (based on the most frequent form/cluster in each habitat), GEN & Habitat, individuals with a genetic classification ($PP \geq 0.90$) typical of their habitat but with discordant morphological classification; and GM & Habitat, individuals with a morphological classification ($PP \geq 0.90$) typical of their habitat but with discordant genetic classification. Population codes are the same as in Figure A2.1 and Table A2.1.

	Total analysed (N)	**GEN & GM (N)**	**GEN & GM (%)**	**GEN & GM & Habitat (N)**	**GEN & GM & Habitat (%)**	**GEN & Habitat (N)**	**GEN & Habitat (%)**	**GM & Habitat (N)**	**GM & Habitat (%)**
Sel_she	4	4	100	4	100	0	0	0	0
Sel_exp	9	7	78	7	78	0	0	1	11
Syl_she	12	10	83	9	75	0	0	0	0
Syl_exp	3	1	33	1	33	1	33	1	33
Lok_she	7	4	57	4	57	0	0	1	14
Lok_exp	3	1	33	1	33	1	33	0	0
AngN_she	14	6	43	6	43	4	29	0	0
AngN_exp	10	4	40	3	30	1	10	0	0
AngS_she	4	1	25	1	25	1	25	0	0
AngS_exp	17	11	65	11	65	1	6	2	12
Total	83	49	59	47	57	9	11	5	6

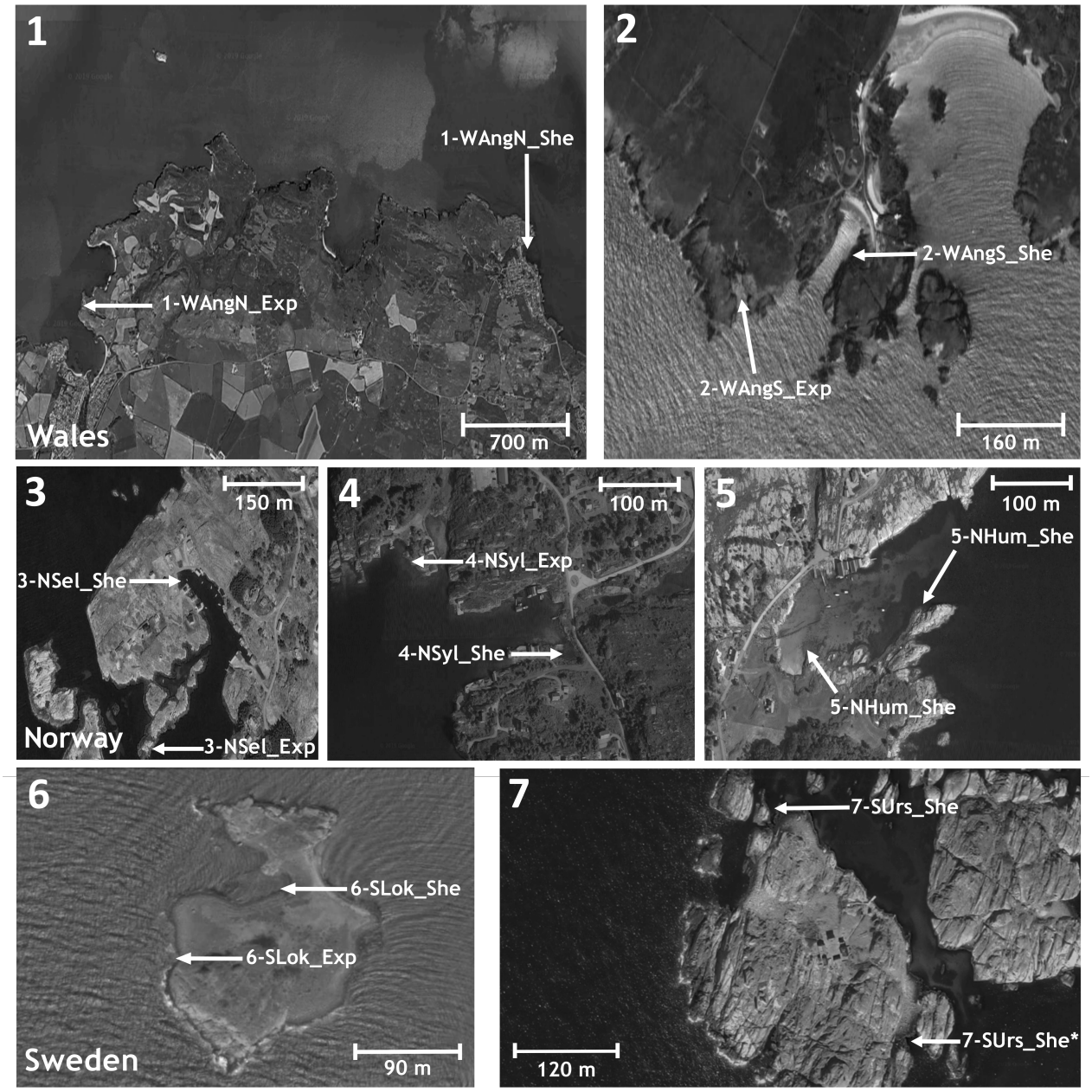


Figure A2.1: Detailed view of the exposed and sheltered sites within each country, for Anglesey North (1) and Anglesey South (2) in Wales; Seløyna (3), Syltøyna (4) and Hummelsund (5) in Norway; and Lökholmen (6) and Ursholmen (7) in Sweden. Numbers are the same as in Figure 2.1. Sheltered sites are named with the abbreviation "She" and moderately-exposed sites with "Exp".

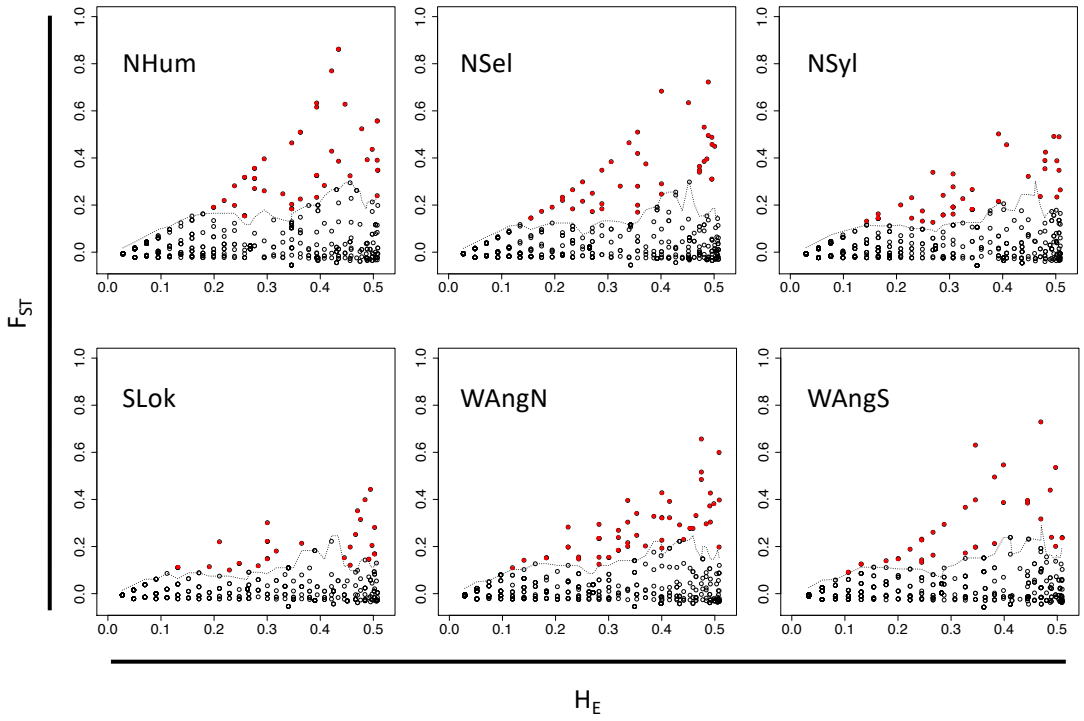


Figure A2.2: Plot of the outlier DFDIST results for *Littorina fabalis*. Shown is the distribution (F_{ST}/heterozygosity) of the empirical AFLP loci for each pairwise comparison within location (moderately-exposed vs sheltered habitat). The solid line represents the 95 quantile of the distribution of simulated loci (see Methods). The red dots represent the outliers ($p > 0.95$).

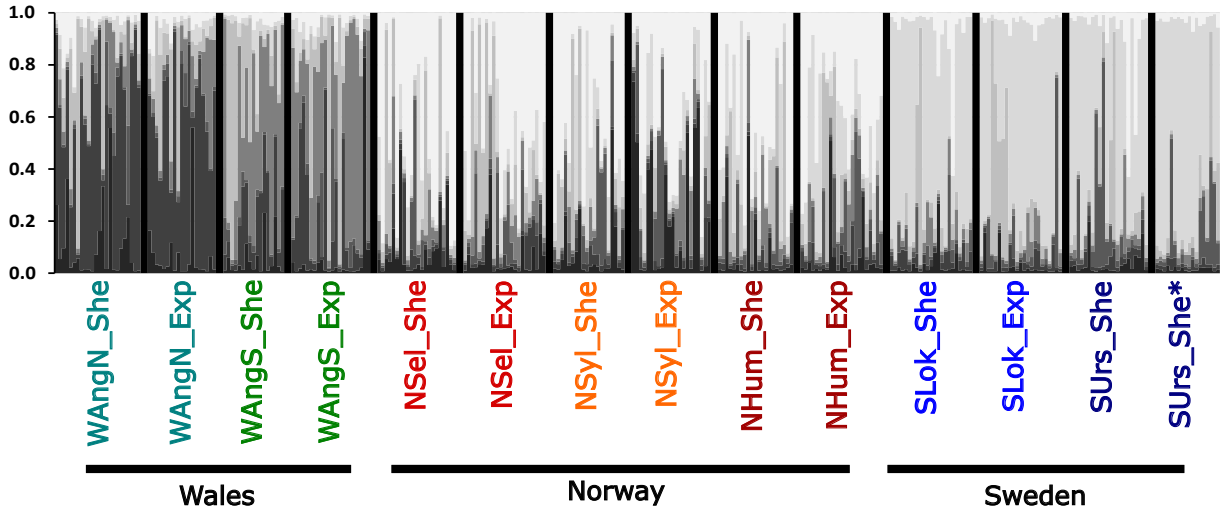


Figure A2.3: STRUCTURE plots for non-outlier loci (N=380, $p < 0.80$) for $K = 7$, the most likely number of clusters according to the Evanno's method.

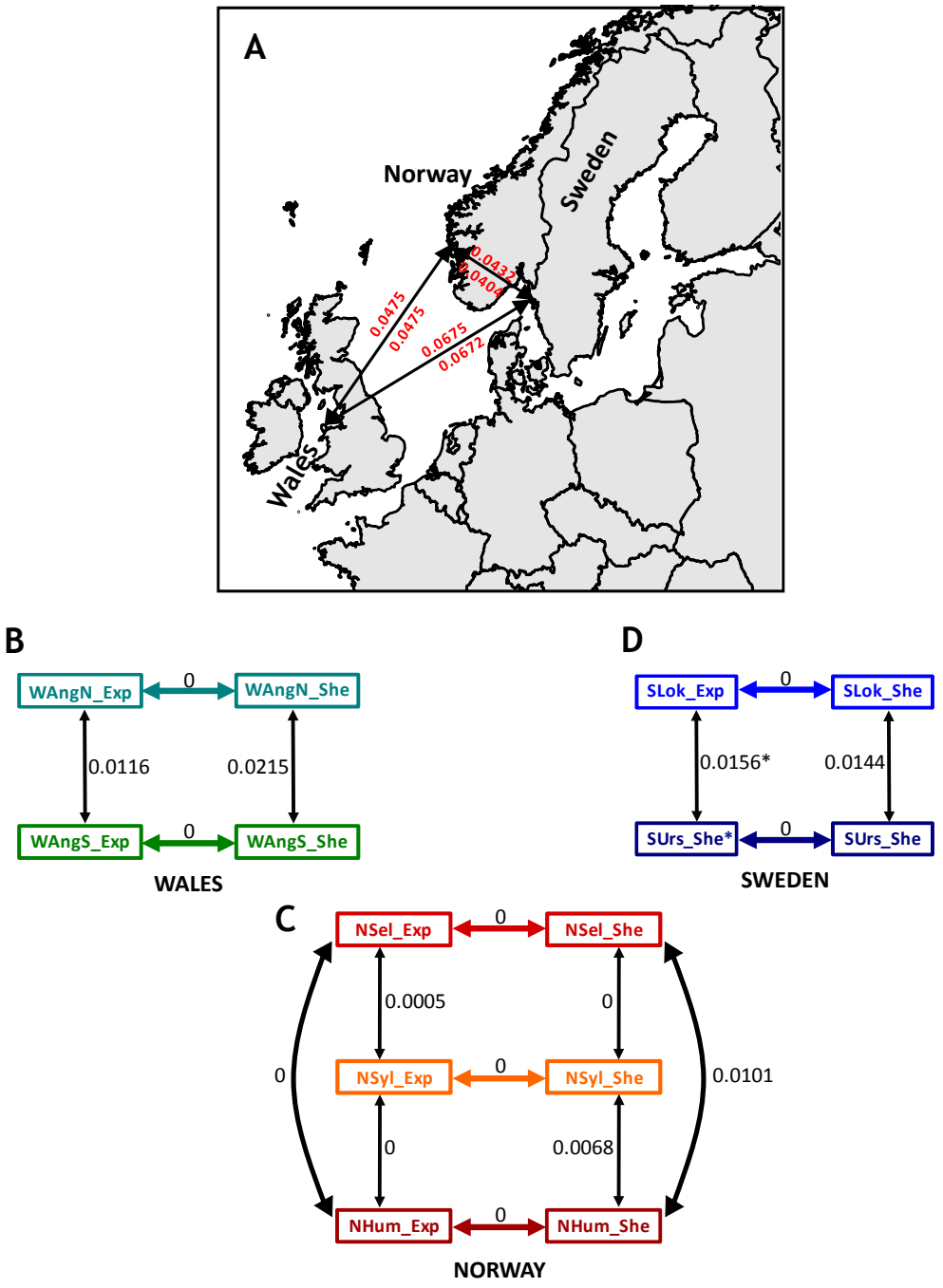


Figure A2.4: Differentiation between populations measured as F_{ST} based on non-outlier loci (N=380, $p < 0.80$) . A) Map of the three studied countries with mean pairwise F_{ST} between populations of the same ecotype from two different countries (above the arrows) and mean pairwise F_{ST} between all populations from two different countries (below the arrows). Mean F_{ST} between ecotypes within each location and between the same ecotype across locations is shown for Wales, United Kingdom (B), Norway (C) and Sweden (D). Population codes are the same as in Figure A2.1 and Table A2.1.

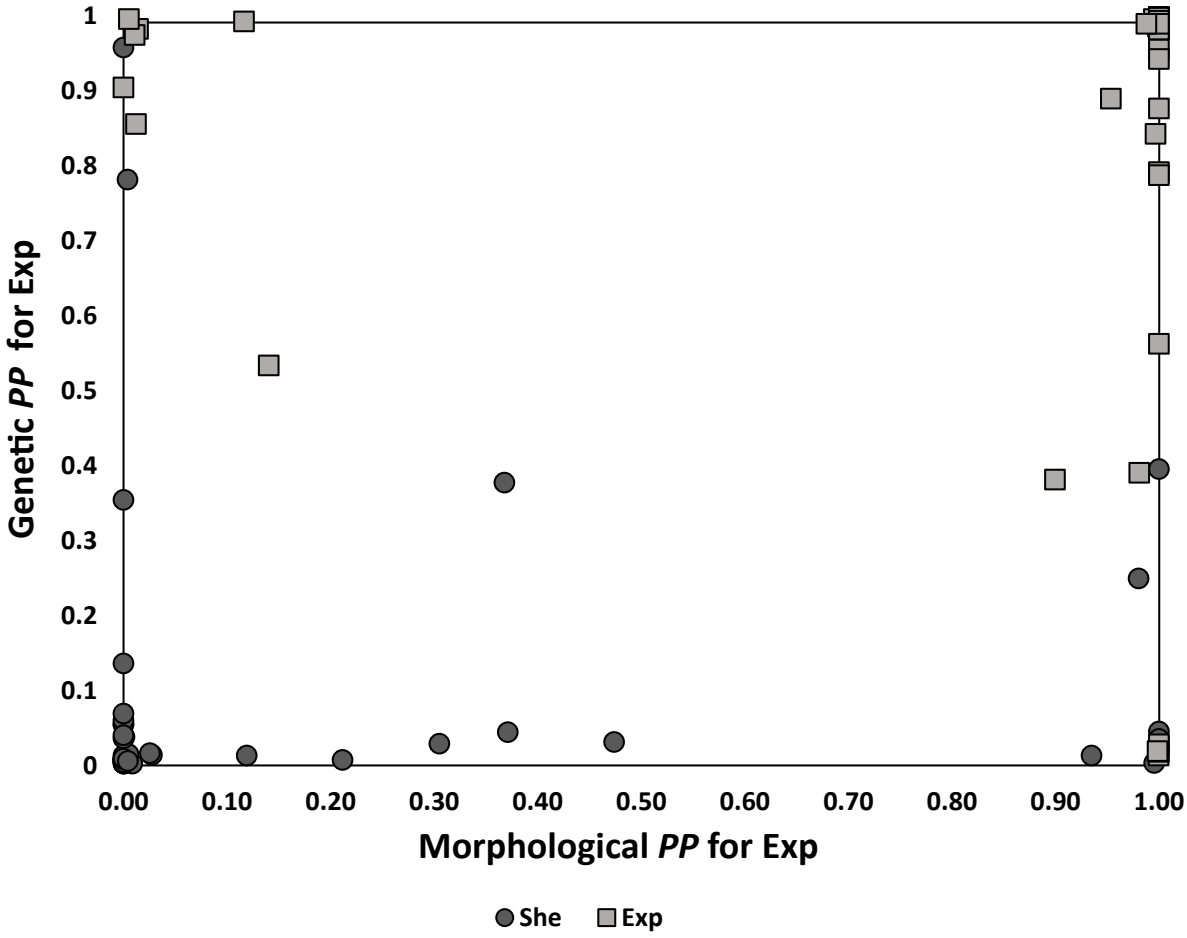


Figure A2.5: Plot of the posterior probability (PP) obtained with the DFA for shape against the genetic assignment obtained with the STRUCTURE outlier analysis (147 loci, $p > 0.95$) for each individual (N=83). For both variables, PP values represent the assignment probability to the moderately-exposed habitat.

CHAPTER 3

poolHelper: an R package to help in designing Pool-Seq studies

João Carvalho, Rui Faria, Roger K. Butlin & Vitor C. Sousa

Published in *Methods in Ecology and Evolution* (2023)
DOI: 10.1111/2041-210X.14185

3.1 ABSTRACT

Next-generation sequencing of pooled samples (Pool-seq) is an important tool in population genomics and molecular ecology. In Pool-seq, the relative number of reads with an allele reflects the allele frequencies in the sample. However, unequal individual contributions to the pool and sequencing errors can lead to inaccurate allele frequency estimates, influencing downstream analysis. When designing Pool-seq studies, researchers need to decide the pool size (number of individuals) and average depth of coverage (sequencing effort). An efficient sampling design should maximise the accuracy of allele frequency estimates while minimising the sequencing effort. We describe a novel tool to simulate single nucleotide polymorphism (SNP) data using coalescent theory and account for sources of uncertainty in Pool-seq. We introduce an R package, *poolHelper*, enabling users to simulate Pool-seq data under different combinations of average depth of coverage and pool size, accounting for unequal individual contributions and sequencing errors, modelled by adjustable parameters. The mean absolute error is computed by comparing the sample allele frequencies obtained based on individual genotypes with the frequency estimates obtained with Pool-seq. *poolHelper* enables users to simulate multiple combinations of pooling errors, average depth of coverage, pool sizes and number of pools to assess how they influence the error of sample allele frequencies and expected heterozygosity. Using simulations under a single population model, we illustrate that increasing the depth of coverage does not necessarily lead to more accurate estimates, reinforcing that finding the best Pool-seq study design is not straightforward. Moreover, we show that simulations can be used to identify different combinations of parameters with similarly low mean absolute errors. This can help users to define an effective sampling design by using those combinations of parameters that minimise the sequencing effort. The *poolHelper* package provides tools for performing simulations with different combinations of parameters (e.g., pool size, depth of coverage, unequal individual contribution) before sampling and generating data, allowing users to define sampling schemes based on simulations. This allows researchers to focus on the best sampling scheme to answer their research questions. *poolHelper* is comprehensively documented with examples to guide effective use.

Keywords: experimental design, open source, Pool-seq, R package, simulations

3.2 INTRODUCTION

Next Generation Sequencing (NGS) is an important tool for many biologists, providing access to polymorphism data across a wide range of model and non-model species (Ellegren, 2014). Al-

though the cost of sequencing is continuously decreasing, high coverage sequencing of multiple individuals is still expensive. Furthermore, it is challenging to obtain individual genomic data for certain species (e.g., small organisms) or in Evolve-and-Resequence experiments involving a large number of populations or many points along a time series. In those instances, next-generation sequencing of pooled samples (Pool-seq) might be the only viable alternative, as it requires less DNA per individual. Pool-seq is a sequencing technique that provides a cost-effective approach to quantify genetic variation within a population. It involves pooling multiple individual DNA samples together and sequencing them collectively. A typical Pool-seq analysis requires several steps. First, researchers should determine the pool size (i.e., the number of individuals included in the pool) and the desired sequencing depth of coverage during the experimental design step. Next, DNA extracted from individual samples is combined into pools. In situations where obtaining DNA from each individual sample is impractical, an alternative approach is to group several individuals together prior to DNA extraction. For instance, muscle tissue from multiple individuals can be combined, extracting DNA from the entire group of individuals (Morales et al., 2019; Ross, Endersby-Harshman, & Hoffmann, 2019). Then, DNA extracted from multiple groups of individuals can be merged into a single, final pool. Non-equimolar quantities of DNA between these groups of multiple individuals, or between individuals within a group, can lead to unequal contributions. This disparity in contribution may result in certain groups of individuals having a disproportionate impact on the overall allele frequencies, leading to inaccurate estimation of sample allele frequencies, potentially affecting downstream analysis (Anderson, Skaug, & Barshis, 2014; Ellegren, 2014). Subsequently, for each pool, a single library is generated prior to sequencing with NGS technologies. Note that Pool-seq does not require individual tagging of sequences, reducing the laboratory work required for library preparation, while still generating population-level genomic data (Schlötterer, Tobler, Kofler, & Nolte, 2014). During this library preparation step, stochastic variation in amplification efficiency can also result in unequal contributions of individuals, and lead to inaccurate sample allele frequencies. Finally, the pooled libraries are sequenced. This step also introduces uncertainties in the analysis due to variation in sequencing depth along the genome, and sequencing errors. The next steps, such as quality control, read alignment, and variant calling, are similar to individual-based sequencing.

Despite these potential sources of uncertainty (e.g., unequal individual contribution), Pool-seq has been extensively used in a variety of settings (Begun et al., 2007; Ferretti, Ramos-Onsins, & Pérez-Enciso, 2013; Prescott et al., 2015; Zhou et al., 2011). This has lead to the development of tools such as the R package poolSeq (Taus, Futschik, & Schlötterer, 2017) and the DIYABC - RF software (Collin et al., 2021) that simulate Pool-seq data, as well as data analysis tools (e.g., Kofler, Pandey, and Schlötterer 2011). Nonetheless, to the best of our knowledge, no tool currently exists

that can simultaneously and explicitly account for variation in depth of coverage, unequal contribution and sequencing errors, which are known sources of Pool-seq uncertainty (see Table A3.1 for more details). It is worth noting that unequal contribution occurs due to variations in DNA concentration or amplification efficiency among the pooled samples, resulting in an uneven representation of genetic material from each sample. Here, we use the term pooling error to quantify the error caused by unevenly combining multiple DNA samples into a single pool, which we explicitly model as the dispersion around the expected proportion of reads from each sample. This pooling error can introduce biases in estimates of sample allele frequencies. As mentioned, two key parameters in the experimental design step of a Pool-seq study are the number of individuals in each pool, and the average depth of coverage. These two parameters determine how much the sample allele frequencies are affected by Pool-seq associated errors. On one hand, increasing the number of individuals allows estimating more accurate allele frequencies, but more individuals in the pool might not avoid errors associated with unequal individual contribution when the pooling error is high. On the other hand, increasing the depth of coverage should lead to more reliable estimates, but it can amplify pooling errors and increase the frequency of sequencing errors, which can make it challenging to differentiate true low-frequency variants from sequencing errors. Moreover, due to its costs, the depth of coverage is typically the limiting resource. Simulations of single nucleotide polymorphism (SNP) data accounting for sources of uncertainty with Pool-seq data under different sampling schemes can thus provide a tool to help researchers design Pool-seq experiments and to minimise the error associated with the sample allele frequencies.

Here, we introduce an R package (Team, 2020), *poolHelper*, to simulate Pool-seq data according to different sampling designs. Our approach relies on coalescent simulations under neutrality using *scrm* (Staab, Zhu, Metzler, & Lunter, 2015). The *poolHelper* package provides tools and functions to simulate Pool-seq datasets, accounting for potential sources of error in the Pool-seq analysis process. Importantly, these errors are modelled by parameters that users can adjust. *poolHelper* models the unequal contribution resulting from differences in DNA concentration and amplification efficiency during DNA extraction and library preparation. Additionally, it accounts for sequencing depth variation across SNPs, sequencing errors, and mapping errors during read alignment. This allows comparing the allele frequencies obtained directly from the simulated individual genotypes with the frequencies obtained from Pool-seq data. Since R is a free and collaborative project, users can use available tools to handle, analyse and visualise genomic datasets. Our goal is to provide a flexible method of simulating Pool-seq data, allowing researchers to design their experiments with a better *a priori* knowledge of possible errors associated with Pool-seq, thus contributing to the recognition of Pool-seq as a valuable source of data to reconstruct the evolutionary history of populations.

3.3 IMPLEMENTATION

The main steps of our pipeline follow a relatively simple scheme: coalescent simulations of individual genotypes under a single population model with a constant size, computation of alternative allele frequencies directly from the genotypes, simulation of Pool-seq given the genotypes, and computation of alternative allele frequencies from the Pool-seq data, assuming that it corresponds to the proportion of reads with that allele. To measure the error associated with Pool-seq we computed the average absolute difference between the actual allele frequencies based on individual genotypes in the sample and the allele frequencies obtained with Pool-seq. Thus, note that we measure the difference between two estimates of the allele frequencies in the sample, one based on the sampled individual genotypes and the other obtained with Pool-seq of the same sample. The *poolHelper* package provides functions to simulate Pool-seq data, under a variety of user-defined conditions. More specifically, users can vary the average and variance of the depth of coverage, the pool size, sequencing error and the pooling error (see below). Additionally, they can also vary the number of groups of individuals contributing to the final sequenced pool. By varying all of these conditions, it is possible to assess how they influence the accuracy of allele frequency estimations. No external R objects are needed to use the package. Users can use the implemented coalescent simulations to obtain genotypes, or provide genotypes directly. The resulting Pool-seq data can be output as R objects with counts of reads, or converted to commonly used file formats (.vcf and .sync), allowing users to analyse simulated Pool-seq data with existing downstream methods.

3.3.1 COALESCENT SIMULATIONS OF INDIVIDUAL GENOTYPES

To obtain individual genotypes, we used *scrm* to simulate coalescent gene trees under a model of a single population with constant effective size N_e. To model different effective population sizes and mutation rates, users can vary $\theta = 4N_e\mu$, where μ is the neutral mutation rate per locus per generation. This allows to investigate Pool-seq associated uncertainties in populations with varying levels of expected genetic diversity, which is proportional to θ. We assumed that the sample size was the same for each locus, corresponding to the total number of individuals sampled in the Pool-seq experiment. The effective size of the population from which the sample is taken is defined by θ, which users can modify. Additionally, we assumed that the actual haplotypes of all individuals in the pool were known. The effect of pooling is simulated in posterior steps (see next section). To obtain individual genotypes, we assumed random mating in the population and paired haplotypes at each locus at random for each biallelic single nucleotide polymorphic (SNP) site.

3.3.2 SIMULATION OF POOL-SEQ DATA

We follow a series of steps (Figure 3.1) to model and simulate allele frequencies obtained with Pool-seq for biallelic SNPs, as described in **?**. The variation in depth of coverage across SNPs is assumed to follow a negative binomial distribution (*nBin*, following e.g., Hardcastle and Kelly 2010). Thus, the number of reads c at each site is:

$$c \sim nBin(s, \psi) \tag{3.1}$$

where $s = mean(c)/var(c)$ and $\psi = mean(c)^2/var(c) - mean(c)$. The $mean(c)$ and $var(c)$ represent, respectively, the mean and variance of the depth of coverage across all SNPs. We assumed that the sequenced pool can have resulted from merging DNA extracted from K different groups of individuals, where each group could have a different number of individuals. To account for variability in the contribution of each individual to the pool, we assumed that the number of reads follows a multinomial-Dirichlet distribution. That is, at each site, reads from the i^{th} individual in the k^{th} group ($r_{k,i}$) follow a multinomial distribution:

$$r_{k,i} \sim mult(c, p_{k,i}) \tag{3.2}$$

where $p_{k,i}$ denotes the proportion of reads from individual i in group k, which is assumed to follow a Dirichlet distribution:

$$p_{k,i} \sim Dir\left(\rho_i \frac{1}{N}\right) \tag{3.3}$$

where N denotes the total number of sequenced individuals in group k, and ρ_i models the variance of contribution, reflecting the unequal contribution of individuals. Note that the contribution is expected to be equal for all individuals. If DNA extraction is performed for K multiple groups of individuals that are then combined into a larger pool, uneven contributions between these groups of individuals may also occur. To account for this, we modelled the unequal contribution of each group of individuals by assuming that the number of reads from the k^{th} group (r_k) follows a multinomial-Dirichlet distribution, such that $r_k \sim mult(c, p_k)$, where p_k is the proportion of reads from a given group, assumed to follow a Dirichlet distribution:

$$p_k \sim \ Dir\left(\rho_g \frac{n_k}{N}\right) \tag{3.4}$$

where n_k is the number of individuals in pool k, and ρ_g models the variance of contribution due to unequal contribution of groups of individuals. Following Gautier et al. (2013), we model explicitly the pooling error due to unequal contribution with the parameters ρ_i and ρ_g, which reflect the variance of contribution of individuals and groups of individuals, respectively, as:

$$\rho_i = \frac{N - 1 - \varepsilon_i^2}{\varepsilon_i^2} \tag{3.5}$$

$$\rho_g = \frac{(N/n_k) - 1 - \varepsilon_g^2}{\varepsilon_g^2} \tag{3.6}$$

where ρ_i and ρ_g are the unequal contribution parameters for individuals within a group, and among groups of individuals, respectively. All groups of individuals are assumed to have the same ρ_g, and all individuals are assumed to have the same ρ_i. These depend on pooling error parameters ε_i and ε_g for individuals and groups of individuals, respectively (Gautier et al., 2013). Larger values of ε_i and ε_g lead to a larger dispersion, resulting in more unequal contributions. The variance of contribution depends on the experimental error as $var(p_{k,i}) = \left(\varepsilon_i E[p_{k,i}]\right)^2$ and $var(p_k) = \left(\varepsilon_g E[p_k]\right)^2$. Although the selection of an appropriate pooling error might be potentially hard, given its unknown nature, we previously estimated values ranging from 24 to 236 (**?**). Furthermore, previous studies have also considered values ranging from 0 to 250 (Gautier et al., 2013). Thus, the pooling errors used here are within the reasonable ranges for this parameter (see Figure A3.1 for an example of how different pooling errors impact individual contribution). Note that the ε_i and ε_g which reflect the maximum dispersion, i.e., maximum unequal contribution when just one individual or one group of individuals contribute to the pool, correspond to ρ_i and ρ_g of zero. This implies that the upper limit for ε_i^2 is $N-1$ (eq. 3.5), and for ε_g^2 is $(N/n_k)-1$ (eq. 3.6). Users can use these values as a reference to determine the maximum error values based on their sample sizes.

We also accounted for sequencing and mapping errors by assuming that the reference allele R may be incorrectly called as an alternative allele A or vice versa with an error rate ε_{seq}. We modelled the number of reads A_i with the alternative allele for the i^{th} individual at a particular site following a binomial distribution: we assumed $A_i \sim \ Bin(r_{k,i}, \varepsilon_{seq})$ if the individual is homozygous for the reference allele and $A_i \sim \ Bin(r_{k,i}, 1-\varepsilon_{seq})$ if the individual is homozygous for the alternative allele. We also assumed that there are only two alleles at each site and that each base has an equal chance

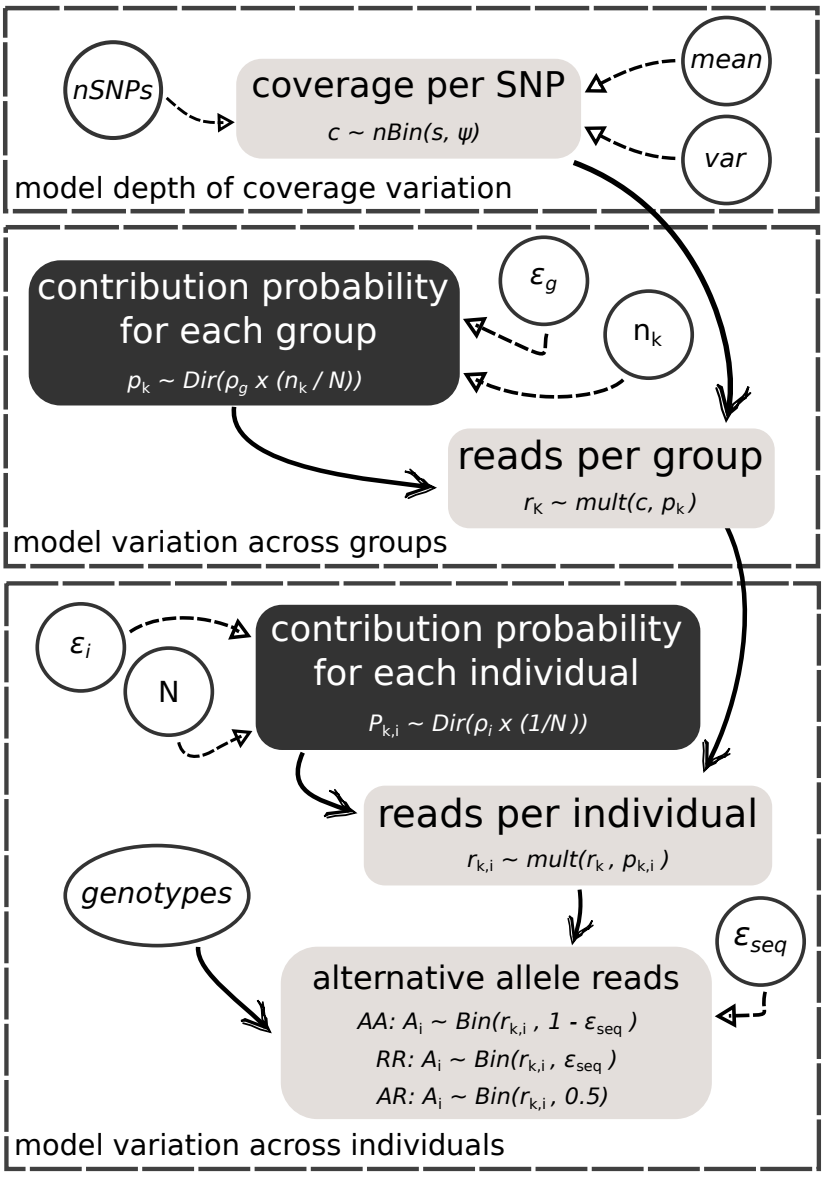


Figure 3.1: Diagram of the required steps to simulate Pool-seq data. The steps related to contribution probabilities are depicted by dark colored boxes, while circles represent the required inputs for each corresponding step. Each box contains the relevant formulas for its corresponding step.

of being miscalled. Therefore, for heterozygous individuals, each read originates from either the reference or alternative allele with equal probability (Li et al., 2012) and $A_i \sim Bin(r_{k,i}, 0.5)$, where $r_{k,i}$ represents the total number of reads contributed by an individual. A commonly used filter can also be applied, discarding SNPs with less than the required number of minor-allele reads. The allele frequencies estimated for the Pool-seq data correspond to the proportion of reads with the alternative allele.

3.3.3 MEASURING ERROR OF ESTIMATES

To measure the error of Pool-seq estimates of allele frequencies or expected heterozygosity, we compared the estimates obtained from the individual genotypes in the sample with the estimates obtained from Pool-seq. We calculate the mean absolute error as:

$$\varepsilon = \frac{1}{n} \times \sum |y_i - x_i| \tag{3.7}$$

where n indicates the total number of SNPs. When calculating the error of Pool-seq estimates of allele frequencies, x_i and y_i correspond to the frequencies of the alternative allele at the i^{th} SNP in the sample, obtained with individual genotypes (x_i) or with Pool-seq (y_i). When measuring the error of expected heterozygosity, x_i and y_i represent the expected heterozygosity obtained based on the sample of either individual genotypes (x_i) or Pool-seq (y_i).

3.4 MAIN FUNCTIONALITY

The *poolHelper* package allows users to compute the mean absolute error of allele frequencies and expected heterozygosity under a variety of conditions. Users can vary the mean depth of coverage and the associated variance, the value of the pooling error and the number of sampled individuals. Additionally, it is possible to evaluate the effect of combinations of parameters, for instance, various mean depths of coverage combined with several pooling error values. Thus, the *poolHelper* package provides users with a tool to aid in the design of pooled sequencing experiments, by allowing researchers to evaluate the best strategy, in terms of pool sizes or depth of coverage, to obtain accurate estimates of allelic frequencies, while minimising the sampling effort and costs.

3.4.1 EFFECT OF COMBINING MULTIPLE GROUPS OF INDIVIDUALS

An important consideration is whether DNA extraction should involve multiple groups of individuals, which are then combined into a final pool for library preparation and sequencing, or if DNA should be extracted individually from each sample and subsequently combined into a final pool. Users can test the effect of this choice by using the ”maePool” function. This function computes the mean absolute error for a given sample size sequenced using a pool with a single group of indi-

viduals or a pool combining multiple groups of individuals (supplementary Figure S2). By varying the mean coverage and the pooling error, it is possible to evaluate the effect of using a single or multiple groups under different conditions.

3.4.2 IMPACT OF MEAN DEPTH OF COVERAGE

Another critical decision is defining the mean depth of coverage used to sequence a pool of individuals. The ”maeFreqs” function implements the calculation of the mean absolute error between allele frequencies computed from genotypes and Pool-seq allele frequencies simulated under different mean depth of coverage. By varying the mean depth of coverage and the associated variance, users can determine which coverage produces more accurate allele frequency estimates for a given sample size and pooling error (Figure 3.2).

3.4.3 IMPACT OF POOL SIZES

When designing a Pool-seq experiment, it is essential to define the number of individuals to include in the pool, i.e., the pool size. The calculation of the mean absolute error between allele frequencies for different pools sizes can be carried out using the ”maeFreqs” function. This allows users to evaluate what is the optimal pool size for a fixed coverage and/or pooling error (Figure 3.2). Thus, the ”maeFreqs” function allows users to decide how many individuals to pool to obtain the most accurate allele frequencies estimates for a given mean depth of coverage.

3.4.4 EXAMPLE OF AN EFFECTIVE POOL-SEQ DESIGN USING SIMULATIONS

By performing simulations in a single panmictic population, assuming that pooling error is intermediate to high (150 or 300) and after applying a commonly used filter (removing sites with less than two minor-allele reads), it is not obvious that one should always increase the average depth of coverage per individual in the pool (Figure 3.2). For instance, when pooling error is 150, we observe the same mean absolute error with a pool of 50 individuals sequenced at 10x than with a pool of 10 individuals sequenced at 50x. This suggests that it may be more cost-effective to use a pool of 50 individuals at 10x (expected individual contribution of 10/50) than using fewer individuals with a higher expected coverage per individual. This holds true for larger pool sizes and

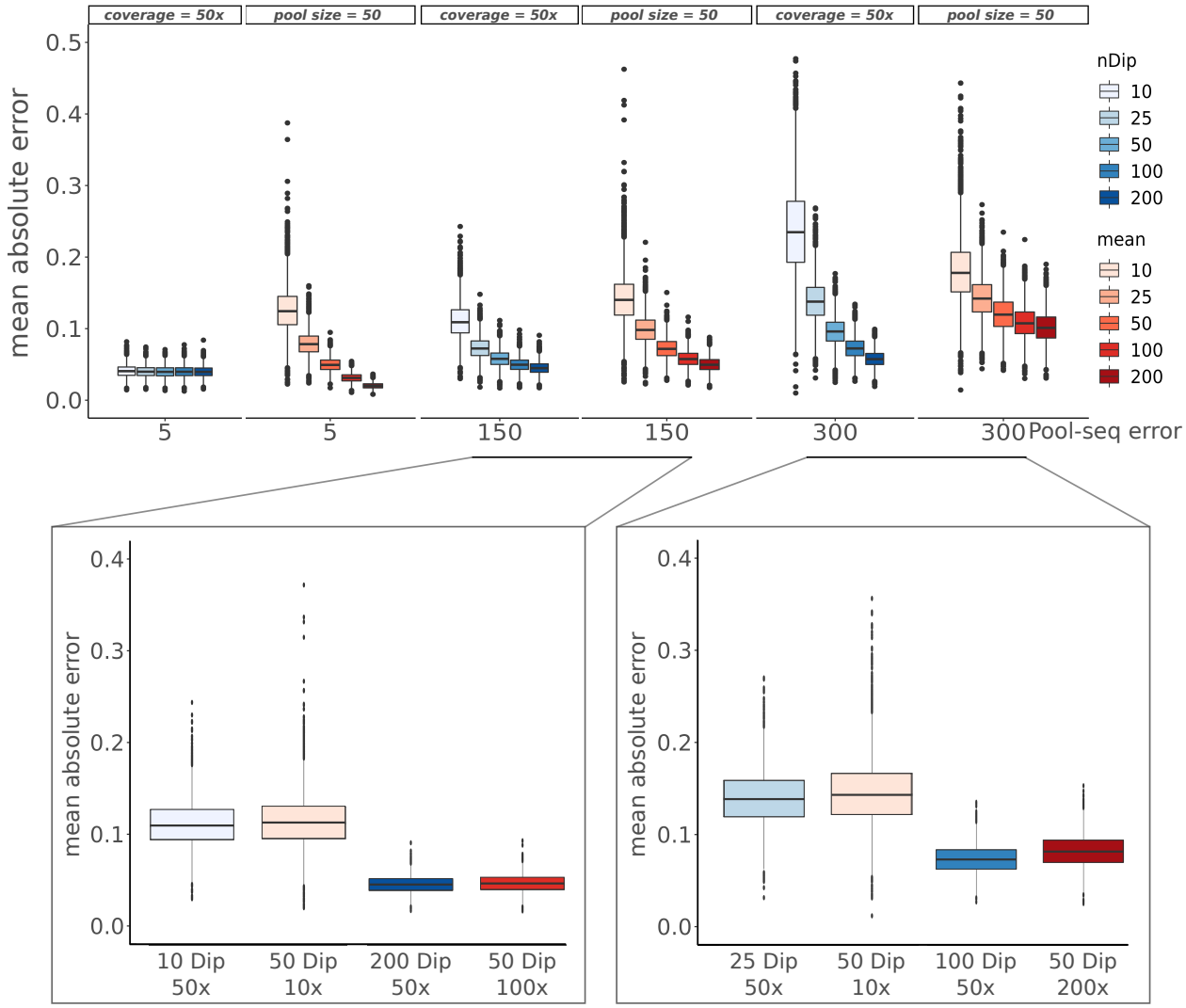


Figure 3.2: Mean absolute error between the allele frequencies obtained from the individual genotypes in the sample and those obtained from Pool-seq data under a variety of conditions. For all conditions, sites with less than two minor-allele reads were removed. In all plots, the y-axis represents the mean absolute error between the allele frequencies estimates. The top panel shows the mean absolute error for three different pooling error values (x-axis). For each plot, either the pool size or the coverage were fixed (the fixed value is indicated on the top of each plot). Thus, when pool size was fixed, the average coverage varied and vice-versa. In the bottom panel, we highlight comparisons that lead to similar mean absolute errors for intermediate values of pooling error (150 in the bottom left panel) and high pooling error (300 in the bottom right panel). In all plots, the pool size, defined by the *nDip* parameter, is represented in shades of blue, with darker shades indicating a larger pool and the average coverage, defined by the *mean* parameter, is represented in shades of red, with darker shades indicating higher coverage.

depths of coverage, given that we also get the same mean absolute error when comparing a pool of 200 individuals sequenced at 50x with a pool of 50 individuals sequenced at 100x (Figure 3.2). If pooling error is even higher (i.e., 300) a pool of 100 individuals sequenced at 100x leads to a slightly lower mean absolute error than a pool of 50 individuals sequenced at double the coverage (200x) (Figure 3.2). Thus, similar errors of allele frequencies in the sample can be obtained with different combinations of pool sizes and average depth of coverage. Therefore, the design of an effective Pool-seq study is not straightforward and an *a priori* simulation study can help assess an efficient sampling scheme to obtain accurate allele frequencies while minimising the sequencing effort (mean depth of coverage).

3.5 CONCLUSIONS

We present an R package, *poolHelper*, to simulate pooled sequencing data under a model of a single panmictic population, and compute the error in sample allele frequencies and expected heterozygosity obtained with Pool-seq for different study designs and commonly used filters (e.g., filters on minimum and maximum depth of coverage and on minimum number of minor-allele reads). The package relies on coalescent simulations performed with *scrm* (Staab et al., 2015). Currently, data is simulated under a single population with a constant effective population size. However, our package allows users to simulate genotypes under different models and use those genotypes as input to compute the mean absolute error or simulate Pool-seq data. This enables users to focus on their specific scenarios of interest and then simulate Pool-seq data under a wide range of user-defined parameters. This package is implemented in the R environment, providing tools for data visualisation, allowing users to produce graphics and quickly visualise the effect of multiple combinations of Pool-seq parameters. The *poolHelper* package's vignette contains a comprehensive explanation of the functions in the package, as well as examples detailing its usage.

3.6 ACKNOWLEDGEMENTS

We thank the editor and two anonymous reviewers for their comments and suggestions and Beatriz Portinha for suggesting the package name. This work was funded by the strategic project UIDB/00329/2020 granted to cE3c by Fundação para a Ciência e a Tecnologia (FCT). JC was supported by an FCT Ph.D. scholarship (PD/BD/128350/2017). RF is funded by a FCT CEEC (Fundação para a Ciência e a Tecnologia, Concurso Estímulo ao Emprego Científico) contract

(2020.00275.CEECIND) and by a FCT research project (PTDC/BIA-EVL/1614/2021). RKB was funded by the European Research Council (ERC-2015-AdG-693030-BARRIERS). VCS was supported by FCT (CEECINST/00032/2018/CP1523/CT0008) and by the Human Frontier Science Program (RGY0081/2020). We thank the National Network for Advanced Computing (RNCA) and INCD (`https://incd.pt/`) for use of the computing infrastructure, funded by FCT to VCS (2021.09795.CPCA).

3.7 REFERENCES

Anderson, E. C., Skaug, H. J., & Barshis, D. J. (2014). Next-generation sequencing for molecular ecology: a caveat regarding pooled samples. *Molecular Ecology*, *23*(3), 502-512. doi: 10.1111/mec.12609

Begun, D. J., Holloway, A. K., Stevens, K., Hillier, L. W., Poh, Y.-P., Hahn, M. W., ... others (2007). Population genomics: whole-genome analysis of polymorphism and divergence in *Drosophila simulans*. *PLoS biology*, *5*(11), e310. doi: 10.1371/journal.pbio.0050310

Collin, F.-d., Durif, G., Raynal, L., Lombaert, E., Gautier, M., Vitalis, R., ... Estoup, A. (2021). Extending approximate bayesian computation with supervised machine learning to infer demographic history from genetic polymorphisms using diyabc random forest. *Molecular Ecology Resources*, *21*(8), 2598-2613. doi: 10.1111/1755-0998.13413

Ellegren, H. (2014). Genome sequencing and population genomics in non-model organisms. *Trends in Ecology & Evolution*, *29*(1), 51-63. doi: 10.1016/j.tree.2013.09.008

Ferretti, L., Ramos-Onsins, S. E., & Pérez-Enciso, M. (2013). Population genomics from pool sequencing. *Molecular Ecology*, *22*(22), 5561-5576. doi: 10.1111/mec.12522

Gautier, M., Foucaud, J., Gharbi, K., Cézard, T., Galan, M., Loiseau, A., ... Estoup, A. (2013). Estimation of population allele frequencies from next-generation sequencing data: pool-versus individual-based genotyping. *Molecular Ecology*, *22*(14), 3766-3779. doi: 10.1111/mec.12360

Hardcastle, T. J., & Kelly, K. A. (2010). bayseq: empirical bayesian methods for identifying differential expression in sequence count data. *BMC bioinformatics*, *11*(422), 1-14. doi: 10.1186/1471-2105-11-422

Kofler, R., Pandey, R. V., & Schlötterer, C. (2011). Popoolation2: identifying differentiation between populations using sequencing of pooled dna samples (pool-seq). *Bioinformatics*, *27*(24), 3435-3436. doi: 10.1093/bioinformatics/btr589

Li, B., Chen, W., Zhan, X., Busonero, F., Sanna, S., Sidore, C., ... Abecasis, G. R. (2012). A likelihood-based framework for variant calling and de novo mutation detection in families. *PLOS Genetics*, *8*(10), 1-12. doi: 10.1371/journal.pgen.1002944

Morales, H. E., Faria, R., Johannesson, K., Larsson, T., Panova, M., Westram, A. M., & Butlin, R. K. (2019). Genomic architecture of parallel ecological divergence: beyond a single

environmental contrast. *Science advances*, *5*(12), eaav9963. doi: 10.1126/sciadv.aav9963

Prescott, N. J., Lehne, B., Stone, K., Lee, J. C., Taylor, K., Knight, J., ... Consortium, U. I. G. (2015). Pooled sequencing of 531 genes in inflammatory bowel disease identifies an associated rare variant in btnl2 and implicates other immune related genes. *PLOS Genetics*, *11*(2), 1-19. doi: 10.1371/journal.pgen.1004955

Ross, P. A., Endersby-Harshman, N. M., & Hoffmann, A. A. (2019). Shedding light on the grey zone of speciation along a continuum of genomic divergence. *Evolutionary applications*, *12*(3), 572-586. doi: doi.org/10.1111/eva.12740

Schlötterer, C., Tobler, R., Kofler, R., & Nolte, V. (2014). Sequencing pools of individuals — mining genome-wide polymorphism data without big funding. *Nature Reviews Genetics*, *15*(11), 749-763. doi: 10.1038/nrg3803

Staab, P. R., Zhu, S., Metzler, D., & Lunter, G. (2015). scrm: efficiently simulating long sequences using the approximated coalescent with recombination. *Bioinformatics*, *31*(10), 1680-1682. doi: 10.1093/bioinformatics/btu861

Taus, T., Futschik, A., & Schlötterer, C. (2017). Quantifying selection with pool-seq time series data. *Molecular biology and evolution*, *34*(11), 3023-3034. doi: 10.1093/molbev/msx225

Team, R. C. (2020). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `https://www.R-project.org/`

Zhou, D., Udpa, N., Gersten, M., Visk, D. W., Bashir, A., Xue, J., ... Haddad, G. G. (2011). Experimental selection of hypoxia-tolerant *Drosophila melanogaster*. *Proceedings of the National Academy of Sciences*, *108*(6), 2349-2354. doi: 10.1073/pnas.1010643108

3.8 APPENDIX

Table A3.1: **Comparison of different tools to simulate Pool-seq data.** We compared the functionalities of three different packages/tools that can generate simulated Pool-seq data. We assessed if they can be used to model known sources of Pool-seq uncertainty, such as variation in depth of coverage, unequal individual contribution and sequencing errors.

Package/Tool	Variation in depth of coverage?	Unequal individual contribution?	Sequencing error?
poolHelper	Yes. Negative binomial with mean and variance.	Yes. Multinomial-Dirichlet distribution with explicit pool error parameters.	Yes. Binomial with same sequencing error rate for all substitutions.
poolSeq	Yes. Poisson with mean coverage.	Not explicitly. Reads are generated by binomial sampling.	Not explicitly modelled.
DYABC - RF	Yes. Poisson with mean coverage.	Not explicitly. Reads sampled from binomial parameterized with simulated allele counts and coverage.	Not explicitly modelled.

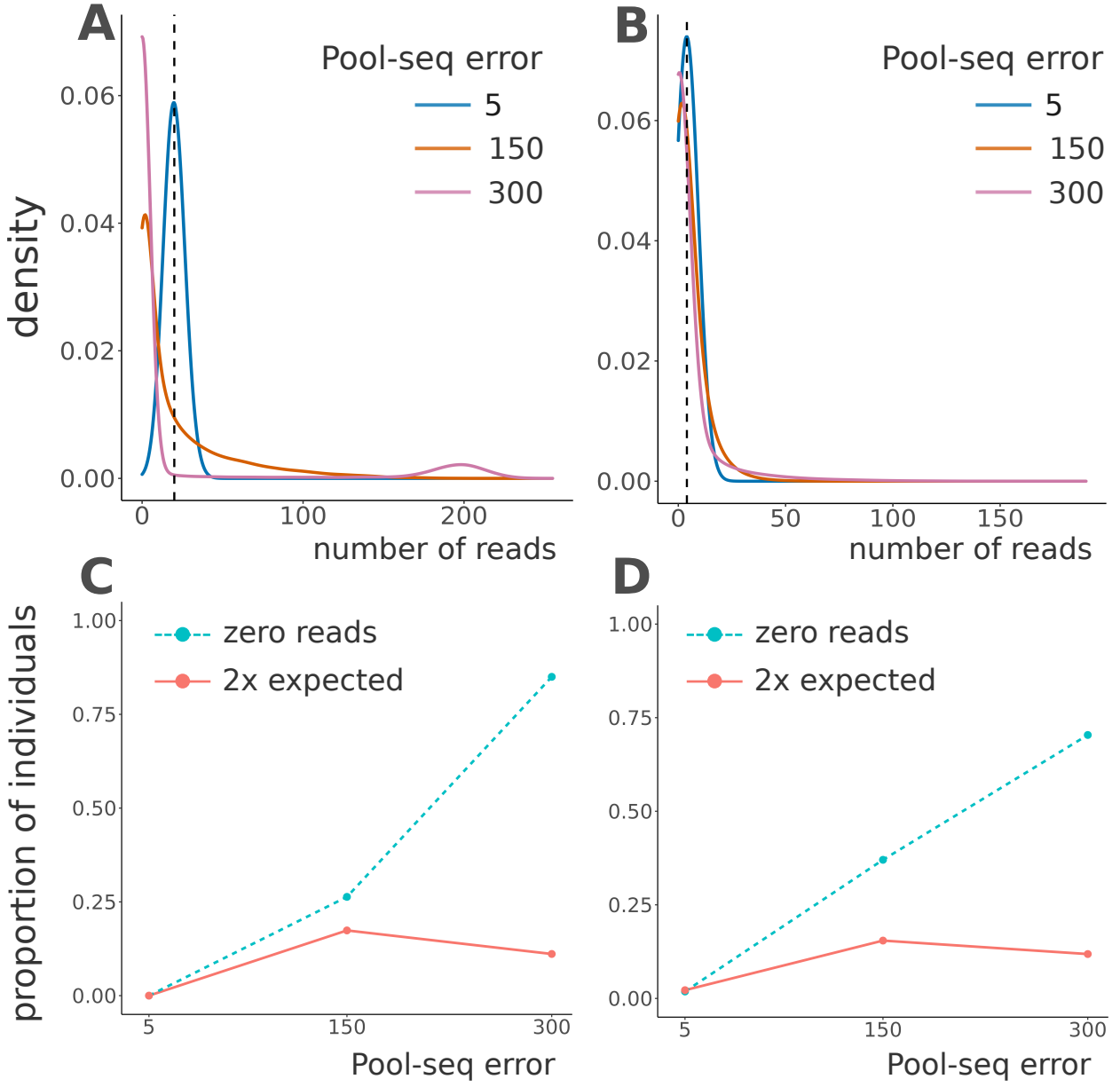


Figure A3.1: Impact of pooling errors in individual contribution. We simulated Pool-seq data obtained with pools of either 10 or 50 individuals, sequenced at a coverage of 200x. Three different levels of pooling errors were considered: 5, 150, and 300. In A and B, the expected contribution of each individual is indicated by the dashed line. The distribution of the number of reads contributed by each individual for a pool of 10 (A) or 50 (B) individuals shows that higher pooling errors result in deviations from the expected value and an increased number of individuals with zero (or near-zero) reads. Note that, with a pooling error of 300, some individuals contributed ∼200 reads. The impact of higher pooling errors is also clear when we analyze the proportion of individuals contributing zero reads or twice the expected number of reads for pools of both 10 (C) and 50 (D) individuals.

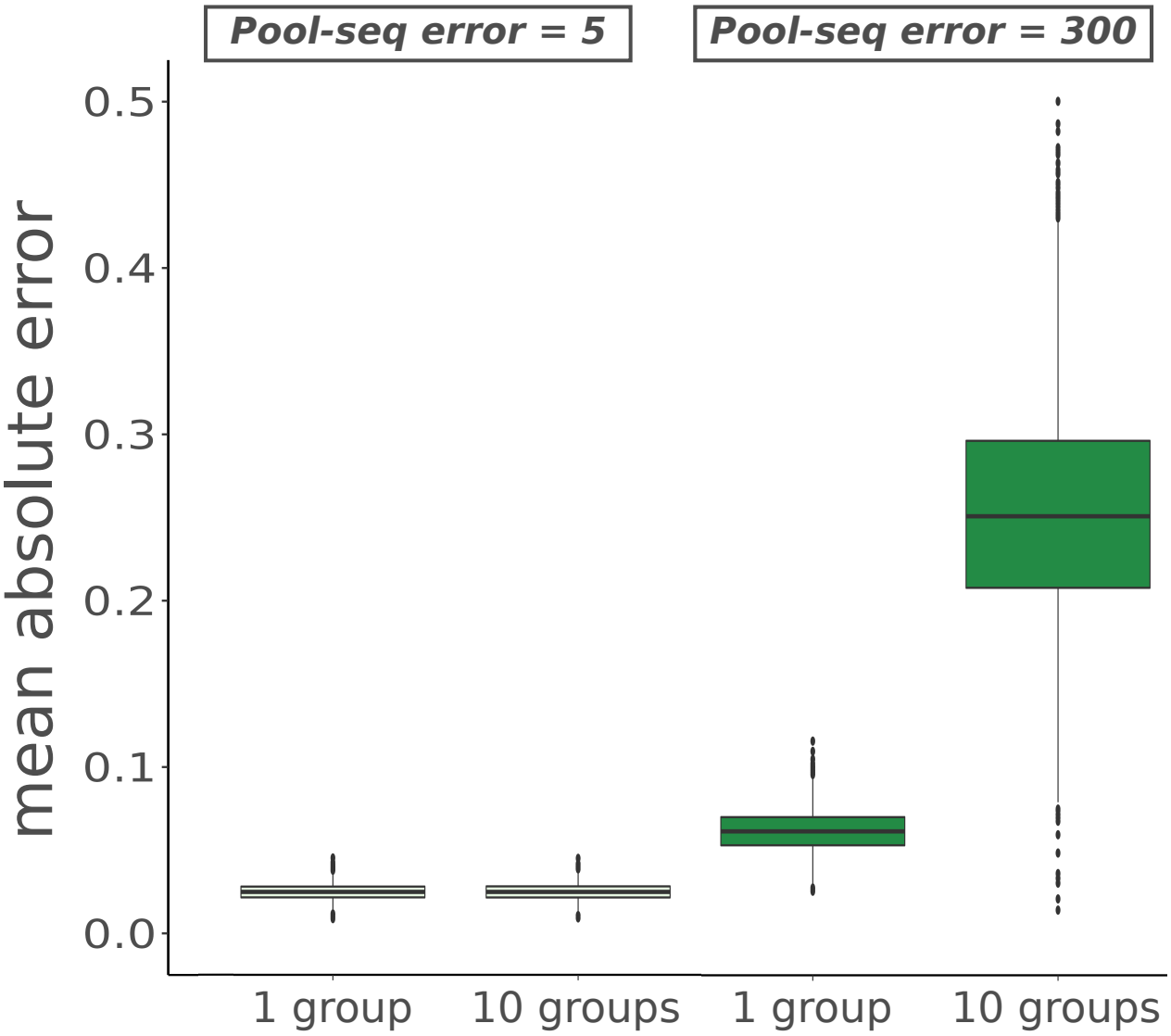


Figure A3.2: Mean absolute error between the allele frequencies obtained from the individual genotypes in the sample and those obtained from Pool-seq data using a single or multiple groups of individuals. Pool-seq data were simulated for 100 individuals. Sequencing was performed with an average coverage of 100x using either a single group of individuals or a pool combining 10 groups of individuals, with each group containing 10 individuals. Two scenarios were considered: one assuming a low pooling error rate of 5, and the other assuming a high pooling error rate of 300. The same pooling error value was used to model the dispersion among pools and individuals. The y-axis represents the mean absolute error between the allele frequencies estimates and the x-axis indicates the number of groups used to sequence the sample.

3.8.1 Vignette for the poolHelper package

A3 - Introduction

A method to simulate pooled sequencing data (Pool-seq) is implemented in the R language in the package `poolHelper`. The aim of this package is to provide users with a tool to chose the appropriate pool size and depth of coverage when conducting experiments that require pool sequencing. This vignette serves as a introduction, explaining how the different functions of the package can be used to assess the impact of different sequencing parameters.

At the end, we also included a section with details of specific functions. At that section, users can find a detailed step-by-step description of how to simulate Pool-seq data. The various subsections describe how to simulate the total depth of coverage and then partition that coverage among different pools and individuals. There is also a subsection describing how the number of reads with the reference allele can be computed according to the genotype of a given individual.

```
library(poolHelper)
```

With the `poolHelper` package, users can evaluate the effect of different pool errors, pool sizes and depths of coverage on the allele frequencies. The frequencies obtained with Pool-seq are compared to the allele frequencies computed directly from genotypes.

A3 - Basic functionality

Briefly, we use `scrm` to simulate a single population at equilibrium and obtain polymorphic sites for each simulated locus. Then, we compute allele frequencies by counting the total number of derived alleles per site and dividing that by the total number of gene copies. After obtaining the allele frequencies computed directly from genotypes, we simulate Pool-seq data and obtain the Pool-seq allele frequencies. Details on this procedure can be found in the last section of this vignette.

We then use the `mae` function from the `Metrics` package to compute the average absolute difference between the Pool-seq allele frequencies and the ones obtained directly from the genotypes. Mean Absolute Error (MAE) is calculated as the sum of absolute errors divided by the sample size.

A3 - Pool-seq experimental design

As mentioned, the main goal of the package `poolHelper` is to provide users with a tool to aid in the experimental design of pooled sequencing. Researchers interested in Pool-seq are concerned in obtaining accurate estimates of allelic frequencies, while keeping the costs down. Thus, it is important to have an idea of how accurate the allele frequencies can be when using different pool sizes or sequencing at different mean coverage values. In the following sections we detail how the `poolHelper` package can help users in answering those questions.

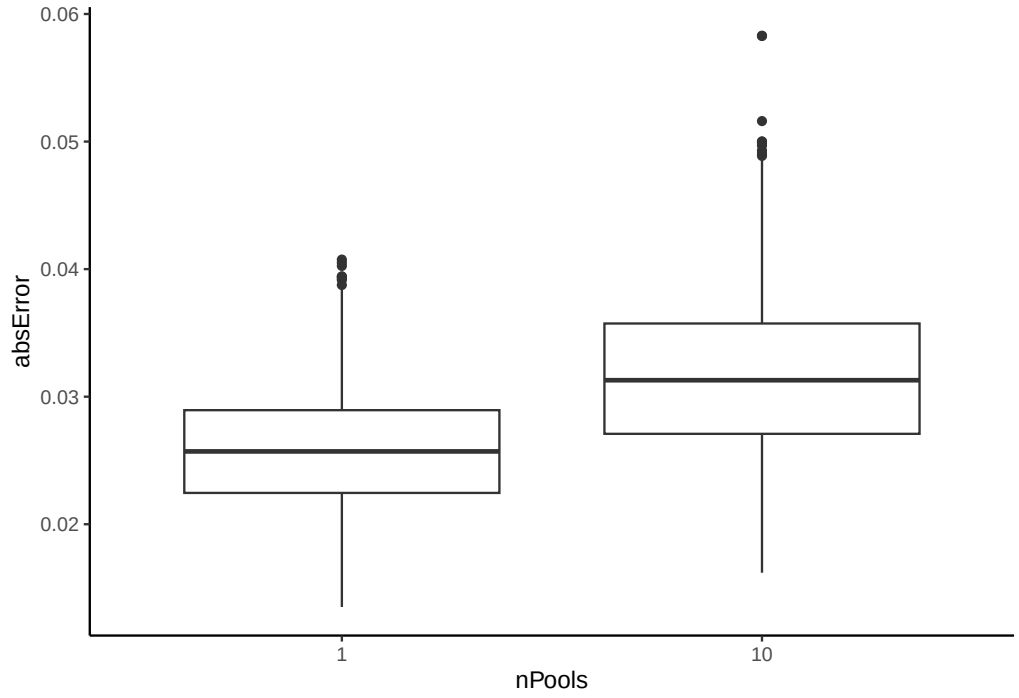
A3 - How many pools should I use?

One important aspect to consider is whether DNA extraction should be done using multiple batches of individuals, combining several of them into larger pools for library preparation and sequencing, or using a single batch of individuals. By using the `maePool` function we can check, under different conditions, what is the effect of using multiple or a single batch of individuals.

The `pools` input argument allows the user to simulate a single pool, by creating a list with a single integer or multiple pools, by creating a list with a vector containing various entries. The `maePool` function assumes that each entry of that vector is the size, in number of diploids individuals, of a given pool.

```
# create a list with a single pool of 100 individuals
pools <- list(100)
# compute average absolute difference between allele frequencies
onePool <- maePool(nDip = 100, nloci = 1000, pools = pools, pError = 100,
                   sError = 0.01, mCov = 100, vCov = 250, min.minor = 0)

# create a list with 10 pools, each with 10 individuals
pools <- list(rep(10, 10))
# compute average absolute difference between allele frequencies
tenPool <- maePool(nDip = 100, nloci = 1000, pools = pools, pError = 100,
                   sError = 0.01, mCov = 100, vCov = 250, min.minor = 0)

# combine both
final <- rbind(onePool, tenPool)
# convert the number of individuals in the pool to a factor
final$nPools <- as.factor(final$nPools)
```

```r
# load the ggplot package
library(ggplot2)
# MAE value in the y-axis
# and the number of individuals in the pool in the x-axis
ggplot(final, aes(x = nPools, y = absError)) +
  geom_boxplot() + theme_classic()
```



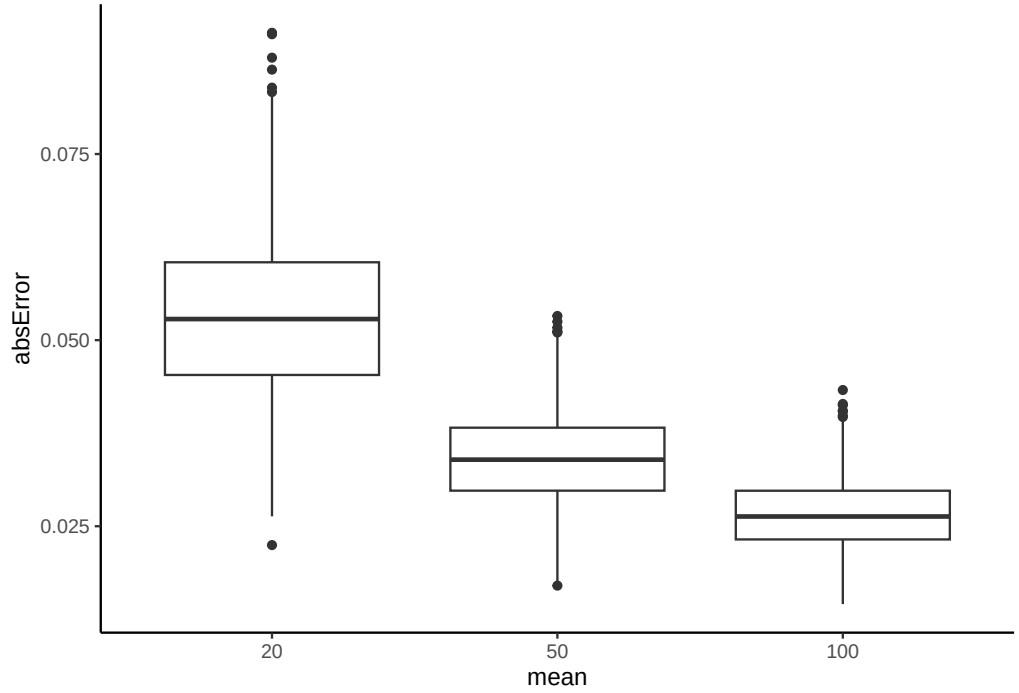
In this example, we can see the effect of using a single or multiple pools when a sample of 100 individuals is sequenced at a mean coverage of 100x and for a given pool error. By varying the `pError` and `mCov` input arguments, users can evaluate the effect of using a single or multiple pools at various pool error values and at different coverages.

A3 - What coverage should I use?

Another fundamental decision is what mean coverage should we try to obtain when sequencing a pool of individuals. By using the `maeFreqs` function we can look at the average absolute difference between genotype allele frequencies and Pool-seq allele frequencies obtained using different mean coverages.

```r
# create a vector with various mean coverages
mCov <- c(20, 50, 100)
```

```r
# create a  vector with the variance of the coverage
vCov <- c(100, 250, 500)

# compute average absolute difference between allele frequencies
mydf <- maeFreqs(nDip = 100, nloci = 1000, pError = 100, sError = 0.01,
                 mCov, vCov, min.minor = 0)

# convert the mean coverage into a factor
mydf$mean <- as.factor(mydf$mean)
# boxplot the MAE value in the y-axis and the coverage in the x-axis
ggplot(mydf, aes(x = mean, y = absError)) +
  geom_boxplot() + theme_classic()
```

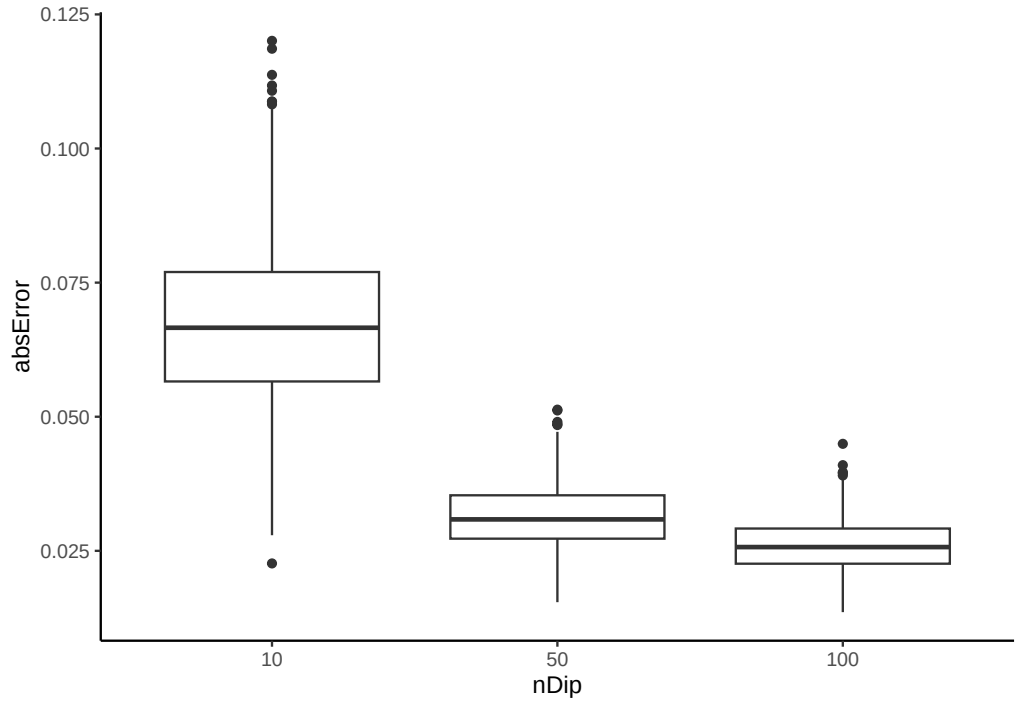


Note that the `mCov` input argument is a vector with various mean coverage values. The `maeFreqs` function computes the average absolute difference for each user-defined coverage. Additionally, `vCov` should also be a vector, with each entry being the variance of the corresponding coverage in `mCov`. In this example, we can see the effect of sequencing a sample of 100 individuals at 20x, 50x or 100x mean coverage. By varying the `mCov` or `pError` input arguments, users can evaluate the impact of different mean coverages at various pool error values.

A3 - What pool size should I use?

It is also important to define the number of individuals to sequence or, in other words, the pool size. The `maeFreqs` function can also be used to compute the average absolute difference between the allele frequencies computed from genotypes and Pool-seq allele frequencies obtained with different pool sizes.

```
# create a vector with various mean coverages
nDip <- c(10, 50, 100)

# compute average absolute difference between allele frequencies
mydf <- maeFreqs(nDip = nDip, nloci = 1000, pError = 100, sError = 0.01,
                 mCov = 100, vCov = 250, min.minor = 0)

# convert the number of individuals into a factor
mydf$nDip <- as.factor(mydf$nDip)
# boxplot the MAE value in the y-axis and the coverage in the x-axis
ggplot(mydf, aes(x = nDip, y = absError)) +
  geom_boxplot() + theme_classic()
```



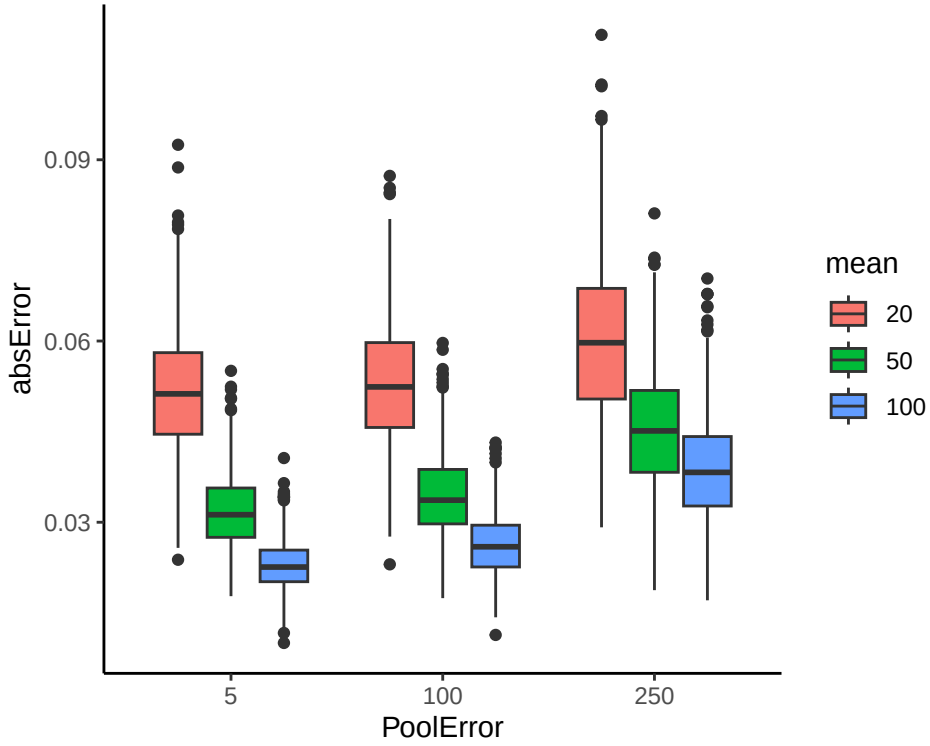
As you can see, by varying the `nDip` input argument, we can evaluate what is the optimal pool size. In this example, we can see the effect of sequencing a sample of 10, 50 or 100 individuals at 100x coverage. For this coverage and pool error value, it is clear that doubling the pool size, from 50 to

100 individuals, does not lead to a significant decrease in the average absolute difference between allele frequencies. The `maeFreqs` function assumes that only a single pool was used to sequence the population and so, for this example, a single pool of 10, 50 or 100 individuals was used.

A3 - How to test different combinations?

The `maeFreqs` function can also be used to simultaneously test different combinations of parameters. By varying the `mCov`, `pError` and/or `nDip` input arguments, the impact of multiple combinations of those parameters can be quickly assessed. The `maeFreqs` function will simulate all possible combinations of those parameters and compute the average absolute difference between allele frequencies.

```
# create a vector with various mean coverages
mCov <- c(20, 50, 100)
# create a  vector with the variance of the coverage
vCov <- c(100, 250, 500)
# create a vector with various pool errors
pError <- c(5, 100, 250)

# compute average absolute difference between allele frequencies
mydf <- maeFreqs(nDip = 100, nloci = 1000, pError, sError = 0.01, mCov,
                 vCov, min.minor = 0)

# convert the mean coverage into a factor
mydf$mean <- as.factor(mydf$mean)
# convert the pooling error to a factor
mydf$PoolError <- as.factor(mydf$PoolError)

# boxplot the MAE value in the y-axis and the pool error in the x-axis
# producing one boxplot for each of the different coverages
ggplot(mydf, aes(x = PoolError, y = absError, fill = mean)) +
  geom_boxplot() + theme_classic()
```



In this example, the number of sampled individuals was kept constant, meaning that the population was always sequenced using a pool of 100 individuals. Those 100 individuals were sequenced at 20x, 50x or 100x mean coverage and assuming a pool error value of 5%, 100% or 250%. By selecting multiple combinations of parameters, users can select a sequencing design that minimizes the average absolute difference between allele frequencies or get an idea of how much mismatch to expect in their Pool-seq data.

A3 - Simulate Pool-seq data

The `poolHelper` package can also be used to simulate Pool-seq data without computing the average absolute difference. Thus, it is possible to use the package to simply obtain simulated Pool-seq data. This Pool-seq data is simulated for a given set of genotypes and so users should provide genotypes. Those genotypes can be obtained with coalescent-based or other type of simulators and using different demographic models. Pool-seq data can be simulated under a variety of parameter combinations, such as different pool sizes and mean coverage.

The `simPoolseq` function is used to simulate pooled sequencing data given a set of parameters and individual genotypes. Consider the following example:

```
# simulate genotypes for 100 individuals sampled at 5 loci
genotypes <- run_scrm(nDip = 100, nloci = 5, theta = 5)
```

```
# simulate Pool-seq data assuming a coverage of 100x
# and a single pool of 100 individuals
pool <- simPoolseq(genotypes = genotypes, pools = 100, pError = 100,
                   sError = 0.001, mCov = 100, vCov = 250, min.minor = 0)

# check the structure of the pool object
str(pool)
#> List of 3
#>  $ reference  :List of 5
#>   ..$ : num [1, 1:21] 90 98 95 85 91 88 70 83 91 108 ...
#>   ..$ : num [1, 1:40] 23 61 101 107 114 29 101 89 53 79 ...
#>   ..$ : num [1, 1:35] 113 91 109 80 4 100 57 92 89 79 ...
#>   ..$ : num [1, 1:20] 104 69 84 123 81 82 92 99 87 92 ...
#>   ..$ : num [1, 1:28] 102 108 87 87 94 85 92 80 108 101 ...
#>  $ alternative:List of 5
#>   ..$ : num [1, 1:21] 0 1 1 5 3 0 27 3 8 1 ...
#>   ..$ : num [1, 1:40] 68 29 3 0 3 62 3 6 60 0 ...
#>   ..$ : num [1, 1:35] 3 3 2 1 86 2 22 1 11 3 ...
#>   ..$ : num [1, 1:20] 0 0 12 2 0 2 1 8 0 4 ...
#>   ..$ : num [1, 1:28] 0 0 1 31 11 1 8 1 0 0 ...
#>  $ total      :List of 5
#>   ..$ : num [1, 1:21] 90 99 96 90 94 88 97 86 99 109 ...
#>   ..$ : num [1, 1:40] 91 90 104 107 117 91 104 95 113 79 ...
#>   ..$ : num [1, 1:35] 116 94 111 81 90 102 79 93 100 82 ...
#>   ..$ : num [1, 1:20] 104 69 96 125 81 84 93 107 87 96 ...
#>   ..$ : num [1, 1:28] 102 108 88 118 105 86 100 81 108 101 ...
```

The simulated Pool-seq data is organized as a list with three named entries: `reference`, `alternative` and `total`. Note that each of those entries has 5 different entries because we simulated 5 loci. Thus, each of the main list entries contains one entry per locus. Each of those entries is a matrix where column represents a different site. The `reference` entry contains the list with the number of reference allele reads, the `alternative` entry contains the list with number of alternative allele reads and the `total` entry contains the total depth of coverage per site.

Users can vary the pooling error (`pError`), the sequencing error (`sError`), the mean (`mCov`) and variance (`vCov`) of the coverage. It is also possible to filter the simulated Pool-seq data by selecting

a value for the `min.minor` input. This value should be an integer representing the minimum allowed number of minor-allele reads. Sites that, across all populations, have less minor-allele reads than this threshold will be removed from the data. Additionally, it is also possible to define an `minimum` and `maximum` input arguments. These optional arguments will define the minimum and maximum coverage allowed. Sites where the coverage is below or above those thresholds will be removed from the data. For instance:

```
# simulate genotypes for 100 individuals sampled at 50 loci
genotypes <- run_scrm(nDip = 100, nloci = 50, theta = 5)

# simulate Pool-seq data assuming a coverage of 100x
# and a single pool of 100 individuals
# remove all sites with a coverage below 80x or above 115x
pool <- simPoolseq(genotypes = genotypes, pools = 100, pError = 100,
                   sError = 0.001, mCov = 100, vCov = 500, min.minor = 0,
                   minimum = 80, maximum = 115)

# check the minimum and maximum coverage
range(unlist(pool$total))
#> [1]  80 115
```

The previous chunk will remove all sites with a depth of coverage below 80x and above 115x. Thus, all the remaining sites will have a coverage comprised between those values.

A3 - Convert to other formats

The simulated Pool-seq data can be converted to other commonly used file formats, specifically the `.vcf` and `.sync` formats. This allows users to simulate Pool-seq data, using different combinations of parameters and genotypes simulated under different demographic scenarios, convert the simulated Pool-seq data into `.vcf` or `.sync` and then use those files to analyse simulated Pool-seq data with existing downstream methods.

The `poolHelper` package includes the `pool2vcf` and the `pool2sync` functions to convert the simulated Pool-seq data into `.vcf` or `.sync` files, respectively. Note that both those functions will create and save the file in the current working directory. Please refer to the manual for more details on the functioning of both functions.

A3 - Details on specific functions

In this section and until the end of the vignette, we go over the steps required to simulate Pool-seq data and give details on some of the specific functions included in the package.

A3 - Simulate depth of coverage

The `simulateCoverage` function can be used to simulate the total depth of coverage at each site. The `mean` and `variance` input arguments of the function represent, respectively the mean coverage and the variance of the coverage to simulate. `nLoci` represents the number of independent loci to simulate and `nSNPs` is the number of polymorphic sites to simulate per locus.

```
# simulate number of reads for one population
reads <- simulateCoverage(mean = 50, variance = 250, nSNPs = 100, nLoci = 1)
# display the structure of the reads object
str(reads)
#> List of 1
#>  $ : int [1, 1:100] 63 49 38 75 56 44 92 61 70 66 ...
```

As you can see, the resulting output is a list with one entry because `nLoci = 1`. That entry is a vector with `length = 100` because that was the number of `nSNPs`. We can also use this function to simulate the coverage of multiple populations at the same time. To do that, the `mean` and `variance` input arguments of the function should be vectors. The function will assume that each entry of those vectors is the mean and variance of a different population. For instance, in the next example we set `mcov <- c(50, 100)`, meaning that we wish to simulate two populations, the first with a mean coverage of 50x and the second with a mean coverage of 100x.

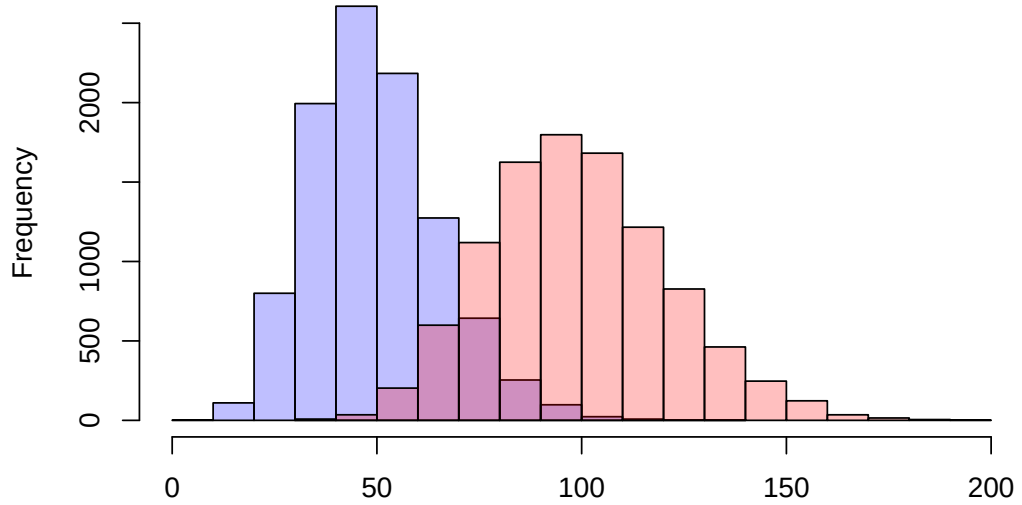
```
# create a vector with the mean coverage of each population
mcov <- c(50, 100)
# create a vector with the variance of the coverage for each population
vcov <- c(250, 500)
# simulate number of reads for two populations
reads <- simulateCoverage(mean=mcov, variance=vcov, nSNPs=100, nLoci=1)
# display the structure of the reads object
str(reads)
#> List of 1
#>  $ : int [1:2, 1:100] 50 154 46 111 57 99 59 95 70 73 ...
```

Now, the output of the function is slightly different. We still have a single locus (`nLoci = 1`)

and 100 sites on that locus (`nSNPs = 100`) but now that one list entry is a matrix with two rows. Each row is the coverage per site for one population. Thus, the first row is the coverage for the first population of the `mcov` input argument and the second row is the coverage for the second population in that argument. If `mcov` had the mean coverage for more populations, the logic would remain the same.

The difference in the mean coverage of the two population can be quickly visualized. In the following we simulate two populations, one with 50x mean coverage and the other with 100x. We set `nSNPs = 10000` and visualize the coverage distribution using a histogram.

```
# create a vector with the mean coverage of each population
mcov <- c(50, 100)
# create a vector with the variance of the coverage for each population
vcov <- c(250, 500)
# simulate number of reads for two populations
reads <- simulateCoverage(mean=mcov, variance=vcov, nSNPs=10000, nLoci=1)
# plot the coverage of the first population
hist(reads[[1]][1,], col=rgb(0,0,1,1/4), xlim=c(0, 200), main="", xlab="")
# add the coverage of the second population
hist(reads[[1]][2,], col = rgb(1,0,0,1/4), add = TRUE)
```



The coverage distribution of the population simulated with a mean of 50x is shown in blue and the distribution of the 100x population is shown in red.

It is also possible to remove sites with low or high coverage by using the `remove_by_reads` function. This function will completely remove any site from the data (in this instance, the site will be removed from both populations). Sites will be removed if their coverage is below the `minimum` allowed or if it is above the `maximum` allowed. In the next bit, we use the `reads` simulated before

and remove all sites with a coverage below 25x and above 150x.

```
# check the minimum and maximum coverage before removal
x <- range(unlist(reads))
# remove sites with coverage below 25x and above 150x
reads <- remove_by_reads(nLoci=1, reads=reads, minimum=25, maximum=150)
# display the structure of the reads object after removal
str(reads)
#> List of 1
#>  $ : int [1:2, 1:9512] 59 106 38 76 28 109 65 113 25 108 ...
# check the minimum and maximum coverage after removal
range(unlist(reads))
#> [1]  25 150
```

Accordingly, the minimum simulated coverage before running the `remove_by_reads` function was `8` and the maximum was `197` but after removal of sites with a coverage below 25x and above 150x, the minimum and maximum coverage are, obviously, `25` and `150` respectively. It is also clear that we no longer have `nSNPs = 10000` in the data.

A3 - Reads contributed by each pool

It is also possible to simulate the contribution of each pool, assuming that a single population was sequenced using multiple pools. Before computing the actual number of reads contributed by each pool, we first need to simulate the proportion of contribution.

To do this, we use the `poolProbs` function. The `nPools` input argument of this function should represent the number of pools used to sequence the population, while the `vector_np` contains the number of individuals per pool. Thus, in the following example `vector_np = c(10, 10, 10, 10)` means that four pools were used to sequence the population, each comprised of 10 individuals. The `pError` input argument defines the degree of pooling error. Briefly, this pooling error controls the dispersion of the pool contribution, centred around the expected value. Higher values of `pError` lead to a higher dispersion and thus, the contributions will vary more between pools. In other words, with higher values of `pError` some pools will contribute a lot of reads and others will not contribute much.

In the next chunk, we see the difference in proportion of contribution when 4 pools of 10 individuals were used to sequence a single population and the pooling error is either low (`pError = 5`) or high (`pError = 250`). We can also assess the impact of different pool sizes by including one pool with 100 individuals instead of only 10.

```
# four pools with low sequencing error
poolProbs(nPools = 4, vector_np = c(10, 10, 10, 10), nSNPs = 6, pError = 5)
#>           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
#> [1,] 0.2770748 0.2645803 0.2617985 0.2469693 0.2528087 0.2428078
#> [2,] 0.2592213 0.2486401 0.2489134 0.2447933 0.2686069 0.2335425
#> [3,] 0.2270433 0.2521062 0.2465722 0.2463870 0.2399388 0.2467867
#> [4,] 0.2366606 0.2346734 0.2427159 0.2618504 0.2386456 0.2768629
# four pools with high sequencing error
poolProbs(nPools = 4, vector_np = c(10, 10, 10, 10), nSNPs = 6, pError = 250)
#> Warning: pError was too high. It was replaced by 163.205080756888
#>              [,1]         [,2]         [,3]         [,4]         [,5]
#> [1,] 9.997940e-01 2.745366e-35 1.515829e-15 3.256760e-07 9.999994e-01
#> [2,] 1.519048e-30 1.269217e-01 9.999989e-01 9.999997e-01 6.293280e-07
#> [3,] 1.494290e-15 4.255151e-16 2.971420e-17 1.078756e-34 1.153420e-27
#> [4,] 2.060070e-04 8.730783e-01 1.110925e-06 3.860350e-09 4.846799e-14
#>              [,6]
#> [1,] 1.688829e-47
#> [2,] 1.000000e+00
#> [3,] 4.842499e-17
#> [4,] 4.846993e-14
# four pools but one is much larger
poolProbs(nPools = 4, vector_np = c(10, 100, 10, 10), nSNPs = 6, pError = 5)
#>           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
#> [1,] 0.07698561 0.08073943 0.07613430 0.08575555 0.07973231 0.07558553
#> [2,] 0.77689898 0.76427916 0.77173400 0.75609041 0.76781451 0.76700344
#> [3,] 0.07624493 0.07663565 0.07855448 0.08175773 0.07905299 0.08562723
#> [4,] 0.06987047 0.07834576 0.07357723 0.07639631 0.07340019 0.07178380
```

The output of the `poolProbs` function is a matrix with the proportion of contribution for each pool. Each row of the matrix corresponds to a different pool and each column is a different site. You can see that in the first example, the proportion of contribution is roughly the same for all pools. The next example is similar but with `pError = 250`. With this higher pool error, it is clear that some pools have a higher proportion of contribution and others have a smaller. Thus, with higher pool errors, the proportion of contribution is no longer the same for all pools. This also happens when pool error is low but one of the pools is much larger. In the last example, the second pool has 100 individuals, while the other pools only have 10. In this instance, it is clear the the proportion of

contribution of the larger pool is always higher.

After computing the proportion of contribution of each pool, this can be used to simulate the actual number of reads contributed by each pool. To do this, we use the `pReads` function. This functions requires as input argument the total number of pools used to sequence the population (`nPools`), a vector with the total `coverage` per site and the probabilities of contribution computed with the `poolProbs` function (`probs`). In the next chunk, we simulate coverage for 10 SNPs of a single population, compute the probability of contribution for 4 pools used to sequence that population and then simulate the actual number of reads per pool.

```
# simulate total coverage per site
reads <- unlist(simulateCoverage(mean=100, variance=250, nSNPs=10, nLoci=1))
# compute the proportion of contribution of each pool
probs <- poolProbs(nPools=4, vector_np=rep(10, 4), nSNPs=10, pError=5)
# simulate the contribution in actual read numbers
pReads <- poolReads(nPools = 4, coverage = reads, probs = probs)
# output the number of reads per pool and per site
pReads
#>      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
#> [1,]   25   20   17   32   30   21   19   36   28    28
#> [2,]   26   19   25   36   17   23   35   22   15    28
#> [3,]   23   28   21   36   27   25   40   38   21    21
#> [4,]   14   24   15   28   22   27   26   31   25    20
```

It is clear that, when pool error is quite low (`pError = 5` in the previous chunk), the number of reads contributed by each pool is quite similar. Thus, the total coverage of any given site is well distributed among all pools. On the other hand, if pool error is high (`pError = 250` in the next chunk).

```
# simulate total coverage per site
reads <- unlist(simulateCoverage(mean=100, variance=250, nSNPs=10, nLoci=1))
# compute the proportion of contribution of each pool
probs <- poolProbs(nPools=4, vector_np=rep(10, 4), nSNPs=10, pError=250)
#> Warning: pError was too high. It was replaced by 163.205080756888
# simulate the contribution in actual read numbers
pReads <- poolReads(nPools = 4, coverage = reads, probs = probs)
# output the number of reads per pool and per site
pReads
```

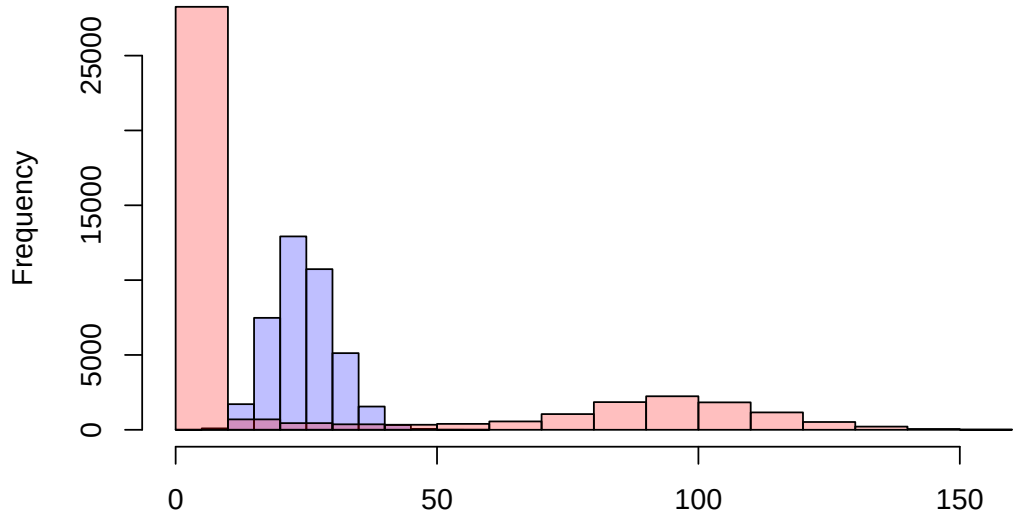
```
#>      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
#> [1,]   98    0    0    0    0    0   90  100   80    92
#> [2,]    0   98    0    0   86   36    0    0    0     0
#> [3,]    0    4    0   76    0    0    0    0    0     0
#> [4,]    0    0  111    0    0   88    1    0    0     0
```

Then the contributions are more uneven. In this instance, there are some sites where one or two pools contribute most of the reads while the remaining pools have few or even zero reads. Thus, the total coverage is not very well distributed among all pools when pool error is higher.

The difference between low or high pool errors can be (roughly) inspected with a histogram. In the next chunk we simulate the total coverage and then use the same coverage to compute the contribution of each pool, using either a low or a high pool error. We then plot the distribution of the number of reads contributed by each pool.

```
# simulate total coverage per site
reads <- simulateCoverage(mean=100, variance=250, nSNPs=10000, nLoci=1)
# unlist to create a vector with the coverage
reads <- unlist(reads)

# compute the proportion of contribution of each pool
probs <- poolProbs(nPools=4, vector_np=rep(10, 4), nSNPs=10000, pError=5)
# simulate the contribution in actual read numbers
low.pReads <- poolReads(nPools = 4, coverage = reads, probs = probs)

# compute the proportion of contribution of each pool
probs <- poolProbs(nPools=4, vector_np=rep(10, 4), nSNPs=10000, pError=250)
#> Warning: pError was too high. It was replaced by 163.205080756888
# simulate the contribution in actual read numbers
high.pReads <- poolReads(nPools = 4, coverage = reads, probs = probs)

# create the plot of the contribution with low pool error
h1 <- hist(unlist(low.pReads), plot = FALSE)
# create the plot of the contribution with high pool error
h2 <- hist(unlist(high.pReads), plot = FALSE)
# get the maximum x-value from the two plots
xmax <- max(h1[["breaks"]], h2[["breaks"]])
```

```
# and the maximum y-value
ymax <- max(h1[["counts"]], h2[["counts"]])
# set the color for the contribution computed with low pool error
col1 <- rgb(0,0,1,1/4)
# set the color for the contribution computed with high pool error
col2 <- rgb(1,0,0,1/4)

# plot the contribution computed with low pool error
plot(h1, col=col1, xlim=c(0, xmax), ylim=c(0, ymax), main="", xlab="")
# add the plot of the contribution computed with high pool error
plot(h2, col = col2, add = TRUE)
```



The distribution of the contribution computed with a low pool error is shown in blue and the distribution computed with a high pool error in red. It is clear that high pool errors lead to more variation in the contribution of each pool towards the total coverage of the population. In particular, the number of pools that contribute zero (or close to zero) reads increases when the pool error is high.

A3 - Reads contributed by each individual

After computing the number of reads contributed by each pool, the next step involves simulating the number of reads contributed by each individual inside their pool. For instance, if a pool of 10 individuals was used to sequence a population, how many reads were contributed by each of those 10 individuals?

As for the pools, the first step requires computing the probability of contribution of each individual. This can be done with the `indProbs` function. This `np` input argument of this function corresponds

to the total number of individuals in the pool, while the `nSNPs` is the number of sites to simulate. As before, the `pError` represents the degree of pooling error and higher values of `pError` mean that some individuals will contribute more reads than others.

In the next chunk, we examine the probability of contribution of 10 individuals, sequenced at 6 sites, when pooling error is quite low.

```
# compute the probability of contribution of each individual
indProbs(np = 10, nSNPs = 6, pError = 5)
#>              [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
#>  [1,] 0.09627053 0.09286627 0.10466446 0.10485763 0.10783017 0.10180182
#>  [2,] 0.09918398 0.10553742 0.10460223 0.10158420 0.10148286 0.10019229
#>  [3,] 0.09467018 0.10631994 0.09205836 0.09492194 0.09873694 0.10040422
#>  [4,] 0.09994063 0.09742962 0.09978582 0.11369760 0.09560072 0.10289944
#>  [5,] 0.10254666 0.10158477 0.09929162 0.10607670 0.10261459 0.10182249
#>  [6,] 0.09991116 0.08760101 0.10491098 0.09758553 0.10075414 0.09185170
#>  [7,] 0.10077137 0.10047420 0.09299284 0.10588431 0.10327234 0.10122942
#>  [8,] 0.10215999 0.10237935 0.09581764 0.09389039 0.09337310 0.10265476
#>  [9,] 0.10054648 0.10324108 0.10596724 0.09034687 0.10027795 0.09935312
#> [10,] 0.10399902 0.10256635 0.09990881 0.09115485 0.09605719 0.09779074
```

In this example, the probability of contribution is very similar across individuals. In fact, the probability is around 0.1 for each individual, meaning that, in a situation with low pooling error, all individuals should contribute equally. If we simulate the same conditions, but increasing the pooling error (`pError = 150`) we should see a different result. Note that we use the `round` function so that the the output is not printed in scientific notation. This is just to make it easier to visualize the differences.

```
# compute the probability of contribution of each individual
round(indProbs(np = 10, nSNPs = 5, pError = 150), digits = 5)
#>          [,1]    [,2]    [,3]    [,4]    [,5]
#>  [1,] 0.01329 0.34953 0.18409 0.00009 0.08225
#>  [2,] 0.04591 0.04627 0.31156 0.00173 0.09286
#>  [3,] 0.00046 0.28590 0.01316 0.00009 0.00375
#>  [4,] 0.00000 0.00677 0.25361 0.03188 0.28474
#>  [5,] 0.30152 0.12283 0.20959 0.12069 0.06045
#>  [6,] 0.14330 0.00337 0.00014 0.00000 0.20501
#>  [7,] 0.00168 0.13068 0.00118 0.17511 0.00000
```

```
#>  [8,] 0.48443 0.03722 0.00022 0.07399 0.11581
#>  [9,] 0.00389 0.01744 0.00003 0.02831 0.00000
#> [10,] 0.00552 0.00001 0.02642 0.56811 0.15513
```

With this higher pooling error, it is evident that the probability of contribution is not the same across all individuals. Some individuals have a much higher probability of contribution while others have a probability of contribution very close to zero.

The probabilities of contribution of each individual can then be used to simulate the total number of reads contributed by each individual, using the `indReads` function. This function requires as input argument the total number of individuals sequenced in that pool (`np`), a vector with the total `coverage` of that particular pool per site and probabilities of contribution computed with the `indProbs` function (`probs`).

In the next chunk, we start by simulating the total coverage per site. This total coverage is then partitioned among the different pools to obtain the total coverage per pool. Finally, we simulate the contribution of the 10 individuals sequenced at one of the pools towards the total coverage of that pool. All these steps are done assuming a low pooling error.

```
# simulate total coverage per site
reads <- unlist(simulateCoverage(mean=100, variance=250, nSNPs=12, nLoci=1))
# compute the proportion of contribution of each pool
probs <- poolProbs(nPools=4, vector_np=rep(10, 4), nSNPs=12, pError=5)
# simulate the contribution in actual read numbers
pReads <- poolReads(nPools = 4, coverage = reads, probs = probs)
# compute the proportion of contribution of each pool
probs <- indProbs(np = 10, nSNPs = 12, pError = 5)
# simulate the contribution in actual read numbers of each individual
indReads(np = 10, coverage = pReads[1,], probs = probs)
#>       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
#>  [1,]    1    5    3    1    4    1    2    0    3     3     1     4
#>  [2,]    0    4    1    4    2    2    2    1    1     1     2     1
#>  [3,]    1    2    3    2    0    2    1    2    4     1     2     3
#>  [4,]    1    3    5    1    0    6    6    2    1     2     3     3
#>  [5,]    0    2    2    2    2    3    2    1    0     2     1     3
#>  [6,]    0    3    0    2    4    2    2    0    2     2     5     2
#>  [7,]    4    4    2    1    1    3    1    3    2     1     3     2
#>  [8,]    1    1    1    2    2    2    4    0    3     6     0     3
```

```
#>  [9,]    3    1    4    1    3    4    3    1    5    8    5    1
#> [10,]    3    3    2    1    4    2    0    3    4    1    5    1
```

It is clear that, when pool error is low (`pError = 5`), each individual contributes roughly the same number of reads towards the total coverage of the pool. Thus, the overall dispersion is quite low. If we repeat the same steps, changing only the pooling error to a much higher value:

```
# simulate total coverage per site
reads <- unlist(simulateCoverage(mean=100, variance=250, nSNPs=12, nLoci=1))
# compute the proportion of contribution of each pool
probs <- poolProbs(nPools=4, vector_np=rep(10, 4), nSNPs=12, pError=150)
# simulate the contribution in actual read numbers
pReads <- poolReads(nPools = 4, coverage = reads, probs = probs)
# compute the proportion of contribution of each pool
probs <- indProbs(np = 10, nSNPs = 12, pError = 150)
# simulate the contribution in actual read numbers of each individual
indReads(np = 10, coverage = pReads[1,], probs = probs)
#>       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
#>  [1,]    0    0    0   33    9    0    0    0    0     0     2     0
#>  [2,]    0    0   37    2    0    0    8    2    0     0     3     0
#>  [3,]    0    0    0    1    4    0    0    2    0     0    35     0
#>  [4,]    0    0   24    0   13    0    0    3    0     0     0     0
#>  [5,]    0    0    2    0    0    0    0    3    0     0     6     0
#>  [6,]    0    0    7    0    0    0    0    0    0     0     0     0
#>  [7,]    0    0    0    0   11    0    1    3    0     0    14     0
#>  [8,]    0    0   12    1    0    0    0    2    0     0     0     0
#>  [9,]    0    0    2    7    0    0    1    0    0     0     3     0
#> [10,]    0    0    0    4    3    0    0    1    0     0     1     0
```

We see that in this instance, there is much more dispersion and the individuals do not contribute the same number of reads. While some individuals do not contribute a single reads towards the total pool coverage, others contribute too many.

A3 - Number of reads with the reference allele

Following the computation of the number of reads contributed by each individual, we should simulate how many of those reads have the reference allele versus how many have the alternative allele. For a single population this can be done using the `computeReference` function.

This function requires as input argument the individual contribution i.e. the number of reads that each individual contributes and the sequencing error - `error`. The sequencing error is defined as a error rate - the higher the error, the more likely it is for an individual that is homozygous for the reference allele (coded as 0 in the `genotypes` matrix) to contribute reads with the alternative allele. Note that this function also requires as input argument the `genotypes` of the individuals. Given that we did not simulate genotypes in this vignette, we are going to create a matrix of genotypes where half the individuals are homozygous for the reference allele and the other half is homozygous for the alternative allele (coded as 2 in the `genotypes` matrix).

In the next chunk we go over all the previous steps, simulating the total coverage for one population, then partitioning that over all pools and computing the contribution of each individual in one of those pools. At the end, we simulate how many of those individually contributed reads have the reference allele.

```
# simulate total coverage per site
reads <- unlist(simulateCoverage(mean=100, variance=250, nSNPs=12, nLoci=1))
# compute the proportion of contribution of each pool
probs <- poolProbs(nPools=4, vector_np=rep(10, 4), nSNPs=12, pError=5)
# simulate the contribution in actual read numbers
pReads <- poolReads(nPools = 4, coverage = reads, probs = probs)
# compute the proportion of contribution of each pool
probs <- indProbs(np = 10, nSNPs = 12, pError = 5)
# simulate the contribution in actual read numbers of each individual
iReads <- indReads(np = 10, coverage = pReads[1,], probs = probs)
# create fake genotypes - half the matrix is 0 and the other half is 2
geno <- rbind(matrix(0, nrow=5, ncol=12), matrix(2, nrow=5, ncol=12))
# simulate the number of reference reads
computeReference(genotypes = geno, indContribution = iReads, error = 0.001)
#>       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
#>  [1,]    1    2    1    4    3    2    4    1    2     6     1     3
#>  [2,]    1    3    1    3    2    2    6    2    1     1     3     3
#>  [3,]    2    3    5    3    3    2    3    1    4     3     5     4
#>  [4,]    2    0    0    1    2    1    4    1    1     3     1     6
#>  [5,]    1    2    0    2    2    1    6    4    3     3     2     3
#>  [6,]    0    0    0    0    0    0    0    0    0     0     0     0
#>  [7,]    0    0    0    0    0    0    0    0    0     0     0     0
#>  [8,]    0    0    0    0    0    0    0    0    0     0     0     0
```

```
#>  [9,]    0    0    0    0    0    0    0    0    0    0    0    0
#> [10,]    0    0    0    0    0    0    0    0    0    0    0    0
```

There is a clear division between the number of reads with the reference allele for the first 5 individuals (coded as 0 in the `genotypes` matrix) and the remaining 5 individuals (coded as 2 in the `genotypes` matrix). This is expected because the `error` was small. If we increased the `error`, then we would expect to see some reference allele reads contributed by individuals that are homozygous for the other allele.

3.8.2 Manual for the Package 'poolHelper'

Title Simulates Pooled Sequencing Genetic Data

Version 1.1.0

Description Simulates pooled sequencing data under a variety of conditions.
Also allows for the evaluation of the average absolute difference between allele frequencies computed from genotypes and those computed from pooled data.
Carvalho et al., (2022) doi:10.1101/2023.01.20.524733 .

License GPL (= 3)

Encoding UTF-8

RoxygenNote 7.2.1

Imports MCMCpack, Metrics, scrm, stats

Suggests knitr, rmarkdown, ggplot2, testthat (= 3.0.0)

VignetteBuilder knitr

URL https://github.com/joao-mcarvalho/poolHelper

BugReports https://github.com/joao-mcarvalho/poolHelper/issues

Config testthat edition 3

NeedsCompilation no

Author Jo o Carvalho [aut, cre] (https://orcid.org/0000-0002-1728-0075),
V tor Sousa [aut]

Maintainer Jo o Carvalho <jgcarvalho@fc.ul.pt>

Repository CRAN

Date Publication 2023-06-29 17:50:02 UTC

R topics documented:

`calculatePi` *Calculate population frequency at each SNP*

Description

The frequency at a given SNP is calculated according to: `pi = c/r`, where `c` = number of minor-allele reads and `r` = total number of observed reads.

Usage

```
calculatePi(listPool, nLoci)
```

Arguments

`listPool`	a list containing the minor element, representing the number of reads with the minor-allele and the total element that contains information about the total number of reads. The list should also contain a major entry with the information about reads containing the major-allele. The output of the `poolPops` function should be used as input here.
`nLoci`	an integer that represents the total number of independent loci in the dataset.

Details

This function takes as input a list that contains the number of reads with the minor allele and the number of total reads per population at a given site. The names of the respective elements of the list should be minor and total. It works with lists containing just one set of minor and total reads, corresponding to a single locus, and with lists where each entry contains a different set of minor and total number of reads, corresponding to different loci.

Value

a list with two named entries

`pi`	a list with the allele frequencies of each population. Each list entry is a matrix, corresponding to a different locus. Each row of a matrix corresponds to a different population and each column to a different site.
`pool`	a list with three different entries: major, minor and total. This list is similar to the one obtained with the `findMinor` function.

Examples

```
# simulate coverage at 5 SNPs for two populations, assuming 20x mean coverage
reads <- simulateCoverage(mean = c(20, 20), variance = c(100, 100), nSNPs = 5, nLoci = 1)

# simulate the number of reads contributed by each individual
# for each population there are two pools, each with 5 individuals
indContribution <- popsReads(list_np = rep(list(rep(5, 2)), 2), coverage = reads, pError = 5)

# set seed and create a random matrix of genotypes for the 20 individuals - 10 per population
set.seed(10)
genotypes <- matrix(rpois(100, 0.5), nrow = 20)

# simulate the number of reference reads for the two populations
readsReference <- numberReferencePop(genotypes = genotypes, indContribution = indContribution,
size = rep(list(rep(5, 2)), 2), error = 0.01)

# create Pooled DNA sequencing data for these two populations and for a single locus
pools <- poolPops(nPops = 2, nLoci = 1, indContribution = indContribution,
readsReference = readsReference)

# define the major and minor alleles for this pool-seq data
# note that we have to select the first entry of the pools list
# because this function works for matrices
pools <- findMinor(reference = pools$reference[[1]], alternative = pools$alternative[[1]],
coverage = pools$total[[1]])

# calculate population frequency at each SNP of this locus
calculatePi(listPool = pools, nLoci = 1)
```

`computeReference` *Compute the number of reference reads over a matrix*

Description

This function works over all the rows and columns of a matrix and computes the number of reads containing the reference allele at each site and for each individual.

Usage

```
computeReference(genotypes, indContribution, error)
```

Arguments

`genotypes`	is a matrix of genotypes. Each column of the matrix should be a different site and each row a different individual. Genotypes should be encoded as 0: reference homozygote, 1: heterozygote and 2: alternative homozygote.
`indContribution`	is a matrix of individual contributions. Each row of that matrix is a different individual and each column is a different site. Thus, each entry of the matrix should contain the number of reads contributed by that individual at that particular site.
`error`	a numeric value with error rate associated with the sequencing and mapping process. This error rate is assumed to be symmetric: error(reference - alternative) = error(alternative - reference). This number should be between 0 and 1.

Value

a matrix with the number of reference allele reads contributed by each individual. Each row of the matrix represents a different individual and each column is a different site.

Examples

```
# probability of contribution for 10 individuals at 5 sites
probs <- indProbs(np = 10, nSNPs = 5, pError = 5)

# simulate the number of reads contributed, assuming 20 coverage for each site
indContribution <- indReads(np = 10, coverage = rep(20, 5), probs = probs)

# set seed and create a random matrix of genotypes
set.seed(10)
genotypes <- matrix(rpois(50, 0.5), nrow = 10)

# simulate the number of reads with the reference allele
computeReference(genotypes = genotypes, indContribution = indContribution, error = 0.01)
```

errorHet *Average absolute difference between expected heterozygosity*

Description

Calculates the average absolute difference between the expected heterozygosity computed directly from genotypes and from pooled sequencing data.

Usage

```
errorHet(
  nDip,
  nloci,
  pools,
  pError,
  sError,
  mCov,
  vCov,
  min.minor,
  minimum = NA,
  maximum = NA,
  theta = 10
)
```

Arguments

nDip	an integer representing the total number of diploid individuals to simulate. Note that scrm::scrm() actually simulates haplotypes, so the number of simulated haplotypes is double of this.
nloci	is an integer that represents how many independent loci should be simulated.
pools	a list with a vector containing the size (in number of diploid individuals) of each pool. Thus, if a population was sequenced using a single pool, the vector should contain only one entry. If a population was sequenced using two pools, each with 10 individuals, this vector should contain two entries and both will be 10.
pError	an integer representing the value of the error associated with DNA pooling. This value is related with the unequal contribution of both individuals and pools towards the total number of reads observed for a given population - the higher the value the more unequal are the individual and pool contributions.
sError	a numeric value with error rate associated with the sequencing and mapping process. This error rate is assumed to be symmetric: error(reference - alternative) = error(alternative - reference). This number should be between 0 and 1.
mCov	an integer that defines the mean depth of coverage to simulate. Please note that this represents the mean coverage across all sites.
vCov	an integer that defines the variance of the depth of coverage across all sites.

min.minor	is an integer representing the minimum allowed number of minor-allele reads. Sites that, across all populations, have less minor-allele reads than this threshold will be removed from the data.
minimum	an optional integer representing the minimum coverage allowed. Sites where the population has a depth of coverage below this threshold are removed from the data.
maximum	an optional integer representing the maximum coverage allowed. Sites where the population has a depth of coverage above this threshold are removed from the data.
theta	a value for the mutation rate assuming theta = 4Nu, where u is the neutral mutation rate per locus.

Details

Different combinations of parameters can be tested to check the effect of the various parameters. The average absolute difference is computed with the mae function, assuming the expected heterozygosity computed directly from the genotypes as the actual input argument and the expected heterozygosity from pooled data as the predicted input argument.

Value

a data.frame with columns detailing the number of diploid individuals, the pool error, the number of pools, the number of individuals per pool, the mean coverage, the variance of the coverage and the average absolute difference between the expected heterozygosity computed from genotypes and from pooled data.

Examples

```
# single population sequenced with a single pool of 100 individuals
errorHet(nDip = 100, nloci = 10, pools = list(100), pError = 100, sError = 0.01,
mCov = 100, vCov = 250, min.minor = 2)

# single population sequenced with two pools, each with 50 individuals
errorHet(nDip = 100, nloci = 10, pools = list(c(50, 50)), pError = 100, sError = 0.01,
mCov = 100, vCov = 250, min.minor = 2)

# single population sequenced with two pools, each with 50 individuals
# removing sites with coverage below 10x or above 180x
errorHet(nDip = 100, nloci = 10, pools = list(c(50, 50)), pError = 100, sError = 0.01,
mCov = 100, vCov = 250, min.minor = 2, minimum = 10, maximum = 180)
```

---

findMinor *Define major and minor alleles*

Description

This function checks which of the two simulated alleles (reference or alternative) corresponds to the minor allele. This function can also be used to remove sites according to a minor-allele reads threshold.

Usage

```
findMinor(reference, alternative, coverage)
```

Arguments

`reference`	is a matrix of reference allele reads. Each row of the matrix should be a different population and each column a different site. Thus, each entry of the matrix contains the number of observed reads with the reference allele for that population at a given site.
`alternative`	is a matrix of alternative allele reads. Each row of the matrix should be a different population and each column a different site. Thus, each entry of the matrix contains the number of observed reads with the alternative allele for that population at a given site.
`coverage`	is a matrix of total coverage. Each row of the matrix should be a different population and each column a different site. Thus, each entry of the matrix contains the total number of observed reads for that population at a given site.

Details

More precisely, this function counts the number of reads with the reference or alternative allele at each site and then sets the minor allele as the least frequent of the two. This is done across all populations and so the major and minor alleles are defined at a global level. Then if the `min.minor` input is not NA, sites where the number of minor allele reads, across all populations, is below the user-defined threshold are removed.

Value

a list with three names entries

`major`	a list with one entry per locus. Each entry is a matrix with the number of major allele reads for each population. Each column represents a different site and each row a different population.
`minor`	a list with one entry per locus. Each entry is a matrix with the number of minor allele reads for each population. Each column represents a different site and each row a different population.
`total`	a list with one entry per locus. Each entry is a matrix with the coverage of each population. Each column represents a different site and each row a different population.

Examples

```
# simulate coverage at 5 SNPs for two populations, assuming 20x mean coverage
reads <- simulateCoverage(mean = c(20, 20), variance = c(100, 100), nSNPs = 5, nLoci = 1)

# simulate the number of reads contributed by each individual
# for each population there are two pools, each with 5 individuals
indContribution <- popsReads(list_np = rep(list(rep(5, 2)), 2), coverage = reads, pError = 5)

# set seed and create a random matrix of genotypes for the 20 individuals - 10 per population
set.seed(10)
genotypes <- matrix(rpois(100, 0.5), nrow = 20)

# simulate the number of reference reads for the two populations
readsReference <- numberReferencePop(genotypes = genotypes, indContribution = indContribution,
size = rep(list(rep(5, 2)), 2), error = 0.01)

# create Pooled DNA sequencing data for these two populations and for a single locus
pools <- poolPops(nPops = 2, nLoci = 1, indContribution = indContribution,
readsReference = readsReference)

# define the major and minor alleles for this Pool-seq data
# we have to select the first entry of the pools list because this function works for matrices
findMinor(reference = pools$reference[[1]], alternative = pools$alternative[[1]],
coverage = pools$total[[1]])
```

---

getNumReadsR_vector *Compute the number of reference reads*

Description

This function takes as input the total depth of coverage and computes how many of those reads are reference allele reads.

Usage

```
getNumReadsR_vector(genotype_v, readCount_v, error)
```

Arguments

genotype_v	is a vector with the genotype of a given individual. Each entry of the vector should be a different site. Genotypes should be encoded as 0: reference homozygote, 1: heterozygote and 2: alternative homozygote.
readCount_v	is a vector with the number of reads contributed by the same given individual. Each entry of that vector should be a different site.
error	a numeric value with error rate associated with the sequencing and mapping process. This error rate is assumed to be symmetric: error(reference - alternative) = error(alternative - reference). This number should be between 0 and 1.

Details

More precisely, this function computes the number of reference reads per site for one individual, given the genotype of the individual at each site, the total number of reads observed for the individual at that site and an error rate.

Value

a vector with the number of reference allele reads. Each entry of the vector corresponds to a different individual.

Examples

```
# number of reference allele reads for three individuals, each with 10x coverage
# one individual is homozygote for the reference allele (0), other is heterozygote (1)
# and the last is homozygote for the alternative allele (2)
getNumReadsR_vector(genotype_v = c(0,1,2), readCount_v = c(10, 10, 10), error = 0.01)
```

Ifreqs *Compute allele frequencies from genotypes*

Description

Computes alternative allele frequencies from genotypes by dividing the total number of alternative alleles by the total number of gene copies.

Usage

```
Ifreqs(nDip, genotypes)
```

Arguments

nDip	an integer representing the total number of diploid individuals to simulate. Note that `scrm::scrm()` actually simulates haplotypes, so the number of simulated haplotypes is double of this.
genotypes	a list of simulated genotypes, where each entry is a matrix corresponding to a different locus. At each matrix, each column is a different SNP and each row is a different individual.

Value

a list of allele frequencies. Each entry of the list corresponds to a different locus.

Examples

```
genotypes <- run_scrm(nDip = 10, nloci = 10)
Ifreqs(nDip = 10, genotypes)
```

indProbs *Probability of contribution of each individual*

Description

This function computes the probability of contribution for each individual of a given pool. Please note that this function works for a single pool and should not be directly applied to situations where multiple pools were used.

Usage

```
indProbs(np, nSNPs, pError)
```

Arguments

np	an integer specifying how many individuals were pooled.
nSNPs	an integer indicating how many SNPs exist in the data.
pError	an integer representing the value of the error associated with DNA pooling. This value is related with the unequal individual contribution towards the total number of reads contributed by a single pool - the higher the value the more unequal are the individual contributions.

Value

a matrix with the probabilities of contribution for each individual. Each row represents a different individual and each column is a different site.

Examples

```
# probability of contribution for 10 individuals at 5 sites
indProbs(np = 10, nSNPs = 5, pError = 100)
```

indReads *Reads contributed by each individual*

Description

This function simulates the contribution, in terms of reads, of each individual of a given pool. Please note that this function works for a single pool and should not be directly applied to situations where multiple pools were used.

Usage

```
indReads(np, coverage, probs)
```

Arguments

np	an integer specifying how many individuals were pooled.
coverage	a vector containing the total depth of coverage of a given pool. Each entry of the vector represents a different site.
probs	a matrix containing the probability of contribution of each individual. This matrix can be obtained with the indProbs function.

Value

a matrix with the number of reads contributed by each individual towards the coverage of its pool. Each row of the matrix is a different individual and each column a different site.

Examples

```
# probability of contribution for 10 individuals at 5 sites
probs <- indProbs(np = 10, nSNPs = 5, pError = 100)

# simulate the number of reads contributed, assuming 10x coverage for each site
indReads(np = 10, coverage = rep(10, 5), probs = probs)
```

---

maeFreqs — *Average absolute difference between allele frequencies computed from genotypes and from Pool-seq data*

Description

Calculates the average absolute difference between the allele frequencies computed directly from genotypes and from pooled sequencing data.

Usage

```
maeFreqs(
  nDip,
  nloci,
  pError,
  sError,
  mCov,
  vCov,
  min.minor,
  minimum = NA,
  maximum = NA,
  theta = 10
)
```

Arguments

nDip	is an integer or a vector representing the total number of diploid individuals to simulate. Note that scrm::scrm() actually simulates haplotypes, so the number of simulated haplotypes is double of this. If it is a vector, then each vector entry will be simulated independently. For instance, if nDip = c(100, 200), simulations will be carried out for samples of 100 and 200 individuals.
nloci	is an integer that represents how many independent loci should be simulated.
pError	an integer or a vector representing the value of the error associated with DNA pooling. This value is related with the unequal contribution of both individuals and pools towards the total number of reads observed for a given population - the higher the value the more unequal are the individual and pool contributions. If it is a vector, then each vector entry will be simulated independently.
sError	a numeric value with error rate associated with the sequencing and mapping process. This error rate is assumed to be symmetric: error(reference - alternative) = error(alternative - reference). This number should be between 0 and 1.
mCov	an integer or a vector that defines the mean depth of coverage to simulate. Please note that this represents the mean coverage across all sites. If it is a vector, then each vector entry will be simulated independently.
vCov	an integer or a vector that defines the variance of the depth of coverage across all sites. If the mCov is a vector, then vCov should also be a vector, with each entry corresponding to the variance of the respective entry in the mCov vector. Thus, the first entry of the vCov vector will be the variance associated with the first entry of the mCov vector.
min.minor	is an integer representing the minimum allowed number of minor-allele reads. Sites that, across all populations, have less minor-allele reads than this threshold will be removed from the data.
minimum	an optional integer representing the minimum coverage allowed. Sites where the population has a depth of coverage below this threshold are removed from the data.
maximum	an optional integer representing the maximum coverage allowed. Sites where the population has a depth of coverage above this threshold are removed from the data.
theta	a value for the mutation rate assuming theta = 4Nu, where u is the neutral mutation rate per locus.

Details

The average absolute difference is computed with the mae function, assuming the frequencies computed directly from the genotypes as the actual input argument and the frequencies from pooled data as the predicted input argument.

Note that this functions allows for different combinations of parameters. Thus, the effect of different combinations of parameters on the average absolute difference can be tested. For instance, it is possible to check what is the effect of different coverages by including more than one value in the mCov input argument. This function will run and compute the average absolute difference for all combinations of the nDip, pError and mCov input arguments. This function assumes that a single pool of size nDip was used to sequence the population.

Value

a data.frame with columns detailing the number of diploid individuals, the pool error, the number of pools, the number of individuals per pool, the mean coverage, the variance of the coverage and the average absolute difference between the frequencies computed from genotypes and from pooled data.

Examples

```
# a simple test with a simple combination of parameters
maeFreqs(nDip = 100, nloci = 10, pError = 100, sError = 0.01, mCov = 100, vCov = 200, min.minor = 1)

# effect of two different pool error values in conjugation with a fixed coverage and pool size
maeFreqs(nDip = 100, nloci = 10, pError = c(100, 200), sError = 0.01,
mCov = 100, vCov = 200, min.minor = 1)

# effect of two different pool error values in conjugation with a fixed pool size
# and two different coverages
maeFreqs(nDip = 100, nloci = 10, pError = c(100, 200), sError = 0.01,
mCov = c(100, 200), vCov = c(200, 500), min.minor = 1)
```

---

maeHet *Average absolute difference between the expected heterozygosity computed from genotypes and from Pool-seq data*

Description

Calculates the average absolute difference between the expected heterozygosity computed directly from genotypes and from pooled sequencing data.

Usage

```
maeHet(
  nDip,
  nloci,
  pError,
  sError,
  mCov,
  vCov,
  min.minor,
  minimum = NA,
  maximum = NA,
  theta = 10
)
```

Arguments

nDip	is an integer or a vector representing the total number of diploid individuals to simulate. Note that scrm::scrm() actually simulates haplotypes, so the number of simulated haplotypes is double of this. If it is a vector, then each vector entry will be simulated independently. For instance, if nDip = c(100, 200), simulations will be carried out for samples of 100 and 200 individuals.
nloci	is an integer that represents how many independent loci should be simulated.
pError	an integer or a vector representing the value of the error associated with DNA pooling. This value is related with the unequal contribution of both individuals and pools towards the total number of reads observed for a given population - the higher the value the more unequal are the individual and pool contributions. If it is a vector, then each vector entry will be simulated independently.
sError	a numeric value with error rate associated with the sequencing and mapping process. This error rate is assumed to be symmetric: error(reference - alternative) = error(alternative - reference). This number should be between 0 and 1.
mCov	an integer or a vector that defines the mean depth of coverage to simulate. Please note that this represents the mean coverage across all sites. If it is a vector, then each vector entry will be simulated independently.
vCov	an integer or a vector that defines the variance of the depth of coverage across all sites. If the mCov is a vector, then vCov should also be a vector, with each entry corresponding to the variance of the respective entry in the mCov vector. Thus, the first entry of the vCov vector will be the variance associated with the first entry of the mCov vector.
min.minor	is an integer representing the minimum allowed number of minor-allele reads. Sites that, across all populations, have less minor-allele reads than this threshold will be removed from the data.
minimum	an optional integer representing the minimum coverage allowed. Sites where the population has a depth of coverage below this threshold are removed from the data.
maximum	an optional integer representing the maximum coverage allowed. Sites where the population has a depth of coverage above this threshold are removed from the data.
theta	a value for the mutation rate assuming theta = 4Nu, where u is the neutral mutation rate per locus.

Details

The average absolute difference is computed with the mae function, assuming the expected heterozygosity computed directly from the genotypes as the actual input argument and the expected heterozygosity from pooled data as the predicted input argument.

Note that this functions allows for different combinations of parameters. Thus, the effect of different combinations of parameters on the average absolute difference can be tested. For instance, it is possible to check what is the effect of different coverages by including more than one value in the mCov input argument. This function will run and compute the average absolute difference for all combinations of the nDip, pError and mCov input arguments. This function assumes that a single pool of size nDip was used to sequence the population.

Value

a data.frame with columns detailing the number of diploid individuals, the pool error, the number of pools, the number of individuals per pool, the mean coverage, the variance of the coverage and the average absolute difference between the expected heterozygosity computed from genotypes and from pooled data.

Examples

```
# a simple test with a simple combination of parameters
maeHet(nDip = 100, nloci = 10, pError = 100, sError = 0.01, mCov = 100, vCov = 200, min.minor = 1)

# effect of two different pool error values in conjugation with a fixed coverage and pool size
maeHet(nDip = 100, nloci = 10, pError = c(100, 200), sError = 0.01,
mCov = 100, vCov = 200, min.minor = 1)

# effect of two different pool error values in conjugation with a fixed pool size
# and two different coverages
maeHet(nDip = 100, nloci = 10, pError = c(100, 200), sError = 0.01,
mCov = c(100, 200), vCov = c(200, 500), min.minor = 1)
```

---

maePool *Average absolute difference between allele frequencies*

Description

Calculates the average absolute difference between the allele frequencies computed directly from genotypes and from pooled sequencing data.

Usage

```
maePool(
  nDip,
  nloci,
  pools,
  pError,
  sError,
  mCov,
  vCov,
  min.minor,
  minimum = NA,
  maximum = NA,
  theta = 10
)
```

Arguments

nDip	an integer representing the total number of diploid individuals to simulate. Note that scrm::scrm() actually simulates haplotypes, so the number of simulated haplotypes is double of this.
nloci	is an integer that represents how many independent loci should be simulated.
pools	a list with a vector containing the size (in number of diploid individuals) of each pool. Thus, if a population was sequenced using a single pool, the vector should contain only one entry. If a population was sequenced using two pools, each with 10 individuals, this vector should contain two entries and both will be 10.
pError	an integer representing the value of the error associated with DNA pooling. This value is related with the unequal contribution of both individuals and pools towards the total number of reads observed for a given population - the higher the value the more unequal are the individual and pool contributions.
sError	a numeric value with error rate associated with the sequencing and mapping process. This error rate is assumed to be symmetric: error(reference - alternative) = error(alternative - reference). This number should be between 0 and 1.
mCov	an integer that defines the mean depth of coverage to simulate. Please note that this represents the mean coverage across all sites.
vCov	an integer that defines the variance of the depth of coverage across all sites.
min.minor	is an integer representing the minimum allowed number of minor-allele reads. Sites that, across all populations, have less minor-allele reads than this threshold will be removed from the data.
minimum	an optional integer representing the minimum coverage allowed. Sites where the population has a depth of coverage below this threshold are removed from the data.
maximum	an optional integer representing the maximum coverage allowed. Sites where the population has a depth of coverage above this threshold are removed from the data.
theta	a value for the mutation rate assuming theta = 4Nu, where u is the neutral mutation rate per locus.

Details

Different combinations of parameters can be tested to check the effect of the various parameters. The average absolute difference is computed with the mae function, assuming the frequencies computed directly from the genotypes as the actual input argument and the frequencies from pooled data as the predicted input argument.

Value

a data.frame with columns detailing the number of diploid individuals, the pool error, the number of pools, the number of individuals per pool, the mean coverage, the variance of the coverage and the average absolute difference between the frequencies computed from genotypes and from pooled data.

Examples

```
# single population sequenced with a single pool of 100 individuals
maePool(nDip = 100, nloci = 10, pools = list(100), pError = 100, sError = 0.01,
mCov = 100, vCov = 250, min.minor = 2)

# single population sequenced with two pools, each with 50 individuals
maePool(nDip = 100, nloci = 10, pools = list(c(50, 50)), pError = 100, sError = 0.01,
mCov = 100, vCov = 250, min.minor = 2)

# single population sequenced with two pools, each with 50 individuals
# removing sites with coverage below 10x or above 180x
maePool(nDip = 100, nloci = 10, pools = list(c(50, 50)), pError = 100, sError = 0.01,
mCov = 100, vCov = 250, min.minor = 2, minimum = 10, maximum = 180)
```

---

mymae *Average absolute difference between allele frequencies computed from genotypes supplied by the user and from Pool-seq data*

Description

Calculates the average absolute difference between the allele frequencies computed directly from genotypes and from pooled sequencing data. The genotypes used should be supplied by the user and can be simulated using different software and under the demographic model of choice.

Usage

```
mymae(
  genotypes,
  pools,
  pError,
  sError,
  mCov,
  vCov,
  min.minor,
  minimum = NA,
  maximum = NA
)
```

Arguments

genotypes	a list of genotypes, where each entry is a matrix corresponding to a different locus. At each matrix, each column is a different SNP and each row is a different individual. Genotypes should be coded as 0, 1 or 2.
pools	a list with a vector containing the size (in number of diploid individuals) of each pool. Thus, if a population was sequenced using a single pool, the vector should contain only one entry. If a population was sequenced using two pools, each with 10 individuals, this vector should contain two entries and both will be 10.

pError	an integer representing the value of the error associated with DNA pooling. This value is related with the unequal contribution of both individuals and pools towards the total number of reads observed for a given population - the higher the value the more unequal are the individual and pool contributions.
sError	a numeric value with error rate associated with the sequencing and mapping process. This error rate is assumed to be symmetric: error(reference - alternative) = error(alternative - reference). This number should be between 0 and 1.
mCov	an integer that defines the mean depth of coverage to simulate. Please note that this represents the mean coverage across all sites.
vCov	an integer that defines the variance of the depth of coverage across all sites.
min.minor	is an integer representing the minimum allowed number of minor-allele reads. Sites that, across all populations, have less minor-allele reads than this threshold will be removed from the data.
minimum	an optional integer representing the minimum coverage allowed. Sites where the population has a depth of coverage below this threshold are removed from the data.
maximum	an optional integer representing the maximum coverage allowed. Sites where the population has a depth of coverage above this threshold are removed from the data.

Details

The average absolute difference is computed with the mae function, assuming the frequencies computed directly from the genotypes as the `actual` input argument and the frequencies from pooled data as the `predicted` input argument.

Note that this functions allows for different combinations of parameters. Thus, the effect of different combinations of parameters on the average absolute difference can be tested. For instance, it is possible to check what is the effect of different coverages by including more than one value in the `mCov` input argument. This function will run and compute the average absolute difference for all combinations of the `pools`, `pError` and `mCov` input arguments.

Value

a data.frame with columns detailing the number of diploid individuals, the pool error, the number of pools, the number of individuals per pool, the mean coverage, the variance of the coverage and the average absolute difference between the frequencies computed from genotypes and from pooled data.

Examples

```
# 100 individuals sampled at a single locus
genotypes <- run_scrm(nDip = 100, nloci = 1, theta = 5)
# compute the mean absolute error assuming a coverage of 100x and two pools of 50 individuals each
mymae(genotypes = genotypes, pools = list(c(50, 50)), pError = 100, sError = 0.001,
mCov = 100, vCov = 250, min.minor = 0)

# 10 individuals sampled at 5 different loci
genotypes <- run_scrm(nDip = 10, nloci = 5, theta = 5)
```

```
# compute the mean absolute error assuming a coverage of 100x and one pool of 10 individuals
mymae(genotypes = genotypes, pools = list(10), pError = 100, sError = 0.001,
mCov = 100, vCov = 250, min.minor = 0)
```

---

numberReferencePop *Compute the number of reference reads for multiple populations*

Description

This function computes the number of reference reads over a single locus for multiple populations.

Usage

```
numberReferencePop(genotypes, indContribution, size, error)
```

Arguments

genotypes	either a list with a single entry (one locus) or a matrix (that the function will convert to a list) containing the genotypes (coded as 0, 1 or 2). Each column of that matrix should be a different site and each row a different individual.
indContribution	a list where each entry contains the information for a single population. Each entry should be a matrix, with as many rows as the number of individuals of that population. Each row contains the number of contributed reads for a given individual and across all sites.
size	a list with one entry per population. Each entry should be a vector containing the size (in number of diploid individuals) of each pool. Thus, if a population was sequenced using a single pool, the vector should contain only one entry. If a population was sequenced using two pools, each with 10 individuals, this vector should contain two entries and both will be 10.
error	a numeric value with error rate associated with the sequencing and mapping process. This error rate is assumed to be symmetric: error(reference - alternative) = error(alternative - reference). This number should be between 0 and 1.

Details

Note that this function will not work as intended if the input consists of multiple loci.

Value

a list with one entry per population. Each entry contains the number of reference allele reads for the individuals of that population and for that locus. Different individuals are in different rows and each columns represents a different site.

Examples

```
# simulate coverage at 5 SNPs for two populations, assuming 20x mean coverage
reads <- simulateCoverage(mean = c(20, 20), variance = c(100, 100), nSNPs = 5, nLoci = 1)

# simulate the number of reads contributed by each individual
# for each population there are two pools, each with 5 individuals
indContribution <- popsReads(list_np = rep(list(rep(5, 2)), 2), coverage = reads, pError = 5)

# set seed and create a random matrix of genotypes for the 20 individuals - 10 per population
set.seed(10)
genotypes <- matrix(rpois(100, 0.5), nrow = 20)

# simulate the number of reference reads for the two populations
numberReferencePop(genotypes = genotypes, indContribution = indContribution,
size = rep(list(rep(5, 2)), 2), error = 0.01)
```

Pfreqs *Compute allele frequencies from pooled sequencing data*

Description

Computes the frequency of the alternative allele in Pool-seq data and removes any site with too few minor-allele reads from both the pool frequencies and the frequencies computed directly from genotypes.

Usage

```
Pfreqs(reference, alternative, coverage, min.minor, ifreqs)
```

Arguments

reference	a matrix with the number of reference allele reads. Each row should be a different population and each column a different site.
alternative	a matrix with the number of alternative allele reads. Each row should be a different population and each column a different site.
coverage	a matrix with the total coverage. Each row should be a different population and each column a different site.
min.minor	is an integer representing the minimum allowed number of minor-allele reads. Sites that, across all populations, have less minor-allele reads than this threshold will be removed from the data.
ifreqs	a vector of allele frequencies computed directly from the genotypes where each entry corresponds to a different site.

Details

The frequency at a given SNP is calculated according to: `pi = c/r`, where c = number of alternative allele reads and r = total number of observed reads. Additionally, if a site has less minor-allele reads than `min.minor` across all populations, that site is removed from the data.

Value

a list with two entries. The `ifreqs` entry contains the allele frequencies computed directly from genotypes and `pfreqs` the allele frequencies computed from pooled sequencing data.

Examples

```
set.seed(10)
# create a vector of allele frequencies
freqs <- runif(20)
set.seed(10)
# create a matrix with the number of reads with the alternative allele
alternative <- matrix(sample(x = c(0,5,10), size = 20, replace = TRUE), nrow = 1)
# create a matrix with the depth of coverage
coverage <- matrix(sample(100:150, size = 20), nrow = 1)
# the number of reads with the reference allele is obtained by subtracting
# the number of alternative allele reads from the depth of coverage
reference <- coverage - alternative
# compute allele frequencies from pooled sequencing data
Pfreqs(reference = reference, alternative = alternative, coverage = coverage,
min.minor = 2, ifreqs = freqs)
```

`pool2sync` *Create 'synchronized' file from Pool-seq data*

Description

Creates and saves a file with the information from Pool-seq data coded in the 'synchronized' format.

Usage

```
pool2sync(reference, alternative, file, pos = NULL)
```

Arguments

`reference`	is a list where each entry corresponds to a different locus. Each list entry is a vector with the number of reads with the reference allele. Each entry of the vector corresponds to a different SNP. This list can have a single entry if the data is comprised of a single locus.
`alternative`	is a list where each entry corresponds to a different locus. Each list entry is a vector with the number of reads with the alternative allele. Each entry of the vector corresponds to a different SNP. This list can have a single entry if the data is comprised of a single locus.

`file` is a character string naming the file to write to.

`pos` is an optional input (default is NULL). If the actual position of the SNPs are known, they can be used as input here. When working with a single locus, this should be a numeric vector with each entry corresponding to the position of each SNP. If the data has multiple loci, this should be a list where each entry is a numeric vector with the position of the SNPs for a different locus.

Details

It starts by converting the number of reads with the `reference` allele and the `alternative` allele to a A-count:T-count:C-count:G-count:N-count:deletion-count string. Here, we assume that the reference allele is always A and the alternative is always T.

Then, this A-count:T-count:C-count:G-count:N-count:deletion-count string is combined with other necessary information such as the chromosome of each SNP, the position of the SNP and the reference character. This step creates a data frame where each row corresponds to a different SNP.

A `file` is then created and saved in the current working directory, with the Pool-seq data coded in the 'synchronized' file format.

Value

a file in the current working directory containing Pool-seq data in the 'synchronized' format.

Examples

```
# simulate Pool-seq data for 100 individuals sampled at a single locus
genotypes <- run_scrm(nDip = 100, nloci = 1, theta = 5)
# simulate Pool-seq data assuming a coverage of 100x and two pools of 50 individuals each
pool <- simPoolseq(genotypes = genotypes, pools = c(50, 50), pError = 100, sError = 0.001,
mCov = 100, vCov = 250, min.minor = 0)
# create a 'synchronized' file of the simulated data - this will create a txt file
# pool2sync(reference = pool$reference, alternative = pool$alternative, file = "mysync.txt")

# simulate Pool-seq data for 10 individuals sampled at 5 loci
genotypes <- run_scrm(nDip = 10, nloci = 5, theta = 5)
# simulate Pool-seq data assuming a coverage of 100x and a single pool of 10 individuals
pool <- simPoolseq(genotypes = genotypes, pools = 10, pError = 100, sError = 0.001,
mCov = 100, vCov = 250, min.minor = 0)

# create a 'synchronized' file of the simulated data - this will create a txt file
# pool2sync(reference = pool$reference, alternative = pool$alternative, file = "mysync.txt")
```

---

pool2vcf *Create VCF file from Pool-seq data*

Description

Creates and saves a file with the information from Pool-seq data coded in the VCF format.

Usage

```
pool2vcf(reference, alternative, total, file, pos = NULL)
```

Arguments

`reference`	is a list where each entry corresponds to a different locus. Each list entry is a vector with the number of reads with the reference allele. Each entry of the vector corresponds to a different SNP. This list can have a single entry if the data is comprised of a single locus.
`alternative`	is a list where each entry corresponds to a different locus. Each list entry is a vector with the number of reads with the alternative allele. Each entry of the vector corresponds to a different SNP. This list can have a single entry if the data is comprised of a single locus.
`total`	is a list where each entry corresponds to a different locus. Each list entry is a vector with the total number of reads observed at each SNP. Each entry of the vector corresponds to a different SNP. This list can have a single entry if the data is comprised of a single locus.
`file`	is a character string naming the file to write to.
`pos`	is an optional input (default is NULL). If the actual position of the SNPs are known, they can be used as input here. When working with a single locus, this should be a numeric vector with each entry corresponding to the position of each SNP. If the data has multiple loci, this should be a list where each entry is a numeric vector with the position of the SNPs for a different locus.

Details

It starts by converting the number of reads with the `reference` allele, the `alternative` allele and the `total` depth of coverage to a R,A:DP string. R is the number of reads of the reference allele, A is the number of reads of the alternative allele and DP is the total depth of coverage.

Then, this information coded as R,A:DP is combined with other necessary information such as the chromosome of each SNP, the position of the SNP and the quality of the genotype among others. This creates a data frame where each row corresponds to a different SNP.

A `file` is then created and saved in the current working directory, with the header lines that go above the table in a VCF file. Finally, the data frame is appended to that file.

Value

a file in the current working directory containing Pool-seq data in the VCF format.

Examples

```
# simulate Pool-seq data for 100 individuals sampled at a single locus
genotypes <- run_scrm(nDip = 100, nloci = 1, theta = 5)
# simulate Pool-seq data assuming a coverage of 100x and two pools of 50 individuals each
pool <- simPoolseq(genotypes = genotypes, pools = c(50, 50), pError = 100, sError = 0.001,
mCov = 100, vCov = 250, min.minor = 0)
# create a vcf file of the simulated data - this will create a txt file
# pool2vcf(reference = pool$reference, alternative = pool$alternative,
```

```
# total = pool$total, file = "myvcf.txt")

# simulate Pool-seq data for 10 individuals sampled at 5 loci
genotypes <- run_scrm(nDip = 10, nloci = 5, theta = 5)
# simulate Pool-seq data assuming a coverage of 100x and a single pool of 10 individuals
pool <- simPoolseq(genotypes = genotypes, pools = 10, pError = 100, sError = 0.001,
mCov = 100, vCov = 250, min.minor = 0)

# create a vcf file of the simulated data - this will create a txt file
# pool2vcf(reference = pool$reference, alternative = pool$alternative,
# total = pool$total, file = "myvcf.txt")
```

`poolPops` *Create Pooled DNA sequencing data for multiple populations*

Description

This function combines the information for each individual of each population into information at the population level.

Usage

```
poolPops(nPops, nLoci, indContribution, readsReference)
```

Arguments

`nPops`	An integer representing the total number of populations in the dataset.
`nLoci`	An integer that represents the total number of independent loci in the dataset.
`indContribution`	Either a list or a matrix (when dealing with a single locus).
`readsReference`	A list, where each entry contains the information for a single locus. Each list entry should then have one separate entry per population. Each of these entries should be a matrix, with each row corresponding to a single individual and each column a different site. Thus, each entry of the matrix contains the number of observed reads with the reference allele for that individual at a given site. The output of the `numberReference` or `numberReferencePop` functions should be the input here.

Details

In other words, the information of all individuals in a given population is combined into a single population value and this is done for the various populations. In this situation, each entry of the `indContribution` and `readsReference` lists should contain one entry per population - being, in essence, a list within a list. Please note that this function is intended to work for multiple populations and should not be used with a single population.

Value

a list with three names entries

reference	a list with one entry per locus. Each entry is a matrix with the number of reference allele reads for each population. Each column represents a different site and each row a different population.
alternative	a list with one entry per locus. Each entry is a matrix with the number of alternative allele reads for each population. Each column represents a different site and each row a different population.
total	a list with one entry per locus. Each entry is a matrix with the coverage of each population. Each column represents a different site and each row a different population.

Examples

```
# simulate coverage at 5 SNPs for two populations, assuming 20x mean coverage
reads <- simulateCoverage(mean = c(20, 20), variance = c(100, 100), nSNPs = 5, nLoci = 1)

# simulate the number of reads contributed by each individual
# for each population there are two pools, each with 5 individuals
indContribution <- popsReads(list_np = rep(list(rep(5, 2)), 2), coverage = reads, pError = 5)

# set seed and create a random matrix of genotypes for the 20 individuals - 10 per population
set.seed(10)
genotypes <- matrix(rpois(100, 0.5), nrow = 20)

# simulate the number of reference reads for the two populations
readsReference <- numberReferencePop(genotypes = genotypes, indContribution = indContribution,
size = rep(list(rep(5, 2)), 2), error = 0.01)

# create Pooled DNA sequencing data for these two populations and for a single locus
poolPops(nPops = 2, nLoci = 1, indContribution = indContribution, readsReference = readsReference)
```

---

poolProbs *Probability of contribution of each pool*

Description

This function computes the probability of contribution of each pool towards the total depth of coverage of a single population. If multiple pools where used to sequence a single population, it is possible that some pools contribute more than others.

Usage

```
poolProbs(nPools, vector_np, nSNPs, pError)
```

Arguments

nPools	an integer indicating how many pools were used to sequence the population.
vector_np	is a vector where each entry contains the number of diploid individuals of a given pool. Thus, if a population was sequenced using two pools, each with 10 individuals, this vector would contain two entries and both will be 10.
nSNPs	an integer indicating how many SNPs exist in the data.
pError	an integer representing the value of the error associated with DNA pooling. This value is related with the unequal pool contribution towards the total number of reads of a population - the higher the value the more unequal are the pool contributions.

Value

a matrix with the probabilities of contribution for each pool. Each row represents a different pool and each column is a different site.

Examples

```
# probability of contribution at 8 SNPs for 5 pools, each with 10 individuals
poolProbs(nPools = 5, vector_np = rep(10, 5), nSNPs = 8, pError = 50)
```

---

poolReads *Reads contributed by each pool*

Description

This function simulates the contribution, in terms of reads, of each pool. The number of reads contributed from all pools is equal to the total coverage of the population.

Usage

```
poolReads(nPools, coverage, probs)
```

Arguments

nPools	an integer indicating how many pools were used to sequence the population.
coverage	a vector containing the total depth of coverage of the population. Each entry of the vector represents a different site.
probs	a matrix containing the probability of contribution of each pool used to sequence the population. This matrix can be obtained with the poolProbs function.

Value

a matrix with the number of reads contributed by each pool towards the total coverage of the population. Each row of the matrix is a different pool and each column a different site.

Examples

```
# simulate the probability of contribution of each pool
probs <- poolProbs(nPools = 5, vector_np = rep(10, 5), nSNPs = 8, pError = 50)

# simulate the number of reads contributed, assuming 10x coverage for each site
poolReads(nPools = 5, coverage = rep(10, 8), probs = probs)
```

---

popReads *Compute number of reads for each individual and across all sites*

Description

This function computes the contribution of each individual towards the total coverage of a given population.

Usage

```
popReads(vector_np, coverage, pError)
```

Arguments

vector_np	is a vector where each entry contains the number of diploid individuals of a given pool. Thus, if a population was sequenced using two pools, each with 10 individuals, this vector would contain two entries and both will be 10.
coverage	a vector containing the total depth of coverage of the population. Each entry of the vector represents a different site.
pError	an integer representing the value of the error associated with DNA pooling. This value is related with the unequal contribution of both individuals and pools towards the total number of reads observed for a given population - the higher the value the more unequal are the individual and pool contributions.

Details

If multiple pools were used to sequence a population, this will compute the contribution of each pool and then use that to calculate how many reads does that pool contribute. Next, the probability of contribution of each individual is computed and utilized to calculate the number of reads that each individual contributes towards the total number of reads observed in the corresponding pool.

Value

a matrix with the number of reads contributed by each individual. Each row of the matrix corresponds to a different individual and each column to a different site.

Examples

```
# simulate number of reads contributed by each individual towards the total population coverage
# assuming a coverage of 10x at 5 sites and two pools, each with 5 individuals
popReads(vector_np = c(5, 5), coverage = rep(10, 5), pError = 100)
```

---

popsReads *Simulate total number of reads for multiple populations*

Description

Simulates the contribution of each individual towards the total coverage of its population.

Usage

```
popsReads(list_np, coverage, pError)
```

Arguments

list_np	is a list where each entry corresponds to a different population. Each entry is a vector and each vector entry contains the number of diploid individuals of a given pool. Thus, if a population was sequenced using two pools, each with 10 individuals, this vector would contain two entries and both will be 10.
coverage	a matrix containing the total depth of coverage of all populations. Each row corresponds to a different population and each column to a different site.
pError	an integer representing the value of the error associated with DNA pooling. This value is related with the unequal contribution of both individuals and pools towards the total number of reads observed for a given population - the higher the value the more unequal are the individual and pool contributions.

Details

If multiple pools were used to sequence a population, this will compute the contribution of each pool and then use that to calculate how many reads does that pool contribute. Next, the probability of contribution of each individual is computed and utilized to calculate the number of reads that each individual contributes towards the total number of reads observed in the corresponding pool. These steps will be performed for each population, thus obtaining the number of reads contributed by each individual for each population.

Value

a list with one entry per population. Each entry represents the number of reads contributed by each individual towards the total coverage of its population. Different individuals correspond to different rows and different sites to different columns.

Examples

```
# simulate coverage for two populations sequenced at 10x at 5 sites
reads <- simulateCoverage(mean = c(10, 10), variance = c(20, 20), nSNPs = 5, nLoci = 1)

# simulate the individual contribution towards that coverage
# assuming that the first population was sequenced using two pools of 5 individuals
# and the second using a single pool with 10 individuals
popsReads(list_np = list(c(5, 5), 10), coverage = reads, pError = 5)
```

`removeSites` *Apply a minor allele reads threshold*

Description

Removes sites where the total number of minor-allele reads is below a certain threshold.

Usage

```
removeSites(freqs, alternative, coverage, minor, min.minor)
```

Arguments

`freqs`	a vector of allele frequencies where each entry corresponds to a different site.
`alternative`	a matrix with the number of reads with the alternative allele. Each row should be a different population and each column a different site.
`coverage`	a matrix with the total coverage. Each row should be a different population and each column a different site.
`minor`	a matrix with the number of minor-allele reads. Each row should be a different population and each column a different site.
`min.minor`	is an integer representing the minimum allowed number of minor-allele reads. Sites that, across all populations, have less minor-allele reads than this threshold will be removed from the data.

Details

If a site has less minor-allele reads than `min.minor` across all populations, that site is removed from the data.

Value

a list with three named entries:

`freqs`	is a vector with the allele frequencies minus the frequency of the removed sites.
`alternative`	is a matrix with the number of alternative-allele reads per site, minus any removed sites.
`coverage`	is a matrix with the depth of coverage minus the coverage of the removed sites.

Examples

```
# create a vector of allele frequencies
freqs <- runif(20)

set.seed(10)
# create a matrix with the number of reads with the alternative allele
alternative <- matrix(sample(x = c(0,5,10), size = 20, replace = TRUE), nrow = 1)
# create a matrix with the depth of coverage
coverage <- matrix(sample(100:150, size = 20), nrow = 1)
# the number of reads with the reference allele is obtained by subtracting
# the number of alternative allele reads from the depth of coverage
reference <- coverage - alternative

# find the minor allele at each site
minor <- findMinor(reference = reference, alternative = alternative, coverage = coverage)
# keep only the matrix with the minor allele reads
minor <- minor[["minor"]]

# remove sites where the number of minor-allele reads is below the threshold
removeSites(freqs = freqs, alternative = alternative, coverage = coverage,
minor = minor, min.minor = 2)
```

---

remove_by_reads *Apply a coverage-based filter over a list*

Description

This function removes sites that have a coverage below a `minimum` value and sites with a coverage above a `maximum` value. This is done over multiple loci, assuming that each entry of the `reads` list is a different locus. If a list of genotypes is also supplied, then those same sites are also removed from each locus of the genotypes.

Usage

```
remove_by_reads(nLoci, reads, minimum, maximum, genotypes = NA)
```

Arguments

`nLoci`	an integer that represents how many independent loci were simulated.
`reads`	a list with the total depth of coverage. Each entry of the list should be a matrix corresponding to a different locus. Each row of that matrix should be the coverage of a different population and each column a different site.
`minimum`	an integer representing the minimum coverage allowed. Sites where any population has a depth of coverage below this threshold are removed from the data.
`maximum`	an integer representing the maximum coverage allowed. Sites where any population has a depth of coverage above this threshold are removed from the data.

genotypes — an optional list input with the genotypes. Each entry of the list should be a matrix corresponding to a different locus. Each column of the matrix should be a different site and each row a different individual.

Value

a list with the total depth of coverage similar to the reads input argument but without sites where the coverage was below the minimum or above the maximum. If the genotypes were included, a second list entry will also be included in the output, containing the genotypes minus the sites that were removed.

Examples

```
set.seed(10)

# simulate coverage for 10 locus
reads <- simulateCoverage(mean = c(25, 25), variance = c(200, 200), nSNPs = 10, nLoci = 10)

# remove sites with coverage below 10x or above 100x
reads <- remove_by_reads(nLoci = 10, reads = reads, minimum = 5, maximum = 100)
# notice that some locus no longer have 10 SNPs - those sites were removed
reads
```

---

remove_by_reads_matrix

Apply a coverage-based filter to a matrix

Description

This function removes sites that have a coverage below a minimum value and sites with a coverage above a maximum value. If a matrix of genotypes is also supplied, then those same sites are also removed from that matrix.

Usage

```
remove_by_reads_matrix(reads, minimum, maximum, genotypes = NA)
```

Arguments

reads — a matrix with the total depth of coverage. Each row of the matrix should be the coverage of a different population and each column a different site.

minimum — an integer representing the minimum coverage allowed. Sites where any population has a depth of coverage below this threshold are removed from the data.

maximum — an integer representing the maximum coverage allowed. Sites where any population has a depth of coverage above this threshold are removed from the data.

genotypes — an optional matrix input with the genotypes. Each column of the matrix should be a different site and each row a different individual.

Value

a matrix with the total depth of coverage minus the sites (i.e. columns) where the coverage for any of the populations was below the minimum or above the maximum. If genotypes were supplied, then the output will be a list, with one entry per locus. Each entry will contain the filtered coverage in the first entry and the genotypes, minus the removed sites, in the second entry.

Examples

```
set.seed(10)

# simulate coverage for a single locus - select the first entry to obtain a matrix
reads <- simulateCoverage(mean = c(25, 25), variance = c(200, 200), nSNPs = 10, nLoci = 1)[[1]]

# check the coverage matrix
reads

# remove sites with coverage below 10x or above 100x
remove_by_reads_matrix(reads = reads, minimum = 10, maximum = 100)
```

---

run_scrm *Simulate a single population*

Description

Simulates the evolution of biological sequences for a single population with variable theta values.

Usage

```
run_scrm(nDip, nloci, theta = 10)
```

Arguments

nDip	an integer representing the total number of diploid individuals to simulate. Note that scrm::scrm() actually simulates haplotypes, so the number of simulated haplotypes is double of this.
nloci	is an integer that represents how many independent loci should be simulated.
theta	a value for the mutation rate assuming theta = 4Nu, where u is the neutral mutation rate per locus.

Value

a list with genotypes. Each entry of the list corresponds to a different locus. For each locus, the genotypes are in a matrix, with each row representing a different individual and each column a different site.

Examples

```
run_scrm(nDip = 100, nloci = 10)
run_scrm(nDip = 100, nloci = 10, theta = 5)
```

`simPoolseq` *Simulate Pool-seq data*

Description

Simulates pooled sequencing data given a set of parameters and individual genotypes.

Usage

```
simPoolseq(
  genotypes,
  pools,
  pError,
  sError,
  mCov,
  vCov,
  min.minor,
  minimum = NA,
  maximum = NA
)
```

Arguments

`genotypes`	a list of genotypes, where each entry is a matrix corresponding to a different locus. At each matrix, each column is a different SNP and each row is a different individual. Genotypes should be coded as 0, 1 or 2.
`pools`	a list with a vector containing the size (in number of diploid individuals) of each pool. Thus, if a population was sequenced using a single pool, the vector should contain only one entry. If a population was sequenced using two pools, each with 10 individuals, this vector should contain two entries and both will be 10.
`pError`	an integer representing the value of the error associated with DNA pooling. This value is related with the unequal contribution of both individuals and pools towards the total number of reads observed for a given population - the higher the value the more unequal are the individual and pool contributions.
`sError`	a numeric value with error rate associated with the sequencing and mapping process. This error rate is assumed to be symmetric: error(reference - alternative) = error(alternative - reference). This number should be between 0 and 1.
`mCov`	an integer that defines the mean depth of coverage to simulate. Please note that this represents the mean coverage across all sites.
`vCov`	an integer that defines the mean depth of coverage to simulate. Please note that this represents the mean coverage across all sites.

`min.minor` is an integer representing the minimum allowed number of minor-allele reads. Sites that, across all populations, have less minor-allele reads than this threshold will be removed from the data.

`minimum` an optional integer representing the minimum coverage allowed. Sites where the population has a depth of coverage below this threshold are removed from the data.

`maximum` an optional integer representing the maximum coverage allowed. Sites where the population has a depth of coverage above this threshold are removed from the data.

Details

Note that this functions allows for different combinations of parameters. Thus, Pool-seq data can be simulated for a variety of parameters. For instance, different mean depths of coverage can be used to simulate Pool-seq data. It is also possible to simulate Pool-seq data using different pool sizes (by changing the `pools` input) and different values of the Pool-seq error parameter (`pError`).

Value

a list with three named entries:

`reference` a list with one entry per locus. Each entry is a matrix with the number of reference allele reads. Each column represents a different site.

`alternative` a list with one entry per locus. Each entry is a matrix with the number of alternative allele reads. Each column represents a different site.

`total` a list with one entry per locus. Each entry is a matrix with the total depth of coverage. Each column represents a different site.

Examples

```
# simulate Pool-seq data for 100 individuals sampled at a single locus
genotypes <- run_scrm(nDip = 100, nloci = 1, theta = 5)
# simulate Pool-seq data assuming a coverage of 100x and two pools of 50 individuals each
simPoolseq(genotypes = genotypes, pools = c(50, 50), pError = 100, sError = 0.001,
mCov = 100, vCov = 250, min.minor = 0)

# simulate Pool-seq data for 10 individuals sampled at 5 loci
genotypes <- run_scrm(nDip = 10, nloci = 5, theta = 5)
# simulate Pool-seq data assuming a coverage of 100x and a single pool of 10 individuals
simPoolseq(genotypes = genotypes, pools = 10, pError = 100, sError = 0.001,
mCov = 100, vCov = 250, min.minor = 0)
```

simReads *Simulate coverage at a single locus*

Description

Simulates the total number of reads, for each polymorphic site of a given locus using a negative binomial distribution.

Usage

```
simReads(mean, variance, nSNPs = NA, genotypes = NA)
```

Arguments

mean	an integer that defines the mean depth of coverage to simulate. Please note that this represents the mean coverage across all sites. If a vector is supplied instead, the function assumes that each entry of the vector is the mean for a different population.
variance	an integer that defines the variance of the depth of coverage across all sites. If a vector is supplied instead, the function assumes that each entry of the vector is the variance for a different population.
nSNPs	an integer representing the number of polymorphic sites per locus to simulate. This is an optional input but either this or the genotypes matrix must be supplied.
genotypes	a matrix of simulated genotypes, where each column is a different SNP and each row is a different individual. This is an optional input but either this or the nSNPs must be supplied.

Details

The total number of reads is simulated with a negative binomial and according to a user-defined mean depth of coverage and variance. This function is intended to work with a matrix of genotypes, simulating the depth of coverage for each site present in the genotypes. However, it can also be used to simulate coverage distributions independent of genotypes, by choosing how many sites should be simulated (with the nSNPs option).

Value

a matrix with the total coverage per population and per site. Different rows represent different populations and each column is a different site.

Examples

```
# coverage for one population at 10 sites
simReads(mean = 20, variance = 100, nSNPs = 10)

# simulate coverage at one locus with 10 SNPs for two populations:
```

```
# the first with 100x and the second with 50x
simReads(mean = c(100, 50), variance = c(250, 150), nSNPs = 10)
```

`simulateCoverage` *Simulate total number of reads per site*

Description

This function simulates the total number of reads, for each polymorphic site using a negative binomial distribution.

Usage

```
simulateCoverage(mean, variance, nSNPs = NA, nLoci = NA, genotypes = NA)
```

Arguments

`mean`	an integer that defines the mean depth of coverage to simulate. Please note that this represents the mean coverage across all sites. If a vector is supplied instead, the function assumes that each entry of the vector is the mean for a different population.
`variance`	an integer that defines the variance of the depth of coverage across all sites. If a vector is supplied instead, the function assumes that each entry of the vector is the variance for a different population.
`nSNPs`	an integer representing the number of polymorphic sites per locus to simulate. This is an optional input but either this or the `genotypes` list must be supplied.
`nLoci`	an optional integer that represents how many independent loci should be simulated.
`genotypes`	a list of simulated genotypes, where each entry is a matrix corresponding to a different locus. At each matrix, each column is a different SNP and each row is a different individual. This is an optional input but either this or the `nSNPs` must be supplied.

Details

The total number of reads is simulated with a negative binomial and according to a user-defined mean depth of coverage and variance. This function is intended to work with a list of genotypes, simulating the depth of coverage for each site present in the genotypes. However, it can also be used to simulate coverage distributions independent of genotypes, by choosing how many loci to simulate (with the `nLoci` option) and choosing how many sites per locus should be simulated (with the `nSNPs` option).

Value

a list with the total coverage per population and per site. Each list entry is a matrix corresponding to a different locus. For each matrix, different rows represent different populations and each column is a different site.

Examples

```
# simulate 10 loci, each with 10 SNPs for a single population
simulateCoverage(mean = 100, variance = 250, nSNPs = 10, nLoci = 10)

# simulate 10 loci, each with 10 SNPs for two populations:
# the first with 100x and the second with 50x
simulateCoverage(mean = c(100, 50), variance = c(250, 150), nSNPs = 10, nLoci = 10)

# simulate coverage given a set of genotypes
# run scrm and obtain genotypes
genotypes <- run_scrm(nDip = 100, nloci = 10)
# simulate coverage
simulateCoverage(mean = 50, variance = 200, genotypes = genotypes)
```

CHAPTER 4

Integrating Pool-seq uncertainties into demographic inference

João Carvalho, Hernán E. Morales, Rui Faria, Roger K. Butlin & Vítor C. Sousa

Published in *Molecular Ecology Resources* (2023)
DOI: 10.1111/1755-0998.13834

4.1 ABSTRACT

Next-generation sequencing of pooled samples (Pool-seq) is a popular method to assess genome-wide diversity patterns in natural and experimental populations. However, Pool-seq is associated with specific sources of noise, such as unequal individual contributions. Consequently, using Pool-seq for the reconstruction of evolutionary history has remained underexplored. Here we describe a novel Approximate Bayesian Computation (ABC) method to infer demographic history, explicitly modeling Pool-seq sources of error. By jointly modeling Pool-seq data, demographic history and the effects of selection due to barrier loci, we obtain estimates of demographic history parameters accounting for technical errors associated with Pool-seq. Our ABC approach is computationally efficient as it relies on simulating subsets of loci (rather than the whole-genome), and on using relative summary statistics and relative model parameters. Our simulation study results indicate Pool-seq data allows distinction between general scenarios of ecotype formation (single versus parallel origin), and to infer relevant demographic parameters (e.g., effective sizes, split times). We exemplify the application of our method to Pool-seq data from the rocky-shore gastropod *Littorina saxatilis*, sampled on a narrow geographical scale at two Swedish locations where two ecotypes (Wave and Crab) are found. Our model choice and parameter estimates show that ecotypes formed before colonization of the two locations (i.e., single origin) and are maintained despite gene flow. These results indicate that demographic modeling and inference can be successful based on pool-sequencing using ABC, contributing to the development of suitable null models that allow for a better understanding of the genetic basis of divergent adaptation.

Keywords: Pool-seq, demographic inference, Approximate Bayesian Computation, R package, ecotype formation

4.2 INTRODUCTION

Population genomics data can be used to infer the complex demographic and adaptive processes that have shaped natural populations. Next Generation Sequencing (NGS) has revolutionized the field of population genomics, allowing reconstruction of evolutionary histories using thousands of SNPs across the genome (Ellegren, 2014). However, generating and sequencing individual libraries can be expensive and difficult for certain species (e.g., small organisms). In such cases, an effective alternative is to combine DNA from various individuals, producing a single library that is then sequenced (Pool-seq). NGS of pooled samples requires less DNA per individual, reducing the necessary laboratory work by decreasing the number of library preparations needed.

This results in decreased costs while still allowing the comparison of populations on a genomic scale (Schlötterer, Tobler, Kofler, & Nolte, 2014). However, pooling introduces challenges in data analysis due to non-equimolar DNA concentrations and stochastic variation in amplification or sequencing efficiency, which can result in loss of accuracy of allele frequency estimates (Anderson, Skaug, & Barshis, 2014; Cutler & Jensen, 2010; Ellegren, 2014). Furthermore, DNA from multiple individuals can be extracted in batches, combining multiple batches into a single pool for library preparation and sequencing (Morales et al., 2019; Ross, Endersby-Harshman, & Hoffmann, 2019), which can lead to unequal representation due to variation in extraction efficiency and/or non-equimolar concentrations of DNA between batches. Nonetheless, theoretical and empirical comparisons of individual-based sequencing and Pool-seq indicate that when an equal sequencing effort is employed, Pool-seq allows the analysis of more individuals which leads to similar or more precise allele frequency estimates (Futschik & Schlötterer, 2010; Gautier et al., 2013). Although empirical studies showed that individual-based sequencing provides more information to detect fine-scale population substructure (e.g., hybrids and migrants) than Pool-seq, both approaches are suitable for inferring population genetic structure (Chen et al., 2022; Dorant et al., 2019). Indeed, when a large number of samples is available, Pool-seq data results in more accurate estimates of effective population sizes and divergence or admixture time events (Collin et al., 2021). Pool-seq has been used in various studies, ranging from population genomic analysis (Begun et al., 2007; Ferretti, Ramos-Onsins, & Pérez-Enciso, 2013; Rubin et al., 2012) to experimental evolution (Parts et al., 2011; Turner, Stewart, Fields, Rice, & Tarone, 2011; Zhou et al., 2011) and human genetics applications to uncover disease-related mutations (Calvo et al., 2010; Lieberman et al., 2014; Prescott et al., 2015). Yet, using Pool-seq to perform demographic history inference has been hampered by a lack of tools that explicitly model this type of data.

Recent developments in population genomics using simulations include machine learning and model-based inference approaches (Schrider, Shanku, & Kern, 2018; Sheehan & Song, 2016). The latter allows comparing alternative models and estimating parameters. Model-based inference methods, such as Approximate Bayesian Computation (ABC), offer important advantages (for a review see Beaumont et al., 2010 and Hickerson, 2014), because they allow for explicit and joint consideration of evolutionary processes and sampling effects. ABC replaces data with summary statistics (e.g., heterozygosity, dxy, F_{ST}) and uses simulations to select models and estimate parameters. The simplest ABC algorithm is based on a rejection approach (Tavaré, Balding, Griffiths, & Donnelly, 1997), where parameter values (and/or models) sampled from the prior are accepted if the distance between the simulated and observed summary statistics is below a given distance threshold (i.e. tolerance) or rejected otherwise. Accepted parameter values provide a sample of independent points from the posterior distribution. Given its flexibility, ABC has been widely

used in ecology (Pontarp, Brännström, & Petchey, 2019; Zhang, Dennis, Landers, Bell, & Perry, 2017), systems biology (Liepe et al., 2014) and population genetics (Cooke & Nakagome, 2018; Rougemont & Bernatchez, 2018), with various software implementations (Boitard, Rodríguez, Jay, Mona, & Austerlitz, 2016; Cornuet et al., 2014; Huang, Takebayashi, Qi, & Hickerson, 2011; Wegmann, Leuenberger, Neuenschwander, & Excoffier, 2010). However, implementing ABC for whole genome data is challenging (Smith & Flaxman, 2020) due to the heavy computational burden and difficulty in simulating recombination and mutation rate variation along the genome (Jay, Boitard, & Austerlitz, 2019).

Genomic data from natural populations has led to recent progress in the field of speciation, particularly through the study of ecotypes, which represent putative initial stages in speciation (Turesson, 1922). Many studies of ecotype evolution (Fang, Kemppainen, Momigliano, Feng, & Merilä, 2020; Ravinet et al., 2016; Riesch et al., 2017; Van Belleghem et al., 2018) aim to infer if the same phenotypes have evolved in multiple times and locations when facing similar divergent pressures, i.e. in parallel (Faria et al., 2014; Schluter, 2000). The support for natural selection in ecotype formation increases with the number of population replicates studied, but individual sequencing can become prohibitively expensive. Therefore, Pool-seq is useful in studies of parallel adaptation and speciation (Morales et al., 2019). Studies of ecotype formation usually consider two scenarios (Faria et al., 2014; Johannesson et al., 2010): (i) initial adaptive divergence occurred once with subsequent colonization of analogous pairs of environments (single origin scenario); and (ii) colonization of multiple environments was followed by independent evolutionary divergence (parallel origin scenario). Lower genetic distance between ecotypes within a locality, inferred by principal component analysis or structure plots, is frequently interpreted as a signal of parallel evolution. However, ongoing or past gene flow between different ecotypes can complicate the distinction between these scenarios (Faria et al., 2014). Rather, distinguishing between these hypotheses requires an explicit contrast of the different scenarios in a model-based framework (Butlin et al., 2012, 2014).

Model-based inference methods are commonly used to test whether divergence occurred with or without gene flow (Klütsch, Manseau, Trim, Polfus, & Wilson, 2016), whether there is ongoing gene flow (Bakovic et al., 2021), as well as in finding the most likely population tree for a given set of sampled populations (Louis et al., 2014) or estimating relevant demographic parameters (Andrew, Kane, Baute, Grassa, & Rieseberg, 2013). However, they have rarely been used explicitly to contrast different demographic scenarios of ecotype formation, despite some examples using coalescent-based approaches (Hume, Recknagel, Bean, Adams, & Mable, 2018) coupled with maximum composite-likelihoods (Le Moan, Gagnaire, & Bonhomme, 2016). Even in recognized model systems for parallel evolution in natural populations, such as the common rocky-shore gas-

tropod, *Littorina saxatilis*, model-based inference methods have seldom been used. This species, found in locations that span the North Atlantic (Reid, 1996), is characterised by the existence of two ecotypes in close proximity: one adapted to crab predation (hereafter "Crab" ecotype) and another to heavier wave exposure ("Wave" ecotype) (Johannesson et al., 2010). Parallel differentiation of these ecotypes has been suggested before (Butlin et al., 2014; Panova, Hollander, & Johannesson, 2006; Rivas et al., 2018; Westram, Panova, Galindo, & Butlin, 2016) but only a single study, based on a limited number of markers, has contrasted the parallel origin scenario against an explicitly defined alternative hypothesis (Butlin et al., 2014). Thus, there is a clear need for efficient and easy-to-use methods that could readily distinguish between the two scenarios, particularly when that distinction might be complicated by recent gene flow.

Here we present a new R package to perform model choice and estimate demographic history parameters tailored to Pool-seq data. The main novelty is that we explicitly model and account for known sources of error associated with pool-based sequencing. We perform simulation studies to assess whether we can leverage pooled sequencing data to infer demographic parameters using ABC under a relatively simple two-population isolation with migration model and to differentiate between alternative scenarios of ecotype formation in more complex models with four populations. Importantly, we consider different migration rates among loci to account for the effects of selection against migrants at neutral markers linked to barriers against gene flow. We illustrate the application of our ABC method to Pool-seq whole genome data from *L. saxatilis* ecotypes, inferring whether the origin of the ecotypes consisted of a single or repeated parallel events in a narrow geographical area of two locations in Sweden.

4.3 MATERIAL AND METHODS

We developed an ABC method to model Pool-seq data explicitly under scenarios with two and four populations. Importantly, in all demographic models, we include an explicit parameter representing the error associated with the pooling process (e.g., unequal individual contribution) and a parameter representing errors associated with sequencing (e.g., sequencing and/or mapping errors). Below, we describe in detail the demographic models considered and the Pool-seq parameters in separate sub-sections.

4.3.1 ISOLATION WITH MIGRATION MODEL WITH TWO POPULATIONS

We started by considering a two population isolation with migration model with eight parameters (Figure 4.1A), assuming that an ancestral population of size N_{ref} (considered the reference effective size) splits T_{div} generations ago into two populations with constant effective population sizes N_1 and N_2 and with constant migration rates m_{12} and m_{21}. To account for the effects of linked selection due to barrier loci (i.e., effect of selection against migrants at neutral markers that are possibly linked to barriers against gene flow), we considered that a proportion of the genome P_{no} has no migration ($m_{12} = m_{21} = 0$).

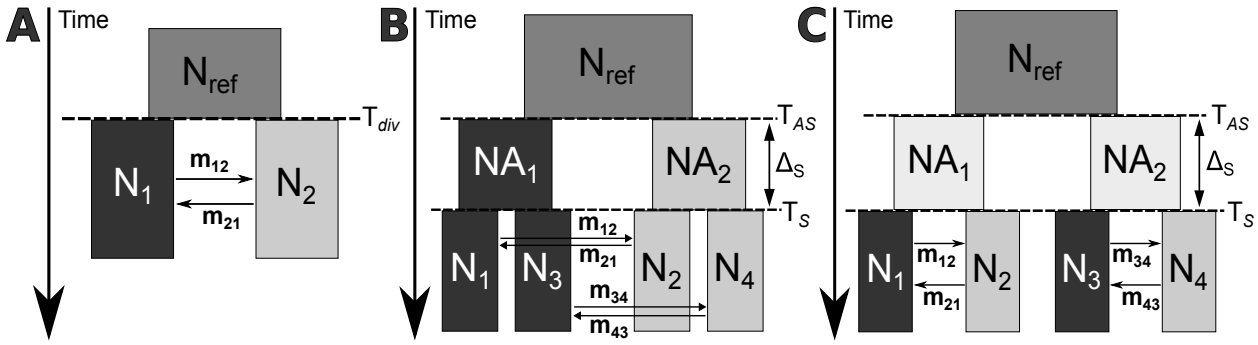


Figure 4.1: Demographic models for the isolation with migration scenario with two populations (A), single (B) and parallel (C) ecotype formation. Dark shading indicates one of the ecotypes, light shading the other ecotype. Parameters used were: N_{ref} - effective size of the ancestral population, NA_1 and NA_2 - size of the two ancestral populations, N_1 - N_4 - sizes of the present-day populations, T_{div} - time of separation of the ecotype populations (in generations), T_s - time of the recent split event (in generations), T_{As} - time of the ancient split event (in generations), Δ_s - time interval between the two split events (in generations), m_{12} - probability per generation that an individual migrates from N_1 to N_2 (forward in time), which corresponds to the probability that lineages move from N_2 to N_1 backwards in time, m_{21} - probability per generation that an individual migrates from N_2 to N_1 (forward in time), which corresponds to the probability that lineages move from N_1 to N_2 backwards in time, m_{34} - probability per generation that an individual migrates from N_3 to N_4 (forward in time), which corresponds to the probability that lineages move from N_4 to N_3 backwards in time and m_{43} - probability per generation that an individual migrates from N_4 to N_3 (forward in time), which corresponds to the probability that lineages move from N_3 to N_4 backwards in time.

4.3.2 MODELS WITH FOUR POPULATIONS: SINGLE VS. PARALLEL ECOTYPE FORMATION

To test the efficiency of our ABC method for distinguishing between different ecotype formation scenarios, we considered two alternative models with four populations. The four extant populations correspond to two ecotypes found at two different locations, i.e., two divergent ecotypes inhabiting

each location. The four-population model has ten relevant demographic history parameters: the population size of the four extant populations (N_1 - N_4) and of the two ancestral populations (NA_1 and NA_2), the time of the recent (T_s) and ancient (T_{As}) split events in generations, and the two migration rates between the two populations ($m_{12} = m_{34}$, $m_{21} = m_{43}$) inhabiting each location. To estimate times of events, we considered as a parameter the time interval between the recent and the ancient split ($\Delta_s = T_{As} - T_s$). Migration rates between divergent ecotypes were assumed to be similar across the two geographic locations (e.g., $m_{12} = m_{34}$ - but note that the scaled migration rate $4Nm$ can be different). A proportion of loci (P_{no}) was also assumed to have no migration between different ecotypes. Depending on the topology, the four-population model can represent: (i) a single origin scenario, where ecotypes are formed in different locations, before dispersing to colonize the two geographic locations (Figure 4.1B); or (ii) a parallel origin scenario, where colonization of each location is followed by independent and parallel divergence of the different ecotypes (Figure 4.1C). Note that for the four-population models we assumed no migration between populations in different locations or between ancestral populations. Thus, the single origin model corresponds to a scenario of divergence of ecotypes without gene flow (i.e., no migration between ancestral populations), whereas in the parallel origin model the divergence of ecotypes occurs within each location with gene flow.

4.3.3 COALESCENT SIMULATIONS OF INDIVIDUAL GENOTYPES

We used coalescent theory to simulate gene trees using *scrm* (Staab, Zhu, Metzler, & Lunter, 2015), under combinations of parameters and models sampled from the priors. Mutations were assumed to occur according to the infinite sites model, with a mutation rate μ per site and per generation. For each locus (i.e., window) in the genome we simulated gene trees with the same sample size, which corresponds to the number of individuals in the pool. In the simulation study we simulated pools of 100 diploid individuals (200 haplotypes) from each population. Thus, when simulating gene trees we assumed the actual haplotypes of all individuals in the pool were known and the effect of pooling was simulated at a later step (see next section). To simulate genotypes, we assumed that individuals within each population were reproducing at random and hence haplotypes were paired at random at each locus to obtain genotypes for each biallelic SNP.

4.3.4 MODELLING POOL-SEQ DATA AND COMBINATION OF POOLS

To model allele frequencies at biallelic SNPs obtained with Pool-seq we follow a series of steps (Figure 4.2). Table 4.1 summarizes the notations used. Sample allele frequencies can be computed as the proportion of reads with a given allele. Thus, they are influenced by the depth of coverage at each single nucleotide polymorphism (SNP), which can vary along the genome due to NGS-associated stochasticity. To account for such variation, we considered that the number of reads at a given site follows a negative binomial distribution (*nBin*), previously shown to fit empirical distributions (e.g., Malaspinas et al. 2016). More precisely, we assumed that, for each SNP, the number of reads C_j for the j^{th} populations follows:

$$C_j \sim nBin(s, \psi) \tag{4.1}$$

where s and ψ are defined as:

$$s = \frac{mean(C_j)}{var(C_j)} \tag{4.2}$$

$$\psi = \frac{mean(C_j)^2}{var(C_j) - mean(C_j)} \tag{4.3}$$

where $mean(C_j)$ and $var(C_j)$ represent, respectively, the mean and variance of the depth of coverage across all SNPs of the j^{th} population. Another source of error in pool-based experiments is heterogeneity on the contribution of each individual to the DNA pool. PCR amplification step(s) during library preparation (e.g., for RAD markers; Baird et al. 2008) can also increase heterogeneity. Moreover, when DNA extraction is performed for several pools of individuals that are combined into a larger pool, uneven contributions between pools might also occur. To account for these uneven individual and pool contributions we assume that, for each site, the number of reads from each individual and each pool follows a multinomial-Dirichlet distribution. We model this sequentially, first obtaining the number of reads for each pool, and then for each individual within each pool.

Table 4.1: **Summary of main notations used.** Note that when we refer to individuals throughout this table, we are referring to diploid individuals.

Notation	Parameter definition
C_j	Total number of reads of the j^{th} population (total coverage)
K	Total number of pools used to sequence the j^{th} population
$\nu_{j,k}$	Number of individuals in the k^{th} pool of the j^{th} population
$\nu_j = \sum_{k=1}^{K} \nu_{j,k}$	Total number of individuals of population j
$E[p_k]$	Expected value of the contribution of the k^{th} pool
$E[p_{k,i}]$	Expected value of the contribution of i^{th} individual of the k^{th} pool
ρ_p	Determines the variance of pools' contribution around their expected value
ρ_i	Determines the variance of individuals' contribution around their expected value
p_k	Contribution (proportion) of reads from the k^{th} pool $\left(\sum_{k=1}^{K} p_k = 1\right)$
$p_{k,i}$	Contribution (proportion) of reads from the i^{th} individual of the k^{th} pool of population j $\left(\sum_{i=1}^{\nu_{j,k}} p_{k,i} = 1\right)$
r_k	Number of reads from the k^{th} pool $\left(r_k = \sum_{i=1}^{\nu_{j,k}} r_{k,i}\right)$ of population j (pool coverage). Note that $C_j = \sum_{k=1}^{K} r_k = \sum_{k=1}^{K} \sum_{i=1}^{\nu_{j,k}} r_{k,i}$
$r_{k,i}$	Number of reads from the i^{th} individual of the k^{th} pool of a given population
D_i	Number of derived allele reads of the i^{th} individual

Following Gautier et al. (2013), we explicitly model the unequal contributions of individuals and pools by increasing the variance of the proportion of reads with two experimental error parameters ε_i and ε_k, for individuals and pools, respectively. As detailed below, these parameters affect the variance of the proportion of reads, as higher variances correspond to higher unequal contributions. Specifically, the number of reads from the k^{th} pool (r_k) follows a multinomial distribution

$$r_k \sim \ mult(C_j, p_k) \tag{4.4}$$

where p_k is the proportion of reads from that pool, which follows a Dirichlet distribution:

$$p_k \sim \ Dir\left(\rho_p \frac{\nu_{j,k}}{\nu_j}\right) \tag{4.5}$$

where $\nu_{j,k}$ is the number of individuals in pool k of population j, ν_j is the number of individuals of the j^{th} population, and ρ_p is a parameter (assumed to be the same for all pools) that determines the variance, and depends on the Pool-seq error of pools ε_k. Following Gautier et al. (2013), we consider that the variance is proportional to the error (ε_k) and expected proportion of reads ($E[p_k]$). We extended their approach to pools of different sizes, considering that the influence of the error ε_k in the variance is partitioned proportionally to the relative size of each pool, i.e., $var(p_k) = \varepsilon_k^2(E[p_k])^2$. The larger the experimental Pool-seq error ε_k, the larger the variance resulting in more unequal contributions. We derived the corresponding ρ_p:

$$\rho_p = \left[\frac{(1-E[p_k])E[p_k]^{-1} - \varepsilon_k^2}{\varepsilon_k^2}\right] = \left[\frac{\frac{\nu_j}{\nu_{j,k}} - 1 - \varepsilon_k^2}{\varepsilon_k^2}\right] \tag{4.6}$$

where $E[p_k] = \nu_{j,k}/\nu_j$. We assumed all pools have the same same ρ_p, which implies that for pools with unequal sizes the pool error ε_k is defined in terms of the pool with a smaller number of individuals (see details in Supplementary Information). This general result simplifies to the results of Gautier et al. (2013) when each pool has a single individual (i.e., $E[p_k] = 1/\nu_j$, Supplementary Information). We used a similar approach to model the individual contribution within each pool. To account for the individual contribution we assumed that, for each site, the number of reads from the i^{th} individual ($r_{k,i}$) of the k^{th} pool follows a multinomial distribution:

$$r_{k,i} \sim \ mult(r_k, p_{k,i}) \tag{4.7}$$

where r_k is the number of reads of pool k, and $p_{k,i}$ is the proportion of reads from individual i in pool k, assumed to follow a Dirichlet distribution:

$$p_{k,i} \sim \ Dir\left(\rho_i \frac{1}{\nu_{j,k}}\right) \tag{4.8}$$

where ρ_i reflects the pool-seq error due to unequal contribution of individuals. Following the same approach as above, ρ_i was obtained based on $var(p_{k,i}) = \varepsilon_i^2(1/\nu_{j,k})^2$ (Supplementary Information):

$$\rho_i = \left[\frac{(1-E[p_{k,i}])E[p_{k,i}]^{-1} - \varepsilon_i^2}{\varepsilon_i^2}\right] = \left[\frac{\nu_{j,k} - 1 - \varepsilon_i^2}{\varepsilon_i^2}\right] \tag{4.9}$$

Note that if there was a single pool, then $\nu_{j,k} = \nu_j$ and $r_k = C_j$, and that the pool-seq error ε_i was assumed equal for all individuals. We derived these results of equations 4.6 to 4.9 in the Supplementary Information, as well as further details on the expected values and variances (Figure A4.1). In sum, this model ensures all individuals are expected to contribute the same number of reads, with errors due to unequal contribution modelled through the dispersion parameters ρ_i and ρ_k. When the experimental error rate tends to zero, the dispersion parameter tends to infinity, resulting in no pooling error as all individuals contribute exactly the same expected number of reads (Gautier et al., 2013). Finally, to account for sequencing and mapping errors, we assumed that, with an error rate ε_{seq}, ancestral allele A will be incorrectly called a derived allele D or vice-versa. More precisely, given the genotype and the total number of reads of the i^{th} individual at a given site, we assumed that the number of reads D_i with the derived allele follows a binomial distribution:

$$D_i \sim \begin{cases} Bin(r_{k,i}, \varepsilon_{seq}) & \text{if individual is AA (homozygous ancestral)} \\ Bin(r_{k,i}, 1-\varepsilon_{seq}) & \text{if individual is DD (homozygous derived)} \\ Bin(r_{k,i}, 0.5) & \text{if individual is AD (heterozygote)} \end{cases} \tag{4.10}$$

where $r_{k,i}$ represents the total number of reads contributed by a particular individual at a given site and ε_{seq} is the combined effect of both the sequencing and mapping errors. We assumed there are only two alleles at each site and that each base has an equal probability of being miscalled. Hence for heterozygotes each allele originates from either the ancestral or derived allele with equal probability (Li et al., 2012).

4.3.5 ABC IMPLEMENTATION USING SUBSETS OF LOCI

To avoid the computational burden of simulating whole genomes, we simulated sets of L independent loci with 2000 sites. We assumed that loci were independent, i.e., with free recombination between all pairs of loci ($r_b = 0.5$), and that within each locus of 2000 sites there was no recombination ($r_w = 0.0$). Our ABC implementation, based on a rejection algorithm, involved several steps: (i) sample demographic and pool-seq parameters from prior distributions (Table 4.2); (ii) simulate genotypes for each individual at L loci using coalescent gene trees based on demographic history parameters; (iii) simulate the number of reads and pooling of individuals for each biallelic SNP, applying filters (e.g., depth of coverage and minor allele frequency); (iv) compute summary statistics for observed and simulated data; (v) calculate Euclidean distance between observed and

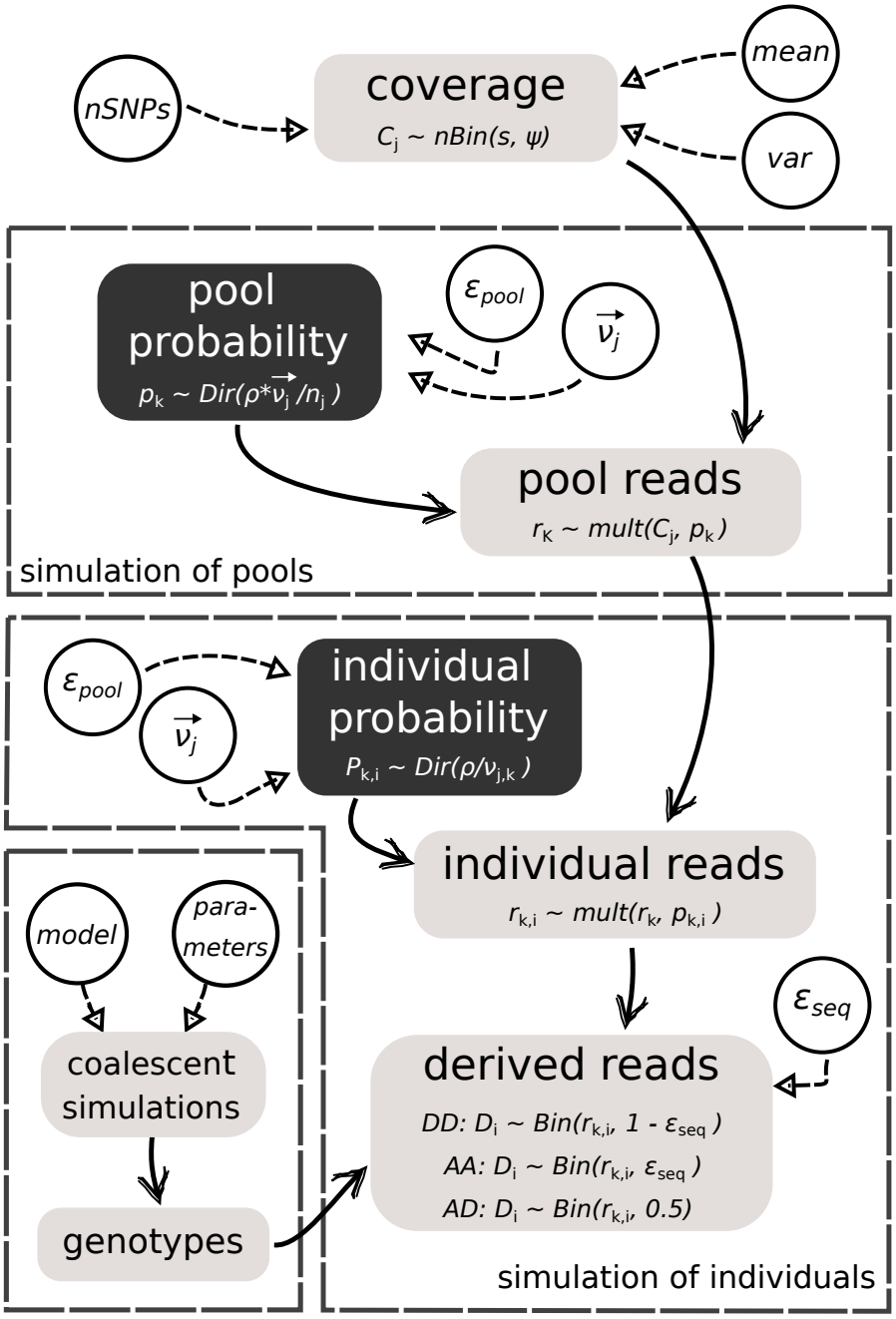


Figure 4.2: Schematics of the steps needed to simulated Pool-seq data. Dark colored boxes denote steps related with probabilities of contribution and circles represent necessary inputs for the corresponding step. Important formulas for each step are included inside the relevant box.

simulated summary statistics, standardizing to ensure that all summary statistics have the same mean and variance; (vi) reject parameters with distances above a tolerance threshold; (vii) apply a post-processing regression to adjust accepted parameter values (Beaumont, Zhang, & Balding, 2002). To simulate coalescent gene trees, we assumed all loci within a subset share the same demographic history, but set migration rate to zero at a proportion of loci P_{no} to account for selection

effects due to barrier loci. For each resulting SNP, pool-seq data were simulated (Figure 4.2) by sampling depth of coverage from a negative binomial (equation 4.1) based on the observed mean and variance of the coverage of each population. To mimic common filter steps, we discarded SNPs with a depth of coverage outside a given range. For instance, for the *L. saxatilis* data, we kept only sites with a depth of coverage between 50x and 150x (see below). We then simulated each pool's contribution (equations 4.4 and 4.5) to the total coverage of a population and each individual's contribution to their pool's coverage (equations 4.7 and 4.8) by randomly sampling values from their respective distributions. Finally, we randomly drew the number of reads from the derived and ancestral alleles for each individual (equation 4.10), and then applied a filter to discard SNPs with fewer than two minor-allele reads. Note that we did not consider sequencing errors at invariant sites, as Pool-seq data were only simulated for polymorphic sites, and any such errors would likely be removed by the minor-allele frequency filter. However, the frequency of sequencing errors at invariant sites increases with increasing coverage and a higher minor-allele count threshold might be required when working with high coverage data (Figure A4.2).

Table 4.2: **Prior distributions and their ranges for each parameter.** Parameters are presented for the four-population models and, when relevant, for the two-population model. n_i - relative sizes of the extant populations (n_1, n_2, n_3, n_4); na_i - relative sizes of the ancestral populations (na_1, na_2); t_{div} - relative time of the split event in the two-population model, t_s - relative time of the recent split event; δ_s - relative time interval between t_s and the ancient split event (t_{As}); ε_{pool} - experimental error introduced by the pooling procedures; ε_{seq} - error associated with sequencing and mapping errors; m_{ij} - probability per generation that an individual migrates from the N_1 or N_3 (Crab) population to the N_2 or N_4 (Wave) population (forward in time), m_{ji} - probability per generation that an individual migrates from the N_2 or N_4 (Wave) population to the N_1 or N_3 (Crab) population (forward in time) and P_{no} - proportion of the simulated loci where no migration occurs between ecotypes.

		Two-population model		Four-population models	
parameter	distribution	minimum	maximum	minimum	maximum
n_i	log-uniform	0.1	3	0.1	3
na_i	log-uniform	-	-	0.1	3
t_{div}	uniform	0	3	-	-
t_s	uniform	-	-	0	3
δ_s	uniform	-	-	0	3
ε_{pool}	uniform	5	250	5	250
ε_{seq}	uniform	0.0001	0.001	0.0001	0.001
m_{ij}	uniform	10^{-13}	10^{-3}	10^{-13}	10^{-3}
m_{ji}	uniform	10^{-13}	10^{-3}	10^{-13}	10^{-3}
P_{no}	beta	0	0.5	0	0.5

For each model, at least 5×10^5 simulations of $L = 300$ loci with $b = 2000$ base pairs were conducted. To reduce computational burden, parameter and summary statistic tables were saved and reused to analyze different subsets of loci from the observed data. To obtain posterior dis-

tributions, we combined 1000 subsets of $L = 300$ loci randomly selected from the observed data. Each subset was processed through steps (v) to (vii) of the ABC algorithm, resulting in a sample of independent points from the posterior of each parameter or model. We combined the independent posterior samples from the 1000 subsets of loci, taking into account the distance between the mean summary statistics of each subset and the overall mean across all loci in the genome. This was done using the Epanechnikov kernel, which assigns more weight to subsets of loci with means closer to the overall mean (Figure A4.3). Since demographic history is expected to affect all loci similarly across the genome, this approach aimed to minimize the impact of outlier subsets of loci on the posterior estimates. All steps were performed using custom-made functions and scripts in R, adapted from Beaumont et al. (2002).

4.3.6 RELATIVE SUMMARY STATISTICS AND SCALED PARAMETERS

We selected a set of statistics (Table A4.1) to summarize the patterns of relative diversity and differentiation within and among populations (Fraïsse et al., 2021; Jay et al., 2019), computed only for polymorphic sites across all populations. Namely, we considered: (i) expected heterozygosity per population and between all pairs of populations (Nei & Roychoudhury, 1974); (ii) pairwise F_{ST} between all pairs of populations (Bhatia, Patterson, Sankararaman, & Price, 2013); (iii) proportion of SNPs with fixed differences between populations (Fraïsse et al., 2021); (iv) proportion of exclusive SNPs within each population (Fraïsse et al., 2021); and for the four population models (v) several D-statistics with different combinations of P1, P2 and P3 populations (adapted from Malinsky, Matschiner, and Svardal (2021)). To capture the distribution across loci, we considered the mean and standard deviation of the above statistics. For F_{ST}, we further considered the 5% and the 95% quantiles because these should capture the effect of barriers to gene flow. In sum, we considered 13 summary statistics for the two-population model and 57 for the four-population models (Table A4.1).

Importantly, all these summary statistics are relative measures of diversity and differentiation that depend on relative branch lengths of coalescent trees (e.g., F_{ST}). Hence, we increased the efficiency of simulations by inferring relative demographic parameters scaled by the ancestral effective population size N_{ref}. We estimated relative effective sizes (e.g., $n_1 = N_1/N_{ref}$), relative times of divergence (e.g., $\delta_s = \Delta_s/4N_{ref}$), and scaled migration rates (e.g., $4N_1m_{21}$). To clarify, note that all relative parameters are represented with a lower case (e.g., n_1), while the absolute parameters are indicated with upper case letters (e.g., N_1) and that scaled migration rates specify which population is receiving immigrants by the subscript next to N. Estimation of relative parameters was done by performing coalescent simulations, fixing the ancestral effective population size to $N_{ref} = 25000$

and the mutation rate to $\mu = 1.5 \times 10^{-8}$ per site, as previously used for *L. saxatilis* (Butlin et al., 2014). To obtain absolute parameter estimates, we re-scaled parameters based on a re-scaling factor $f = obs[S]/E[S]$ that depends on the observed number of SNPs ($obs[S]$), and on the expected number of SNPs according to parameter estimates of a given model ($E[S]$). Assuming the infinite sites mutation model, the expected number of segregating sites was calculated based on the expected total branch length ($E[T]$), mutation rate per site (μ) and number of sites (L) as $E[S] = E[T]\mu L$ (Hudson, 1990). To obtain $E[T]$ we simulated 100,000 gene trees according to parameter estimates of a given model. The absolute effective population sizes and times of events in generations were obtained by multiplying by the rescaling factor f, i.e., $N_e = f \times n_e$ and $T_s = f \times t_s$, respectively.

4.3.7 SIMULATION STUDY

For the two-population model, estimates were based on 10^6 simulations, whereas for the four-population scenarios they were based on 5×10^5 simulations for each scenario of ecotype formation. For each simulation, we generated 300 independent loci with 2000 base pairs, sampling 100 diploid individuals from each population. For each population, Pool-seq data were simulated assuming 20 pools, each with 5 individuals (i.e., $\nu_{j,k}$ = 5). Parameter values were sampled from uniform or log-uniform prior distributions summarized in Table 4.2. The exception was the proportion without migration (P_{no}), which was sampled from a Beta distribution reflecting a low proportion of loci without migration *a priori*. For P_{no} we truncated the distribution, replacing values below 0.01 and above 0.50 by 0.00 and 0.50, respectively (Table 4.2). For the sequencing error, ranges for the prior distribution were based on error rate reports of the technology used (Illumina HiSeq2500; Stoler and Nekrutenko 2021), as well as previous filtering steps (see "*Littorina saxatilis* Pool-seq data" section below).

To evaluate the accuracy of our ABC implementation for Pool-Seq data to estimate parameters and model choice, we performed a leave-one-out cross-validation (Csilléry, François, & Blum, 2012). Hereafter, we use the term accuracy to indicate how close (or far off) a particular point estimate is to the true parameter value. Briefly, a random simulation was picked, and its summary statistics were used as pseudo-observed data. The remaining simulations were used to infer the parameters of the selected simulation. The ABC estimation was repeated for n pseudo-observed datasets.

The prediction error was computed as:

$$\varepsilon_{pred} = \frac{1}{n} \cdot \frac{\sum_{i=1}^{n} (\hat{\Theta}_i - \Theta_i)^2}{var(\Theta)} \tag{4.11}$$

where Θ_i is the true parameter value of the i^{th} pseudo-observed dataset, $\hat{\Theta}_i$ is the estimated parameter value, and $var(\Theta)$ is the variance of the true parameter values. For parameter inference, we assessed the prediction error with $n = 5000$, considering three different point estimates (mode, median and mean of the posterior distribution), at two tolerance values (0.005 or 0.01). For comparison, we computed prediction errors using the mean of the prior distribution as point estimates. For evaluating the model choice we used $n = 1000$ pseudo-observed datasets. To define the model estimated for each pseudo-observed dataset, we considered two posterior probability thresholds: (i) 0.5, assigning a dataset to the model with posterior probability larger than 0.5; (ii) 0.9, a more stringent criterion assigning a dataset to a model only if the posterior was larger than 0.9, classifying it as "unclear" otherwise.

4.3.8 EFFECT OF EXPLICITLY MODELING POOL-SEQ ERRORS

By assuming that the proportion of reads with a given allele corresponds to the allele frequencies, it is possible to analyse Pool-seq data with existing model-based methods, e.g., fastsimcoal2 (Excoffier et al., 2021) and DIYABC random forest (DIYABC-RF) (Collin et al., 2021). Yet, ignoring Pool-seq associated errors due to unequal individual contribution might result in biased demographic estimates. To assess whether this is the case, and whether accounting for Pool-seq errors improves inference of demographic parameters, we compared estimates obtained either ignoring or explicitly modelling Pool-seq errors. We simulated a pseudo-observed Pool-seq dataset (i.e., with variable depth of coverage at each site and unequal individual contribution) according to the parameter estimates obtained for *L. saxatilis* with the two population model. We performed parameter inference using the regression adjustment with priors defined in Table 4.2 using $L = 100$ loci, 500k simulations and a tolerance of 0.01, either: (i) ignoring Pool-Seq errors by computing summary statistics directly from the simulated haplotypes; (ii) explicitly accounting for depth of coverage variation, unequal individual contribution and sequencing errors by computing summary statistics after simulating Pool-seq data as described above.

4.3.9 EFFECT OF NUMBER OF LOCI

To increase computational efficiency we simulate multiple subsets of $L = 300$ loci, rather than entire genomes. To assess the impact of varying the number of loci in a single subset (i.e., varying L), we conducted 100k simulations with 10, 30, 100 or 300 simulated loci per subset using the two population isolation with migration model and the priors defined in Table 4.2. We then performed a leave-one-out cross-validation, without combining multiple subsets of loci, as described above. We computed the prediction error using the mean of the regression-adjusted posterior as a point estimate for $n = 5000$ pseudo-observed datasets, with a tolerance of 0.01. To obtain the 95% confidence interval of the prediction error, we used a non-parametric bootstrap approach resampling 10k times the $n = 5000$ point estimates and re-calculating the prediction error.

4.3.10 EFFECT OF COMBINING MULTIPLE SUBSETS OF LOCI TO OBTAIN POSTERIORS

Our method relies on combining posteriors obtained from multiple subsets of loci, giving more weight to subsets of loci with summary statistics closer to the mean whole-genome values. To evaluate the impact of this strategy we compared estimates obtained with the whole-genome with estimates obtained by merging the posteriors of random subsets of loci, varying the proportion of the genome sampled (10%, 30%, or 50% of the genome). To reduce the computational burden, we assumed that the whole-genome consisted of 100 loci. Using the two-population isolation with migration model, we generated 100 pseudo-observed whole-genomes according to the parameter estimates of *L. saxatilis*. Using the same model and the priors defined in Table 4.2, we conducted 100k simulations with 10, 30, or 50 loci per subset. Then, for each whole-genome, we sampled with replacement 100 subsets corresponding to either 10%, 30%, or 50% of the genome (i.e., 10, 30 or 50 loci). Note that this means that the same locus can appear in multiple subsets, as the goal was to evaluate the impact of combining posterior probabilities obtained from various subsets of loci, even if certain loci may appear in multiple subsets.

We performed parameter inference by merging the posteriors of the 100 subsets and using the regression adjustment with a tolerance of 0.01. These estimates were compared to the approach of Boitard et al. (2016), by using summary statistics computed from the whole-genome as a target to perform parameter inference, but using for inference summary statistics computed from a proportion of the genome, either with 10, 30 or 50 loci. We computed the bias of the estimates using $\frac{1}{n} \cdot \sum(\hat{\Theta}_i - \Theta_i)$, where $\hat{\Theta}_i$ is the estimated mean posterior with subsets of loci, and Θ_i is the mean

posterior with 100 loci (mimicking the whole genome) for the i^{th} pseudo-observed dataset, while $n = 100$ is the number of simulated pseudo-observed datasets.

4.3.11 IMPACT OF IGNORING WITHIN-LOCUS RECOMBINATION

Our models assume free recombination between loci ($r_b = 0.5$) but no recombination within loci ($r_w = 0.0$). We evaluated the effect of this assumption on parameter estimates by comparing the posteriors obtained for pseudo-observed datasets with within-locus recombination to those obtained for datasets simulated without recombination to assess if ignoring within-locus recombination leads to changes in posteriors and thus impacts our estimates. This was done by simulating 100 pseudo-observed datasets according to the estimates obtained for *L. saxatilis*. Each dataset contained 100 loci with within-locus recombination rate equal to the mutation rate ($r_w = \mu$). We then estimated the parameters using the regression adjustment with 500k simulations and a tolerance of 0.01, under our assumption of no within-locus recombination.

4.3.12 *LITTORINA SAXATILIS* POOL-SEQ DATA

We illustrate the application of our ABC implementation to previously published Pool-seq data (Morales et al., 2018) from *L. saxatilis* populations sampled at two different sites in Sweden (Arsklovet and Ramsö). At each of those sites, 100 females of the Crab and another 100 females of the Wave ecotype were sequenced in two separate pools (Morales et al., 2019). DNA extraction was performed for batches of five individuals by combining pieces of foot muscle tissue from five snails in one tube. Reads were trimmed with Trimmomatic v.0.36 (Bolger, Lohse, & Usadel, 2014) and mapped against the *L. saxatilis* reference genome, produced from a single Crab ecotype individual (Westram et al., 2018), using CLC v5.0.3 (`www.qiagenbioinformatics.com`). Only those reads with a mapping score higher than Q20 were retained. Bam files were processed with SAMtools v1.3.1 (Danecek et al., 2021), BEDtools v2.25.0 (Quinlan & Hall, 2010), and Picard tools v2.7.1 (`http://broadinstitute.github.io/picard`) and, for each set of bam files, reads with base quality lower than 30, mapping quality lower than 20 and those that mapped to very short contigs (<500 bp) were filtered out. We removed sites with a coverage lower than 50x or higher than 150x, ensuring we discarded low-coverage sites that would not contain reads for most individuals (<50x) and sites at putative repetitive or duplicated regions leading to unusually high depth of coverage (>150x). We also removed sites with fewer than two minor-allele reads observed across all populations.

Recent studies have uncovered an important role of chromosomal inversions in the adaptive divergence of *L. saxatilis* ecotypes (Faria et al., 2019; Koch et al., 2021; Morales et al., 2019). Each inversion likely has its unique evolutionary history that may be influenced by various demographic and selective processes, such as divergent and balancing selection, and may differ from the population history. Therefore, to avoid biased estimates, inversion-tailored inference methods would be required, accounting for specific features such as varying recombination rates between homozygotes and heterozygotes. Since our aim was to infer the demographic history, an approach tailored to inversions is outside the scope of this study. Thus, we took a conservative approach removing regions that could be associated or linked with the reported inversions (Westram, Faria, Johannesson, & Butlin, 2021) (list of kept and removed contigs in Supplementary File S2). As breakpoints are not yet defined for many inversions, we removed 3671 contigs within inversions or in buffer regions. This corresponds to 3.3% of the whole genome Pool-seq dataset, distributed across the genome but with approximately 1/3 of the removed contigs located in chromosomes 10 and 12. To maximize the number of SNPs we kept all the remaining contigs, although only 20% of them map to known collinear regions (Westram et al., 2018).

We estimated parameters of the two-population model for the two ecotypes from Arsklovet using the prior distributions and 10^6 simulations used for the simulation study. Similarly, we performed model choice and estimated parameters for the four-population models using 5×10^5 simulations, estimating parameters for the model with the highest posterior probability. Keeping in line with our strategy of using subsets of loci, we considered each contig in the *L. saxatilis* dataset as an independent locus. We obtained posteriors by merging 1000 subsets of 300 loci (i.e. $L = 300$), which were randomly selected. For each subset, we implemented a random selection process (without replacement) to choose 300 contigs. Subsequently, from each of these selected contigs, we further randomly selected a window of $b = 2000$ base pairs. Summary statistics were then computed for each subset based on the 300 loci (i.e., the 300 selected windows). Although the number of contigs in the genome was large (54,201), contigs were re-used in different subsets. However, even in such cases, it is highly probable that a different window of 2000 base pairs would be chosen, resulting in a unique combination of loci. To estimate parameters we computed the mean posterior (point estimate) and 95% credible intervals based on the weighted quantiles. Since this dataset only contained SNPs, remaining sites could be monomorphic or missing data. To re-scale the parameters, we calculated the number of SNPs per window assuming that the remaining sites were monomorphic. We converted time of events in generations to years, assuming a generation time of 0.5 years (Butlin et al., 2014).

4.4 RESULTS

4.4.1 PERFORMANCE OF ABC POINT ESTIMATES

To evaluate the performance of our ABC implementation we performed a simulation study, summarizing the posterior distributions with three point estimates (mean, median and mode). When using $L = 300$ loci, prediction errors were lower using the mean or median with the regression-based adjustment for all the parameters (Tables A4.2, A4.3 and A4.4). As expected, with the regression, tolerance had a negligible effect in the prediction error. Additionally, prediction errors decreased with increasing number of simulated loci in the subsets, despite a clear trend of diminishing returns with more than $L = 100$ loci (Figure A4.4). Thus, unless specified, hereafter we summarize results obtained with subsets of $L = 300$ loci, using the regression-based adjustment and the mean as a point estimate, with a tolerance of 0.01.

Although the set of summary statistics was different for the two and four-population models, the prediction errors were similar for most parameters (Figures A4.5 and A4.6). For the relative effective sizes of extant populations (Figure 4.1), prediction errors ranged from 0.110 to 0.119 for the two-population model (Table 4.3, panel A in Figure 4.3), from 0.111 to 0.127 for the single origin (Figure 4.3B), and from 0.121 to 0.140 for the parallel origin (Table 4.3), indicating that the mean of posteriors provide accurate point estimates. For the sizes of ancestral populations in the four-population models (absolute values indicated by NA_1 and NA_2 in Figure 4.1), prediction errors were higher in the single origin than in the parallel origin (Table 4.3). For both models, the relative sizes of ancestral populations, na_1 and na_2, attained the highest prediction errors across all parameters, ranging from 0.530 to 0.616, indicating that point estimates are less accurate for ancestral effective sizes. Nevertheless, since prediction errors are smaller than the ones obtained when using the mean of the prior (close to 1), the shape of the posterior indicates that the summary statistics provide information about such parameters. For the relative timing of the split events, prediction errors were higher in the two-population model (0.34, Table 4.3, Figure 4.3D), than in the four-population models (ranging from 0.036 to 0.182). For the relative time of recent split (t_s), prediction errors were lower in the single origin model (0.036) than in the parallel model (0.172, Table 4.3), whereas for the relative time interval between split events (δ_s), prediction errors were similar for both models (0.182 for single, 0.179 for parallel) (Figure 4.3C-F and Figure A4.6B-C and H-I).

Table 4.3: **Prediction errors for parameter estimation.** Prediction errors were computed using the mean of the posterior distribution, obtained after the regression adjustment and a tolerance of 0.01. Prior mean indicates the prediction error if the mean of the prior distribution were used as point estimates. n_1 to n_4 - relative population sizes of the extant populations; na_1 and na_2 - relative population sizes of the ancestral populations; t_{div} - relative time of the split event in the two-population model; t_s - relative time of the split event that lead to the origin of the current populations; δ_s - relative time interval between t_s and the ancient split event (t_{As}); ε_{pool} - experimental error introduced by the pooling procedures; ε_{seq} - error associated with sequencing and mapping errors; m_{12}, m_{34} - probability per generation that an individual migrates from the N_1 or N_3 (Crab) population to the N_2 or N_4 (Wave) population (forward in time), m_{21}, m_{43} - probability per generation that an individual migrates from the N_2 or N_4 (Wave) population to the N_1 or N_3 (Crab) population (forward in time); $4N_2m_{12}$ and $4N_1m_{21}$ - average number of immigrants per generation ($4Nm$) from N_1 to N_2 and from N_2 to N_1 (respectively) at the first site; $4N_4m_{34}$ and $4N_3m_{43}$ - equivalent immigration rates at the second site and P_{no} - proportion of the simulated loci where no migration occurs between ecotypes.

parameter	prior mean	two-population	single origin	parallel origin
n_1	0.997	0.119	0.111	0.128
n_2	0.998	0.110	0.113	0.121
n_3	0.997	–	0.121	0.140
n_4	0.999	–	0.127	0.129
na_1	0.998	–	0.596	0.530
na_2	1.000	–	0.616	0.549
t_{div}	1.000	0.342	–	–
t_s	1.000	–	0.036	0.172
δ_s	1.001	–	0.182	0.179
ε_{pool}	1.000	0.242	0.243	0.241
ε_{seq}	1.001	0.592	0.062	0.042
m_{12}, m_{34}	1.000	0.401	0.396	0.448
m_{21}, m_{43}	1.001	0.448	0.399	0.439
$4N_2m_{12}$	0.999	0.284	0.325	0.311
$4N_1m_{21}$	0.998	0.293	0.287	0.329
$4N_4m_{34}$	0.996	–	0.298	0.319
$4N_3m_{43}$	1.000	–	0.298	0.340
P_{no}	1.000	0.072	0.041	0.124

Regarding the migration rates, although we specified prior immigration rates m_{ij} (probability that a lineage migrates from population i to j forward in time per generation), we focus on the average number of immigrants per generation ($4N_jm_{ij}$, where N_j is the effective size of the population receiving immigrants) as it accounts for both migration (proportional to m_{ij}) and drift (proportional to N_j), with $4N_jm_{ij} > 1$ indicating that migration occurs at a higher rate than drift. Prediction errors for $4N_jm_{ij}$ were similar for the two and four-population models, ranging from 0.284 to 0.340 (Table 4.3), although slightly higher in the parallel origin model. Across all models, the accuracy of the mean of the posterior decreased when the immigration used in simulations was too high, with poorer estimates when true $4N_jm_{ij} >> 10$. Overall, prediction errors for $4N_jm_{ij}$ were higher than for times of split and extant effective sizes, indicating that it is harder to accurately infer migration.

The proportion of loci without migration (P_{no}) was accurately estimated, as supported by the very low prediction errors for the two and four-population models (Table 4.3).

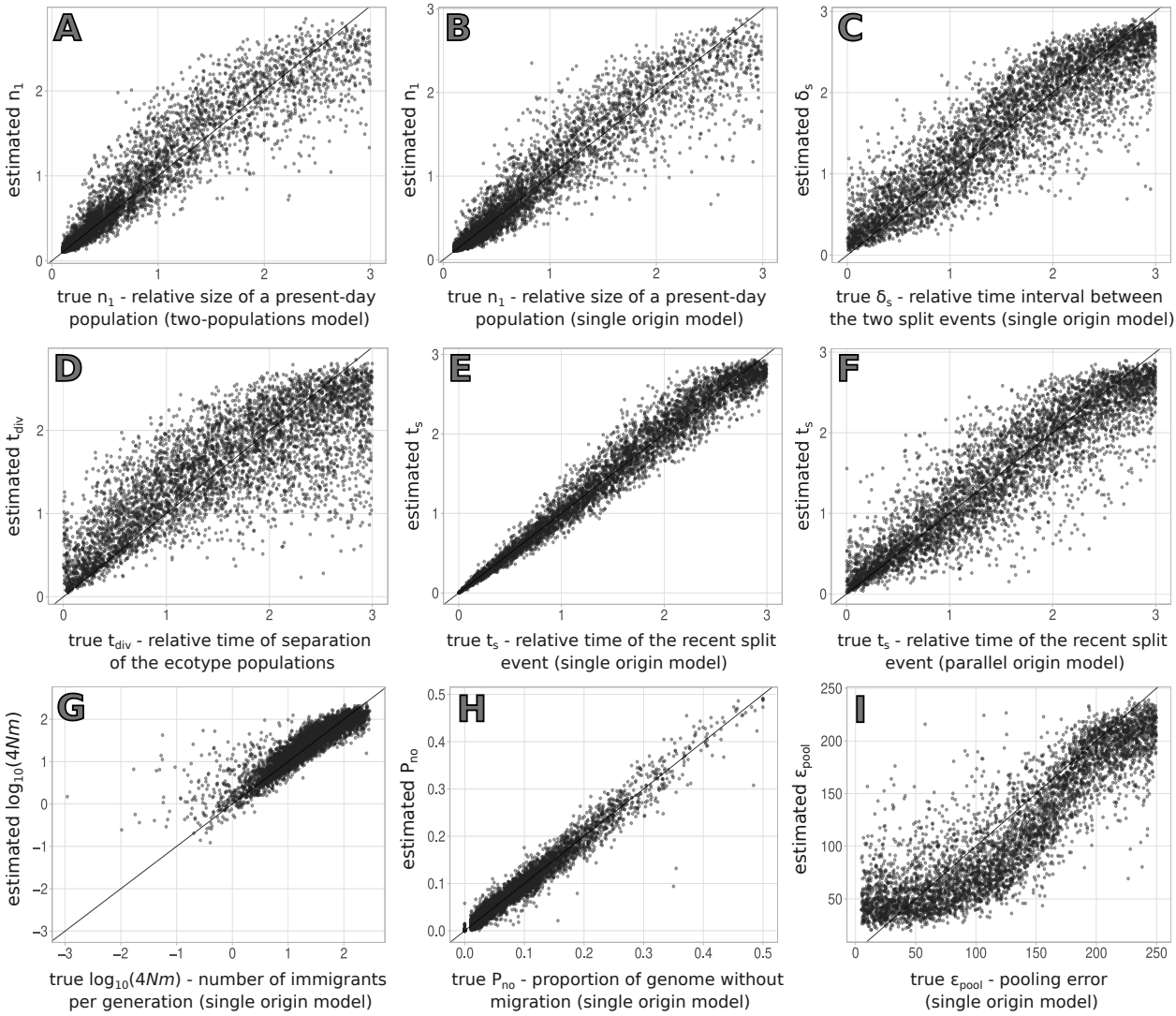


Figure 4.3: Results of the cross-validation for parameter estimation. The y-axis displays the estimated values, plotted against the true parameter values on the x-axis. Estimates correspond to the mean of the posterior obtained with a tolerance rate of 0.01. Parameters shown here are: A - relative size of a present-day population (n_1) of the two-population model; B - relative size of a present-day population (n_1) of the single origin model; C - time interval between the two split events (δ_s); D to F - time of the split event (t_{div}) for the two-population model and time of the recent split (t_s) for the single origin model and the parallel origin model (respectively); G - average number of immigrants per generation in log_{10} scale ($4Nm$); H - proportion of the genome without migration between different populations (P_{no}) and I - pooling error.

Ignoring pooling and sequencing errors resulted in biased estimates for most demographic parameters (Figure 4.4 and Table A4.5), when pseudo-observed Pool-seq data were analysed without modelling explicitly the joint effect of variation in depth of coverage, unequal individual contri-

bution and sequencing errors. Importantly, this is ignored by current demographic inference approaches (e.g., DIYABC-RF or fastsimcoal2). In contrast, our ABC approach based on explicitly modelling these sources of Pool-seq error provides accurate estimates (Figure 4.4).

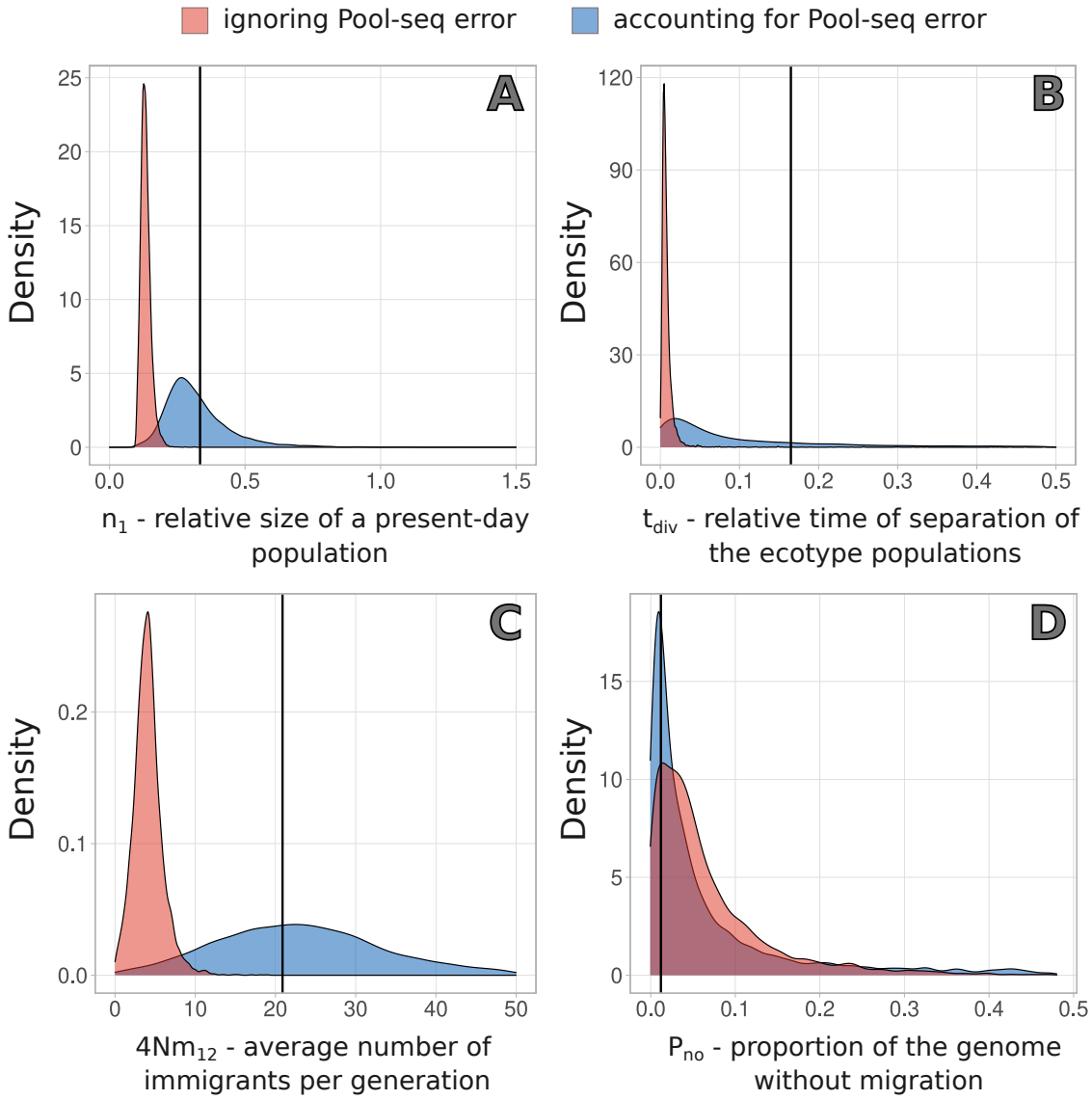


Figure 4.4: Impact of ignoring Pool-seq errors on demographic parameter estimates. Posterior obtained for a pseudo-observed Pool-seq dataset using either our ABC approach that explicitly accounts for Pool-seq errors (blue), or ignoring Pool-seq errors by using directly simulated allele frequencies (red). The parameters shown here are: A - relative size of a present-day population (n_1), B - relative time of separation of the ecotype populations (t_{div}), C - average number of immigrants per generation ($4Nm_{12}$) and D - proportion of the genome without migration (P_{no}). The black line represents the true parameter value used to simulate the pseudo-observed dataset with $L = 100$ loci.

Although our aim was to demonstrate the implementation of an ABC method to perform parameter inference and model selection while explicitly modelling Pool-Seq data, treating pooling and sequencing errors as nuisance parameters, we report the prediction error for those parameters. The accuracy of the inference of the pooling error was similar to that of other parameters, with errors ranging from 0.241 to 0.243 (Table 4.3). This parameter was reasonably well estimated by the posterior mean when simulations were done with pooling errors above 150% (Figure 4.3I and Figures A4.5F and A4.6F-L). For the sequencing error, prediction error was higher for the two population (0.592) than for the four population models (0.042 - 0.062, Table 4.3), probably because there is more information in models with more individuals.

4.4.2 PERFORMANCE OF MODEL CHOICE

Results of the simulation study indicate that our ABC implementation allows a distinction between the single and parallel origin scenarios considered. Out of the 1,000 pseudo-observed datasets analysed under each model, using a 50% posterior probability threshold, the model was correctly inferred for 975 datasets of parallel origin (mean posterior probability of 0.952), and for 937 of single origin (mean posterior probability of 0.927, Figure 4.5A). When the model with the highest posterior was incorrect, its posterior probability was substantially lower (0.703 when parallel was inferred as single, and 0.755 when single was inferred as parallel). Using a more stringent threshold of 90% posterior probability, ABC still allowed to disentangle the two scenarios. The number of pseudo-observed datasets for which the model was correctly inferred was 877 for the parallel origin (one incorrectly assigned to the single model and 122 classified as unclear), and 854 for the single origin (12 incorrectly assigned to parallel and 134 classified as unclear (Figure 4.5B).

4.4.3 APPLICATION TO *L. SAXATILIS* DATASET: EFFECT OF MERGING SUBSETS OF LOCI AND RECOMBINATION

For simplicity, we discuss results after re-scaling relative parameters to absolute effective sizes and time of events in years, using k to indicate thousands (Table 4.4 but see Table A4.6 for the relative estimates). Re-scaling was performed after combining the posterior distributions from multiple subsets of loci, giving more weight to subsets of loci with mean summary statistics closer to the mean over the whole genome.

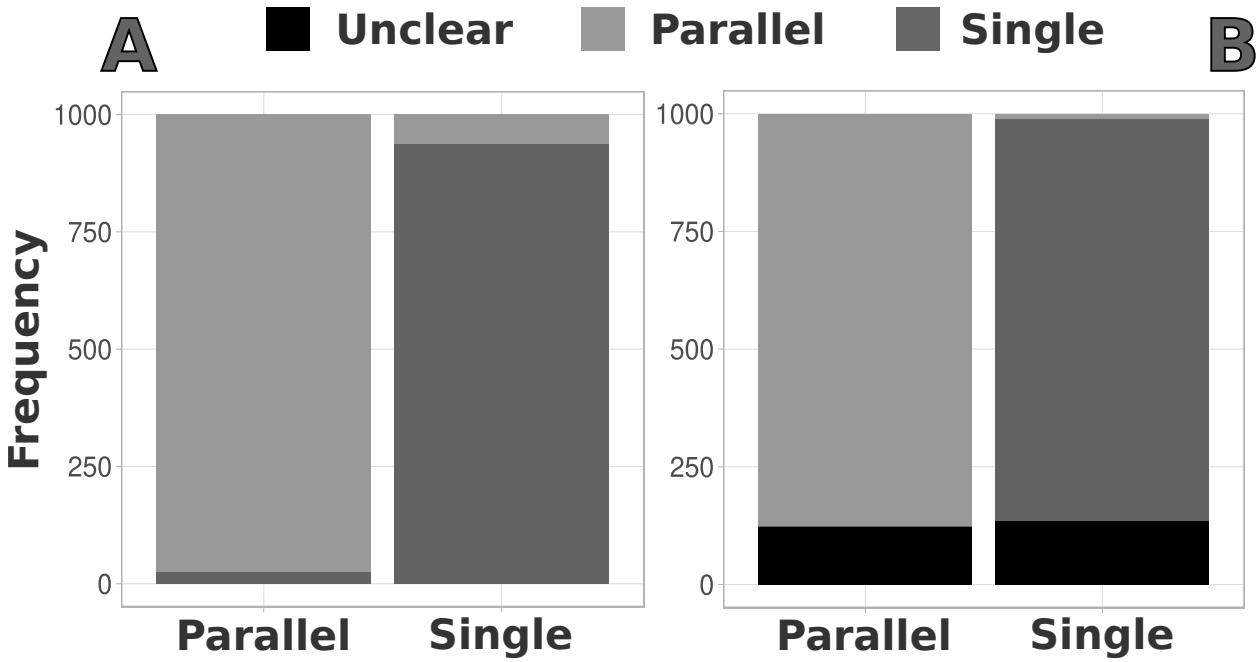


Figure 4.5: Model misclassification for the four-population models. Confusion matrix assuming that a simulation is assigned to a given model when the posterior probability is above 0.5 (A) or assuming that a simulation is only assigned to a model when the posterior probability is above 0.9 (B).

By comparing posteriors obtained by merging subsets with varying numbers of loci, we found that using subsets of loci led to posteriors similar to those obtained with the whole genome (i.e. 100 simulated loci), but with a wider variance, i.e., higher uncertainty. Yet, even with subsets representing only 10% of the genome, posteriors were similar to those obtained using all loci, becoming closer as the number of loci in subsets increases (Figure 4.6, Table A4.7). Additionally, for all parameters, the bias obtained when merging posteriors is similar to, or lower than the bias obtained using the summary statistics from all SNPs to estimate parameters simulating just a subset of loci (as proposed by Boitard et al. (2016), Table A4.7).

Estimates based on the two-population model with Crab and Wave populations from Arsklovet indicate (Figure 4.7 and Figure A4.7): (i) a slightly larger effective size for Crab (mean $\sim$18k, 95% CI: 12k - 33k) than Wave (mean $\sim$15k, 95% CI: 10K - 28k) which, despite the large overlap of the CIs, is in line with previous studies using individual genotypes (a combination of mtDNA, amplified fragment length polymorphism markers and three nuclear genes) (Butlin et al., 2014); (ii) a split between Crab and Wave ecotype populations $\sim$18k years ago, but with a wide credible interval (95% CI: 2.2k - 111k); and that (iii) divergence was accompanied by gene flow, with higher immigration from the Crab into Wave ecotype, which is in agreement with reported cline shifts in these populations (Westram et al., 2021). Analysis of pseudo-observed datasets simulated under this scenario suggests that estimates are unlikely to be significantly biased by assuming no

within-locus recombination ($r_w = 0$) since we obtained identical posterior distributions for pseudo-observed datasets simulated without ($r_w = 0$) or with a within-locus recombination rate equal to the mutation rate ($r_w = \mu$, Figure A4.8).

Table 4.4: **Absolute parameter estimates for *Littorina saxatilis* populations.** Results are shown for the Arsklovet population for the two-population model and for Arsklovet and Ramsö for the single origin model. For this model N_1 and N_2 correspond, respectively, to the absolute size of the Arsklovet Crab and Wave populations, while N_3 and N_4 correspond to the absolute size of the Ramsö Crab and Wave populations, respectively. For each parameter, the value outside brackets corresponds to the re-scaled mean of the posterior distribution and in-between brackets is the 95% credible interval. T_{div}, T_s and Δ_s are presented in years. Parameters indicated here are the same as in table 4.3, except for P_{no}, which is converted to the percentage of the genome where no migration occurs between ecotypes.

parameter	two-population	single origin
N_1	18489 (12106 - 32956)	10336 (4617 - 34148)
N_2	15793 (10167 - 27613)	5486 (2936 - 18424)
N_3	–	12648 (5488 - 35603)
N_4	–	15309 (6245 - 41201)
NA_1	–	40854 (8516 - 53242)
NA_2	–	21118 (3866 - 47367)
T_{div}	18211 (2210 - 111264)	–
T_s	–	521 (316 - 818)
Δ_s	–	14308 (4790 - 42954)
$4N_2m_{12}$	22.8 (5.9 - 60.8)	30.6 (10.3 - 105.1)
$4N_1m_{21}$	16.3 (2.3 - 52.6)	32.1 (10.1 - 108.0)
$4N_4m_{34}$	–	34.3 (11.0 - 117.2)
$4N_3m_{43}$	–	19.9 (6.3 - 71.4)
P_{no}	1.2 (0.1 - 6.6)	1.3 (0.2 - 5.4)

Our analysis of Crab and Wave ecotypes from two locations in Sweden (Arsklovet and Ramsö) supports the single origin model with strong posterior probabilities of 0.967 using the rejection algorithm and 1.000 using logistic regression. Our parameter estimates under the single origin model (Table 4.4 and Figure A4.9) suggest that the two ecotypes diverged approximately 15,000 years ago (95% CI: 5000 to 43000 years), followed by a recent colonization of both locations by populations from both ecotypes about $\sim$500 years ago (95% CI: 300 to 800 years). Under the single origin model, we estimated high and similar immigration rates between ecotypes in Arsklovet and lower migration from Wave into Crab in Ramsö (Figure 4.7H,I and Table A4.6). The point estimates supported larger ancestral effective sizes for the Crab population (mean 40k, 95% CI: 9K - 53k) than the Wave population (mean 21k, 95% CI: 4K - 48k), but the posteriors were wide and overlapping, indicating high uncertainty (Figure 4.7C). Nevertheless, the joint posteriors of present-day and ancestral populations indicate a population decline for the Crab ecotype in both

locations, and for the Wave ecotype at Arsklovet. Finally, we inferred a proportion of loci without migration P_{no} close to zero, with a mean of approximately 1% and an upper CI close to 6% (Table A4.6).

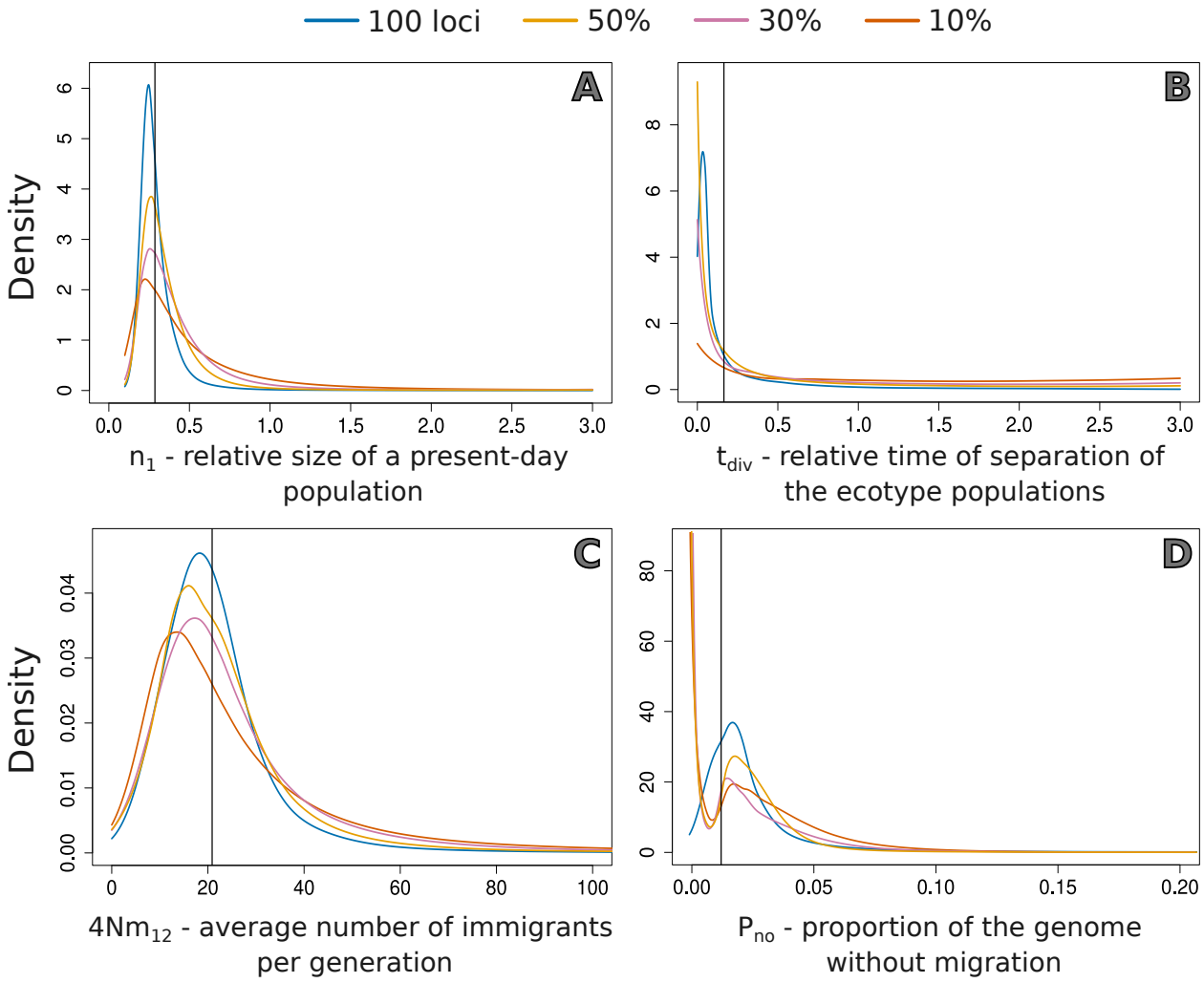


Figure 4.6: Impact of merging posteriors. We generated a pseudo-observed dataset of 100 loci and inferred parameters using the full dataset or subsets representing 10%, 30%, or 50% of the genome. The x-axis shows the estimated parameter value, and the y-axis shows the density of the posterior distribution obtained with the full dataset and the weighted combination of posteriors from the subsets. The solid vertical line represents the true parameter value. Parameters shown are: A - relative size of a present-day population (n_1), B - relative time of separation of the ecotype populations (t_{div}), C - average number of immigrants per generation ($4Nm_{12}$) and D - proportion of the genome without migration (P_{no}).

4.5 DISCUSSION

We developed a model-based method to analyse pooled-sequencing data, explicitly modeling various sources of error (e.g., variation in depth of coverage, unequal individual contribution, merging

multiple pools) by extending the framework of Gautier et al. (2013) into an ABC inference framework. We implemented this into a freely available R package, allowing users to perform model choice and parameter inference of demographic history based on Pool-seq data from natural populations. Our approach is based on simulating subsets of loci, estimating relative parameters and using relative summary statistics. These included summary statistics that are widely used in ABC, such as the mean and standard deviation of expected heterozygosity per population and between all pairs of populations (Jay et al., 2019), relative genetic differentiation between population pairs (F_{ST}), and others that capture parts of the joint site frequency spectrum (Wakeley & Hey, 1997), such as the proportion of SNPs with fixed difference between populations (Fraïsse et al., 2021).

To increase computational efficiency we fixed the ancestral effective population size (N_{ref}) and inferred relative demographic parameters, which were converted to absolute values based on an average mutation rate and number of observed SNPs. This circumvented the simulation of combinations of parameters leading to similar diversity and differentiation values, e.g., identical $\theta = 4N_e\mu$ and hence identical summary statistics due to low N_e with high μ or high N_e with low μ. Moreover, by combining multiple posterior distributions, obtained from different subsets of independent loci, and weighting them according to the distance to the genome-wide mean summary statistics, we minimized the impact of non-neutral processes (e.g., background selection) in the inference of demographic history.

Our simulation study shows that, for the datasets analysed here, the means of the posterior distributions provide accurate point estimates for most demographic history parameters of the two- and four-population models. In fact, the prediction errors for most parameters were similar for both models (Table 4.3), with the exception of migration rates, for which we found higher prediction errors for the parallel origin model (Table 4.3). This can be explained by the recent divergence of ecotypes with gene flow in each location, implying that it is harder to disentangle gene flow from incomplete lineage sorting under the parallel origin model. Importantly, our prediction errors based on Pool-seq were within the range of those of recent ABC methods based on individual genotypes (Fraïsse et al., 2021). Although the aim was to infer demographic history accounting for the effects of barrier loci, results indicate that the proportion of loci without migration (P_{no}) was well estimated in the two- and four-population models, suggesting it is possible to estimate the number of barrier loci under selection.

Additionally, and despite concerns about model choice and estimation of Bayes factors with ABC (Marin, Pillai, Robert, & Rousseau, 2014; Robert, Cornuet, Marin, & Pillai, 2011), our model choice results indicate that Pool-seq provides enough information to distinguish between scenarios of ecotype formation with high posterior probabilities (proportion of correctly assigned simulations

with 90% posterior probability above 0.85 for both models, Figure 4.5). This is explained by the fact that the single and parallel origin models considered have different mean values for several summary statistics (Figure A4.10), which is required to distinguish models in an ABC framework (Marin et al., 2014), and was expected given that gene flow occurs between populations with different shared ancestries in the alternative models (Figure 4.1).

Importantly, our R package includes functions to compute prediction errors, allowing users to perform simulation studies based on their specific set of models, prior distributions, sample sizes, depths of coverage and numbers of pools. Thus, users can evaluate the accuracy of ABC results for their specific datasets and models. Also, the R package includes functions to assess the fit of the models to the data, visually plotting the fit of simulations to the observed summary statistics. Below we discuss the application to *L. saxatilis* ecotypes, as well as limitations and future perspectives.

4.5.1 RECENT SINGLE ORIGIN OF *LITTORINA SAXATILIS* ECOTYPES IN SWEDEN

To illustrate the application of our method to Pool-Seq data, we analysed data from pools of *L. saxatilis* ecotypes, exploring the effects of obtaining posteriors by merging subsets of loci and assumptions about within-locus recombination. Using subsets of 300 loci, we found evidence supporting a single origin of Crab and Wave ecotypes in Sweden. Our results indicate that the ecotypes diverged relatively recently, followed by a split of the populations in different locations about 1,000 generations ago (approximately 500 years ago), with high gene flow between ecotypes. This is consistent with a recent postglacial colonization of Swedish islands (Panova et al., 2011). The estimates from both the two- and four-population models were consistent, with the divergence time for Crab and Wave ecotypes being approximately 15,000 years ago.

Both models also indicate high migration rates between ecotypes ($4Nm > 10$), with slightly higher rates from Crab to Wave ecotypes (Figure 4.7G-I). This supports the hypothesis of a higher net dispersal from Crab to Wave, which may explain the observed shift in cline centres towards the Wave habitat on Swedish islands (Westram et al., 2021).

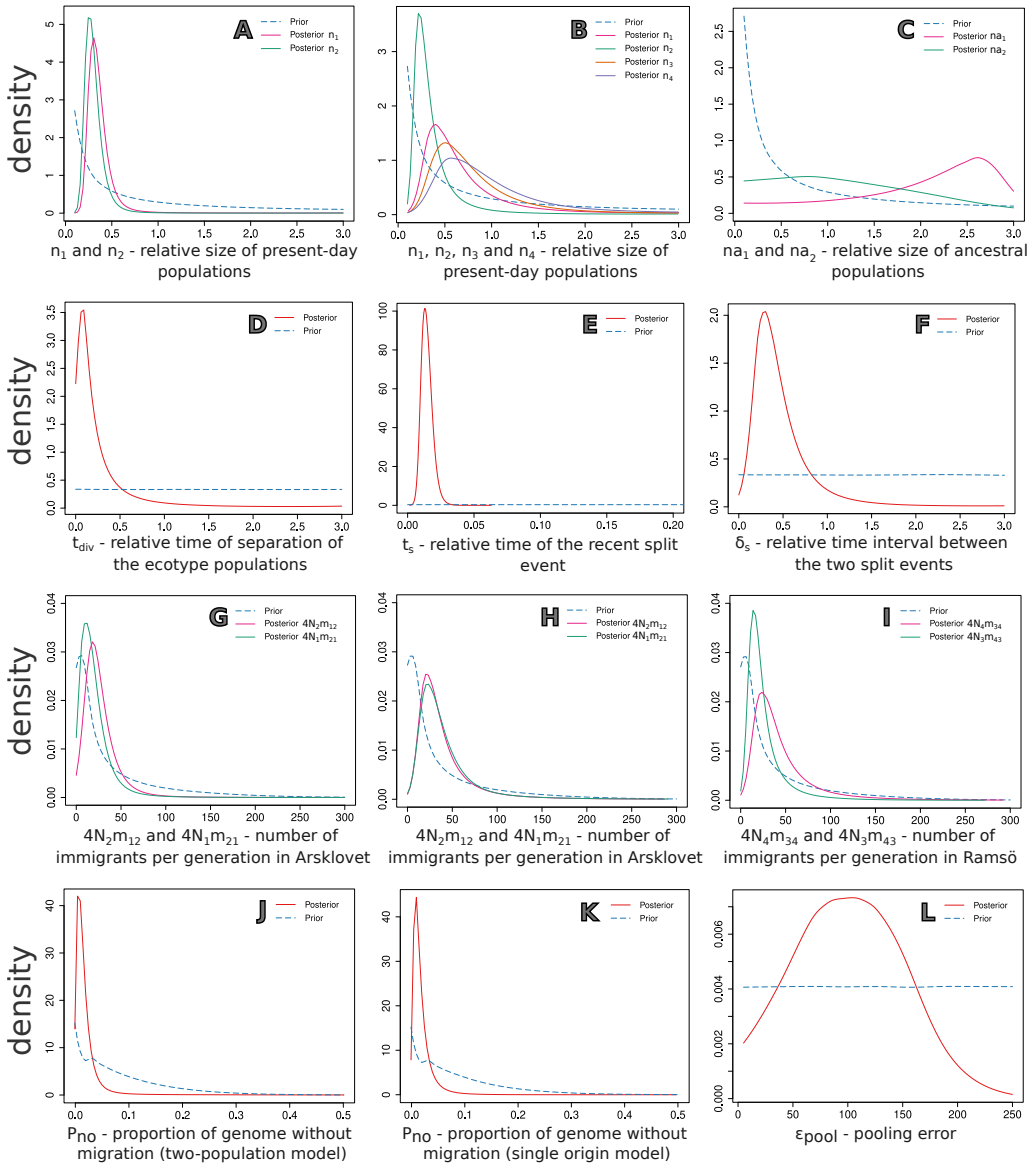


Figure 4.7: Posterior distributions of relative *L. saxatilis* parameters with regression adjustment and a tolerance of 0.01. Prior distributions are displayed as a dotted blue line for reference. The first column (A, D, G, and J) corresponds to the two-population model, while the remaining columns represent the single origin model. A - relative size of Arsklovet Crab (n_1) and Wave populations (n_2), B - relative size of Arsklovet Crab (n_1), Arsklovet Wave (n_2), Ramsö Crab (n_3) and Ramsö Wave (n_4) populations, C - relative size of ancestral populations (na_1 and na_2), D - relative time of separation of the ecotype populations (t_{div}), E - relative time of the recent split event (t_s), F - relative time interval between the two split events (δ_s), G and H - average number of immigrants per generation ($4N_2m_{12}$ and $4N_1m_{21}$) in Arsklovet, I - average number of immigrants per generation ($4N_4m_{34}$ and $4N_3m_{43}$) in Ramsö, J and K - proportion of the genome without migration (P_{no}) and L - pooling error. Relative parameter values were converted to absolute values using a re-scaling factor f, calculated as the ratio of the observed number of SNPs ($obs[S]$) to the expected number of SNPs ($E[S]$). The absolute parameter values were obtained by multiplying the point estimate of the posteriors shown here by the rescaling factor f.

We found slightly larger effective sizes for Crab than Wave ecotype populations, together with lower effective sizes for present-day than ancestral populations, in agreement with a previously

reported lack of support for past expansions based on individual genotypes (Butlin et al., 2014). Despite the high uncertainty in the posteriors for ancestral population sizes, our estimates suggest a higher density of individuals in Crab than Wave habitats, which is also consistent with the reported shifts in cline centres (Westram et al., 2021). Finally, we found that a low proportion of the genome was linked to complete barriers to gene flow between the two ecotypes ($P_{no} < 6\%$). This low proportion of barrier loci was not surprising since we excluded SNPs from all known regions associated with chromosomal inversions in *L. saxatilis*, which play an important role in the non-neutral ecotype divergence process (Westram et al., 2021). Thus, a possible explanation for our estimates is that barrier loci also occur outside inversions. However, given the lack of a chromosome level reference genome with a clear mapping of collinear and inverted regions, we cannot exclude that some of the SNPs included in our analysis are actually linked with chromosomal inversions.

The inferred high gene flow ($4Nm > 10$) between Wave and Crab populations may limit our ability to distinguish between alternative models (Bierne, Gagnaire, & David, 2013), but our results and ABC model choice based on individual genotypes (Butlin et al., 2014) both support a single origin for *L. saxatilis* ecotypes in Sweden. Indeed, simulations under the single origin model fit the observed summary statistics (Figure A4.11), but caution is needed due to the simplified nature of our models. Due to the limited spatial scale of our study, our results may reflect recent postglacial colonization of the two locations, rather than ecotype formation. Indeed, it is probable that ecotype formation in these Swedish locations predates their colonization. To determine if ecotype formation occurred in parallel, the ABC approach developed here could be applied to compare Wave and Crab ecotypes from more distant locations.

4.5.2 LIMITATIONS AND FUTURE PERSPECTIVES

Our aim was to implement an ABC method using Pool-seq data and test its performance under generic divergence models. These models are relatively simple, and probably fail to capture the complexity of ecotype formation in these geographically restricted *L. saxatilis* Crab and Wave ecotypes. For instance, we assumed a simultaneous divergence of the four extant populations, and no migration between ancestral populations, which is unlikely to hold. More complex models, implying different strengths of selection at barrier loci or the possibility of one ecotype acting as a reservoir of standing genetic variation (Jones et al., 2012; Liu, Ferchaud, Grønkjær, Nygaard, & Hansen, 2018) could also be considered. It remains to be tested whether an ABC framework allows distinguishing between more complex models with Pool-seq data. Nonetheless, a recent study has highlighted the potential of Pool-seq data to infer demographic histories by combining ABC with supervised machine learning in the DIYABC-RF software (Collin et al., 2021). Similarly to

our approach, DIYABC-RF enables the simulation and analysis of Pool-seq data by first simulating individual SNP genotypes and then using the corresponding allele frequencies to generate pool read counts from a binomial distribution. However, DIYABC-RF does not explicitly model all possible sources of Pool-seq errors, as it only models variation in read coverage across SNPs (by randomly drawing coverages from the vectors of SNP coverages in the observed data set). Here, we explicitly model the different sources of errors with specific error parameters, such as variation in depth of coverage, unequal individual and pool contributions, and sequencing errors. Our results show that ignoring Pool-Seq errors might lead to incorrect estimates, but that demographic parameters are estimated accurately by explicitly modeling Pool-Seq errors (Figure 4.4). The low prediction errors found in our simulation study in models with up to four populations indicate that Pool-seq data might be suitable to infer demographic history under more complex models.

Our modular approach allows users to integrate our R package seamlessly with other packages at different steps. First, here we used the coalescent simulator implemented in the R package *scrm*, but it is possible to consider other demographic scenarios and simulate genetic data with coalescent-based methods for recombining chromosomes (Kelleher, Etheridge, & McVean, 2016), or forward simulators that explicitly model positive and background selection (Haller & Messer, 2019) and then use our functions to simulate Pool-seq data. Second, after simulating Pool-seq data, users can feed the reference tables with parameters and summary statistics to other tools using more sophisticate algorithms, such as neural networks or random forest ABC. Third, after the ABC rejection step, users can perform post-processing adjustment using other tools (e.g., abc R package, Csilléry et al. 2012). Despite some limitations, our results show that combining Pool-seq with ABC is an effective approach for investigating parallel evolution in taxa where similar ecotypes are found at multiple locations. We illustrated this by applying our method to Swedish populations of *L. saxatilis* ecotypes. The demographic history models considered provide suitable null models for a better comprehension of the genetic basis of divergent adaptation across many taxa.

4.6 ACKNOWLEDGEMENTS

We thank the editor and three anonymous reviewers for their comments and suggestions. This work was funded by the strategic project UIDB/00329/2020 granted to cE3c by Fundação para a Ciência e a Tecnologia (FCT). JC was supported by an FCT Ph.D. scholarship (PD/BD/128350/2017). RF is funded by a FCT CEEC (Fundação para a Ciência e a Tecnologia, Concurso Estímulo ao Emprego Científico) contract (2020.00275.CEECIND) and by a FCT research project (PTDC/BIA-EVL/1614/2021). RKB was funded by the European Research Council (ERC-2015-AdG-693030-

BARRIERS). VCS was supported by the Human Frontier Science Program (RGY0081/2020) and by FCT (CEECINST/00032/2018/CP1523/CT0008). We thank the National Network for Advanced Computing (RNCA) and INCD (https://incd.pt/) for access and use of their computing infrastructure, funded by FCT to VCS (2021.09795.CPCA).

4.7 REFERENCES

Anderson, E. C., Skaug, H. J., & Barshis, D. J. (2014). Next-generation sequencing for molecular ecology: a caveat regarding pooled samples. *Molecular Ecology*, *23*(3), 502-512. doi: 10.1111/mec.12609

Andrew, R. L., Kane, N. C., Baute, G. J., Grassa, C. J., & Rieseberg, L. H. (2013). Recent nonhybrid origin of sunflower ecotypes in a novel habitat. *Molecular Ecology*, *22*(3), 799-813. doi: 10.1111/mec.12038

Baird, N. A., Etter, P. D., Atwood, T. S., Currey, M. C., Shiver, A. L., Lewis, Z. A., ... Johnson, E. A. (2008). Rapid snp discovery and genetic mapping using sequenced rad markers. *PLOS ONE*, *3*(10), 1-7. doi: 10.1371/journal.pone.0003376

Bakovic, V., Martin Cerezo, M. L., Höglund, A., Fogelholm, J., Henriksen, R., Hargeby, A., & Wright, D. (2021). The genomics of phenotypically differentiated *Asellus aquaticus* cave, surface stream and lake ecotypes. *Molecular Ecology*, *30*(14), 3530-3547. doi: 10.1111/mec.15987

Beaumont, M. A., Nielsen, R., Robert, C., Hey, J., Gaggiotti, O., Knowles, L., ... Excoffier, L. (2010). In defence of model-based inference in phylogeography. *Molecular Ecology*, *19*(3), 436-446. doi: 10.1111/j.1365-294X.2009.04515.x

Beaumont, M. A., Zhang, W., & Balding, D. J. (2002). Approximate bayesian computation in population genetics. *Genetics*, *162*(4), 2025-2035. doi: 10.1111/j.1937-2817.2010.tb01236.x

Begun, D. J., Holloway, A. K., Stevens, K., Hillier, L. W., Poh, Y.-P., Hahn, M. W., ... others (2007). Population genomics: whole-genome analysis of polymorphism and divergence in *Drosophila simulans*. *PLoS biology*, *5*(11), e310. doi: 10.1371/journal.pbio.0050310

Bhatia, G., Patterson, N., Sankararaman, S., & Price, A. L. (2013). Estimating and interpreting fst: the impact of rare variants. *Genome research*, *23*(9), 1514-1521. doi: 10.1101/gr.154831.113

Bierne, N., Gagnaire, P.-A., & David, P. (2013). The geography of introgression in a patchy environment and the thorn in the side of ecological speciation. *Current Zoology*, *59*(1), 72-86. doi: 10.1093/czoolo/59.1.72

Boitard, S., Rodríguez, W., Jay, F., Mona, S., & Austerlitz, F. (2016). Inferring population size history from large samples of genome-wide molecular data - an approximate bayesian computation approach. *PLOS Genetics*, *12*(3), 1-36. doi: 10.1371/journal.pgen.1005877

Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: a flexible trimmer for illumina sequence data. *Bioinformatics*, *30*(15), 2114-2120. doi: 10.1093/bioinformatics/btu170

Butlin, R. K., Debelle, A., Kerth, C., Snook, R. R., Beukeboom, L. W., Cajas, R. C., ... Schilthuizen, M. (2012). What do we need to know about speciation? *Trends in Ecology & Evolution*, *27*(1), 27-39. doi: 10.1016/j.tree.2011.09.002

Butlin, R. K., Saura, M., Charrier, G., Jackson, B., André, C., Caballero, A., ... Rolán-Alvarez, E. (2014). Parallel evolution of local adaptation and reproductive isolation in the face of gene flow. *Evolution*, *68*(4), 935-949. doi: 10.1111/evo.12329

Calvo, S. E., Tucker, E. J., Compton, A. G., Kirby, D. M., Crawford, G., Burtt, N. P., ... Mootha, V. K. (2010). High-throughput, pooled sequencing identifies mutations in nubpl and foxred1 in human complex i deficiency. *Nature Genetics*, *42*(10), 851-858. doi: 10.1038/ng.659

Chen, C., Parejo, M., Momeni, J., Langa, J., Nielsen, R. O., Shi, W., ... others (2022). Population structure and diversity in european honey bees (*Apis mellifera L.*)—an empirical comparison of pool and individual whole-genome sequencing. *Genes*, *13*(2), 182. doi: 10.3390/genes13020182

Collin, F.-d., Durif, G., Raynal, L., Lombaert, E., Gautier, M., Vitalis, R., ... Estoup, A. (2021). Extending approximate bayesian computation with supervised machine learning to infer demographic history from genetic polymorphisms using diyabc random forest. *Molecular Ecology Resources*, *21*(8), 2598-2613. doi: 10.1111/1755-0998.13413

Cooke, N. P., & Nakagome, S. (2018). Fine-tuning of approximate bayesian computation for human population genomics. *Current Opinion in Genetics and Development*, *53*, 60-69. doi: 10.1016/j.gde.2018.06.016

Cornuet, J. M., Pudlo, P., Veyssier, J., Dehne-Garcia, A., Gautier, M., Leblois, R., ... Estoup, A. (2014). Diyabc v2.0: A software to make approximate bayesian computation inferences about population history using single nucleotide polymorphism, dna sequence and microsatellite data. *Bioinformatics*, *30*(8), 1187-1189. doi: 10.1093/bioinformatics/btt763

Csilléry, K., François, O., & Blum, M. G. (2012). Abc: An r package for approximate bayesian computation (abc). *Methods in Ecology and Evolution*, *3*(3), 475-479. doi: 10.1111/j.2041-210X.2011.00179.x

Cutler, D. J., & Jensen, J. D. (2010). To pool, or not to pool? *Genetics*, *186*(1), 41-43. doi: 10.1534/genetics.110.121012

Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., ... Li, H. (2021). Twelve years of samtools and bcftools. *Gigascience*, *10*(2), giab008. doi: 10.1093/gigascience/giab008

Dorant, Y., Benestan, L., Rougemont, Q., Normandeau, E., Boyle, B., Rochette, R., & Bernatchez, L. (2019). Comparing pool-seq, rapture, and gbs genotyping for inferring weak population structure: The american lobster (*Homarus americanus*) as a case study. *Ecology and evolution*, *9*(11), 6606-6623. doi: 10.1002/ece3.5240

Ellegren, H. (2014). Genome sequencing and population genomics in non-model organisms. *Trends*

in Ecology & Evolution, *29*(1), 51-63. doi: 10.1016/j.tree.2013.09.008

Excoffier, L., Marchi, N., Marques, D. A., Matthey-Doret, R., Gouy, A., & Sousa, V. C. (2021). fastsimcoal2: demographic inference under complex evolutionary scenarios. *Bioinformatics*, *37*(24), 4882-4885. doi: 10.1093/bioinformatics/btab468

Fang, B., Kemppainen, P., Momigliano, P., Feng, X., & Merilä, J. (2020). On the causes of geographically heterogeneous parallel evolution in sticklebacks. *Nature ecology & evolution*, *4*(8), 1105-1115. doi: 10.1038/s41559-020-1222-6

Faria, R., Chaube, P., Morales, H. E., Larsson, T., Lemmon, A. R., Lemmon, E. M., ... Butlin, R. K. (2019). Multiple chromosomal rearrangements in a hybrid zone between *Littorina saxatilis* ecotypes. *Molecular Ecology*, *28*(6), 1375-1393. doi: 10.1111/mec.14972

Faria, R., Renaut, S., Galindo, J., Pinho, C., Melo-Ferreira, J., Melo, M., ... Butlin, R. K. (2014). Advances in ecological speciation: an integrative approach. *Molecular Ecology*, *23*(3), 513-521. doi: 10.1111/mec.12616

Ferretti, L., Ramos-Onsins, S. E., & Pérez-Enciso, M. (2013). Population genomics from pool sequencing. *Molecular Ecology*, *22*(22), 5561-5576. doi: 10.1111/mec.12522

Fraïsse, C., Popovic, I., Mazoyer, C., Spataro, B., Delmotte, S., Romiguier, J., ... Roux, C. (2021). Dils: Demographic inferences with linked selection by using abc. *Molecular Ecology Resources*, *21*(8), 2629-2644. doi: 10.1111/1755-0998.13323

Futschik, A., & Schlötterer, C. (2010). The next generation of molecular markers from massively parallel sequencing of pooled dna samples. *Genetics*, *186*(1), 207-218. doi: 10.1534/genetics.110.114397

Gautier, M., Foucaud, J., Gharbi, K., Cézard, T., Galan, M., Loiseau, A., ... Estoup, A. (2013). Estimation of population allele frequencies from next-generation sequencing data: pool-versus individual-based genotyping. *Molecular Ecology*, *22*(14), 3766-3779. doi: 10.1111/mec.12360

Haller, B. C., & Messer, P. W. (2019). Slim 3: forward genetic simulations beyond the wright-fisher model. *Molecular biology and evolution*, *36*(3), 632-637. doi: 10.1093/molbev/msy228

Hickerson, M. J. (2014). All models are wrong. *Molecular Ecology*, *23*(12), 2887-2889. doi: 10.1111/mec.12794

Huang, W., Takebayashi, N., Qi, Y., & Hickerson, M. J. (2011). Mtml-msbayes: Approximate bayesian comparative phylogeographic inference from multiple taxa and multiple loci with rate heterogeneity. *BMC Bioinformatics*, *12*(1), 1-14. doi: 10.1186/1471-2105-12-1

Hudson, R. R. (1990). Gene genealogies and the coalescent process. *Oxford surveys in evolutionary biology*, *7*(1), 1-44.

Hume, J. B., Recknagel, H., Bean, C. W., Adams, C. E., & Mable, B. K. (2018). Radseq and mate choice assays reveal unidirectional gene flow among three lamprey ecotypes despite weak assortative mating: insights into the formation and stability of multiple ecotypes in sympatry. *Molecular ecology*, *27*(22), 4572-4590. doi: 10.1111/mec.14881

Jay, F., Boitard, S., & Austerlitz, F. (2019). An abc method for whole-genome sequence data: inferring paleolithic and neolithic human expansions. *Molecular biology and evolution*, *36*(7), 1565-1579. doi: 10.1093/molbev/msz038

Johannesson, K., Panova, M., Kemppainen, P., André, C., Rolán-Alvarez, E., & Butlin, R. K. (2010). Repeated evolution of reproductive isolation in a marine snail: unveiling mechanisms of speciation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *365*(1547), 1735-1747. doi: 10.1098/rstb.2009.0256

Jones, F. C., Grabherr, M. G., Chan, Y. F., Russell, P., Mauceli, E., Johnson, J., ... Team, B. I. G. S. P. . W. G. A. (2012). The genomic basis of adaptive evolution in threespine sticklebacks. *Nature*, *484*(7392), 55-61. doi: 10.1038/nature10944

Kelleher, J., Etheridge, A. M., & McVean, G. (2016). Efficient coalescent simulation and genealogical analysis for large sample sizes. *PLoS computational biology*, *12*(5), e1004842. doi: 10.1371/journal.pcbi.1004842

Klütsch, C. F. C., Manseau, M., Trim, V., Polfus, J., & Wilson, P. J. (2016). The eastern migratory caribou: the role of genetic introgression in ecotype evolution. *Royal Society Open Science*, *3*(2), 150469. doi: 10.1098/rsos.150469

Koch, E. L., Morales, H. E., Larsson, J., Westram, A. M., Faria, R., Lemmon, A. R., ... Butlin, R. K. (2021). Genetic variation for adaptive traits is associated with polymorphic inversions in *Littorina saxatilis*. *Evolution Letters*, *5*(3), 196-213. doi: 10.1002/evl3.227

Kofler, R., Pandey, R. V., & Schlötterer, C. (2011). Popoolation2: identifying differentiation between populations using sequencing of pooled dna samples (pool-seq). *Bioinformatics*, *27*(24), 3435-3436. doi: 10.1093/bioinformatics/btr589

Le Moan, A., Gagnaire, P.-A., & Bonhomme, F. (2016). Parallel genetic divergence among coastal-marine ecotype pairs of european anchovy explained by differential introgression after secondary contact. *Molecular Ecology*, *25*(13), 3187-3202. doi: 10.1111/mec.13627

Li, B., Chen, W., Zhan, X., Busonero, F., Sanna, S., Sidore, C., ... Abecasis, G. R. (2012). A likelihood-based framework for variant calling and de novo mutation detection in families. *PLOS Genetics*, *8*(10), 1-12. doi: 10.1371/journal.pgen.1002944

Lieberman, T. D., Flett, K. B., Yelin, I., Martin, T. R., McAdam, A. J., Priebe, G. P., & Kishony, R. (2014). Genetic variation of a bacterial pathogen within individuals with cystic fibrosis provides a record of selective pressures. *Nature Genetics*, *46*(1), 82-87. doi: 10.1038/ng.2848

Liepe, J., Kirk, P., Filippi, S., Toni, T., Barnes, C. P., & Stumpf, M. P. (2014). A framework for parameter estimation and model selection from experimental data in systems biology using approximate bayesian computation. *Nature Protocols*, *9*(2), 439-456. doi: 10.1038/nprot.2014.025

Liu, S., Ferchaud, A.-L., Grønkjær, P., Nygaard, R., & Hansen, M. M. (2018). Genomic parallelism and lack thereof in contrasting systems of three-spined sticklebacks. *Molecular ecology*, *27*(23), 4725-4743. doi: 10.1111/mec.14782

Louis, M., Fontaine, M. C., Spitz, J., Schlund, E., Dabin, W., Deaville, R., ... Simon-Bouhet,

B. (2014). Ecological opportunities and specializations shaped genetic divergence in a highly mobile marine top predator. *Proceedings of the Royal Society B: Biological Sciences*, *281*(1795), 20141558. doi: 10.1098/rspb.2014.1558

Malaspinas, A.-S., Westaway, M. C., Muller, C., Sousa, V. C., Lao, O., Alves, I., ... others (2016). A genomic history of aboriginal australia. *Nature*, *538*(7624), 207-214. doi: 10.1038/nature18299

Malinsky, M., Matschiner, M., & Svardal, H. (2021). Dsuite-fast d-statistics and related admixture evidence from vcf files. *Molecular ecology resources*, *21*(2), 584-595. doi: 10.1111/1755-0998.13265

Marin, J.-M., Pillai, N. S., Robert, C. P., & Rousseau, J. (2014). Relevant statistics for bayesian model choice. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *76*(5), 833-859. doi: 10.1111/rssb.12056

Morales, H. E., Faria, R., Johannesson, K., Larsson, T., Panova, M., Westram, A. M., & Butlin, R. K. (2018). *Littorina saxatilis genome sequencing and population re-sequencing*. Retrieved from `https://www.ncbi.nlm.nih.gov/bioproject/PRJNA494650`

Morales, H. E., Faria, R., Johannesson, K., Larsson, T., Panova, M., Westram, A. M., & Butlin, R. K. (2019). Genomic architecture of parallel ecological divergence: beyond a single environmental contrast. *Science advances*, *5*(12), eaav9963. doi: 10.1126/sciadv.aav9963

Nei, M., & Roychoudhury, A. K. (1974). Sampling variances of heterozygosity and genetic distance. *Genetics*, *76*(2), 379-390. doi: 10.1093/genetics/76.2.379

Panova, M., Blakeslee, A. M., Miller, A. W., Mäkinen, T., Ruiz, G. M., Johannesson, K., & André, C. (2011). Glacial history of the north atlantic marine snail, *Littorina saxatilis*, inferred from distribution of mitochondrial dna lineages. *PLoS One*, *6*(3), e17511. doi: 10.1371/journal.pone.0017511

Panova, M., Hollander, J., & Johannesson, K. (2006). Site-specific genetic divergence in parallel hybrid zones suggests nonallopatric evolution of reproductive barriers. *Molecular Ecology*, *15*(13), 4021-4031. doi: 10.1111/j.1365-294X.2006.03067.x

Parts, L., Cubillos, F. A., Warringer, J., Jain, K., Salinas, F., Bumpstead, S. J., ... Liti, G. (2011). Revealing the genetic structure of a trait by sequencing a population under selection. *Genome Research*, *21*(7), 1131-1138. doi: 10.1101/gr.116731.110

Pontarp, M., Brännström, Å., & Petchey, O. L. (2019). Inferring community assembly processes from macroscopic patterns using dynamic eco-evolutionary models and approximate bayesian computation (abc). *Methods in Ecology and Evolution*, *10*(4), 450-460. doi: 10.1111/2041-210X.13129

Prescott, N. J., Lehne, B., Stone, K., Lee, J. C., Taylor, K., Knight, J., ... Consortium, U. I. G. (2015). Pooled sequencing of 531 genes in inflammatory bowel disease identifies an associated rare variant in btnl2 and implicates other immune related genes. *PLOS Genetics*, *11*(2), 1-19. doi: 10.1371/journal.pgen.1004955

Quinlan, A. R., & Hall, I. M. (2010). Bedtools: a flexible suite of utilities for comparing genomic

features. *Bioinformatics*, *26*(6), 841-842. doi: 10.1093/bioinformatics/btq033

Ravinet, M., Westram, A., Johannesson, K., Butlin, R., André, C., & Panova, M. (2016). Shared and nonshared genomic divergence in parallel ecotypes of *Littorina saxatilis* at a local scale. *Molecular ecology*, *25*(1), 287-305. doi: 10.1111/mec.13332

Reid, D. G. (1996). *Systematics and evolution of Littorina*. London: Ray Society.

Riesch, R., Muschick, M., Lindtke, D., Villoutreix, R., Comeault, A. A., Farkas, T. E., ... others (2017). Transitions between phases of genomic differentiation during stick-insect speciation. *Nature ecology & evolution*, *1*(4), 1-13. doi: 10.1038/s41559-017-0082

Rivas, M. J., Saura, M., Pérez-Figueroa, A., Panova, M., Johansson, T., André, C., ... Quesada, H. (2018). Population genomics of parallel evolution in gene expression and gene sequence during ecological adaptation. *Scientific reports*, *8*(1), 1-12. doi: 10.1038/s41598-018-33897-8

Robert, C. P., Cornuet, J.-M., Marin, J.-M., & Pillai, N. S. (2011). Lack of confidence in approximate bayesian computation model choice. *Proceedings of the National Academy of Sciences*, *108*(37), 15112-15117. doi: 10.1073/pnas.1102900108

Ross, P. A., Endersby-Harshman, N. M., & Hoffmann, A. A. (2019). Shedding light on the grey zone of speciation along a continuum of genomic divergence. *Evolutionary applications*, *12*(3), 572-586. doi: doi.org/10.1111/eva.12740

Rougemont, Q., & Bernatchez, L. (2018). The demographic history of atlantic salmon (*Salmo salar*) across its distribution range reconstructed from approximate bayesian computations. *Evolution*, *72*(6), 1261-1277. doi: 10.1111/evo.13486

Rubin, C.-J., Megens, H.-J., Barrio, A. M., Maqbool, K., Sayyab, S., Schwochow, D., ... Andersson, L. (2012). Strong signatures of selection in the domestic pig genome. *Proceedings of the National Academy of Sciences*, *109*(48), 19529-19536. doi: 10.1073/pnas.1217149109

Schlötterer, C., Tobler, R., Kofler, R., & Nolte, V. (2014). Sequencing pools of individuals — mining genome-wide polymorphism data without big funding. *Nature Reviews Genetics*, *15*(11), 749-763. doi: 10.1038/nrg3803

Schluter, D. (2000). *The ecology of adaptive radiation*. Oxford University Press.

Schrider, D. R., Shanku, A. G., & Kern, A. D. (2018). Supervised machine learning reveals introgressed loci in the genomes of *Drosophila simulans* and *D. sechellia*. *PLoS genetics*, *14*(4), e1007341. doi: 10.1371/journal.pgen.1007341

Sheehan, S., & Song, Y. S. (2016). Deep learning for population genetic inference. *PLoS computational biology*, *12*(3), e1004845. doi: 10.1371/journal.pcbi.1004845

Smith, C. C., & Flaxman, S. M. (2020). Leveraging whole genome sequencing data for demographic inference with approximate bayesian computation. *Molecular ecology resources*, *20*(1), 125-139. doi: 10.1111/1755-0998.13092

Staab, P. R., Zhu, S., Metzler, D., & Lunter, G. (2015). scrm: efficiently simulating long sequences using the approximated coalescent with recombination. *Bioinformatics*, *31*(10), 1680-1682.

doi: 10.1093/bioinformatics/btu861

Stoler, N., & Nekrutenko, A. (2021). Sequencing error profiles of illumina sequencing instruments. *NAR genomics and bioinformatics*, *3*(1), lqab019. doi: 10.1093/nargab/lqab019

Tavaré, S., Balding, D. J., Griffiths, R. C., & Donnelly, P. (1997). Inferring coalescence times from dna sequence data. *Genetics*, *145*(2), 505-518.

Turesson, G. (1922). The genotypical response of the plant species to the habitat. *Hereditas*, *3*(3), 211-350.

Turner, T. L., Stewart, A. D., Fields, A. T., Rice, W. R., & Tarone, A. M. (2011). Population-based resequencing of experimentally evolved populations reveals the genetic basis of body size variation in *Drosophila melanogaster*. *PLOS Genetics*, *7*(3), 1-10. doi: 10.1371/journal.pgen.1001336

Van Belleghem, S. M., Vangestel, C., De Wolf, K., De Corte, Z., Möst, M., Rastas, P., ... Hendrickx, F. (2018). Evolution at two time frames: polymorphisms from an ancient singular divergence event fuel contemporary parallel evolution. *PLoS genetics*, *14*(11), e1007796. doi: 10.1371/journal.pgen.1007796

Wakeley, J., & Hey, J. (1997). Estimating ancestral population parameters. *Genetics*, *145*(3), 847-855. doi: 10.1093/genetics/145.3.847

Wegmann, D., Leuenberger, C., Neuenschwander, S., & Excoffier, L. (2010). Abctoolbox: a versatile toolkit for approximate bayesian computations. *BMC Bioinformatics*, *11*(1), 1-7. doi: 10.1186/1471-2105-11-116

Westram, A. M., Faria, R., Johannesson, K., & Butlin, R. K. (2021). Using replicate hybrid zones to understand the genomic basis of adaptive divergence. *Molecular ecology*, *30*(15), 3797-3814. doi: 10.1111/mec.15861

Westram, A. M., Panova, M., Galindo, J., & Butlin, R. K. (2016). Targeted resequencing reveals geographical patterns of differentiation for loci implicated in parallel evolution. *Molecular ecology*, *25*(13), 3169-3186. doi: 10.1111/mec.13640

Westram, A. M., Rafajlović, M., Chaube, P., Faria, R., Larsson, T., Panova, M., ... Butlin, R. K. (2018). Clines on the seashore: The genomic architecture underlying rapid divergence in the face of gene flow. *Evolution letters*, *2*(4), 297-309. doi: 10.1002/evl3.74

Zhang, J., Dennis, T. E., Landers, T. J., Bell, E., & Perry, G. L. (2017). Linking individual-based and statistical inferential models in movement ecology: A case study with black petrels (*Procellaria parkinsoni*). *Ecological Modelling*, *360*, 425-436. doi: 10.1016/j.ecolmodel.2017.07.017

Zhou, D., Udpa, N., Gersten, M., Visk, D. W., Bashir, A., Xue, J., ... Haddad, G. G. (2011). Experimental selection of hypoxia-tolerant *Drosophila melanogaster*. *Proceedings of the National Academy of Sciences*, *108*(6), 2349-2354. doi: 10.1073/pnas.1010643108

4.8 APPENDIX

Table A4.1: **Set of summary statistics considered.** The D-statistics combinations tested if there was more introgression between the divergent ecotypes at the same location or between the same ecotypes at different locations: for D-statistic 1, P1 was the Wave population in the first location (N_2), P2 was the Wave population in the second location (N_4) and P3 was the Crab population at the first location (N_1); for D-statistic 2, P1 was again the Wave population in the first location (N_2) but P2 was the Crab population in the second location (N_3) and P3 was the Crab population at the first location (N_1); for D-statistic 3, P1 was also the Wave population at the first location (N_2), P2 was the Crab population at the first location (N_1) and P3 was the Wave population at the second location (N_4). For all combinations, P4 was assumed to be an outgroup fixed, at all sites, for the major allele. Note that for the four-population models we only considered the proportion of SNPs with fixed differences between the two populations that inhabit the same location. For the proportion of exclusive SNPs, we also computed this per location i.e. checking if each site was segregating in one population but not in the other population inhabiting the same location, but we also computed the proportion of sites that were segregating in only one population and not in the other three.

summary statistic	two-population	four-populations
mean heterozygosity [1]	2 values (1 per population)	4 values (1 per population)
SD heterozygosity [1]	2 values (1 per population)	4 values (1 per population)
mean heterozygosity between populations [1]	1 pairwise value	6 pairwise values
SD heterozygosity between populations [1]	1 pairwise value	6 pairwise values
pairwise F_{ST} [2]	1 pairwise value	6 pairwise values
SD F_{ST} [2]	1 pairwise value	6 pairwise values
5%F_{ST} [2]	1 pairwise value	6 pairwise values
95%F_{ST} [2]	1 pairwise value	6 pairwise values
proportion of fixed differences [3]	1 pairwise value	2 values
proportion of exclusive SNPs [3]	2 values (1 per population)	5 values
mean D-statistic 1 [4]	–	1 value
mean D-statistic 2 [4]	–	1 value
mean D-statistic 3 [4]	–	1 value
SD D-statistic 1 [4]	–	1 value
SD D-statistic 2 [4]	–	1 value
SD D-statistic 3 [4]	–	1 value
total	13	57

[1] - Nei and Roychoudhury (1974); [2] - Bhatia et al. (2013); [3] - Fraïsse et al. (2021); [4] - Adapted from Malinsky et al. (2021) assuming that the outgroup was fixed for an allele different from P3, using $nABBA = \sum_{i=1}^{L}(p_{i1}(1-p_{i2})(1-p_{i3})) + ((1-p_{i1})p_{i2}p_{i3})$, $nBABA = \sum_{i=1}^{L}((1-p_{i1})p_{i2}(1-p_{i3})) + (p_{i1}(1-p_{i2})p_{i3})$, where pij denotes the minor-allele frequency at site i for population j.

Table A4.2: **Prediction errors for two-population model parameters.** Parameter inference was performed using a simple rejection or a regression adjustment using a local linear regression. For each method, values are presented for two different tolerance rates. n_1 and n_2 - relative population sizes of the extant populations, t_{div} - relative time of separation of the ecotype populations, ε_{pool} - experimental error introduced by the pooling procedures, ε_{seq} - error associated with sequencing and mapping errors, m_{12} - probability per generation that an individual migrates from N_1 to N_2 (forward in time), m_{21} - probability per generation that an individual migrates from N_2 to N_1 (forward in time), $4N_2m_{12}$ and $4N_1m_{21}$ - average number of immigrants per generation ($4Nm$) from N_1 to N_2 and from N_2 to N_1 (respectively) and P_{no} - proportion of the simulated loci where no migration occurs between ecotypes.

	REJECTION						**REGRESSION**					
	tolerance of 0.005			*tolerance of 0.01*			*tolerance of 0.005*			*tolerance of 0.01*		
parameter	mode	median	mean	mode	median	mean	mode	median	mean	mode	median	mean
n_1	0.312	0.219	0.220	0.349	0.243	0.243	0.113	0.106	0.106	0.127	0.119	0.119
n_2	0.310	0.213	0.213	0.356	0.239	0.239	0.118	0.110	0.110	0.120	0.111	0.110
t_{div}	1.023	0.564	0.589	1.225	0.607	0.634	0.456	0.340	0.319	0.500	0.367	0.342
ε_{pool}	0.625	0.414	0.432	0.658	0.461	0.487	0.261	0.239	0.236	0.262	0.242	0.242
ε_{seq}	2.527	0.966	0.974	2.651	0.974	0.981	0.914	0.613	0.591	0.884	0.611	0.592
m_{12}	1.404	0.648	0.674	1.390	0.651	0.687	0.655	0.436	0.428	0.609	0.402	0.401
m_{21}	1.301	0.627	0.656	1.368	0.668	0.697	0.668	0.432	0.423	0.710	0.462	0.448
$4N_2m_{12}$	0.762	0.501	0.485	0.800	0.528	0.512	0.338	0.287	0.283	0.336	0.286	0.284
$4N_1m_{21}$	0.768	0.486	0.466	0.837	0.555	0.525	0.308	0.262	0.259	0.351	0.297	0.293
P_{no}	0.323	0.233	0.214	0.327	0.232	0.212	0.117	0.102	0.090	0.089	0.080	0.072

Table A4.3: **Prediction errors for the single origin parameters.** Parameter inference was performed using a simple rejection or a regression adjustment using a local linear regression. For each method, values are presented for two different tolerance rates. n_1 to n_4 - relative population sizes of the extant populations, na_1 and na_2 - relative population sizes of the ancestral populations, t_s - relative time of the split event that lead to the origin of the current populations, δ_s - relative time interval between t_s and the ancient split event (t_{As}), ε_{pool} - experimental error introduced by the pooling procedures, ε_{seq} - error associated with sequencing and mapping errors, m_{12}, m_{34} - probability per generation that an individual migrates from the N_1 or N_3 (Crab) population to the N_2 or N_4 (Wave) population (forward in time), m_{21}, m_{43} - probability per generation that an individual migrates from the N_2 or N_4 (Wave) population to the N_1 or N_3 (Crab) population (forward in time), $4N_2m_{12}$ and $4N_1m_{21}$ - average number of immigrants per generation ($4Nm$) from N_1 to N_2 and from N_2 to N_1 (respectively) at the first site, $4N_4m_{34}$ and $4N_3m_{43}$ - equivalent immigration rates at the second site and P_{no} - proportion of the simulated loci where no migration occurs between ecotypes.

	REJECTION						**REGRESSION**					
	tolerance of 0.005			*tolerance of 0.01*			*tolerance of 0.005*			*tolerance of 0.01*		
parameter	mode	median	mean	mode	median	mean	mode	median	mean	mode	median	mean
n_1	0.759	0.465	0.417	0.830	0.489	0.447	0.142	0.127	0.122	0.126	0.114	0.111
n_2	0.857	0.513	0.451	0.934	0.546	0.490	0.138	0.123	0.119	0.133	0.118	0.113
n_3	0.734	0.452	0.409	0.880	0.530	0.474	0.126	0.113	0.110	0.140	0.125	0.121
n_4	0.821	0.501	0.448	0.957	0.563	0.495	0.127	0.115	0.112	0.149	0.134	0.127
na_1	1.949	1.109	0.954	1.945	1.119	0.963	1.316	0.613	0.583	1.407	0.627	0.596
na_2	1.943	1.103	0.955	1.933	1.112	0.963	1.383	0.643	0.615	1.415	0.646	0.616
t_s	0.070	0.063	0.067	0.075	0.071	0.078	0.039	0.037	0.036	0.039	0.037	0.036
δ_s	1.327	0.694	0.734	1.452	0.741	0.778	0.228	0.193	0.185	0.223	0.188	0.182
ε_{pool}	1.256	0.704	0.767	1.429	0.766	0.822	0.266	0.253	0.236	0.271	0.261	0.243
ε_{seq}	0.539	0.550	0.629	0.619	0.627	0.703	0.084	0.070	0.062	0.088	0.071	0.062
m_{12}, m_{34}	1.579	0.744	0.794	1.569	0.781	0.827	0.523	0.386	0.379	0.559	0.401	0.396
m_{21}, m_{43}	1.410	0.738	0.790	1.528	0.798	0.842	0.522	0.384	0.377	0.549	0.401	0.399
$4N_2m_{12}$	1.072	0.759	0.659	1.156	0.843	0.720	0.357	0.299	0.276	0.426	0.357	0.325
$4N_1m_{21}$	1.113	0.773	0.657	1.123	0.808	0.709	0.396	0.330	0.299	0.367	0.307	0.287
$4N_4m_{34}$	1.129	0.811	0.696	1.188	0.865	0.731	0.365	0.306	0.280	0.393	0.328	0.298
$4N_3m_{43}$	1.153	0.818	0.687	1.149	0.840	0.727	0.358	0.299	0.274	0.388	0.323	0.298
P_{no}	0.190	0.135	0.125	0.235	0.162	0.149	0.044	0.042	0.041	0.045	0.043	0.041

Table A4.4: **Prediction errors for the parallel origin parameters.** Parameter inference was performed using a simple rejection or a regression adjustment using a local linear regression. For each method, values are presented for two different tolerance rates. n_1 to n_4 - relative population sizes of the extant populations, na_1 and na_2 - relative population sizes of the ancestral populations, t_s - relative time of the split event that lead to the origin of the current populations, δ_s - relative time interval between t_s and the ancient split event (t_{As}), ε_{pool} - experimental error introduced by the pooling procedures, ε_{seq} - error associated with sequencing and mapping errors, m_{12}, m_{34} - probability per generation that an individual migrates from the N_1 or N_3 (Crab) population to the N_2 or N_4 (Wave) population (forward in time), m_{21}, m_{43} - probability per generation that an individual migrates from the N_2 or N_4 (Wave) population to the N_1 or N_3 (Crab) population (forward in time), $4N_2m_{12}$ and $4N_1m_{21}$ - average number of immigrants per generation ($4Nm$) from N_1 to N_2 and from N_2 to N_1 (respectively) at the first site, $4N_4m_{34}$ and $4N_3m_{43}$ - equivalent immigration rates at the second site and P_{no} - proportion of the simulated loci where no migration occurs between ecotypes.

	REJECTION						**REGRESSION**					
	tolerance of 0.005			*tolerance of 0.01*			*tolerance of 0.005*			*tolerance of 0.01*		
parameter	mode	median	mean	mode	median	mean	mode	median	mean	mode	median	mean
n_1	0.743	0.437	0.395	0.888	0.512	0.455	0.157	0.138	0.131	0.149	0.134	0.128
n_2	0.731	0.444	0.397	0.858	0.497	0.445	0.145	0.128	0.123	0.141	0.126	0.121
n_3	0.910	0.525	0.456	0.968	0.563	0.494	0.154	0.136	0.130	0.171	0.149	0.140
n_4	0.836	0.480	0.423	0.961	0.553	0.487	0.156	0.137	0.129	0.153	0.135	0.129
na_1	1.913	0.899	0.806	1.914	0.942	0.834	1.161	0.562	0.533	1.199	0.560	0.530
na_2	1.925	0.923	0.813	1.958	0.962	0.840	1.116	0.547	0.524	1.194	0.582	0.549
t_s	0.549	0.360	0.385	0.603	0.389	0.415	0.204	0.171	0.158	0.223	0.189	0.172
δ_s	0.474	0.334	0.347	0.493	0.353	0.367	0.202	0.176	0.167	0.212	0.188	0.179
ε_{pool}	1.270	0.710	0.760	1.346	0.753	0.801	0.263	0.245	0.240	0.261	0.244	0.241
ε_{seq}	0.531	0.471	0.539	0.600	0.542	0.611	0.061	0.051	0.044	0.061	0.049	0.042
m_{12}, m_{34}	1.505	0.767	0.809	1.582	0.803	0.843	0.609	0.454	0.447	0.606	0.447	0.448
m_{21}, m_{43}	1.514	0.781	0.822	1.570	0.800	0.841	0.606	0.450	0.443	0.580	0.438	0.439
$4N_2m_{12}$	1.118	0.772	0.658	1.161	0.799	0.685	0.387	0.331	0.311	0.396	0.333	0.311
$4N_1m_{21}$	1.119	0.764	0.658	1.184	0.827	0.704	0.407	0.345	0.319	0.417	0.353	0.329
$4N_4m_{34}$	1.150	0.789	0.653	1.201	0.860	0.726	0.417	0.345	0.331	0.400	0.340	0.319
$4N_3m_{43}$	1.179	0.834	0.667	1.208	0.870	0.733	0.412	0.350	0.326	0.432	0.367	0.340
P_{no}	0.484	0.314	0.278	0.579	0.368	0.327	0.129	0.120	0.118	0.134	0.126	0.124

Table A4.5: **Biases of the estimates obtained when explicitly modeling or ignoring Pool-seq errors.** We simulated pseudo-observed Pool-seq data and inferred parameters using either a table of summary statistics computed directly from simulated haplotypes without accounting for Pool-seq errors or a table of summary statistics computed after simulating Pool-seq data and explicitly considering depth of coverage variation, unequal individual contribution, and sequencing errors. We computed the bias of the estimates using $\frac{1}{n} \cdot \sum(|\hat{\Theta}_i - \Theta_i|)$, where $\hat{\Theta}_i$ is the estimated mean posterior, and Θ_i is the true parameter value for the i^{th} pseudo-observed dataset, while $n = 100$ is the number of simulated pseudo-observed datasets. n_1 and n_2 - relative population sizes of the present-day populations, t_{div} - relative time of separation of the ecotype populations, $4Nm_{12}$ and $4Nm_{21}$ - average number of immigrants per generation and P_{no} - proportion of the genome without migration.

parameter	ignoring Pool-seq data	accounting for Pool-seq data
n_1	0.605	0.144
n_2	0.584	0.192
t_{div}	0.939	0.630
$4Nm_{12}$	0.797	0.187
$4Nm_{21}$	0.719	0.239
P_{no}	0.018	0.010

Table A4.6: **Estimates for relative parameters of *Littorina saxatilis* populations.** Results are shown for the Arsklovet population for the two-population model and for Arsklovet and Ramsö for the single origin and parallel origin models. For these models n_1 and n_2 correspond to the Arsklovet Crab and Wave population respectively, while n_3 and n_4 correspond to the Ramsö Crab and Wave population respectively. For each parameter, the value outside brackets corresponds to the mean of the posterior distribution and in-between brackets is the 95% credible interval.

parameter	two-population	single origin	parallel origin
n_1	0.334 (0.219 - 0.596)	0.557 (0.249 - 1.841)	0.315 (0.134 - 0.732)
n_2	0.286 (0.184 - 0.499)	0.296 (0.158 - 0.993)	0.754 (0.241 - 1.895)
n_3	–	0.682 (0.296 - 1.919)	0.662 (0.208 - 1.718)
n_4	–	0.825 (0.337 - 2.221)	0.939 (0.277 - 2.189)
na_1	–	2.203 (0.459 - 2.870)	2.641 (1.554 - 2.980)
na_2	–	1.139 (0.208 - 2.554)	2.396 (0.873 - 2.963)
t_{div}	0.165 (0.020 - 1.517)	–	–
t_s	–	0.014 (0.009 - 0.022)	0.007 (0.005 - 0.018)
δ_s	–	0.386 (0.129 - 1.158)	0.029 (0.002 - 0.070)
m_{12}, m_{34}	0.00073 (0.00013 - 0.0009)	0.00048 (0.00012 - 0.00094)	0.00024 (0.00002 - 0.00076)
m_{21}, m_{43}	0.00049 (0.00005 - 0.00096)	0.00058 (0.00016 - 0.00096)	0.00077 (0.00028 - 0.00099)
P_{no}	0.012 (0.001 - 0.066)	0.013 (0.002 - 0.054)	0.205 (0.015 - 0.428)
ε_{pool}	182 (67 - 236)	102 (24 - 183)	130 (23 - 222)
ε_{seq}	0.00100 (0.00098 - 0.00100)	0.00092 (0.00059 - 0.00099)	0.00099 (0.00097 - 0.00100)

Table A4.7: **Biases of the estimates obtained with subsets of loci.** We simulated a pseudo-observed dataset of 100 loci and inferred parameters using the full dataset or subsets representing 10%, 30%, or 50% of the genome. To compute the bias, we contrasted the mean of the posterior distribution obtained with subsets of loci with the mean posterior obtained with 100 loci. The bias was computed a) after weighted combination of posteriors obtained with subsets representing 10%, 30% or 50% of the genome and b) by using the summary statistics of the full dataset as the target for parameter inference performed with 10%, 30% or 50% of the genome. n_1 and n_2 - relative population sizes of the present-day populations, t_{div} - relative time of separation of the ecotype populations, $4Nm_{12}$ and $4Nm_{21}$ - average number of immigrants per generation and P_{no} - proportion of the genome without migration.

	a) merging posteriors			b) whole-genome		
parameter	10%	30%	50%	10%	30%	50%
n_1	0.136	0.043	0.008	0.256	0.043	0.006
n_2	0.186	0.093	0.046	0.209	0.080	0.042
t_{div}	0.867	0.521	0.242	0.845	0.426	0.212
$4Nm_{12}$	4.564	1.584	1.863	13.261	0.730	1.878
$4Nm_{21}$	5.061	1.479	0.107	25.674	1.052	0.106
P_{no}	0.009	0.012	0.010	-0.020	0.003	0.006

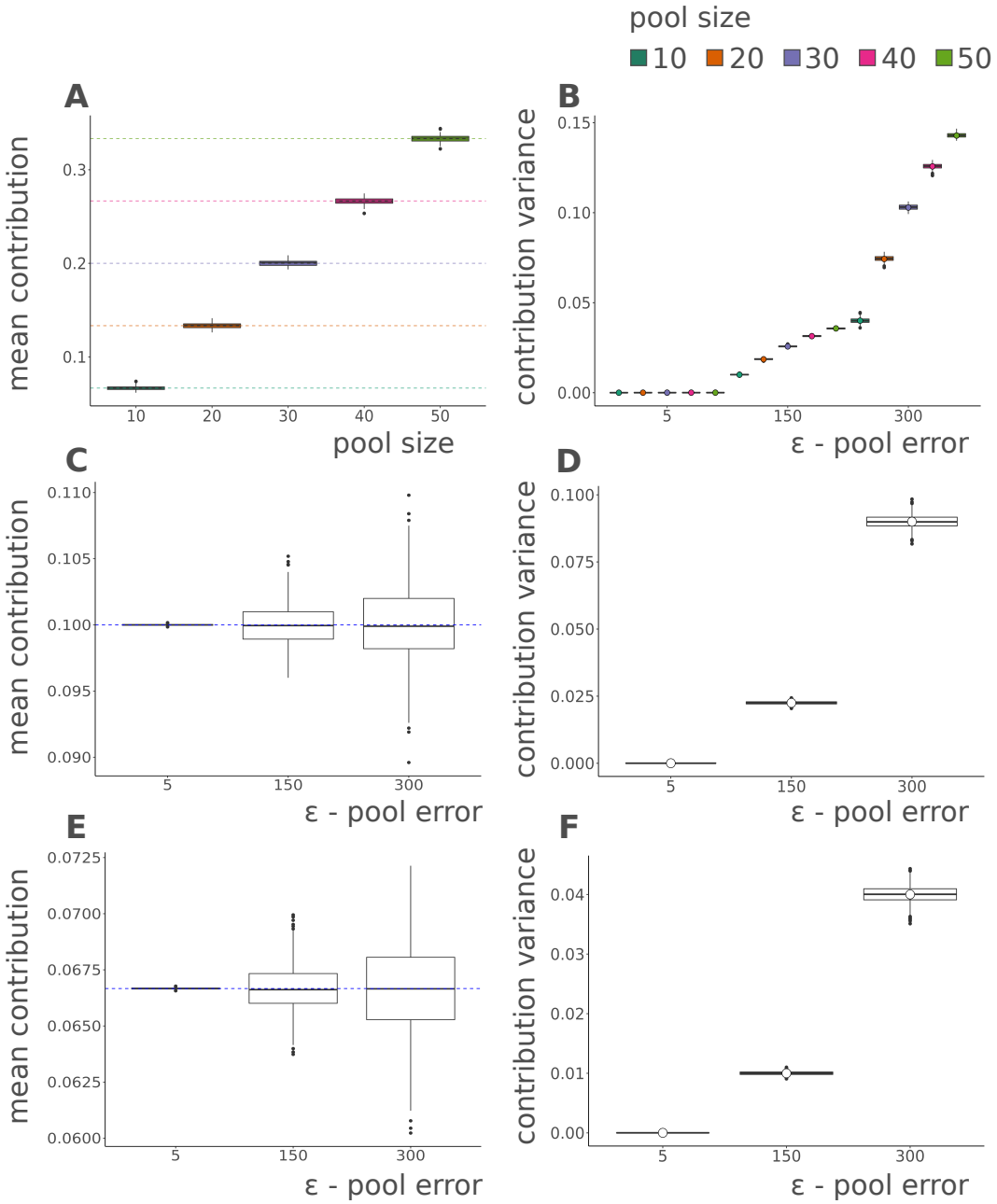


Figure A4.1: Simulations converge to the expected mean and variance in all cases. We simulated the proportion of contribution at 100 loci, each with 10k SNPs, and calculated the mean and variance of contribution per locus. Expected mean proportions are represented by dashed lines, and expected variances by dots on the box-plots. Simulations with 5 pools of varying size show that, even when pool error is high (300), each pool has the expected contribution (A) and variance (B). Expected values for A and B were obtained with eqs. A4.2 and A4.8. Simulations with 10 pools of the same size (10 individuals per pool) show that the contribution of each pool converges to the expected mean (C) and variance (D) across pool errors. Expected values for C and D were obtained with eqs. A4.10-A4.12. Simulations of the contribution of individuals (15) within a pool show that the mean contribution (E) and variance (F) also converge to the values of eqs. A4.13-A4.15.

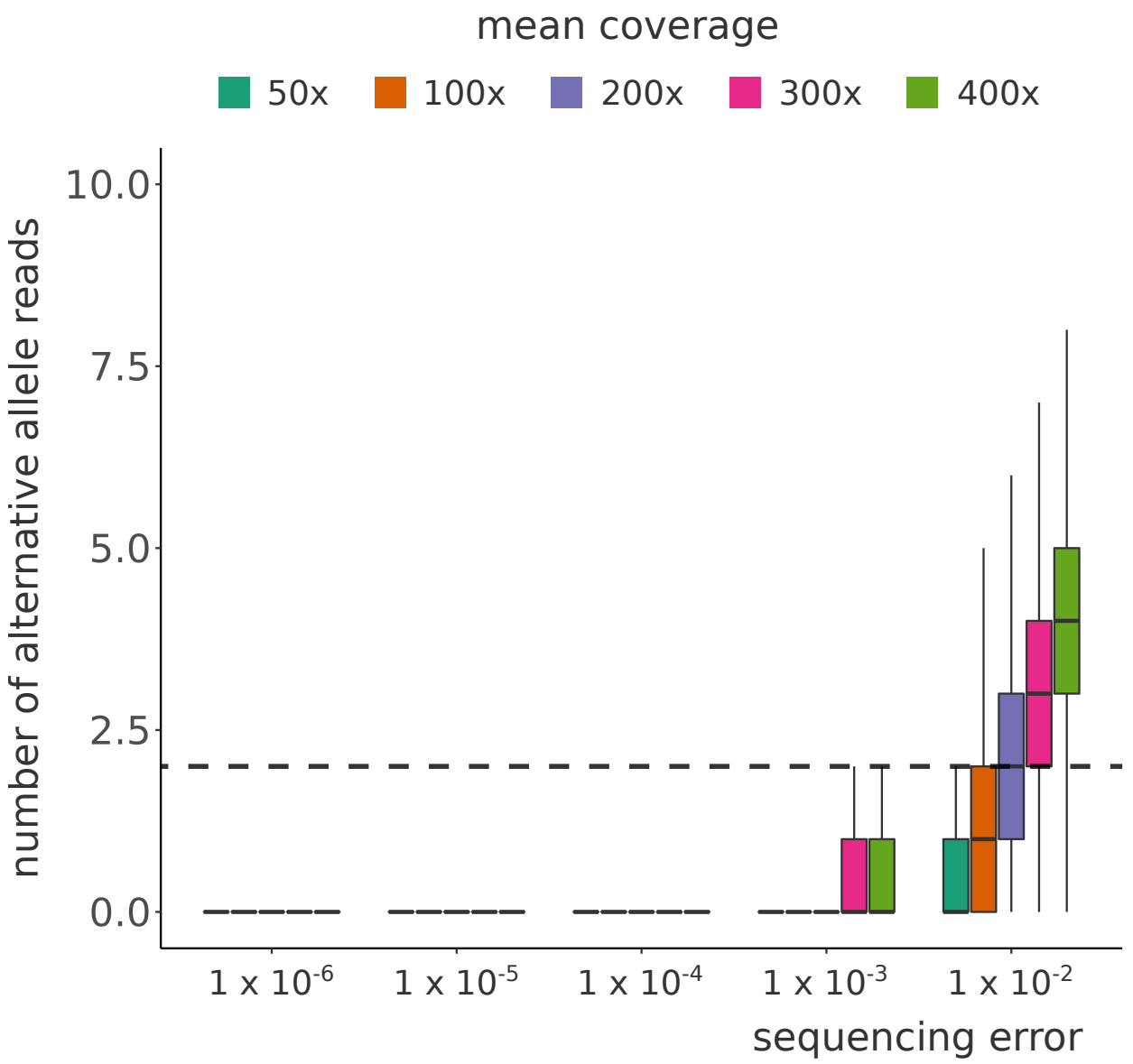


Figure A4.2: Impact of the sequencing error and depth of coverage on the detection of false SNPs. We simulated Pool-seq data for a single pool of 100 individuals, sampled at 300 loci, each with 50 SNPs. Since all individuals were homozygous for the reference allele, any alternative allele reads were considered false SNPs. We plotted the number of alternative allele reads (y-axis) observed at different levels of sequencing errors and depths of coverage. The dashed line represents the threshold we applied, discarding SNPs with fewer than two minor-allele reads. Note that all *L. saxatilis* populations analysed here had a mean coverage below 100x and thus, our minor-allele threshold should eliminate most false SNPs even when sequencing error is high.

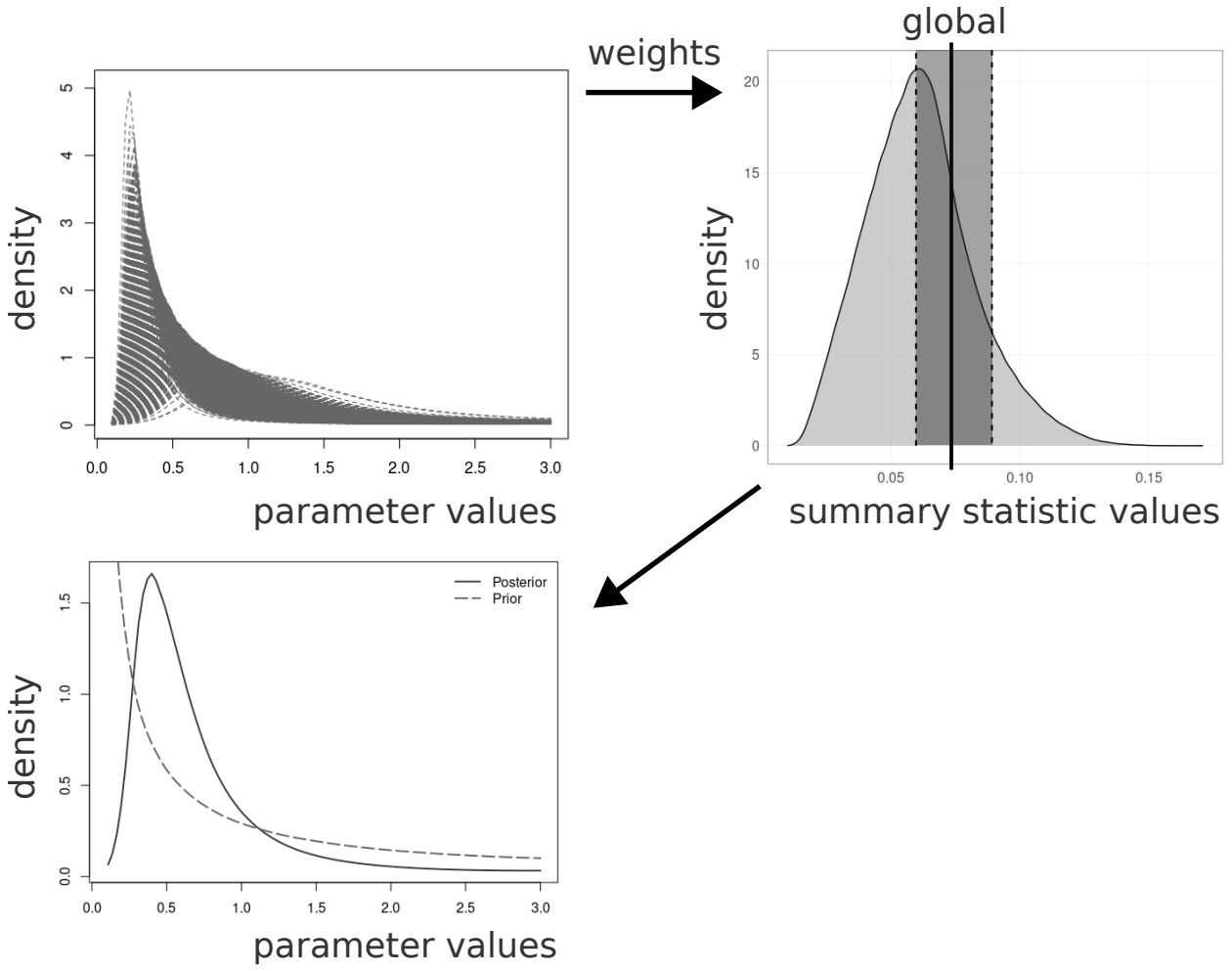


Figure A4.3: Merging of multiple posterior distributions. This represents an example of how the posteriors obtained for each set of loci are combined to obtain a single estimate per parameter. In the top-left plot several posteriors distributions are shown, one for each set of loci and for a given parameter. These multiple posteriors are weighted according to the distance between the summary statistics of the corresponding simulations and the mean across the genome, giving more weight to sets of loci with a mean closer to the overall mean. The top-right plot represents an example of this, where the simulations with values closer to the global value (represented by the black line) will have more weight. Using these weights, the multiple posteriors are combined to obtain a single estimate per parameter, as shown in the bottom-left panel.

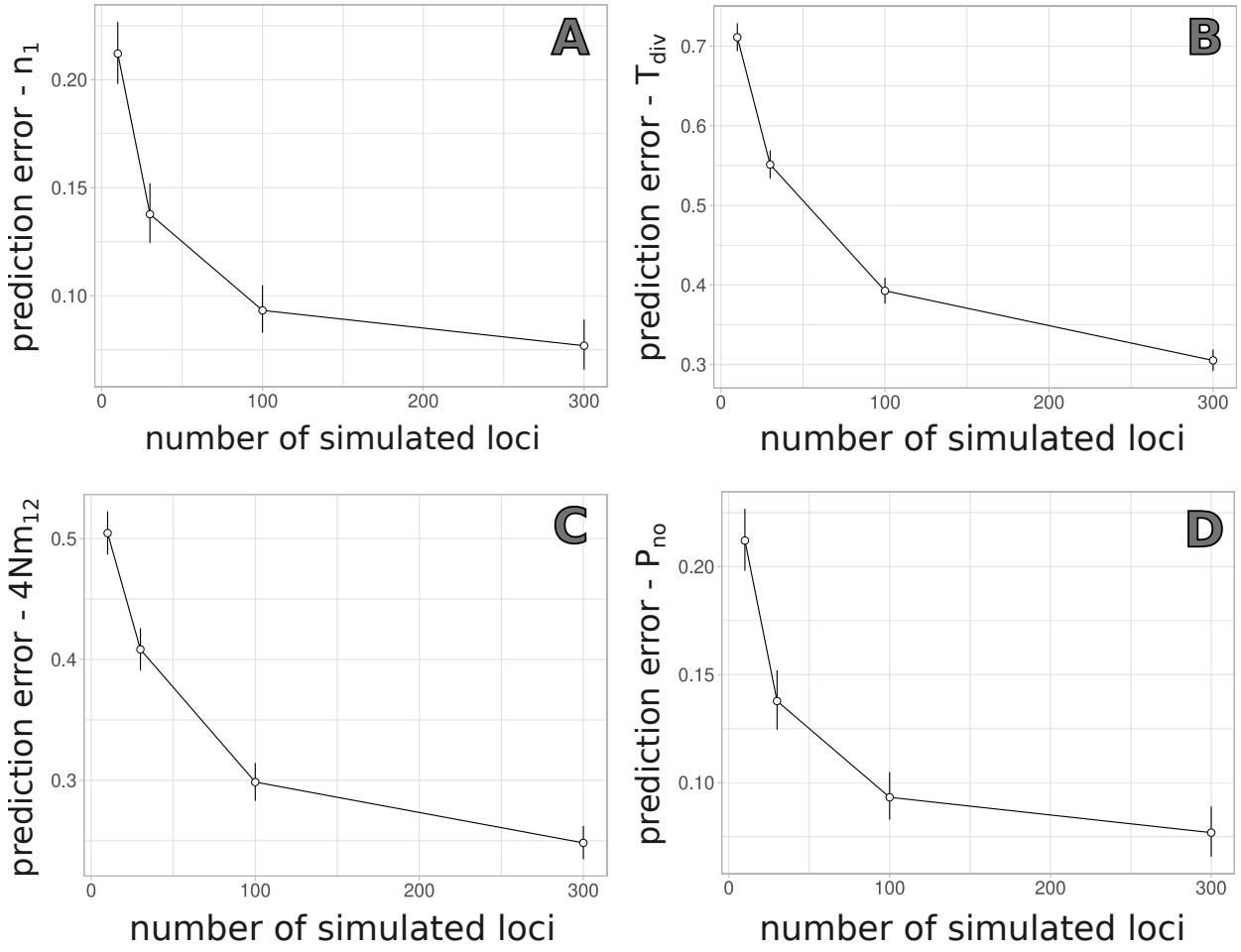


Figure A4.4: Impact of number of loci on the prediction error. A leave-one-out cross-validation simulation study with varying numbers of loci per subset was performed to compute the prediction error for several demographic parameters. The prediction error is shown on the y-axis. The x-axis shows the numbers of loci per subset. Points represent the mean prediction error after bootstrapping and error bars represent 95% confidence intervals. Parameters shown here are: A - relative size of a present-day population (n_1), B - relative time of separation of the ecotype populations (t_{div}), C - average number of immigrants per generation ($4Nm_{12}$) and D - proportion of the genome without migration (P_{no}).

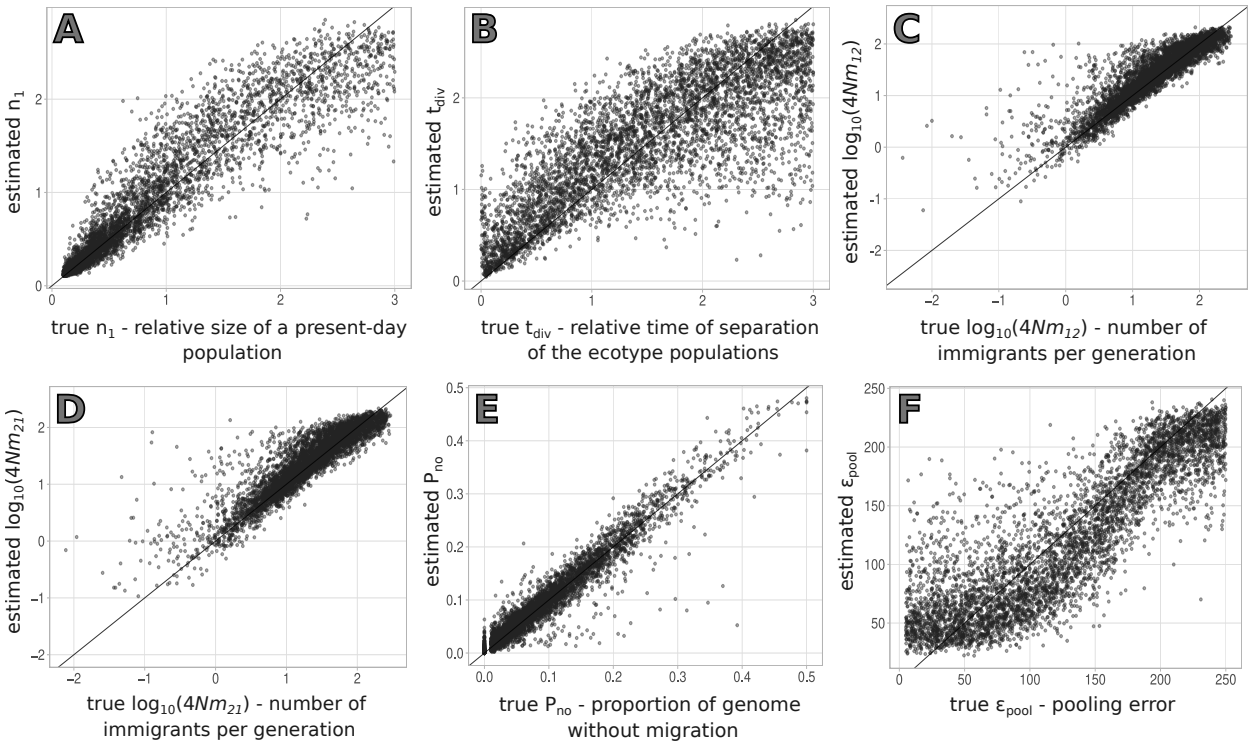


Figure A4.5: Results of the cross-validation for parameter estimation using the two-population model. The y-axis displays the estimated values, plotted against the true parameter values on the x-axis. Estimates correspond to the mean of the posterior obtained with a tolerance rate of 0.01. Parameters shown here are: A - relative size of a present-day population (n_1), B - relative time of separation of the ecotype populations (t_{div}), C and D - average number of immigrants per generation ($4Nm_{12}$ and $4Nm_{21}$, respectively), E - proportion of the genome without migration between different populations (P_{no}) and F - pooling error

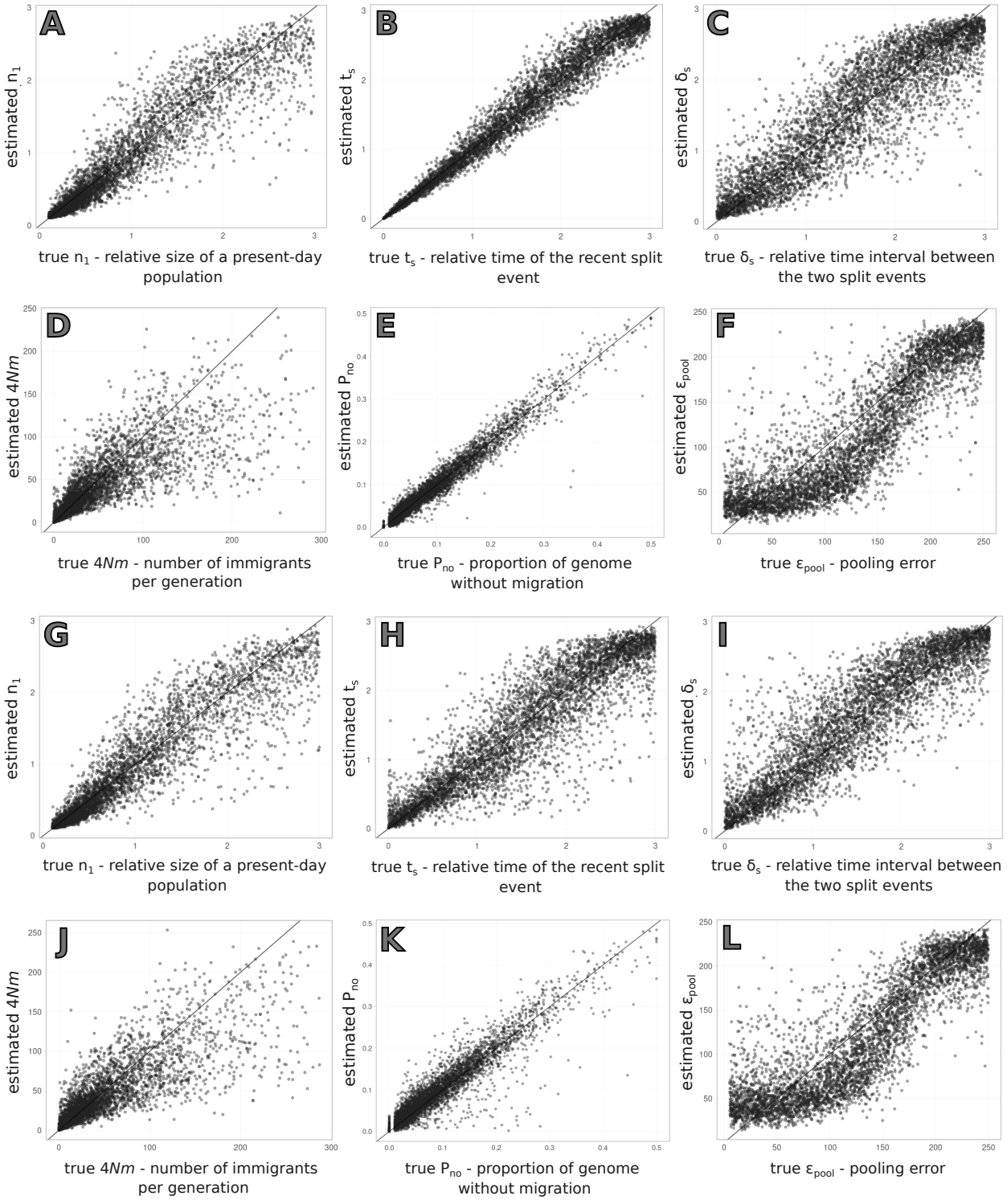


Figure A4.6: Results of the cross-validation for parameter estimation using the four-population models. Panels from A to F show the results for the single origin model, while panels G to L show the results for the parallel origin model. The y-axis displays the estimated values, plotted against the true parameter values on the x-axis. Estimates correspond to the mean of the posterior obtained with a tolerance rate of 0.01. Parameters shown here are: A and G - relative size of a present-day population (n_1), B and H - relative time of the recent split event (t_s), C and I - relative time interval between the two split events (δ_s), D and J - average number of immigrants per generation ($4Nm$), E and K - proportion of the genome without migration between different populations (P_{no}) and F and L - pooling error

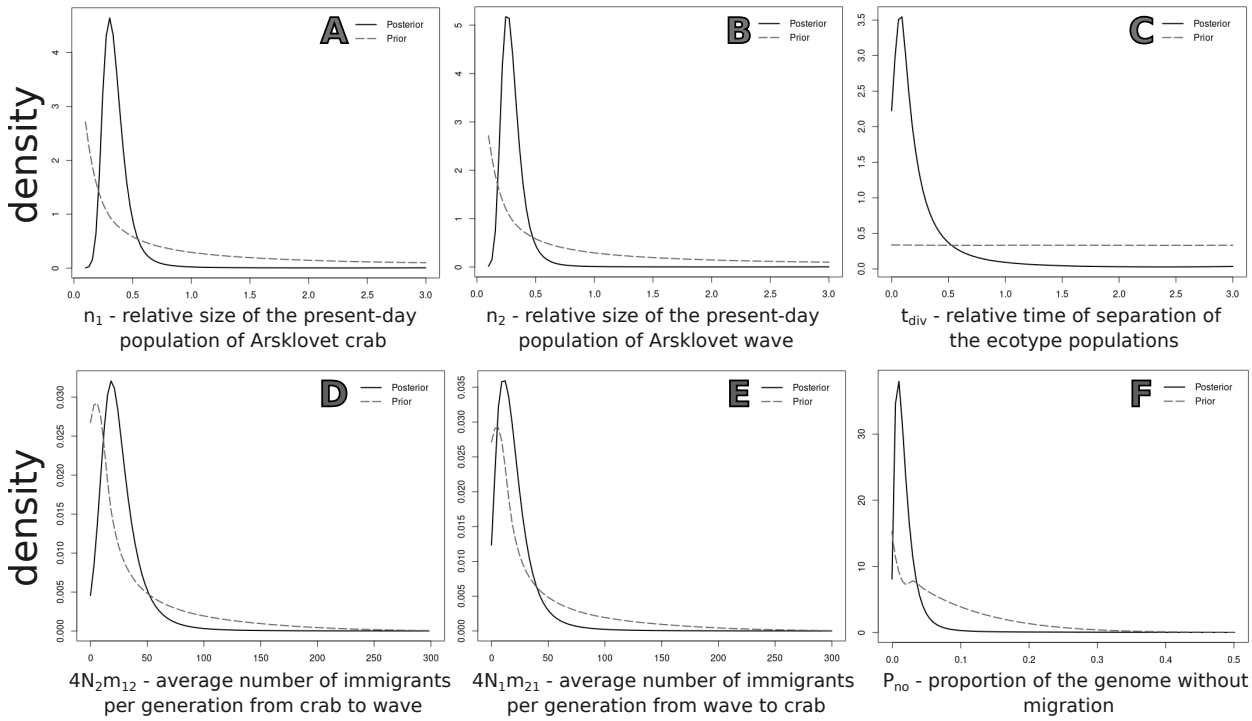


Figure A4.7: Posterior distribution of relative *L. saxatilis* parameters using the two-population model. The posterior distributions were obtained with the regression adjustment method and using a tolerance rate of 0.01. For reference, the prior distribution of each parameter is shown (dotted line). Parameters shown here are: A - relative size of the Arsklovet Crab population (n_1), B - relative size of the Arsklovet Wave population (n_2), C - relative time of separation of the ecotype populations (t_{div}), D and E - average number of immigrants per generation ($4Nm_{12}$ and $4Nm_{21}$, respectively) and F - proportion of the genome without migration between different populations (P_{no}). The relative parameter values presented here were converted to absolute values using a re-scaling factor $f = obs[S]/E[S]$, where $obs[S]$ corresponds to the observed number of SNPs and $E[S]$ is the expected number of SNPs. Absolute parameter values were obtained by multiplying the point estimate of the posteriors shown here by the rescaling factor f.

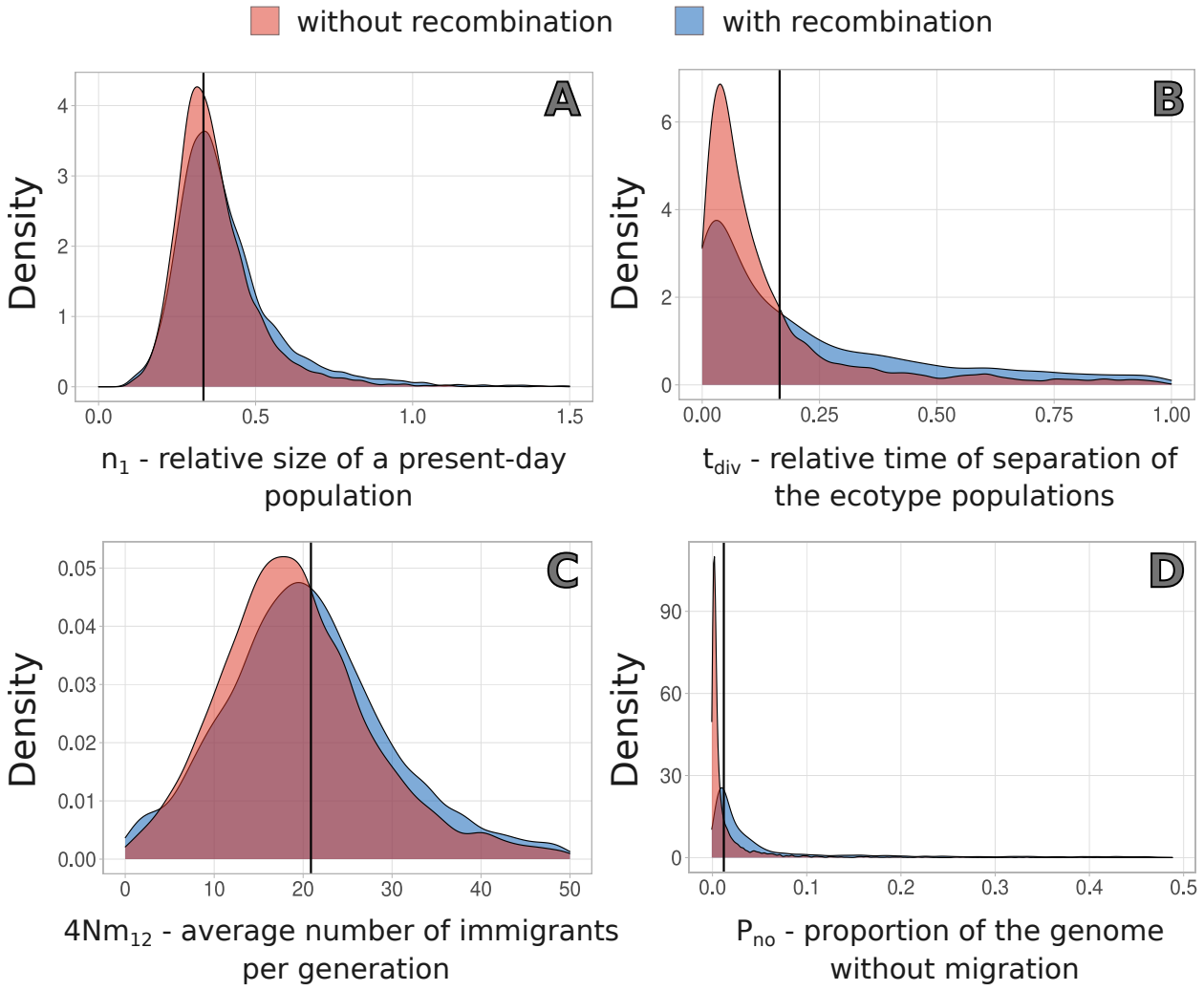


Figure A4.8: Impact of within-locus recombination on parameter estimates. We used simulations that excluded within-locus recombination to estimate the parameters of two pseudo-observed datasets: one with and another without within-locus recombination. The x-axis shows the estimated parameter value, and the y-axis shows the density of the posterior distribution. The posterior obtained for the pseudo-observed dataset without within-locus recombination is shown in red and the posterior for the pseudo-observed dataset with within-locus recombination in blue. The solid vertical line represents the true parameter value. Parameters shown here are: A - relative size of a present-day population (n_1), B - relative time of separation of the ecotype populations (t_{div}), C - average number of immigrants per generation ($4Nm_{12}$) and D - proportion of the genome without migration (P_{no}).

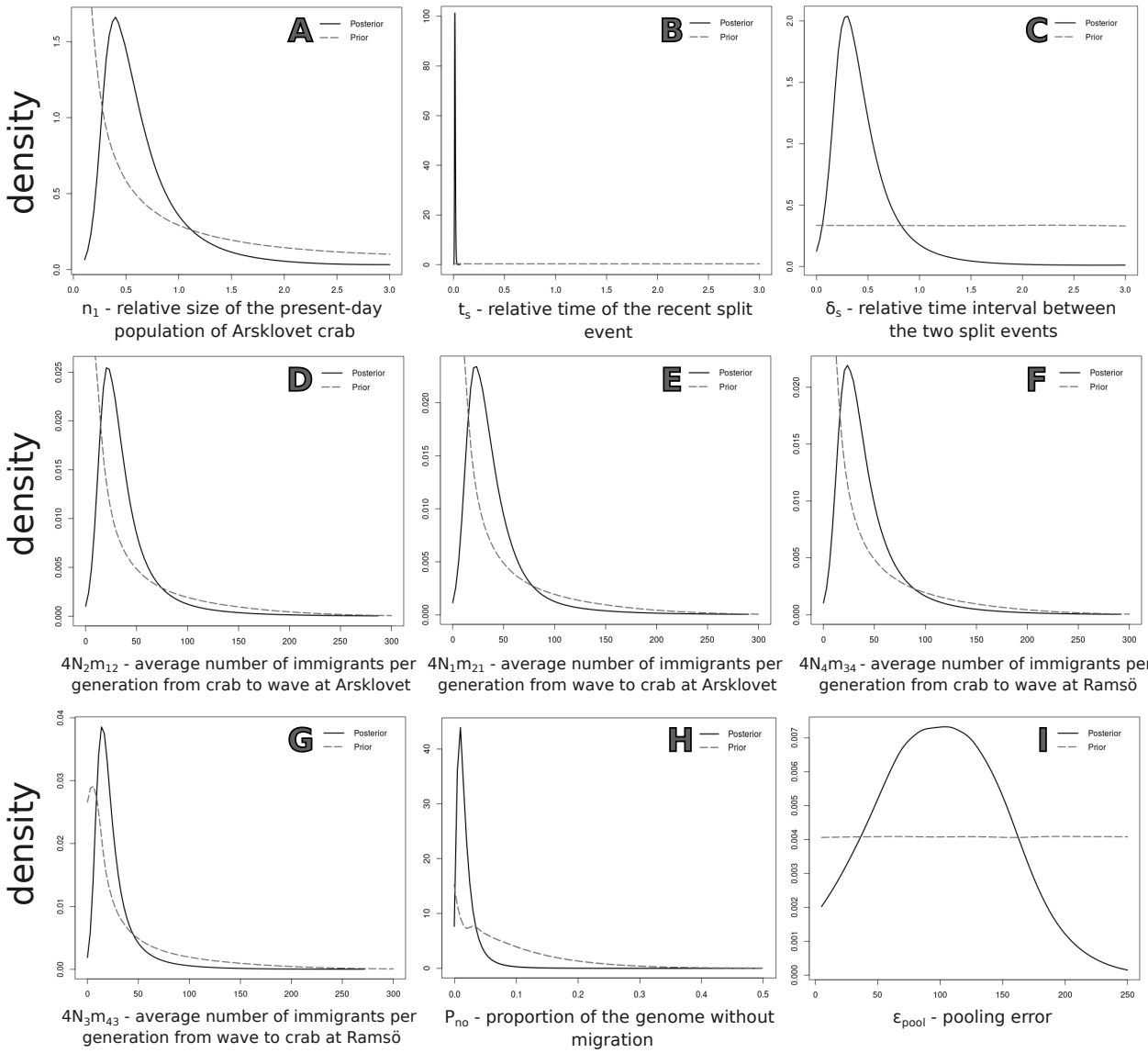


Figure A4.9: Posterior distribution of relative *L. saxatilis* parameters using the single origin model. The posterior distributions were obtained with the regression adjustment method and using a tolerance rate of 0.01. For reference, the prior distribution of each parameter is shown (dotted line). Parameters shown here are: A - relative size of the Arsklovet Crab population (n_1), B - relative time of the recent split event (t_s), C - relative time interval between the two split events (δ_s), D and E - average number of immigrants per generation ($4Nm$) from Crab to Wave and from Wave to Crab (respectively) at Arsklovet, F and G - average number of immigrants per generation ($4Nm$) from Crab to Wave and from Wave to Crab (respectively) at Ramsö, H - proportion of the genome without migration between different populations (P_{no}) and I - pooling error. The relative parameter values presented here were converted to absolute values using a re-scaling factor $f = obs[S]/E[S]$, where $obs[S]$ corresponds to the observed number of SNPs and $E[S]$ is the expected number of SNPs. Absolute parameter values were obtained by multiplying the point estimate of the posteriors shown here by the rescaling factor f.

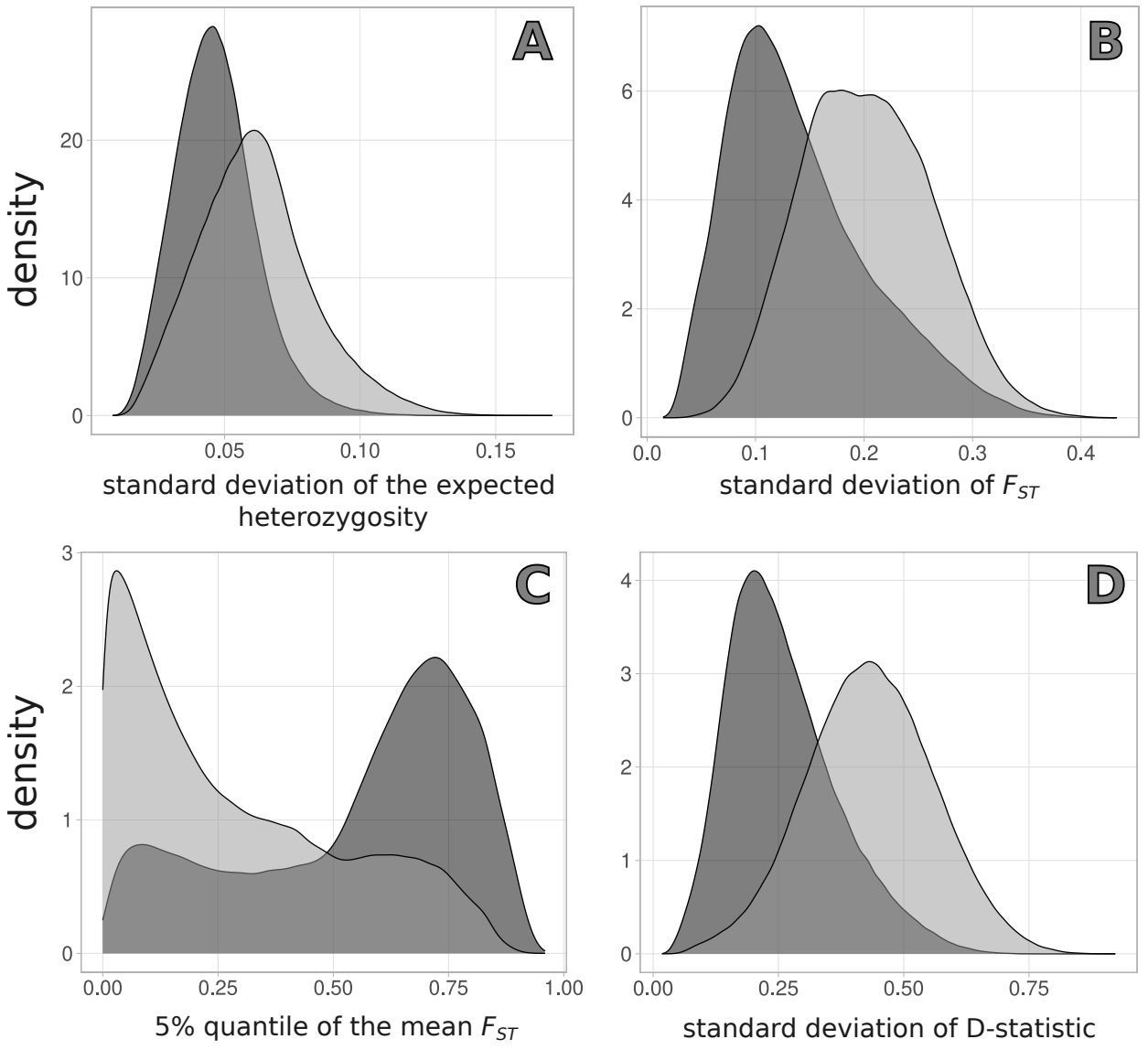


Figure A4.10: Distribution of summary statistics obtained for the single and parallel origin models. Dark shading indicates the parallel origin model and light shading the single origin model. Summary statistics are: A - standard deviation of the expected heterozygosity for a given population, B - standard deviation of mean pairwise F_{ST}, C - 5% quantile of the mean pairwise F_{ST} and D - standard deviation of D-statistic

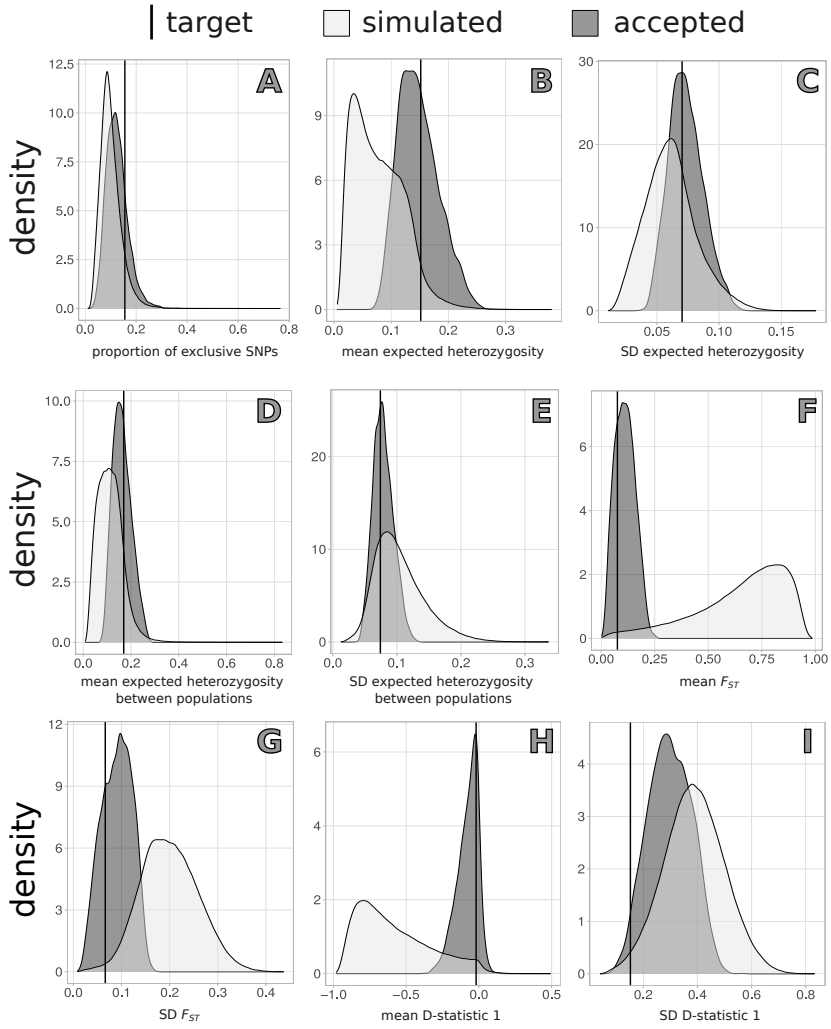


Figure A4.11: Distribution of accepted summary statistics. The black line represents the target for the parameter inference, the light shading is the distribution of the complete set of simulated summary statistics and the dark shading is the distribution of the accepted summary statistics for that particular target. Summary statistics include examples of all those analysed here: A - proportion of exclusive sites, B - mean heterozygosity, C - standard deviation of the mean heterozygosity, D - mean heterozygosity between a pair of populations, E - standard deviation of the mean heterozygosity between a pair of populations, F - mean F_{ST} between a pair of populations, G - standard deviation of F_{ST}, H - D-statistic and I - standard deviation of D-statistic.

4.8.1 MODELLING UNEQUAL CONTRIBUTION OF INDIVIDUALS AND POOLS

Here we describe the explicit modeling of Pool-seq errors associated with unequal contribution of individuals and pools, deriving expected values and variances for the proportion and counts of reads from each individual and pool. We extend the approach of Gautier et al. (2013) to pools with unequal number of individuals, using a general notation that can be applied to model unequal contribution from both individuals and pools. Namely, we refer to p_k as the proportion of reads from pool k, ν_k as the number of individuals of pool k, K to the number of pools, and ν to the total number of individuals across pools ($\nu = \sum_{k=1}^{K} \nu_k$). We use $E[.]$ and $V[.]$ to denote expected value and variance, respectively. We derive results for pools of unequal size (with relative size ν_k/ν). We then derive results for pools of the same size as a special case when $\nu_k/\nu = 1/K$. Finally, we derive the results for individuals as a special case by considering that the number of pools is equal to the number of individuals (i.e., $K = \nu$) with the size of 1 individual per pool (i.e., with a relative size of $\nu_k = 1/\nu$). Note that in the material and methods section of the manuscript, the notation also includes the population label j, but for simplicity we drop it here, e.g. ν_k rather than $\nu_{j,k}$.

We start by considering the general case where pools have different sizes, assuming that the proportion p_k of reads from pool k follows a Dirichlet distribution, where the dispersion is controlled by the parameter ρ,

$$p_k \sim Dir\left(\rho \cdot \frac{\nu_k}{\nu}\right) \tag{A4.1}$$

Given the properties of the Dirichlet distribution, the expected value of the proportion of reads from pool k is

$$E[p_k] = \frac{\rho \frac{\nu_k}{\nu}}{\sum_{p=1}^{K} \rho \frac{\nu_k}{\nu}} = \frac{\nu_k}{\sum_{p=1}^{K} \nu_k} = \frac{\nu_k}{\nu} \tag{A4.2}$$

and the variance of p_k is

$$V[p_k] = \frac{\frac{\rho \frac{\nu_k}{\nu}}{\sum_{p=1}^{K} \rho \frac{\nu_k}{\nu}}(1 - \frac{\rho \frac{\nu_k}{\nu}}{\sum_{p=1}^{K} \rho \frac{\nu_k}{\nu}})}{(\sum_{p=1}^{K} \rho \frac{\nu_k}{\nu}) + 1} = \frac{\frac{\nu_k}{\nu}(1 - \frac{\nu_k}{\nu})}{\rho + 1} = \frac{E[p_k](1 - E[p_k])}{\rho + 1} \tag{A4.3}$$

The parameter ρ is the same for all pools, as it reflects the dispersion around the expected values, with values closer to zero indicating a higher variance. As ρ goes to infinity, the variance tends to zero ($\lim_{\rho\to+\infty}\frac{E[p_k](1-E[p_k])}{\rho+1}=0$), whereas in the limit as ρ tends to zero, the variance tends to $E[p_k](1-E[p_k])$, ($\lim_{\rho\to 0}\frac{E[p_k](1-E[p_k])}{\rho+1}=E[p_k](1-E[p_k])$), which is its maximum value. Following Gautier et al. (2013), this variance can be expressed in terms of a pool-Seq error parameter ε_k, which is proportional to the expected value:

$$V[p_k]=\varepsilon_k^2\cdot E[p_k]^2 \tag{A4.4}$$

Replacing $V[p_k]$ in A4.4 by its definition in A4.3, we obtained:

$$\frac{E[p_k](1-E[p_k])}{\rho+1}=\varepsilon_k^2\cdot E[p_k]^2 \tag{A4.5}$$

Solving this for ρ, we obtained the general solution:

$$\begin{aligned}
&(\rho+1)\varepsilon_k^2E[p_k]^2=E[p_k](1-E[p_k])\\
&\varepsilon_k^2(\rho+1)=\frac{\cancel{E[p_k]}(1-E[p_k])}{E[p_k]^{\cancel{2}}}\\
&\varepsilon_k^2\rho+\varepsilon_k^2=\frac{(1-E[p_k])}{E[p_k]}\\
&\rho=\frac{(1-E[p_k])-\varepsilon_k^2}{E[p_k]\varepsilon_k^2}\\
&\rho=\frac{E[p_k]^{-1}(1-E[p_k])-\varepsilon_k^2}{\varepsilon_k^2}
\end{aligned} \tag{A4.6}$$

By definition ρ needs to be a positive value. Hence, the range of possible values for ε_k is bounded, i.e., $\varepsilon_k^2<E[p_k]^{-1}(1-E[p_k])$. Since $E[p_k]=\nu_k/\nu$ and $E[p_k]^{-1}=\nu/\nu_k$ we can obtain the following expression for ρ

$$\rho=\frac{(1-\frac{\nu_k}{\nu})\frac{\nu}{\nu_k}-\varepsilon_k^2}{\varepsilon_k^2}=\frac{\frac{\nu}{\nu_k}-1-\varepsilon_k^2}{\varepsilon_k^2} \tag{A4.7}$$

Thus, in the limit as ε_k tends to zero, ρ will tend to infinity ($\lim_{\varepsilon_k\to 0}\frac{\frac{\nu}{\nu_k}-1-\varepsilon^2}{\varepsilon^2}=\infty$), whereas as

ε tends to the upper bound of its range $\sqrt{\frac{\nu}{\nu_k}-1}$, ρ tends to zero ($\lim_{\varepsilon_k \to \sqrt{(\frac{\nu}{\nu_k}-1)}} \frac{\frac{\nu}{\nu_k}-1-\varepsilon_k^2}{\varepsilon_k^2} = 0$).

To ensure that the expected value of the proportion of reads $E[p_k]$ corresponds to the relative size of each pool (i.e., $E[p_k] = \frac{\nu_k}{\nu}$), ρ needs to be the same for all pools (eq. A4.2). This implies that pools with different sizes have different ε_k. Given that the maximum value of ε_k depends on the relative number of individuals in each pool (i.e., $\varepsilon_k^2 < \frac{\nu}{\nu_k} - 1$), we compute ρ using eq. A4.7 based on the pool with fewer individuals, as it corresponds to the case of a larger upper bound for ε_k (i.e., larger $\frac{\nu}{\nu_k} - 1$). Setting ε_k as the error of the pool with the lowest size, the error ε_j for another pool j can be written using a scaling factor α_j. Below, we derive α_j considering two pools k and j of different sizes ν_k and ν_j, respectively, with $\nu_k \leq \nu_j$

$$\begin{cases} V[p_k] = \varepsilon_k^2 \cdot E[p_k]^2 \\ V[p_j] = \varepsilon_j^2 \cdot E[p_j]^2 = (\alpha_j \varepsilon_k)^2 \cdot E[p_j]^2 \end{cases} \tag{A4.8}$$

Replacing the variances given in eq. A4.3 and expected values in eq. A4.2, and solving for ε_k^2 and α_j^2

$$\begin{cases} \frac{(\nu_k/\nu)(1-(\nu_k/\nu))}{\rho+1} = \varepsilon_k^2(\nu_k/\nu)^2 \\ \frac{(\nu_j/\nu)(1-(\nu_j/\nu))}{\rho+1} = \alpha_j^2\varepsilon_k^2(\nu_j/\nu)^2 \end{cases} \equiv \begin{cases} \varepsilon_k^2 = \frac{(1-(\nu_k/\nu))}{(\rho+1)(\nu_k/\nu)} \\ \alpha_j^2 = \frac{(1-(\nu_j/\nu))(\nu_k/\nu)}{(1-(\nu_k/\nu))(\nu_j/\nu)} \end{cases} \tag{A4.9}$$

The result of eq. A4.9 is general for any number of pools. We now consider the special case of pools of the same size, i.e., same ν_k for all K pools. In this case, $\nu_k/\nu = 1/K$. Hence, the expected values and variances are the same for all pools:

$$E[p_k] = \frac{\rho(1/K)}{\sum_{p=1}^{K}\rho(1/K)} = \frac{1}{K} \tag{A4.10}$$

$$V[p_k] = \frac{(1/K)(1-(1/K))}{\rho+1} \tag{A4.11}$$

and ρ is given as a function of the pool-seq error $\varepsilon = \varepsilon_k$, which is the same for all pools.

$$\rho = \frac{(1-\frac{1}{K})K - \varepsilon^2}{\varepsilon^2} = \frac{K-1-\varepsilon^2}{\varepsilon^2} \tag{A4.12}$$

In this case, all the pools have the same ε, as the scaling factor α is equal to 1, since $\nu_k = \nu_j$ and $\nu_k/\nu = 1/K$ for all pools. Finally, we obtain the result for the unequal contribution of individuals inside a single pool solving for the special case of $K = \nu$ with 1 individual per pool (i.e., $\nu_k = 1$). That is, one pool with ν individuals is treated as if we had a mixture of ν pools, each with one individual. We use the notation p_i to refer to proportion of reads from each individual in a single pool, as in the main text of the manuscript.

$$E[p_i] = \frac{\rho(1/\nu)}{\sum_{i=1}^{\nu}\rho(1/\nu)} = \frac{1}{\nu} \tag{A4.13}$$

$$V[p_i] = \frac{(1/\nu)(1-(1/\nu))}{\rho+1} \tag{A4.14}$$

To clarify the link with the main text, we also use an explicit notation for the parameters related to the pool-seq error of individuals, ρ_i and ε_i

$$\rho_i = \frac{(1-\frac{1}{\nu})\nu - \varepsilon_i^2}{\varepsilon_i^2} = \frac{\nu - 1 - \varepsilon_i^2}{\varepsilon_i^2} \tag{A4.15}$$

These results are the same as those obtained by Gautier et al. (2013) for the expected values, variances and ρ obtained for the contribution of individuals in a single pool. The Supplementary Figure A4.1 shows that simulations done under our model converge to these analytical results. In the main manuscript, we treat the error ε_k (for pools) and ε_i (for individuals) as nuisance parameters in our simulations, sampled from a uniform prior distribution. All simulations in the main manuscript were done assuming pools of identical size. Hence, we computed the constant ρ (ρ_p in the notation of the main manuscript) that describes the unequal contribution of the K pools using eq. A4.12. For the constant ρ_i that describes the unequal contribution of individuals within each pool we used eq. A4.15. In the estimation of parameters, we considered a single parameter ε that was set to be the error associated with the unequal contribution of individuals, i.e. $\varepsilon = \varepsilon_i$. Rather than considering that the error associated with the unequal contribution of pools was the same as the unequal contribution of individuals (i.e., setting $\varepsilon_k = \varepsilon_i$), we considered that $\varepsilon_k^2 = (K-1)/(\nu-1)\varepsilon_i^2$. The rationale is that the range of values for ε_k^2 is between zero and $K-1$ (see eq. A4.12); whereas for ε_i^2 is between zero and $\nu - 1$ (see eq. A4.15).

4.8.2 INPUT AND OUTPUT FILES

For the simulation of Pool-seq data, our method relies on custom-made R functions that do not require a particular input file but instead require a set of user inputs at each appropriate function. To simulate the total depth of coverage for each population, the user must define the mean and the variance of the depth of coverage for each population, as well as the total number of SNPs to simulate. To simulate pools, the user must define the pool error to use in the simulation (ε_{pool}). Finally, to obtain the number of reads with the derived allele, D_i, the user must also supply a value for the sequencing/mapping error (ε_{seq}) and the genotypes, ideally obtained using coalescent theory to simulate gene trees. After this step, our method provides a function to translate the number of ancestral/derived alleles into major/minor alleles, ensuring that the minor allele is the one for which we have fewer reads across all the populations. At this step, the user also has the choice to remove sites with fewer than x minor allele reads, where x is a user-defined threshold. The output of this section of our method are two different matrices, one containing the number of minor allele reads and the other containing the total depth of coverage. Both matrices are in the $nPop \times nSNP$ format, meaning that each row contains the information for a given population, while each column is a different site. These matrices can be used to compute allele frequencies and calculate several summary statistics.

Our ABC method is designed to work with the *_rc* files produced by the snp-frequency-diff.pl script from the PoPoolation2 suite (Kofler, Pandey, & Schlötterer, 2011). This file contains the number of major and minor allele reads for every SNP in a concise format (for more information see: `https://sourceforge.net/p/popoolation2/wiki/Manual/`). Given the modular nature of the method, it can also accommodate inputs in the form of matrices, where one of the matrices contains the number of minor allele reads and the other contains the total depth of coverage. These matrices should be in the format $nPop \times nSNP$, meaning that each row should contain the information for a given population, while each column is a different site. Note that an additional matrix, of the same dimensions, containing SNP position and contig information should also be available. The input files can then be filtered, removing sites with high or low coverage and sites with too few minor allele reads. For ABC parameter inference and model selection, summary statistics are computed for several random blocks of windows and used as the target. The final output of model selection includes the proportion of accepted simulations for a model under a rejection algorithm and the posterior model probabilities of each model after a local linear regression adjustment. For parameter inference, the output includes the estimates under the rejection algorithm, the regression adjusted estimates if a local linear regression was performed and the median, mean, mode and 95% confidence interval of the weighted posteriors for each parameter.

4.8.3 Vignette for the poolABC package

A4 - Introduction

An implementation of Approximate Bayesian Computation (ABC) methods applied to pooled sequencing (Pool-seq) data is available in R language in the package `poolABC`. The purpose of this vignette is to provide an in-depth overview of the capabilities of the package, highlighting the usage of its main functions.

```
library(poolABC)
```

The initial sections of this vignette detail how to import pooled sequencing data from files in the `_rc` format. We also show how to randomly select multiple subsets of the observed data, compute summary statistics for those subsets and use those summary statistics as target for parameter estimation and model selection with ABC.

This vignette also teaches users how to simulate Pool-seq data under pre-defined models. Note that the simulation of Pool-seq data requires functions included in the `poolHelper` package. We then exemplify how the imported data and the simulated data can be used to perform parameter inference. The `poolABC` package allows the use of genome-wide multilocus data for ABC by using multiple subsets of simulated and observed loci.

Briefly, we obtain one set of summary statistics for each set of simulated loci and for each random subset of observed data. Each set of summary statistics computed from a unique subset of observed data is used as an independent target for parameter estimation. Thus, with the `poolABC` package, users obtain one posterior distribution, for each parameter of interest and for each subset of observed loci. Then, our package allows users to combine those multiple posteriors to obtain a single estimate per parameter. This merging is performed with the Epanechnikov kernel and weighting according to the distance between the mean summary statistics of a subset of loci and the mean across the genome, giving more weight to subsets of loci with a mean closer to the overall mean. Finally, the `poolABC` package also includes functions to compute point estimates and produce plots of those merged posterior distributions.

A4 - Import dataset

This package uses pooled sequencing data stored on `_rc` format files. These `_rc` files are created by running the `SNP-frequency-diff.pl` function of `popoolation2`. Briefly, this is an example of a typical `_rc` file with only two populations:

```
data.frame(chr=c("NC297", "NC297"), pos=c(3530, 5450), rc=c("A", "T"),
           allele_count=2, allele_states=c("A/G", "T/A"), deletion_sum=0,
           snp_type="pop", major_alleles=c("AA","TT"),
           minor_alleles=c("GG", "AN"), maa_1=c("54/55", "51/54"),
           maa_2=c("76/78", "96/96"), mia_1=c("1/55", "3/54"),
           mia_2=c("2/78", "0/96"))
```

More information about these files can be found at: https://sourceforge.net/p/popoolation2/wiki/Main/

If you have your data in `_rc` files in a folder of your computer, you can simply use the `importContigs` function.

```
# load multiple files and organize information by contig
files <- importContigs(path = "/home/myawesomedata", pops = c(1, 4, 7, 10))
```

The `path` input of this function indicates the path to the folder where the `_rc` files are located. By default, the `importContigs` function will import all files present in the folder that include the `_rc` pattern in their name. The index of populations to import is defined by the `pops` input argument. For instance, the input `pops = c(1, 4, 7, 10)` will import the major and minor allele for the first, fourth, seventh and tenth population in the `_rc` files.

The `importContigs` function also includes several optional input arguments. The `files` input argument allows you to specify the index of the files to import. For instance, by setting `files = 1:5`, only the first five files listed in the output of `list.files(path = path)` will be imported. Additionally, specific contigs can be removed from the data by adding their names to the `remove` input argument.

The `min.minor` input argument allows you to filter the data by the number of minor allele reads. For instance, if `min.minor = 2` all sites where the total number of minor allele reads across all populations of the `pops` input argument is below 2, will be removed from the data. Alternatively, by setting `filter = TRUE`, you can filter the data by the frequency of the minor allele. When `filter = TRUE`, the user can define a `threshold` for the minimum allowed frequency of the minor allele. If no `threshold` is defined, the `importContigs` function will assume that at least one minor allele

read per site should exist. Finally, it is possible to include an header when importing the data. This header can be created with the `createHeader` function.

A4 - Random subset of loci

Random windows of a given size (in base pairs) can be selected from the imported data with the `pickWindows` function. The data imported with the `importContigs` function is a list with all the elements required for the `pickWindows` function. You can assign each of those list elements to an individual object or use them directly as input argument of the `pickWindows` function.

```
# randomly select blocks of a given size from several contigs
blocks <- pickWindows(freqs, positions, range, rMajor, rMinor, coverage,
                      window = 1000, nLoci = 100)
```

With this function, users can randomly select a subset of the complete pooled sequencing data at their disposal. More specifically, the `pickWindows` function allows users to randomly select `nLoci` blocks of `window` size (in base pairs) from the data imported in the previous section. In other words, this function will randomly select select `nLoci` contigs and then select one random block with a user defined size (defined by the `window` input) per contig.

A4 - Compute stats for observed data

The next step is the computation of a set of summary statistics from the observed data. To compute summary statistics from the observed data, we can use the `statsContig` function. This function will compute the same set of summary statistics used in the simulations from the multiple random subset of loci obtained in the previous step.

```
# compute a set of observed summary statistics
obs <- statsContig(randomWindows=blocks, nPops=4, stat.names=stat.names)
```

Note that we are using the `blocks` object created with the `pickWindows` function as the `randomWindows` input argument. The `statsContig` function will compute summary statistics from those randomly selected blocks of observed data. Also, the use of names for the summary statistics is strongly recommended. To ensure that the set of observed summary statistics is named, we should obtain the name of the simulated summary statistics and include those in the `stat.names` input argument of the `statsContig` function.

A4 - Simulate data for two or four-population models

With this package, you also have the ability to simulate pooled sequencing data under three different models by using the `poolSim` function. The `model` input argument allows the user to define which model to simulate. At the moment, this package includes three different models: an isolation with migration model with two populations, a model representing a single origin of two divergent ecotypes and a third model representing a parallel origin of those ecotypes.

To simulate data using the two populations model, you have to define the mean depth of coverage and the variance of the coverage for those two populations. You also need to create a list with the number of individuals per pool and per population. In the next chunk, you can see how to simulate data using this model:

```
# set the mean and variance of the coverage
sMean <- c(84.34, 66.76); sVars <- c(1437.22, 912.43)

# create a list containing the information of the pool sizes by population
size <- rep(list(rep(10, 10)), 2)

# run simulation for a two-populations model
sims <- poolSim(model="2pops", nDip=200, nPops=2, nLoci=100, nSites=2000,
                mutrate=2e-8, size=size, mean=sMean, variance=sVars,
                minimum=15, maximum=180, min.minor=1, Nref=c(25000, 25000),
                ratio=c(0.1, 3), pool=c(5, 180), seq=c(0.0001, 0.001),
                split=c(0, 3), CW=c(1e-13, 1e-3), WC=c(1e-13, 1e-3),
                bT=c(0, 0.5))
```

The `poolSim` function requires several input arguments, that are explained in detail in the help page of the function. However, note that most of those input arguments define the minimum and maximum values for a variety of relevant parameters. To simulate data using a four populations model:

```
# set the mean
sMean <- c(84.34, 66.76, 65.69, 68.83)
# and variance of the coverage
sVars <- c(1437.22, 912.43, 848.02, 1028.23)

# create a list containing the information of the pool sizes by population
size <- rep(list(rep(5, 20)), 4)
```

```
# run simulation for a four-populations model
sims <- poolSim(model="Single", nDip=400, nPops=4, nLoci=100, nSites=2000,
                mutrate=2e-8, size=size, mean=sMean, variance=sVars,
                minimum=15, maximum=180, min.minor=2, Nref=c(25000, 25000),
                ratio=c(0.1, 3), pool=c(5, 180), seq=c(0.0001, 0.001),
                split=c(0, 3), CW=c(1e-13, 1e-3), WC=c(1e-13, 1e-3),
                CC=c(1e-13, 1e-3), WW=c(1e-13, 1e-3), ANC=c(1e-13, 1e-3),
                bT=c(0, 0.5), bCW=c(0, 0.5), bWC=c(0, 0.5))
```

A4 - multiple simulations

The `poolSim` function can be used to perform a single simulation. However, most of the times, you will want to perform thousands of simulations. One way to accomplish this is to use `replicate` function together with our `poolSim` function. We recommend that you do the following:

```
# set the mean and variance of the coverage
sMean <- c(84.34, 66.76); sVars <- c(1437.22, 912.43)

# create a list containing the information of the pool sizes by population
size <- rep(list(rep(5, 20)), 2)

# define how many simulations to run
nSims <- 10

# run one batch of simulations
sims <- t(replicate(n=nSims,
                    unlist(poolSim(model="2pops", nDip=200, nPops=2,
                                   nLoci=100, nSites=1000, mutrate=2e-8,
                                   size=size, mean=sMean, variance=sVars,
                                   minimum=20, maximum=185, min.minor=2,
                                   Nref=c(25000, 25000), ratio=c(0.1, 3),
                                   pool=c(5, 180), seq=c(0.0001, 0.001),
                                   split=c(0, 3), CW=c(1e-13, 1e-3),
                                   WC=c(1e-13, 1e-3), bT=c(0, 0.5)))))
```

By using the `replicate` function, you can perform multiple simulations. By unlisting and then transposing the output of those simulations, you obtain a matrix where each row corresponds to a different simulation and each column is a different parameter or summary statistic.

A4 - Perform parameter estimation

The observed summary statistics computed in the previous sections and the simulations performed in the previous one can then be used to perform parameter estimation with Approximate Bayesian Computation (ABC).

We included with this package a small dataset simulated under the two populations model. This includes one matrix (`sumstats`) with the summary statistics computed from the simulated data, one matrix (`params`) with the simulated parameter values and a final matrix (`limits`) with the minimum and maximum value of the prior distribution for each parameter.

```
# load the data included in the package
data("sumstats"); data("params"); data("limits")
```

The `poolABC` package aims at streamlining the process of parameter inference with Pool-seq data. One of the key components of that design is the `ABC` function.

By using this function, users can simultaneously perform parameter estimation with ABC for multiple targets. The `ABC` function requires the data, imported with the `importContigs` function and then uses both the `pickWindows` and `statsContig` functions to select multiple random subset of loci from the observed data and compute a set of observed summary statistics for each of those subsets. Thus, for each subset of loci we obtain a vector of summary statistics and each vector acts as an independent target for parameter estimation. The `ABC` function can be used by doing:

```
# parameter estimation with ÃBC function
myabc <- ABC(nPops=2, ntrials=10, freqs=freqs, positions=positions,
             range=range, rMajor=rMajor, rMinor=rMinor, coverage=coverage,
             window=1000, nLoci=100, limits=limits, params=params,
             sumstats=sumstats, tol=0.01, method="regression")
```

The `ntrials` input argument defines the number of independent targets for parameter estimation. In this example, we are performing parameter inference for 10 different targets. Each of those targets was obtained by computing summary statistics from windows of 1000 base pairs (`window = 1000`) from 100 (`nLoci = 100`) randomly selected contigs of the observed data.

Note that you should define the `method` and tolerance rate, `tol`, to use. The `tol` is defined as the percentage of accepted simulation. You should strive to keep a low tolerance rate, to avoid accepting simulations that are too distant from the observed data, but it is also important to avoid very stringent tolerance rates that may lead to few accepted values. A typical value of `tol = 0.01` or `tol = 0.05` is recommended but you should test different `tol` values in the cross-validation analysis (see more in subsequent sections).

This package implements two ABC algorithms for constructing the posterior distribution from the accepted simulations: a rejection method and a regression-based correction using a local linear regression. When `method` is "rejection", simulations are accepted if the Euclidean distance between the set of summary statistics computed from the simulated data and the target is sufficiently small and these accepted simulations are considered a sample from the posterior distribution. When `method` is "regression", an additional step is used to correct for the imperfect match between the summary statistics computed from the simulated data and the summary statistics computed from the observed data. For this reason, we recommend that you select the regression method because it will, most often than not, lead to more precise parameter estimates.

A4 - Merge multiple posteriors

After using the `ABC` function to perform parameter estimation with Approximate Bayesian Computation for several targets, we need to merge the multiple posteriors obtained (one for each target) into a single posterior per parameter.

This can be performed with the `mergepost` function. One of the required input arguments of this function is the `global` input. This input should be a vector with the observed summary statistics computed from the entire dataset. We recommend that you use the `pickWindows` function to select a large number of loci and then use that random selection as the input argument of the `statsContig` function.

```
# load multiple files and organize information by contig
blocks <- pickWindows(freqs = freqs, positions = positions, range = range,
                      rMajor = rMajor, rMinor = rMinor, coverage = coverage,
                      window = 1000, nLoci = 800)

# compute a set of summary statistics from the observed data
global <- statsContig(randomWindows=blocks, nPops=2, stat.names=stat.names)
```

The global vector can then be used in the `mergepost` function. The remaining required input arguments are the matrix with the `target` for the parameter inference and the list containing the posteriors (`post`) for each target and parameter. It is also possible to include the regression weights in the `wtreg` option.

```
# merge posterior distributions
myabc <- mergepost(target = myabc$target, global = global,
                   post = myabc$adjusted, wtreg = myabc$weights)
```

Briefly, this function will merge the different posteriors into a single one, using different weighting methods for the merging. Details about the various elements of the `mergepost` output can be found in the help page of the function. Note that the `merge`, `weighted`, `merge_reg` and `weighted_reg` entries contain, for each parameter, a `locfit` object, obtained after merging the multiple posteriors using the corresponding method. The `merged_stat`, `weighted_stat`, `merge_reg_stat` and `weighted_reg_stat` are the posterior point estimates for the corresponding merging method.

A4 - Posterior point estimates and plots

Users can then plot the resulting merged posterior distribution with the `plot_weighted` function. You should include your matrix with the simulated parameter values in the `prior` input argument of this function to also plot the prior distribution of the chosen parameter. This allows for a comparison, in the same plot, of the prior and posterior shape.

You should include the output of the `mergepost` function as the `merged_posterior` input argument and a matrix with the `limits` of the prior distribution for each parameter. Then, you just need to define which parameter to plot with the `index` input argument.

```
# plot the density estimation of a parameter
plot_weighted(prior=params, merged_posterior=myabc, limits=limits, index=2)
```

The `plot_weighted` function plots the posterior density of the chosen parameter, together with the prior distribution of the same plot.

A4 - Model selection

The `poolABC` package also allows users to perform model selection by estimating the posterior model probabilities, comparing two scenarios of ecotype formation: the single and the parallel origin scenario. The `modelSelect` function can be used to perform model selection with ABC.

One of the required input arguments of the `modelSelect` function is the `index`. This is a vector of model indices that should have the same `length` as `nrow(sumstats)` to indicate to which model a particular row of `sumstats` belongs. The remaining input arguments are explained in the help page of the function. As before, you should also define the tolerance rate (`tol`) and the `method` to use. A tolerance of 0.01 and the "regression" method are recommended.

```
# create a vecto of model indices
index <- c(rep("model1", nrow(sumstats)/2), rep("model2", nrow(sumstats)/2))
# select a random simulation to act as target
target <- sumstats[10, ]
# perform model selection with ABC
mysel <- modelSelect(target = target, index = index, sumstats = sumstats,
                     tol = 0.1, method = "regression")

# display the structure of the mysel object
str(mysel)
#> List of 6
#>  $ method : chr "regression"
#>  $ indices: Factor w/ 2 levels "model1","model2": 1 1 1 1 1 1 1 1 1 1 ...
#>  $ pred   : 'table' num [1:2(1d)] 0.541 0.459
#>   ..- attr(*, "dimnames")=List of 1
#>   .. ..$ : chr [1:2] "model1" "model2"
#>  $ ss     : num [1:1000, 1:14] 0 0.0106 0.00118 0.01243 0 ...
#>   ..- attr(*, "dimnames")=List of 2
#>   .. ..$ : NULL
#>   .. ..$ : chr [1:14] "Sf" "Sx1" "Sx2" "SS" ...
#>  $ weights: num [1:1000] 0.2723 1 0.0146 0.0171 0.3713 ...
#>  $ nmodels: Named int [1:2] 5000 5000
#>   ..- attr(*, "names")= chr [1:2] "model1" "model2"
```

The output of the `modelSelect` function is a list with six entries. To quickly view the results of the model selection, you can use the `summary_modelSelect` function. This function will provide an easy to read display of the posterior model probabilities and Bayes factors. The only required input argument of the `summary_modelSelect` function is the object created with the the `modelSelect` function.

```
# check results of model selection
msel <- summary_modelSelect(object = mysel)
#> Data:
#>  results based on 1000 posterior samples
#>
#> Models a priori:
#>  model1, model2
#>
#> Models a posteriori:
#>  model1, model2
#>
#> Proportion of accepted simulations (rejection):
#> model1 model2
#>   0.51   0.49
#>
#> Posterior model probabilities (mnlogistic):
#> model1 model2
#>  0.541  0.459
```

As you can see, by running the `summary.modelSelect` function we get an output with the proportion of accepted simulation for each model under a rejection method and posterior model probabilities under the regression method. If we print the object itself

```
# print results of model selection
msel
#> $rejection
#> $rejection$Prob
#> model1 model2
#>   0.51   0.49
#>
#> $rejection$BayesF
#>           model1   model2
```

```
#> model1 1.0000000 1.040816
#> model2 0.9607843 1.000000
#>
#>
#> $mnlogistic
#> $mnlogistic$Prob
#>    model1    model2
#> 0.5406397 0.4593603
#>
#> $mnlogistic$BayesF
#>           model1   model2
#> model1 1.0000000 1.176941
#> model2 0.8496605 1.000000
```

we can also see the Bayes factors between pairs of models for both the rejection and the regression methods.

A4 - Cross validation for Approximate Bayesian Computation

A fundamental part of any ABC analysis is the validation of the results obtained in the parameter estimation and model selection steps. The `poolABC` package includes tools to perform cross validation for both analysis, computing prediction errors for both parameter inference and model selection.

A4 - Parameter inference

One important component of this validation process is the calculation of prediction errors for each parameter. This allows us to evaluate the confidence of the estimates and the effect of various point estimates and/or tolerance rates.

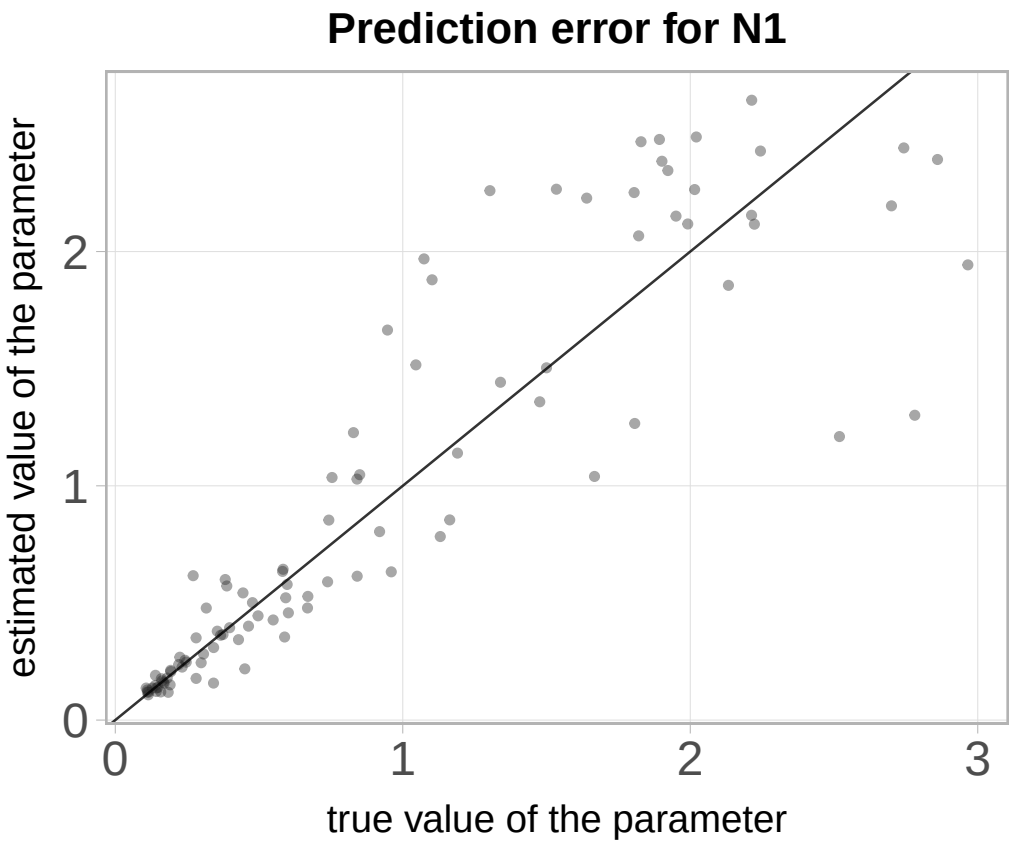
To perform a leave-one-out cross validation for ABC, you can use the `simulationABC` function. This functions requires the simulated parameter values, `params`, the simulated summary statistics, `sumstats` and a matrix with the `limits` of the prior distributions. You should also define the size of the cross-validation sample, `nval`, the tolerance rate, `tol`, and the type of ABC algorithm to be applied in the `method` input.

```
# perform an Approximate Bayesian Computation simulation study
mycv <- simulationABC(params = params, sumstats = sumstats, limits = limits,
                      nval = 100, tol = 0.01, method = "regression")

# display the structure of the mycv object
str(mycv, max.level = 2)
#> List of 3
#>  $ true: num [1:100, 1:8] 0.163 1.902 1.821 0.225 0.375 ...
#>   ..- attr(*, "dimnames")=List of 2
#>  $ rej :List of 4
#>   ..$ mode  : num [1:100, 1:8] 0.099 3 1.873 0.186 0.25 ...
#>   .. ..- attr(*, "dimnames")=List of 2
#>   ..$ median: num [1:100, 1:8] 0.579 1.725 1.906 0.216 0.326 ...
#>   .. ..- attr(*, "dimnames")=List of 2
#>   ..$ mean  : num [1:100, 1:8] 0.681 1.798 1.895 0.267 0.381 ...
#>   .. ..- attr(*, "dimnames")=List of 2
#>   ..$ error : num [1:3, 1:8] 0.443 0.246 0.227 0.405 0.292 ...
#>   .. ..- attr(*, "dimnames")=List of 2
#>  $ reg :List of 4
#>   ..$ mode  : num [1:100, 1:8] 0.163 2.576 2.018 0.25 0.337 ...
#>   .. ..- attr(*, "dimnames")=List of 2
#>   ..$ median: num [1:100, 1:8] 0.167 2.386 2.067 0.268 0.365 ...
#>   .. ..- attr(*, "dimnames")=List of 2
#>   ..$ mean  : num [1:100, 1:8] 0.171 2.295 2.082 0.279 0.381 ...
#>   .. ..- attr(*, "dimnames")=List of 2
#>   ..$ error : num [1:3, 1:8] 0.245 0.209 0.188 0.173 0.149 ...
#>   .. ..- attr(*, "dimnames")=List of 2
```

The output of the `simulationABC` function is a list with three elements. Details about each list element are available in the help page of the function. A quick way to visualize the results of the leave-one-out cross validation is to plot the the cross-validation results.

The `poolABC` package includes the `plot_errorABC` function to allow this visual evaluation of the quality of the estimation. This function requires as input the output of the `simulationABC` function. Additionally, you need to define the ABC algorithm (either "reg" for regression or "rej" for rejection) and which point estimate ("mode", "median" or "mean") to plot. You should also define which parameter to plot, by selecting the corresponding `index`.

```
# plot the prediction errors
plot_errorABC(x = mycv, method = "reg", statistic = "median", index = 1)
```



This produces a plot with the true parameter value in x-axis and the estimate value of the parameter in the y-axis. The closer the points are to the diagonal line, the higher is the accuracy of the estimation.

A4 - Model selection

It is also possible to evaluate how much confidence we should place in the model selection results by performing a leave-one-out cross validation for model selection with ABC via subsequent calls to the function `modelSelect`.

Briefly, several simulations from each model are selected to act as validation simulations, while the remaining simulations are used as training simulations. For each validation simulation, the function `modelSelect` is called to estimate the posterior model probabilities.

```
# perform a leave-one-out cross validation for model selection
modelSim <- sim_modelSel(index=index, sumstats=sumstats, nval=100, tol=0.1)
# display the structure of the modelSim object
str(modelSim, vec.len=2, list.len=2)
```

```
#> List of 5
#>  $ cusamples  : Named int [1:200] 3675 2098 2719 2819 2408 ...
#>   ..- attr(*, "names")= chr [1:200] "model11" "model12" ...
#>  $ true       : chr [1:200] "model1" "model1" ...
#>   [list output truncated]
```

The output of this leave-one-out cross validation for model selection is a list with 5 different elements that can be used in the `error_modelSel` function to compute the confusion matrix and the mean misclassification probabilities of models.

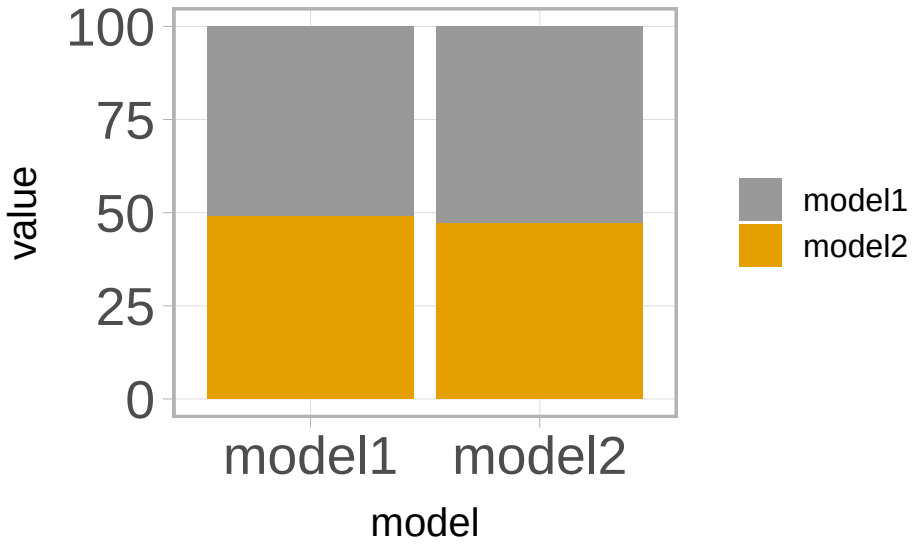
Users can also define a `threshold` for the posterior model probabilities. This `threshold` corresponds to the minimum posterior probability of assignment. Thus, a simulation where the posterior probability of any model is below the threshold will not be assigned to a model and will instead be classified as `unclear`.

```
# compute the mean misclassification probabilities
mselError <- error_modelSel(object = modelSim)
#> Confusion matrix based on 100 samples for each model
#>
#>        model1 model2
#> model1     51     49
#> model2     53     47
#>
#> Mean model posterior probabilities (mnlogistic)
#>
#>        model1 model2
#> model1  0.495  0.505
#> model2  0.503  0.497
#>
#> Posterior probabilities of correctly assigned model1 models
#>
#>    model1 incorrect
#>     0.534     0.466
#>
#> Posterior probabilities of correctly assigned model2 models
#>
#> incorrect    model2
```

```
#>     0.462     0.538
#>
#> Posterior probabilities when model1 is estimated as model2
#>
#> model1 model2
#>  0.455  0.545
#>
#> Posterior probabilities when model2 is estimated as model1
#>
#> model1 model2
#>   0.54   0.46
```

The `error_modelSel` function outputs the confusion matrix and the mean model posterior probabilities obtained with the regression method. It will also output other useful information such as the mean posterior probability of correctly assigned models and the mean posterior probability when each model is not correctly assigned. For a more visual interpretation of these results, it is also possible to display a barplot of the model misclassification. By using the `plot_msel` function we can plot the confusion matrix, either in colour (if `color = TRUE`) or in grey (if `color = FALSE`).

```
# display a barplot of model misclassification
plot_msel(object = mselError)
```



Using the output of the `error_modelSel` function as the input of the `plot_msel` function, we can produce this barplot showing the proportion of simulations classified to any of the models.

4.8.4 Manual for the Package 'poolABC'

Title Approximate Bayesian Computation with Pooled Sequencing Data

Version 1.0.0

Description Provides functions to simulate Pool-seq data under models of demographic formation and to import Pool-seq data from real populations. Implements two ABC algorithms for performing parameter estimation and model selection using Pool-seq data. Cross-validation can also be performed to assess the accuracy of ABC estimates and model choice. Carvalho et al., (2022) doi:10.1111/1755-0998.13834 .

License GPL (= 3)

Encoding UTF-8

RoxygenNote 7.2.1

Imports doParallel, foreach, ggplot2, graphics, locfit, MetricsWeighted, nnet, poolHelper (= 1.1.0), RColorBrewer, rlang, scrm, stats, utils

Depends R (= 2.10)

LazyData true

URL https://github.com/joao-mcarvalho/poolABC

BugReports https://github.com/joao-mcarvalho/poolABC/issues

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Jo o Carvalho [aut, cre, cph] (https://orcid.org/0000-0002-1728-0075), V tor Sousa [aut]

Maintainer Jo o Carvalho <jgcarvalho@fc.ul.pt>

Repository CRAN

Date Publication 2023-08-08 14:00:02 UTC

R topics documented:

ABC — *Parameter estimation with Approximate Bayesian Computation with several targets*

Description

Perform multivariate parameter estimation based on summary statistics using an Approximate Bayesian Computation (ABC) algorithm. This function always uses a rejection sampling algorithm while a local linear regression algorithm might or might not be used.

Usage

```
ABC(
  nPops,
  ntrials,
  freqs,
  positions,
  range,
  rMajor,
  rMinor,
  coverage,
  window,
  nLoci,
  limits,
  params,
  sumstats,
  tol,
  method,
  parallel = FALSE,
  ncores = NA
)
```

Arguments

nPops	is an integer indicating how many different populations are present in the dataset you are analysing.
ntrials	indicates how many different trials should be performed. Each trial corresponds to a different target for the parameter estimation.
freqs	is a list containing the allelic frequencies. Each entry of that list should represent a different contig and be a matrix where each row corresponds to a different site and each column to a different population.
positions	is a list containing the position of the SNPs. Each entry should represent a different contig and be a vector containing the position of each SNP present in the contig.

range	is a list containing the range of the contig. Each entry should represent a different contig and be a vector with two entries: the first detailing the minimum position of the contig and the second the maximum position of the contig.
rMajor	a list containing the number of major allele reads. Each entry should represent a different contig. For each contig (matrix), each row should be a different site and each column a different population.
rMinor	a list containing the number of minor allele reads. Each entry should represent a different contig. For each contig (matrix), each row should be a different site and each column a different population.
coverage	is a list containing the depth of coverage. Each entry should represent a different contig and be a matrix with the sites as rows and the different populations as columns.
window	is a non-negative integer indicating the size, in base pairs, of the block of the contig to keep.
nLoci	is a non-negative integer indicating how many different contigs should be kept in the output. If each randomly selected window is a different loci, then how many different window should be selected?
limits	is a matrix with two columns and as many rows as there are parameters. Each row should contain the minimum value of the prior for a given parameter in the first column and the maximum value in the second column.
params	is a vector or matrix of simulated parameter values i.e. numbers from the simulations. Each row or vector entry should be a different simulation and each column of a matrix should be a different parameter. This is the dependent variable for the regression, if a regression step is performed.
sumstats	is a vector or matrix of simulated summary statistics. Each row or vector entry should be a different simulation and each column of a matrix should be a different statistic. These act as the independent variables if a regression step is performed.
tol	is the tolerance rate, indicating the required proportion of points accepted nearest the target values.
method	either rejection or regression indicating whether a regression step should be performed during ABC parameter estimation.
parallel	logical, indicating whether this function should be run using parallel execution. The default setting is FALSE, meaning that this function will utilize a single core.
ncores	a non-negative integer that is required when parallel is TRUE. It specifies the number of cores to use for parallel execution.

Details

To use this function, the usual steps of ABC parameter estimation have to be performed. Briefly, data should have been simulated based on random draws from the prior distributions of the parameters of interest and a set of summary statistics should have been calculated from that data. This function requires as input the observed data and computes the same set of summary statistics from that observed data. Multiple sets of observed summary statistics are computed from ntrials sets

of nLoci blocks of size window. Parameter estimation is performed for each one of those sets of observed summary statistics i.e. each set corresponds to a different target.

After computing this set of observed summary statistics, a simple rejection is performed by calling the rejABC() function. In this step, parameter values are accepted if the Euclidean distance between the set of summary statistics computed from the simulated data and the set of summary statistics computed from the observed data is sufficiently small. The percentage of accepted simulations is determined by tol.

When method is regression , a local linear regression method is used to correct for the imperfect match between the summary statistics computed from the simulated data and the summary statistics computed from the observed data. The output of the rejABC() function is used as the input of the regABC() function to apply this correction. The parameter values accepted in the rejection step are weighted by a smooth function (kernel) of the distance between the simulated and observed summary statistics and corrected according to a linear transformation.

Value

a list with seven different entries.

target	observed summary statistics.
ss	set of accepted summary statistics from the simulations.
unadjusted	parameter estimates obtained with the rejection sampling.
adjusted	regression adjusted parameter values.
predmean	estimates of the posterior mean for each parameter.
weights	regression weights.
position	position of each SNP used for calculating the observed summary statistics.

See Also

For more details see the poolABC vignette: vignette("poolABC", package = "poolABC")

Examples

```
# Note that this example is limited to a few of the options available
# you should check the poolABC vignette for more details

# this creates a variable with the path for the toy example data
mypath <- system.file('extdata', package = 'poolABC')

# import data for two populations from all files
mydata <- importContigs(path = mypath, pops = c(8, 10))

# to perform parameter inference for two populations using the rejection method
# and with a tolerance of 0.01
myabc <- ABC(nPops = 2, ntrials = 10, freqs = mydata$freqs, positions = mydata$positions,
range = mydata$range, rMajor = mydata$rMajor, rMinor = mydata$rMinor, coverage = mydata$coverage,
window = 1000, nLoci = 4, limits, params, sumstats, tol = 0.01, method = "rejection")

# the previous will perform parameter inference for 10 different targets (ntrials = 100)
```

```
# each of those trials will be comprised of 4 loci, each with 1000 base pairs

# to perform parameter inference for two populations using the regression method
# and with a tolerance of 0.01
myabc <- ABC(nPops = 2, ntrials = 10, freqs = mydata$freqs, positions = mydata$positions,
range = mydata$range, rMajor = mydata$rMajor, rMinor = mydata$rMinor, coverage = mydata$coverage,
window = 1000, nLoci = 4, limits, params, sumstats, tol = 0.01, method = "regression")
```

`cleanData` *Import and clean a single file containing data in* `popoolation2` *format*

Description

Imports data for two or four populations from a single file containing data in the rc format. The data is then split so that the number of major-allele reads, minor-allele reads, total depth of coverage and remaining relevant information are kept on separate matrices.

Usage

```
cleanData(file, pops, header = NA, remove = NA, min.minor = NA)
```

Arguments

`file`	is a character string indicating the path to the file you wish to import.
`pops`	is a vector with the index of the populations that should be imported. This function works for two or four populations and so this vector must have either length 2 or 4.
`header`	is a character vector containing the names for the columns. If set to NA (default), no column names will be added to the output.
`remove`	is a character vector where each entry is a name of a contig to be removed. These contigs are, obviously, removed from the imported dataset. If NA (default), all contigs will be kept in the output.
`min.minor`	what is the minimum allowed number of reads with the minor allele across all populations? Sites where this threshold is not met are removed from the data. The default (NA) means that no sites will be removed because of their number of minor-allele reads.

Details

The information in the rc format is stored in a x/y format, where x represents the observed reads and the y is the coverage. The initial step of this function splits this string to separate the number of reads from the total coverage. Then, the number of major plus minor allele reads is compared to the total coverage and sites where both values are not equal are removed from the dataset. Additionally, sites where any of the populations has an N as the reference character of their major allele, are removed from the data. This function also ensures that the major allele is the same and the most

frequent across all populations. Finally, if the `min.minor` input is supplied, sites where the total number of minor-allele reads is below the specified number, will be removed from the data set.

Note also that all non biallelic sites and sites where the sum of deletions in all populations is not zero will be removed from the dataset. Although this function can only import 2 or 4 populations at the time, it is possible to define which two or four populations to import. For instance, if we define the first population as the first column for which we have data in the x/y format, then you could wish to import the data for the 5th and 6th populations, defined as the populations in the 6th and 7th columns. To do so, you should define the `pops` input as `pops = c(5, 6)`.

Value

a list with the following elements:

`rMajor`	a matrix with the number of major-allele reads. Each row of this matrix is a different site and each column a different population.
`rMinor`	a matrix with the number of minor-allele reads. Each row of this matrix is a different site and each column a different population.
`coverage`	a matrix with the total coverage. Each row of this matrix is a different site and each column a different population.
`info`	a data frame with 5 different columns containing: the contig name, the SNP position, the reference character of the SNP and the reference character of the major and minor allele for each of the populations. Each row of this data frame corresponds to a different site

Examples

```
# load the data from one rc file
data(rc1)
# clean and organize the data in this single file
cleanData(file = rc1, pops = 7:10)
```

`cmd2pops` *Create SCRM command line for a model with two populations*

Description

This function creates a command line tailored for an isolation with migration model with two populations. The command line can then be fed to the scrm package to run the model.

Usage

```
cmd2pops(parameters, nSites, nLoci, nDip, mutrate, extra = FALSE)
```

Arguments

parameters	A vector where each entry corresponds to a different parameter, e.g. one entry is the size of the reference population, another is the time of recent split, etc. Please note that this functions depends on the ordering of the parameters in the vector and thus, it should only be used with a vector created with the createParams function.
nSites	An integer representing the number of base pairs that each locus should have.
nLoci	An integer that represents how many independent loci should be simulated.
nDip	An integer representing the total number of diploid individuals to simulate. Note that scrm actually simulates haplotypes, so the number of simulated haplotypes is double of this. Also note that this is the total number of diploid individuals and this function will distribute the individuals equally by the two populations.
mutrate	A number representing the mutation rate assumed for the simulations.
extra	is a logical value indicating whether the required number of loci should be enforced. The default is FALSE but, if set to TRUE, then additional loci will be simulated. These additional loci are simulated to try to have sufficient loci to keep the required number of loci after filtering.

Value

a character vector with two entries. The first entry is the scrm command line for the loci without any barriers against migration, while the second entry is the scrm command line for the loci without migration between divergent ecotypes.

Examples

```
# create a vector with parameter values for a two populations model
params <- createParams(Nref = c(25000, 25000), ratio = c(0.1, 3), pool = c(5, 250),
seq = c(0.0001, 0.001), split = c(0, 3), CW = c(1e-13, 1e-3), WC = c(1e-13, 1e-3),
bT = c(0, 0.2), model = "2pops")

# create the command line for the scrm package
cmd2pops(parameters = params, nSites = 2000, nLoci = 100, nDip = 100, mutrate = 2e-8)
```

---

cmdParallel *Create SCRM command line for a parallel origin scenario*

Description

This function creates a command line tailored for a scenario of parallel origin to explain ecotype formation. The command line can then be fed to the scrm package to run the model.

Usage

```
cmdParallel(parameters, nSites, nLoci, nDip, mutrate, extra = FALSE)
```

Arguments

parameters	A vector where each entry corresponds to a different parameter, e.g. one entry is the size of the reference population, another is the time of recent split, etc. Please note that this functions depends on the ordering of the parameters in the vector and thus, it should only be used with a vector created with the createParams function.
nSites	An integer representing the number of base pairs that each locus should have.
nLoci	An integer that represents how many independent loci should be simulated.
nDip	An integer representing the total number of diploid individuals to simulate. Note that scrm actually simulates haplotypes, so the number of simulated haplotypes is double of this. Also note that this is the total number of diploid individuals and this function will distribute the individuals equally by the two populations.
mutrate	A number representing the mutation rate assumed for the simulations.
extra	is a logical value indicating whether the required number of loci should be enforced. The default is FALSE but, if set to TRUE, then additional loci will be simulated. These additional loci are simulated to try to have sufficient loci to keep the required number of loci after filtering.

Details

For convenience, imagine we have two divergent ecotypes, named C and W. This model assumes that the first population corresponds to the C ecotype at the first location, the second population to the W ecotype in the first location, the third population to the C ecotype in the second location and the fourth population to the W ecotype in the second location.

Value

a character vector with four entries. The first entry is the scrm command line for the loci without any barriers against migration. The second entry is the command line for the loci without migration from the C towards the W ecotype. The third entry is command line for the loci without migration from the W towards the C ecotype and the last entry is the scrm command line for the loci without migration between divergent ecotypes.

Examples

```
# create a vector with parameter values for the parallel origin scenario
params <- createParams(Nref = c(25000, 25000), ratio = c(0.1, 3), pool = c(5, 250),
seq = c(0.0001, 0.001), split = c(0, 3), CW = c(1e-13, 1e-3), WC = c(1e-13, 1e-3),
CC =  c(1e-13, 1e-3), WW = c(1e-13, 1e-3), ANC = c(1e-13, 1e-3), bT = c(0, 0.2),
bCW = c(0, 0.5), bWC = c(0, 0.5), model = "Parallel")

# create the command line for the scrm package
cmdParallel(parameters = params, nSites = 2000, nLoci = 100, nDip = 400, mutrate = 2-8)
```

`cmdSingle` *Create SCRM command line for a single origin scenario*

Description

This function creates a command line tailored for a scenario of single origin to explain ecotype formation. The command line can then be fed to the scrm package to run the model.

Usage

```
cmdSingle(parameters, nSites, nLoci, nDip, mutrate, extra = FALSE)
```

Arguments

`parameters`	A vector where each entry corresponds to a different parameter, e.g. one entry is the size of the reference population, another is the time of recent split, etc. Please note that this functions depends on the ordering of the parameters in the vector and thus, it should only be used with a vector created with the `createParams` function.
`nSites`	An integer representing the number of base pairs that each locus should have.
`nLoci`	An integer that represents how many independent loci should be simulated.
`nDip`	An integer representing the total number of diploid individuals to simulate. Note that scrm actually simulates haplotypes, so the number of simulated haplotypes is double of this. Also note that this is the total number of diploid individuals and this function will distribute the individuals equally by the two populations.
`mutrate`	A number representing the mutation rate assumed for the simulations.
`extra`	is a logical value indicating whether the required number of loci should be enforced. The default is FALSE but, if set to TRUE, then additional loci will be simulated. These additional loci are simulated to try to have sufficient loci to keep the required number of loci after filtering.

Details

For convenience, imagine we have two divergent ecotypes, named C and W. This model assumes that the first population corresponds to the C ecotype at the first location, the second population to the C ecotype in the second location, the third population to the W ecotype in the first location and the fourth population to the W ecotype in the second location.

Value

a character vector with four entries. The first entry is the scrm command line for the loci without any barriers against migration. The second entry is the command line for the loci without migration from the C towards the W ecotype. The third entry is command line for the loci without migration from the W towards the C ecotype and the last entry is the scrm command line for the loci without migration between divergent ecotypes.

Examples

```
# create a vector with parameter values for the single origin scenario
params <- createParams(Nref = c(25000, 25000), ratio = c(0.1, 3), pool = c(5, 250),
seq = c(0.0001, 0.001), split = c(0, 3), CW = c(1e-13, 1e-3), WC = c(1e-13, 1e-3),
CC =  c(1e-13, 1e-3), WW = c(1e-13, 1e-3), ANC = c(1e-13, 1e-3), bT = c(0, 0.2),
bCW = c(0, 0.5), bWC = c(0, 0.5), model = "Single")

# create the command line for the scrm package
cmdSingle(parameters = params, nSites = 2000, nLoci = 100, nDip = 400, mutrate = 2-8)
```

---

createHeader *Create a header for a rc file of popoolation2*

Description

Creates a header for files in the rc format of the popoolation2 software. This header can be applied to a matrix as column names.

Usage

```
createHeader(nPops)
```

Arguments

nPops	is an integer specifying how many different populations exist in the rc file.

Details

Please note that the first 9 columns are a default output of the popoolation2 software and thus this functions maintains the same names.

Value

a character vector with the column names for a rc popoolation2 file.

Examples

```
createHeader(nPops = 10)
```

`createParams` *Draw parameters from the priors*

Description

This function creates a named vector of parameters that can be used as input in the command line of the scrm package. Please note that this function needs to be adjusted if you wish to test the effect of different prior distributions.

Usage

```
createParams(
  Nref,
  ratio,
  split,
  pool,
  seq,
  CW,
  WC,
  CC = NA,
  WW = NA,
  ANC = NA,
  bT,
  bCW = NA,
  bWC = NA,
  model,
  digits = 5
)
```

Arguments

`Nref`	The minimum and maximum value of the uniform distribution for the effective population size of the reference population (Nref).
`ratio`	The minimum and maximum value of the distribution from which the relative size of the present-day and ancestral populations are drawn. The size of these populations is set as a ratio of the size of the Nref population. All of these ratios are drawn from a log10 uniform distribution.
`split`	The minimum and maximum values, at the 4Nref scale, of the uniform distribution from which the values of the times of the split events are draw. Both the time of the recent split event and the distance between the two split events are drawn from this distribution.
`pool`	The minimum and maximum values of the uniform distribution from which the value of the error associated with DNA pooling is drawn. More specifically, this value is related with the unequal individual contribution to the pool.

seq	The minimum and maximum values of the uniform distribution from which the value of the error associated with DNA sequencing is drawn. This parameter should be supplied as a decimal number between zero and one.
CW	The minimum and maximum value of the uniform distribution from which the migration rate between the two divergent ecotypes inhabiting the same location is drawn. We consider that this parameter is drawn on a m scale. This is the migration rate from ecotype C to ecotype W.
WC	The minimum and maximum value of the uniform distribution from which the migration rate between the two divergent ecotypes inhabiting the same location is drawn. We consider that this parameter is drawn on a m scale. This is the migration rate from ecotype W to ecotype C.
CC	The minimum and maximum value of the uniform distribution from which the migration rate between similar ecotypes inhabiting different locations is drawn. We consider that this parameter is drawn on a m scale. This is the migration between the two C ecotypes at two different locations.
WW	The minimum and maximum value of the uniform distribution from which the migration rate between similar ecotypes inhabiting different locations is drawn. We consider that this parameter is drawn on a m scale. This is the migration between the two W ecotypes at two different locations.
ANC	The minimum and maximum value of the uniform distribution from which the migration rate between the two ancestral populations is drawn. We consider that this parameter is drawn on a m scale.
bT	The minimum and maximum values of the distribution from which the proportion of the simulated loci where no migration occurs between divergent ecotypes is drawn. The maximum value should not be higher than one.
bCW	The minimum and maximum values of the distribution from which the proportion of the simulated loci where no migration occurs from the C ecotype towards the W ecotype is drawn. The maximum value should not be higher than one.
bWC	The minimum and maximum values of the distribution from which the proportion of the simulated loci where no migration occurs from the W ecotype towards the C ecotype is drawn. The maximum value should not be higher than one.
model	Either 2pops , Single or Parallel indicating for which model should parameters be drawn.
digits	An optional integer indicating the number of decimal places to use when rounding certain parameters. The default is five.

Value

a vector with one named entry per relevant parameter. Each entry is the sampled value from the prior for that particular parameter.

Examples

```
# for a model with two populations
createParams(Nref = c(25000, 25000), ratio = c(0.1, 3), pool = c(5, 250), seq = c(0.0001, 0.001),
split = c(0, 3), CW = c(1e-13, 1e-3), WC = c(1e-13, 1e-3), bT = c(0, 0.2), model = "2pops")
```

```
# for a single origin scenario
createParams(Nref = c(25000, 25000), ratio = c(0.1, 3), pool = c(5, 250), seq = c(0.0001, 0.001),
split = c(0, 3), CW = c(1e-13, 1e-3), WC = c(1e-13, 1e-3), CC =  c(1e-13, 1e-3),
WW = c(1e-13, 1e-3), ANC = c(1e-13, 1e-3), bT = c(0, 0.2), bCW = c(0, 0.5),
bWC = c(0, 0.5), model = "Single")
```

`error_modelSel` *Compute error in model selection with Approximate Bayesian Computation*

Description

This function calculates the confusion matrix and the mean misclassification probabilities of models from the output of the `sim_modelSel()` function.

Usage

```
error_modelSel(object, threshold = NA, print = TRUE)
```

Arguments

`object`	a list created by the `sim_modelSel()` function, containing results of a simulation study to evaluate the quality of model selection with Approximate Bayesian Computation.
`threshold`	numeric value between 0 and 1 representing the minimum posterior probability of assignment.
`print`	logical, if TRUE (default), then this function prints the mean models probabilities.

Details

It is also possible to define a `threshold` for the posterior model probabilities. This threshold sets the minimum posterior probability of assignment. Thus, a simulation where the posterior probability of any model is below the threshold will not be assigned to a model and will instead be classified as unclear .

Value

apart from directly displaying the results if print is TRUE, the output object of this function is a list with the following elements:

`confusion.matrix`	the confusion matrix.
`probs`	the mean model misclassification probabilities.
`postmeans`	the mean model misclassification probabilities when each model is correctly or incorrectly estimated.

Examples

```
# load the matrix with simulated parameter values
data(sumstats)

# select a random simulation to act as target just to test the function
target <- sumstats[10 ,]

# create a "fake" vector of model indices
# this assumes that half the simulations were from one model and the other half from other model
# this is not true but serves as an example of how to use this function
index <- c(rep("model1", nrow(sumstats)/2), rep("model2", nrow(sumstats)/2))

# perform a leave-one-out cross validation of model selection
mysim <- sim_modelSel(index = index, sumstats = sumstats, nval = 10, tol = 0.1)

# compute the confusion matrix and the mean misclassification probabilities
error_modelSel(object = mysim)
```

---

forceLocus *Force the simulations to contain the required number of loci*

Description

This function attempts to force the required number of loci after the filtering steps are performed.

Usage

```
forceLocus(
  model,
  parameters,
  nSites,
  nLoci,
  nDip,
  mutrate,
  mean,
  variance,
  minimum,
  maximum,
  size,
  min.minor
)
```

Arguments

model	a character, either 2pops , Single or Parallel indicating which model should be simulated.

`parameters` a vector of parameters used to create the command line for the scrm package. Each entry of the vector is a different parameter. Note that each vector entry should be named with the name of the corresponding parameter. The output of the `CreateParameters` function is the intended input.

`nSites` is an integer that specifies how many base pairs should scrm simulate, i.e. how many sites per locus to simulate.

`nLoci` an integer that represents how many independent loci should be simulated.

`nDip` an integer representing the total number of diploid individuals to simulate. Note that scrm actually simulates haplotypes, so the number of simulated haplotypes is double of this. Also note that this is the total number of diploid individuals and this function will distribute the individuals equally by the simulated populations.

`mutrate` an integer representing the mutation rate assumed for the simulations.

`mean` an integer or a vector defining the mean value of the negative binomial distribution from which different number of reads are drawn. It represents the mean coverage across all sites. If a vector is supplied, the function assumes that each entry of the vector is the mean for a different population.

`variance` an integer or a vector defining the variance of the negative binomial distribution from which different number of reads are drawn. It represents the variance of the total coverage across all sites. If a vector is supplied, the function assumes that each entry of the vector is the variance for a different population.

`minimum` an integer representing the minimum coverage allowed. Sites where any population has a depth of coverage below this threshold are removed from the data.

`maximum` an integer representing the maximum coverage allowed. Sites where any population has a depth of coverage above this threshold are removed from the data.

`size` a list with one entry per population. Each entry should be a vector containing the size (in number of diploid individuals) of each pool. Thus, if a population was sequenced using a single pool, the vector should contain only one entry. If a population was sequenced using two pools, each with 10 individuals, this vector should contain two entries and both will be 10.

`min.minor` is an integer representing the minimum allowed number of minor-allele reads. Sites that, across all populations, have less minor-allele reads than this threshold will be removed from the data.

Details

This is done by simulating extra loci for each of the different types of simulations performed. The possible types of simulations include loci without barriers against migration between divergent ecotypes, loci without migration from the C towards the W ecotype, loci without migration from the W towards the C ecotypes and loci where no migration occurs between divergent ecotypes. Using this function, more loci than required are simulated for each of those types of simulations.

Then, a coverage-based filter is applied to the data, followed by a filter based on a required number of minor-allele reads per site. Those filters remove some loci from the data. The extra simulated loci should allow us to keep the required number of loci per type of simulation even after filtering.

Value

a list with two names entries

pool	a list with three different entries: major, minor and total. This list is obtained by running the forcePool function.
nPoly	a numeric value indicating the mean number of polymorphic sites across all simulated locus.

Examples

```
# create a vector with parameter values for a two populations model
params <- createParams(Nref = c(25000, 25000), ratio = c(0.1, 3), pool = c(5, 250),
seq = c(0.0001, 0.001), split = c(0, 3), CW = c(1e-13, 1e-3), WC = c(1e-13, 1e-3),
bT = c(0, 0.2), model = "2pops")

# simulate exactly 10 loci - using an isolation with migration model with two populations
forceLocus(model = "2pops", parameters = params, nSites = 1000, nLoci = 10, nDip = 100,
mutrate = 2e-8, mean = c(100, 100), variance = c(250, 250), minimum = 10, maximum = 200,
size = list(50, 50), min.minor = 0)
```

---

getmode *Calculate the mode of a distribution*

Description

Computes and outputs the mode of the input distribution.

Usage

```
getmode(x, xlim, weights = NULL, alpha = 0.7, precision = 1000)
```

Arguments

x	is a numeric vector containing the values of the distribution.
xlim	is a vector with two entries.The first entry is the minimum of the x distribution and the second entry is the maximum value of the x distribution. Ideally these values should be the minimum and maximum value of the prior for this particular parameter.
weights	this is an optional input consisting of a vector with the prior weights for the locfit function.
alpha	numeric value with the alpha parameter of the locfit function. The default value is 0.7
precision	value indicating the number of entries evaluated. The larger the value the higher the precision. The default value is 1000.

Details

The `locfit::locfit()` function is used to fit a local regression to the distribution. The `stats::predict()` function is then used to predict the y-axis values of the locfit and the mode is defined as the value where that prediction is maximized. Note that if this function is not able to fit a local regression to the distribution, then the mode of the distribution will be assumed to be equal to the median.

Value

a numeric value of the mode of the input distribution.

Examples

```
# create a random distribution
x <- rnorm(n = 100, mean = 2, sd = 25)

# compute the mode of the distribution
getmode(x = x, xlim = c(min(x), max(x)))
```

---

importContigs *Import multiple files containing data in PoPoolation2 format*

Description

Imports multiple files containing data in PoPoolation2 format and organize that information into different entries for each contig.

Usage

```
importContigs(
  path,
  pops,
  files = NA,
  header = NA,
  remove = NA,
  min.minor = NA,
  filter = FALSE,
  threshold = NA
)
```

Arguments

`path`	is a character string indicating the path to the folder where the data you wish to import is located.
`pops`	is a vector with the index of the populations that should be imported. This function works for two or four populations and so this vector must have either length 2 or 4.

files	is an integer or a numeric vector with the index of the files you wish to import.
header	is a character vector containing the names for the columns. If set to NA (default), no column names will be added to the output.
remove	is a character vector where each entry is a name of a contig to be removed. These contigs are, obviously, removed from the imported dataset. If NA (default), all contigs will be kept in the output.
min.minor	what is the minimum allowed number of reads with the minor allele across all populations? Sites where this threshold is not met are removed from the data.
filter	is a logical switch, either TRUE or FALSE. If TRUE, then the data is filtered by the frequency of the minor allele and if FALSE, that filter is not applied.
threshold	is the minimum allowed frequency for the minor allele. Sites where the allelic frequency is below this threshold are removed from the data.

Details

The data from two or four populations is split so that the number of major-allele reads, minor-allele reads, total depth of coverage and remaining relevant information are kept on separate list entries. Sites where the sum of the major and minor allele reads does not match the total coverage and sites where any population has an N as the reference character of their major allele, are removed from the data. This function also ensures that the major allele is the same and the most frequent across all populations. Note also that all non biallelic sites and sites where the sum of deletions in all populations is not zero will be removed from the dataset.

If the min.minor input is supplied, sites where the total number of minor-allele reads is below the specified number, will be removed from the data set. Alternatively, if the filter input is set to TRUE, data will be filtered by the frequency of the minor-allele. If a threshold is supplied, the computed frequency is compared to that threshold and sites where the frequency is below the threshold are removed from the dataset. If no threshold is supplied, the threshold is assumed to be 1/total coverage, meaning that a site should have, at least, one minor-allele read.

Finally, the name of each contig is used to organize the information in a per contig basis. Thus, each output will be organized by contig. For example, the list with the number of minor-allele reads will contain several entries and each of those entries is a different contig.

Value

a list with six named entries:

freqs	a list with the allele frequencies, computed by dividing the number of minor-allele reads by the total coverage. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.
positions	a list with the positions of each SNP. Each entry of this list is a vector corresponding to a different contig.
range	a list with the minimum and maximum SNP position of each contig. Each entry of this list is a vector corresponding to a different contig.
rMajor	a list with the number of major-allele reads. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.

rMinor	a list with the number of minor-allele reads. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.
coverage	a list with the total coverage. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.

See Also

For more details see the poolABC vignette: `vignette("poolABC", package = "poolABC")`

Examples

```
# this function should be used to import your data
# you should include the path to the folder your PoPoolation2 data is

# this creates a variable with the path for the toy example data
mypath <- system.file('extdata', package = 'poolABC')

# an example of how to import data for two populations from all files
importContigs(path = mypath, pops = c(8, 10))

# to remove contigs from the data
importContigs(path = mypath, pops = c(8, 10), remove = "Contig1708")
```

---

index.rejABC — *Parameter estimation with Approximate Bayesian Computation using rejection sampling and recording just the index of accepted simulations*

Description

This function performs multivariate parameter estimation based on summary statistics using an Approximate Bayesian Computation (ABC) algorithm. The algorithm used here is the rejection sampling algorithm. This is a simplified version of the `rejABC()` function that records only the index of the accepted simulations.

Usage

```
index.rejABC(target, params, sumstats, tol)
```

Arguments

target	a vector with the target summary statistics. These are usually the set of observed summary statistics.
params	is a vector or matrix of simulated parameter values i.e. numbers from the simulations. Each row or vector entry should be a different simulation and each column of a matrix should be a different parameter.

sumstats	is a vector or matrix of simulated summary statistics. Each row or vector entry should be a different simulation and each column of a matrix should be a different statistic.
tol	is the tolerance rate, indicating the required proportion of points accepted nearest the target values.

Details

The rejection sampling algorithm generates random samples from the posterior distributions of the parameters of interest. Note that to use this function, the usual steps of ABC parameter estimation have to be performed. Briefly, data should have been simulated based on random draws from the prior distributions of the parameters of interest and a set of summary statistics should have been calculated from that data. The same set of summary statistics should have been calculated from the observed data to be used as the `target` input in this function. Parameter values are accepted if the Euclidean distance between the set of summary statistics computed from the simulated data and the set of summary statistics computed from the observed data is sufficiently small. The percentage of accepted simulations is determined by `tol`.

Value

a list with two named entries

index	the index of the accepted simulations.
dst	euclidean distances in the region of interest.

Examples

```
# load the matrix with parameter values
data(params)
# load the matrix with simulated parameter values
data(sumstats)

# select a random simulation to act as target just to test the function
target <- sumstats[10 ,]

# Parameter estimation using rejection sampling
index.rejABC(target = target, params = params, sumstats = sumstats[-10, ], tol = 0.01)
```

---

limits *Matrix of prior limits*

Description

this imports a matrix with the limits of the prior distribution for each parameter. Each row of the matrix is a different parameter, indicated by the row name. The matrix contains two columns, the first being the minimum value of the distribution and the second being the maximum value.

Usage

```
limits
```

Format

a matrix with 8 rows and 2 columns. Each of the rows corresponds to a different parameter:

N1 relative size of the first population. This population corresponds to the C ecotype.

N2 relative size of the second population. This population corresponds to the W ecotype.

Split time, in 4Nref scale, of the split event that creates the two populations.

PoolError error associated with DNA pooling.

SeqError error associated with DNA sequencing.

pM proportion of the genome with no barriers against gene flow. This is the proportion of simulated loci where migration occurs in both directions between the divergent ecotypes.

mig CW scaled migration rate between the two divergent ecotypes This is the migration rate from ecotype C to ecotype W.

mig WC scaled migration rate between the two divergent ecotypes This is the migration rate from ecotype W to ecotype C.

Source

simulations performed

---

mergepost *Merge posterior distributions*

Description

After using the `multipleABC()` function to perform parameter estimation with Approximate Bayesian Computation for several targets, this function can be used to merge the different posterior distributions.

Usage

```
mergepost(target, global, post, a = 0.5, wtreg = NULL)
```

Arguments

`target`	a matrix or a list with target mean sumstat, where each entry corresponds to a vector of size n (n = number of summary statistics) with the summary statistics of each subset of loci.
`global`	numeric vector of size n with mean summary statistics across all loci.
`post`	list with sample of posterior obtained for each subset of loci. Each entry of the list is a matrix where each line corresponds to an accepted simulations (size S) and each column corresponds to a parameter.

a	numeric value with the alpha parameter of the locfit function.
wtreg	(optional) list with the weights of regression method. Each entry of the list is a numeric vector with weights for each accepted simulation (size S).

Details

The posterior density will be estimated after simply merging the posteriors computed from all target subset of loci and after weighting the posterior of each target by its distance to the overall summary statistic mean. In other words, each posterior will be weighted according to the distance between the mean summary statistics of the subset of loci for which that posterior was computed and the mean across all loci, giving more weight to sets of loci with a mean closer to the overall mean.

Additionally, if the regression weights are available, each accepted point will be weighted by its regression weight and by distance of its associated target. The combination of these weights will be used to merge the multiple posteriors. The weighted mean, median, mode and quantiles will be computed for each of these different posterior merging methods by using the `weighted_stats()` and `mode_locfit()` functions. Note that this function requires the package **locfit**.

Value

list of locfit objects with the density of the posterior for each parameter and of mean, mode and quantiles obtained using weighted quantiles. The list has the following elements:

merge	obtained by simply merging all the posteriors into a single one and fitting a local regression without any prior weighting.
merged_stat	posterior point estimates for the corresponding merging method, merge. This includes the median, mean, mode and various quantiles of the posterior.
weighted	each target was weighted by its distance to the global summary statistics mean, giving more weight to the target subset of loci with mean summary statistics closer to the mean across the genome.
weighted_stat	posterior point estimates for the corresponding merging method, weighted. This includes the median, mean, mode and various quantiles of the posterior.
merge_reg	each accepted point was weighted by its regression weight.
merge_reg_stat	posterior point estimates for the corresponding merging method, merge_reg. This includes the median, mean, mode and various quantiles of the posterior.
weighted_reg	each target was weighted according to its distance to the overall mean and each point was weighted by its regression weight.
weighted_reg_stat	posterior point estimates for the corresponding merging method, weighted_reg. This includes the median, mean, mode and various quantiles of the posterior.

Details about the output can be found at: https://aakinshin.net/posts/weighted-quantiles/ and https://www.rdocumentation.org/packages/reldist/versions/1.6-6/topics/wtd.quantile

Examples

```
# load the matrix with parameter values
data(params)
# load the matrix with simulated parameter values
data(sumstats)
# load the matrix with the prior limits
data(limits)

# select some random simulations to act as target just to test the function
targets <- sumstats[c(11:20) ,]
# we should remove those random simulation from the sumstats and params matrices
sumstats <- sumstats[-c(11:20), ]; params <- params[-c(11:20), ]

# parameter estimation for multiple targets
myabc <- multipleABC(targets = targets, params = params, sumstats = sumstats, limits = limits,
tol = 0.01, method = "regression")

# select a random simulation to act as the global value of the summary statistics
# ideally this should be computed from the entirety of the observed data
global <- sumstats[50, ]

# merge the posterior distributions obtained in the previous step
mergepost(target = targets, global = global, post = myabc$adjusted, wtreg = myabc$weights)
```

`modelSelect` *Perform model selection with Approximate Bayesian Computation*

Description

Estimates posterior model probabilities using Approximate Bayesian Computation (ABC).

Usage

```
modelSelect(target, index, sumstats, tol, method, warning = TRUE)
```

Arguments

`target`	is a vector with the target summary statistics. These are usually computed from observed data.
`index`	is a vector of model indices. This can be a a character vector of model names, repeated as many times as there are simulations for each model. This vector will be coerced to factor and it must have the same length as `nrow(sumstats)` to indicate which row of the `sumstats` matrix belongs to which model.
`sumstats`	is a vector or matrix containing the simulated summary statistics for all the models. Each row or vector entry should be a different simulation and each column of a matrix should be a different statistic. The order must be the same as the order of the models in the `index` vector.

tol	is a numerical value, indicating the required proportion of points nearest the target values (tolerance).
method	a character string, either rejection or regression , indicating which algorithm should be used for model selection.
warning	logical, if TRUE (default) warnings produced while running this function, mainly related with accepting simulations for just one of the models, will be displayed.

Details

Prior to using this function, simulations must have been performed under, at least, two different models. When method is rejection , the posterior posterior probability of a given model is approximated by the proportion of accepted simulations of that particular model. Note that this approximation is only valid if all models where, a priori, equally likely and if the number of simulations performed is the same for all models. When the method is set to regression , a multinomial logistic regression is used to estimate the posterior model probabilities. This multinomial regression is implemented in the multinom function.

Value

a list with the following elements:

method	the method used for model selection.
indices	a vector of model indices in the accepted region. In other words, this vector contains the name of the accepted model for each accepted point.
pred	a vector of model probabilities.
ss	the summary statistics in the accepted region.
weights	vector of regression weights when method is regression.
nmodels	the number of a priori simulations performed for each model.

Examples

```
# load the matrix with simulated parameter values
data(sumstats)

# select a random simulation to act as target just to test the function
target <- sumstats[10 ,]

# create a "fake" vector of model indices
# this assumes that half the simulations were from one model and the other half from other model
# this is not true but serves as an example of how to use this function
index <- c(rep("model1", nrow(sumstats)/2), rep("model2", nrow(sumstats)/2))

# perform model selection with ABC
modelSelect(target = target, index = index, sumstats = sumstats, tol = 0.01, method = "regression")
```

`mode_locfit` *Compute mode of a locfit object*

Description

This function computes and outputs the the mode of a locfit object.

Usage

```
mode_locfit(locx, xlim, precision = 1000)
```

Arguments

`locx`	is a locfit object.
`xlim`	is a vector with two entries.The first entry is the minimum of the distribution and the second entry is the maximum value of the distribution.
`precision`	value indicating the number of entries evaluated. The larger the value the higher the precision. The default value is 1000.

Details

The `stats::predict()` function is used to predict the y-axis values of the locfit object and the mode is defined as the value where that prediction is maximized.

Value

a numeric value of the mode of the input locfit object.

Examples

```
# create a random distribution
x <- rnorm(n = 1000, mean = 2, sd = 25)

# perform a local regression
loc <- locfit::locfit(~x)

# compute the mode of the locfit object
mode_locfit(locx = loc, xlim = c(min(x), max(x)))
```

multipleABC *Parameter estimation with Approximate Bayesian Computation for multiple targets*

Description

Perform multivariate parameter estimation based on summary statistics using an Approximate Bayesian Computation (ABC) algorithm. This function always uses a rejection sampling algorithm while a local linear regression algorithm might or might not be used.

Usage

```
multipleABC(
  targets,
  params,
  sumstats,
  limits,
  tol,
  method,
  parallel = FALSE,
  ncores = NA
)
```

Arguments

targets	a matrix of observed summary statistics. Each row will be considered a different target for parameter estimation. Each column should be a different summary statistics and these statistics should correspond to the statistics in the sumstats input.
params	is a vector or matrix of simulated parameter values i.e. numbers from the simulations. Each row or vector entry should be a different simulation and each column of a matrix should be a different parameter. This is the dependent variable for the regression, if a regression step is performed.
sumstats	is a vector or matrix of simulated summary statistics. Each row or vector entry should be a different simulation and each column of a matrix should be a different statistic. These act as the independent variables if a regression step is performed.
limits	is a matrix with two columns and as many rows as there are parameters. Each row should contain the minimum value of the prior for a given parameter in the first column and the maximum value in the second column.
tol	is the tolerance rate, indicating the required proportion of points accepted nearest the target values.
method	either rejection or regression indicating whether a regression step should be performed during ABC parameter estimation.

parallel	logical, indicating whether this function should be run using parallel execution. The default setting is FALSE, meaning that this function will utilize a single core.
ncores	a non-negative integer that is required when parallel is TRUE. It specifies the number of cores to use for parallel execution.

Details

To use this function, the usual steps of ABC parameter estimation have to be performed. Briefly, data should have been simulated based on random draws from the prior distributions of the parameters of interest and a set of summary statistics should have been calculated from that data. The same set of summary statistics should have been calculated from the observed data to be used as the targets in this function. Parameter values are accepted if the Euclidean distance between the set of summary statistics computed from the simulated data and the set of summary statistics computed from the observed data is sufficiently small. The percentage of accepted simulations is determined by tol. This function performs a simple rejection by calling the rejABC() function.

When method is regression , a local linear regression method is used to correct for the imperfect match between the summary statistics computed from the simulated data and the summary statistics computed from the observed data. The output of the rejABC() function is used as the input of the regABC() function to apply this correction. The parameter values accepted in the rejection step are weighted by a smooth function (kernel) of the distance between the simulated and observed summary statistics and corrected according to a linear transformation.

Please note that this functions performs parameter estimation for multiple targets. The targets should contain multiple rows and each row will be treated as an independent target for parameter estimation.

Value

the returned object is a list containing the following components:

target	parameter estimates obtained with the rejection sampling.
ss	set of accepted summary statistics from the simulations.
unadjusted	parameter estimates obtained with the rejection sampling.
adjusted	regression adjusted parameter values.
predmean	estimates of the posterior mean for each parameter.
weights	regression weights.

Examples

```
# load the matrix with parameter values
data(params)
# load the matrix with simulated parameter values
data(sumstats)
# load the matrix with the prior limits
data(limits)

# select some random simulations to act as target just to test the function
targets <- sumstats[c(11:20) ,]
```

```
# we should remove those random simulation from the sumstats and params matrices
sumstats <- sumstats[-c(11:20), ]; params <- params[-c(11:20), ]

# parameter estimation for multiple targets
multipleABC(targets = targets, params = params, sumstats = sumstats, limits = limits,
tol = 0.01, method = "regression")
```

`myparams` *Matrix of simulated parameter values*

Description

This data set contains a matrix of simulated parameter values. These parameter values were sampled from prior distributions and used to perform simulations under a isolation with migration model with two populations. Each row of this matrix corresponds to a different simulation.

Usage

```
myparams
```

Format

a matrix with 5000 rows and 8 columns:

N1 relative size of the first population. This population corresponds to the C ecotype.

N2 relative size of the second population. This population corresponds to the W ecotype.

Split time, in 4Nref scale, of the split event that creates the two populations.

PoolError error associated with DNA pooling.

SeqError error associated with DNA sequencing.

mCW migration rate between the two divergent ecotypes This is the migration rate from ecotype C to ecotype W.

mWC migration rate between the two divergent ecotypes This is the migration rate from ecotype W to ecotype C.

pM proportion of the genome with no barriers against gene flow. This is the proportion of simulated loci where migration occurs in both directions between the divergent ecotypes.

Source

simulations performed

params *Matrix of simulated parameter values*

Description

This data set contains a matrix of simulated parameter values. These parameter values were sampled from prior distributions and used to perform simulations under a isolation with migration model with two populations. Each row of this matrix corresponds to a different simulation.

Usage

```
params
```

Format

a matrix with 10000 rows and 8 columns:

N1 relative size of the first population. This population corresponds to the C ecotype.

N2 relative size of the second population. This population corresponds to the W ecotype.

Split time, in 4Nref scale, of the split event that creates the two populations.

PoolError error associated with DNA pooling.

SeqError error associated with DNA sequencing.

pM proportion of the genome with no barriers against gene flow. This is the proportion of simulated loci where migration occurs in both directions between the divergent ecotypes.

mig CW scaled migration rate between the two divergent ecotypes This is the migration rate from ecotype C to ecotype W.

mig WC scaled migration rate between the two divergent ecotypes This is the migration rate from ecotype W to ecotype C.

Source

simulations performed

plot_errorABC *Prediction error plots for ABC using a list*

Description

Plots the prediction error computed from a leave-one-out cross validation for ABC parameter inference. This function takes as input a list created when performing cross validation and allows the user to select which ABC algorithm and point estimate statistic to plot.

Usage

```
plot_errorABC(
  x,
  method,
  statistic,
  index,
  transformation = "none",
  main = NULL
)
```

Arguments

x	is a list produced by a leave-one-out cross validation of ABC. This list should contain the prediction errors computed using the rejection and/or regression algorithm. For each of those methods, the prediction error obtained using three different point estimates of the posterior should be included in this list.
method	a character that can be either 'rej' or 'reg' indicating whether you wish to plot the prediction error computed with a rejection or regression based ABC algorithm.
statistic	a character that can be 'mode', 'median' or 'mean' indicating if you wish to plot the prediction error obtained using the mode, median or mean as the point estimate of the posterior.
index	an integer indicating which parameter to look at. It corresponds to a column on a matrix. So, to plot the first parameter, corresponding to the first column, select 1. To plot the second parameter, select 2 and so on.
transformation	default is none. It can also be 'log' if you wish to transform both the true and estimated values using a log10 scale.
main	is an optional character input. It will be used as the title of the plot. If NULL (default), then a generic title will be used instead.

Details

These plots help in visualizing the quality of the estimation and the effect of the chosen tolerance level or point estimate statistic.

Value

a plot of the estimated value of the parameter (in the y-axis) versus the true parameter value (in the x-axis). A line marking the perfect correspondence between the true and estimated values is also plotted. Thus, the closer the points are to that line, the lower the prediction error is.

Examples

```
# load the matrix with parameter values
data(params)
# load the matrix with simulated parameter values
data(sumstats)
# load the matrix with the prior limits
data(limits)
```

```
# perform a leave-one-out cross validation for ABC
mysim <- simulationABC(params = params, sumstats = sumstats, limits, nval = 10,
tol = 0.1, method = "regression")

# plot the prediction error for a given parameter
plot_errorABC(x = mysim, method = "reg", statistic = "median", index = 1)
```

`plot_msel` *Plot model misclassification*

Description

Displays a barplot of the confusion matrix obtained with a leave-one-out cross validation for model selection.

Usage

```
plot_msel(object, color = TRUE)
```

Arguments

`object`	a list created by the `error_modelSel()` function, containing the results of a leave-one-out cross validation for model selection.
`color`	logical, if TRUE (default) then a colour version of the barplot will be produced, if FALSE then a grey scale version will be produced.

Details

The barplot shows the proportion of validation simulations classified to each of the models. This function can produce either a colour or a grey scale barplot. If the classification of models is perfect, meaning that the model probability of each model is one for the correct model, then each bar will have a single colour representing its corresponding model.

Value

a barplot of the proportion of simulations classified to any of the models. In other words, a barplot of the confusion matrix.

Examples

```
# load the matrix with simulated parameter values
data(sumstats)

# select a random simulation to act as target just to test the function
target <- sumstats[10 ,]

# create a "fake" vector of model indices
```

```
# this assumes that half the simulations were from one model and the other half from other model
# this is not true but serves as an example of how to use this function
index <- c(rep("model1", nrow(sumstats)/2), rep("model2", nrow(sumstats)/2))

# perform a leave-one-out cross validation of model selection
mysim <- sim_modelSel(index = index, sumstats = sumstats, nval = 10, tol = 0.1)

# compute the confusion matrix and the mean misclassification probabilities
myerror <- error_modelSel(object = mysim, print = FALSE)

# barplot of model misclassification
plot_msel(object = myerror)
```

`plot_param` *Plot the density estimation of a given parameter*

Description

Plots the density estimation of a single parameter for quick visualization of the quality of an ABC analysis.

Usage

```
plot_param(prior, posterior, limits, index, weights = NULL)
```

Arguments

`prior`	is a vector or matrix of simulated parameter values i.e. numbers from the simulations. Each row or vector entry should be a different simulation and each column of a matrix should be a different parameter. This corresponds to the prior distribution and it should contain all the simulated parameter values.
`posterior`	is either a list or a matrix with samples from the posterior distributions obtained for each target. If in list format, each entry should be a matrix where each row corresponds to a different accepted simulations and each column corresponds to a different parameter.
`limits`	is a matrix with two columns and as many rows as there are parameters. Each row should contain the minimum value of the prior for a given parameter in the first column and the maximum value in the second column.
`index`	is an non-negative integer indicating which parameter to plot. It corresponds to the desired column of a matrix in the `posteriors` input. So, to plot the first parameter, corresponding to the first column in the `posteriors` input select 1. To plot the second parameter, select 2 and so on.
`weights`	is an optional list input containing the weights from the local linear regression method. Each entry of the list should be a numeric vector with the weights for each accepted simulation.

Details

This function can be used for a quick visualization of the posterior distribution obtained for a single target with the singleABC() function. Alternatively, if parameter estimation was performed with the multipleABC() function, the multiple posterior distributions, each obtained for a different target, will be combined into a single matrix and all values will be considered samples from the same posterior distribution.

Value

a plot of the density estimation of a given parameter. This plot will include a title with the name of the parameter. It will also include the density of the prior distribution for that parameter.

Examples

```
# load the matrix with parameter values
data(params)
# load the matrix with simulated parameter values
data(sumstats)
# load the matrix with the prior limits
data(limits)

# select a random simulation to act as target just to test the function
target <- sumstats[15 ,]
# we should remove the random simulation from the sumstats and params matrices
sumstats <- sumstats[-15, ]; params <- params[-15, ]

# parameter estimation for a single target
myabc <- singleABC(target = target, params = params, sumstats = sumstats, limits = limits,
tol = 0.01, method = "regression")

# plot the density estimation of a given parameter
plot_param(prior = params, posterior = myabc$adjusted, limits = limits,
index = 6, weights = myabc$weights)

# note that this is just an example!
# we don't have enough simulations to obtain credible results
```

---

plot_Posteriors *Plot multiple posterior distributions*

Description

Plots, in the same plot, the density of multiple posterior distributions of a given parameter.

Usage

```
plot_Posteriors(posteriors, index, limits, weights = NULL)
```

Arguments

posteriors	is a list with samples from the posterior distributions obtained for each target. Each entry of the list is a matrix where each row corresponds to a different accepted simulations and each column corresponds to a different parameter.
index	an non-negative integer indicating which parameter to plot. It corresponds to the desired column of a matrix in the posteriors input. So, to plot the first parameter, corresponding to the first column in the posteriors input select 1. To plot the second parameter, select 2 and so on.
limits	is a matrix with two columns and as many rows as there are parameters. Each row should contain the minimum value of the prior for a given parameter in the first column and the maximum value in the second column.
weights	is an optional list input containing the weights from the local linear regression method. Each entry of the list should be a numeric vector with the weights for each accepted simulation.

Details

After using the multipleABC() or ABC() functions to perform parameter estimation with Approximate Bayesian Computation with several targets, this function can be used for a quick visualization of the quality of an ABC analysis. Multiple posterior distributions, each obtained for a different target, are plotted in the same plot, allowing for a visualization of the shape of the posteriors and a quick inspection of whether all the posteriors converge to the same estimate.

Value

a plot with multiple posterior distributions, each obtained for a different target, for the selected parameter.

Examples

```
# load the matrix with parameter values
data(params)
# load the matrix with simulated parameter values
data(sumstats)
# load the matrix with the prior limits
data(limits)

# select some random simulations to act as target just to test the function
targets <- sumstats[c(11:20) ,]
# we should remove those random simulation from the sumstats and params matrices
sumstats <- sumstats[-c(11:20), ]; params <- params[-c(11:20), ]

# parameter estimation for a single target
myabc <- multipleABC(targets = targets, params = params, sumstats = sumstats, limits = limits,
tol = 0.01, method = "regression")

# plot multiple posteriors
plot_Posteriors(posteriors = myabc$adjusted, index = 1, limits = limits, weights = myabc$weights)
```

```
# note that this is just an example!
# we don't have enough simulations to obtain credible results
```

`plot_stats` *Plot the fit of a summary statistic to the target*

Description

Plot the fit of a summary statistic to the target

Usage

```
plot_stats(sumstat, target, accepted, index = NA, colour = TRUE)
```

Arguments

`sumstat`	is a vector or matrix of simulated summary statistics. If this input is a vector, then each entry should correspond to a different simulation. If it is a matrix, then each row should be a different simulation and each column a different statistic. Note that this should be the entire set of simulated values.
`target`	is an integer or a numeric vector containing the target of the parameter inference. If a single integer, then this should be the target summary statistic corresponding to the input `sumstat` vector. If this input is a vector, then the order of the entries in the vector should be the same as the order of the columns of the `sumstat` matrix input. Either way, this input should contain the value of the summary statistics calculated from observed data.
`accepted`	is a vector or matrix of accepted summary statistics. If this input is a vector, then each entry should correspond to a different simulation. If it is a matrix, then each row should be a different simulation and each column a different statistic. Note that this should be summary statistics of the accepted simulations during parameter inference.
`index`	is an optional non-negative integer. This input is only required when the `sumstat` and `accepted` inputs are matrices. In that instance, it will indicate which summary statistic to plot. It corresponds to the desired column of the `sumstat` and `accepted` matrices and to the entry of the `target` vector.
`colour`	logical, indicating whether the plot should be a colour version (default) or a grayscale plot.

Value

a plot with the fit of the simulated summary statistics to the observed value. Both the density estimation of the entire simulated summary statistics and the accepted summary statistics are contrasted with the observed value.

Examples

```
# load the matrix with parameter values
data(params)
# load the matrix with simulated parameter values
data(sumstats)
# load the matrix with the prior limits
data(limits)

# select a random simulation to act as target just to test the function
target <- sumstats[10 ,]
# we should remove the random simulation from the sumstats and params matrices
sumstats <- sumstats[-10, ]; params <- params[-10, ]

# parameter estimation for a single target
myabc <- singleABC(target = target, params = params, sumstats = sumstats,
limits = limits, tol = 0.01, method = "regression")

# check the fit of a summary statistic to the target
plot_stats(sumstat = sumstats, target = target, accepted = myabc$ss, index = 5)

# note that we performed parameter estimation for a single target
# because this function will only work when using a matrix
```

---

plot_weighted *Plot the density estimation of a given parameter*

Description

Plots a locfit object obtained after parameter estimation with Approximate Bayesian Computation using the `multipleABC()` function and merging the multiple posteriors with the `mergepost()` function.

Usage

```
plot_weighted(
  prior,
  merged_posterior,
  index,
  limits,
  regWeights = TRUE,
  weighted = TRUE
)
```

Arguments

prior	is a vector or matrix of simulated parameter values i.e. numbers from the simulations. Each row or vector entry should be a different simulation and each

column of a matrix should be a different parameter. This corresponds to the prior distribution and it should contain all the simulated parameter values.

`merged_posterior`
: is a list obtained by the `mergepost()` function. The output of that function produces a list with the locfit of the various parameters. This function plots those locfits.

`index`
: is an non-negative integer indicating which parameter to plot. It corresponds to the desired entry of the `merged_posterior` list. So, to plot the first parameter, corresponding to the first entry in the `merged_posterior` input select 1. To plot the second parameter, select 2 and so on.

`limits`
: is a matrix with two columns and as many rows as there are parameters. Each row should contain the minimum value of the prior for a given parameter in the first column and the maximum value in the second column.

`regWeights`
: logical, indicating whether to plot the posterior density obtained from merging the multiple posteriors with or without the weights of the regression step. The default is TRUE.

`weighted`
: logical, indicating whether to plot the posterior density obtained from merging the multiple posteriors with or without weighting by the overall distance to the global mean. The default is TRUE.

Details

The `mergepost()` function includes different posterior merging methods and produces locfit objects for each parameter and method. It is possible to select which parameter to plot, with the `index` input, and whether to plot the density estimation after each accepted point was weighted by its regression weight and by distance of its associated target to the overall mean of the data. If `regWeights` is set to FALSE, the density estimation obtained without considering the regression weights will be plotted. If `weighted` is set to FALSE, the density estimation obtained without considering the distance between the mean summary statistics of the target and the mean across all loci.

Value

a plot of the density estimation of a given parameter. This plot will include a title with the name of the parameter. It will also include the density of the prior distribution for that parameter. The density estimation shown here is obtained after merging multiple posteriors for that parameter.

Examples

```
# load the matrix with parameter values
data(params)
# load the matrix with simulated parameter values
data(sumstats)
# load the matrix with the prior limits
data(limits)

# select some random simulations to act as target just to test the function
targets <- sumstats[c(11:20) ,]
# we should remove those random simulation from the sumstats and params matrices
sumstats <- sumstats[-c(11:20), ]; params <- params[-c(11:20), ]
```

```
# parameter estimation for multiple targets
myabc <- multipleABC(targets = targets, params = params, sumstats = sumstats, limits = limits,
tol = 0.01, method = "regression")

# select a random simulation to act as the global value of the summary statistics
# ideally this should be computed from the entirety of the observed data
global <- sumstats[50, ]

# merge the posterior distributions obtained in the previous step
mymerge <- mergepost(target = targets, global = global, post = myabc$adjusted,
wtreg = myabc$weights)

# plot the merged posterior distribution
plot_weighted(prior = params, merged_posterior = mymerge, index = 7, limits = limits)

# note that this is just an example!
# we don't have enough simulations to obtain credible results
```

---

poolSim *Simulation of Pooled DNA sequencing*

Description

This is a master function that goes to all the steps required to obtain summary statistics from pooled sequencing data.

Usage

```
poolSim(
  model,
  nDip,
  nPops,
  size,
  nLoci,
  nSites,
  mutrate,
  mean,
  variance,
  minimum,
  maximum,
  min.minor = NA,
  Nref,
  ratio,
  split,
  pool,
  seq,
```

```
  CW = NA,
  WC = NA,
  CC = NA,
  WW = NA,
  ANC = NA,
  bT = NA,
  bCW = NA,
  bWC = NA,
  force = FALSE
)
```

Arguments

model	a character, either 2pops , Single or Parallel indicating which model should be simulated.
nDip	an integer representing the total number of diploid individuals to simulate. Note that scrm actually simulates haplotypes, so the number of simulated haplotypes is double of this. Also note that this is the total number of diploid individuals and this function will distribute the individuals equally by the simulated populations.
nPops	An integer, representing the total number of populations of the simulated model.
size	a list with one entry per population. Each entry should be a vector containing the size (in number of diploid individuals) of each pool. Thus, if a population was sequenced using a single pool, the vector should contain only one entry. If a population was sequenced using two pools, each with 10 individuals, this vector should contain two entries and both will be 10.
nLoci	an integer that represents how many independent loci should be simulated.
nSites	is an integer that specifies how many base pairs should scrm simulate, i.e. how many sites per locus to simulate.
mutrate	an integer representing the mutation rate assumed for the simulations.
mean	an integer or a vector defining the mean value of the negative binomial distribution from which different number of reads are drawn. It represents the mean coverage across all sites. If a vector is supplied, the function assumes that each entry of the vector is the mean for a different population.
variance	an integer or a vector defining the variance of the negative binomial distribution from which different number of reads are drawn. It represents the variance of the total coverage across all sites. If a vector is supplied, the function assumes that each entry of the vector is the variance for a different population.
minimum	an integer representing the minimum coverage allowed. Sites where any population has a depth of coverage below this threshold are removed from the data.
maximum	an integer representing the maximum coverage allowed. Sites where any population has a depth of coverage above this threshold are removed from the data.
min.minor	is an integer representing the minimum allowed number of minor-allele reads. Sites that, across all populations, have less minor-allele reads than this threshold will be removed from the data.
Nref	is the minimum and maximum value of the uniform distribution for the effective population size of the reference population (Nref).

`ratio` is the minimum and maximum value of the distribution from which the relative size of the present-day and ancestral populations are drawn. The size of these populations is set as a ratio of the size of the Nref population. All of these ratios are drawn from a log10 uniform distribution.

`split` is the minimum and maximum values, at the 4Nref scale, of the uniform distribution from which the values of the times of the split events are draw. Both the time of the recent split event and the distance between the two split events are drawn from this distribution.

`pool` is the the minimum and maximum values of the uniform distribution from which the value of the error associated with DNA pooling is drawn. More specifically, this value is related with the unequal individual contribution to the pool. This parameter should be supplied as a decimal number between zero and one.

`seq` is the minimum and maximum values of the uniform distribution from which the value of the error associated with DNA sequencing is drawn. This parameter should be supplied as a decimal number between zero and one.

`CW` is the minimum and maximum value of the uniform distribution from which the migration rate between the two divergent ecotypes inhabiting the same location is drawn. We consider that this parameter is drawn on a m scale. This is the migration rate from ecotype C to ecotype W.

`WC` is the minimum and maximum value of the uniform distribution from which the migration rate between the two divergent ecotypes inhabiting the same location is drawn. We consider that this parameter is drawn on a m scale. This is the migration rate from ecotype W to ecotype C.

`CC` is the minimum and maximum value of the uniform distribution from which the migration rate between similar ecotypes inhabiting different locations is drawn. We consider that this parameter is drawn on a m scale. This is the migration between the two C ecotypes at two different locations.

`WW` is the minimum and maximum value of the uniform distribution from which the migration rate between similar ecotypes inhabiting different locations is drawn. We consider that this parameter is drawn on a m scale. This is the migration between the two W ecotypes at two different locations.

`ANC` is the minimum and maximum value of the uniform distribution from which the migration rate between similar ecotypes inhabiting different locations is drawn. We consider that this parameter is drawn on a m scale. This is the migration between the two W ecotypes at two different locations.

`bT` is the minimum and maximum values of the distribution from which the proportion of the simulated loci where no migration occurs between divergent ecotypes is drawn. The maximum value should not be higher than one.

`bCW` is the minimum and maximum values of the distribution from which the proportion of the simulated loci where no migration occurs from the C ecotype towards the W ecotype is drawn. The maximum value should not be higher than one.

`bWC` is the minimum and maximum values of the distribution from which the proportion of the simulated loci where no migration occurs from the W ecotype towards the C ecotype is drawn. The maximum value should not be higher than one.

`force` is a logical value indicating whether the required number of loci should be enforced. The default is FALSE but, if set to TRUE, then additional loci will be simulated. These additional loci are simulated to try to have sufficient loci to keep the required number of loci after filtering.

Details

Starts by creating a vector of parameters, with values drawn from the respective prior distributions. Then those parameter values are used to simulate genetic data under a coalescent approach. A series of steps is then followed to turn that genetic data into pooled sequencing data. Finally, a set of summary statistics is computed using the simulated pooled sequencing data.

Value

a list with several named entries. The number of entries depends of the chosen model.

`Nref` numeric, sampled value from the prior for the effective population size of the reference population.

`N1` numeric, sampled value from the prior for the relative size of the present-day populations. This is the relative size of the first population.

`N2` numeric, sampled value from the prior for the relative size of the present-day populations. This is the relative size of the second population.

`N3` numeric, sampled value from the prior for the relative size of the present-day populations. This is the relative size of the third population. This entry only exists when the selected model has four populations.

`N4` numeric, sampled value from the prior for the relative size of the present-day populations. This is the relative size of the fourth population. This entry only exists when the selected model has four populations.

`NA1` numeric, sampled value from the prior for the relative size of the ancestral populations. This is the relative size of the ancestral population of N1 and N2. This entry only exists when the selected model has four populations.

`NA2` numeric, sampled value from the prior for the relative size of the ancestral populations. This is the relative size of the ancestral population of N3 and N4. This entry only exists when the selected model has four populations.

`Split` numeric, sampled value from the prior for the time, in 4Nref scale, of the recent split event.

`Dsplit` numeric, sampled value from the prior for the time, in 4Nref scale, of the distance between the two split events.

`PoolError` numeric, sampled value from the prior for the error associated with DNA pooling.

`SeqError` numeric, sampled value from the prior for the error associated with DNA sequencing.

`mCW1` numeric, sampled value from the prior for the migration rate between the two divergent ecotypes inhabiting the first location. This is the migration rate from ecotype C to ecotype W. For a two population model, this entry will be called mCW because that model considers a single location.

mCW2	numeric, sampled value from the prior for the migration rate between the two divergent ecotypes inhabiting the second location. This is the migration rate from ecotype C to ecotype W. For a two population model, this entry will not exist.
mWC1	numeric, sampled value from the prior for the migration rate between the two divergent ecotypes inhabiting the first location. This is the migration rate from ecotype W to ecotype C. For a two population model, this entry will be called mWC because that model considers a single location.
mWC2	numeric, sampled value from the prior for the migration rate between the two divergent ecotypes inhabiting the second location. This is the migration rate from ecotype W to ecotype C. For a two population model, this entry will not exist.
mCC	numeric, sampled value from the prior for the migration rate between similar ecotypes inhabiting different locations. This is the migration between the two C ecotypes at two different locations. For a two population model, this entry will not exist.
mWW	numeric, sampled value from the prior for the migration rate between similar ecotypes inhabiting different locations. This is the migration between the two W ecotypes at two different locations. For a two population model, this entry will not exist.
mAA	numeric, sampled value from the prior for the migration rate between the two ancestral populations. For a two population model, this entry will not exist.
pM	numeric, sampled value from the prior for the proportion of the genome with no barriers against gene flow. This is the proportion of simulated loci where migration occurs in both directions between the divergent ecotypes.
pCW	numeric, sampled value from the prior for the proportion of the genome where no migration occurs from the C ecotype towards the W ecotype. This is the proportion of simulated loci where migration occurs only from W towards C. This entry does not exist for the two populations model.
pWC	numeric, sampled value from the prior for the proportion of the genome where no migration occurs from the W ecotype towards the C ecotype. This is the proportion of simulated loci where migration occurs only from C towards W. This entry does not exist for the two populations model.
pNO	numeric, sampled value from the prior for the proportion of the genome with no gene flow between divergent ecotypes. This is the proportion of simulated loci where migration does not occur in both directions between the C and W ecotypes.
nPoly	numeric, mean number of polymorphic sites across all simulated locus.
nFilter	numeric, mean number of polymorphic sites retained after filtering across all simulated locus.
nLoci	numeric, total number of loci retained after filtering. Summary statistics are calculated for these loci.
Sf	numeric, fraction of sites fixed between populations. For the model with two populations, this is a single value. For the four-population models, this includes three values: the first is the fraction of fixed sites between the two populations

in the first location, the second value is between the populations in the second location and the third value is the overall fraction of fixed sites, obtained by comparing each population against the other three.

`Sx` numeric, fraction of exclusive sites per population. When running the model with two populations, this entry has two values - one per population. For the four-population models, there is also one value per population, followed by a fifth value representing the fraction of sites that are segregating in only one of the populations.

`SS` numeric values representing the fraction of sites shared between populations. For the model with two populations, this is a single value. When running one of the four-population models, this entry has three values. The first is the fraction of shared sites between the two populations in the first location, the second value is between the populations in the second location and the third value is the fraction of shared sites across all four populations.

`Mean_Het` numeric, expected heterozygosity within each population. This entry has two values when using a two populations model and four when running one of the four-populations model.

`SD_Het` numeric, standard deviation of the expected heterozygosity for each population. This entry has two values when using a two populations model and four when running one of the four-populations model.

`Mean_HetBet` numeric, mean heterozygosity between all pairs of populations. For the two populations model, this is a single value representing the heterozygosity between the two populations. For the four-population models, this entry includes six values. The first value is the heterozygosity between the first and the second population, the second value is between the first and the third population, the third value is between the first and fourth population, the fourth value is between the second and third populations, the fifth value is between the second and fourth population and the sixth value is between the third and fourth populations.

`SD_HetBet` numeric, standard deviation of the mean heterozygosity between all pairs of populations. For the two populations model, this is a single value representing the standard deviation of heterozygosity between the two populations. When running one of the four-population models, this entry includes six values. The order of those entries is the same as for `Mean_HetBet`.

`Mean_FST` numeric, mean pairwise FST between populations. For the two populations model, this is a single value representing the mean FST between the two populations. For the four-population models, this entry includes six values. The first value is the mean FST between the first and second populations, the second is between the first and third population, the third is between the second and third populations, the fourth is between the first and fourth populations, the fifth value is between the second and fourth populations and the sixth is between the third and fourth populations.

`SD_FST` numeric, standard deviation of the mean pairwise FST between populations. For the two populations model, this is a single value representing the standard deviation of the FST between the two populations. When running one of the four-population models, this entry includes six values. The order of those entries is the same as for `Mean_FST`.

FSTQ1	numeric, it is the 5% quantile of the mean pairwise FST distribution. For the two populations model, this is a single value representing the 5% quantile of the FST between the two populations. When running one of the four-population models, this entry includes six values. The order of those entries is the same as for Mean_FST.
FSTQ2	numeric, it is the 95% quantile of the mean pairwise FST distribution. For the two populations model, this is a single value representing the 95% quantile of the FST between the two populations. For the four-population models, this entry includes six values. The order of those entries is the same as for Mean_FST.
Dstat	numeric, value of D-statistic for various combinations of populations. This entry only exists if a four-population model was selected. It includes three different values. For the first value, P1 was the W ecotype in the first location P2 was the W ecotype in the second location and P3 was the C ecotype at the first location. For the second value P1 was again the W ecotype in the first location but P2 was the C ecotype in the second ecotype and P3 was the C ecotype at the first location. For the third value, P1 was also the W ecotype at the first location, P2 was the C ecotype at the first location and P3 was the W ecotype at the second location. For all combinations, P4 was assumed to be an outgroup fixed, at all sites, for the major allele.
SD_dstat	numeric, standard deviation of D-statistic for various combinations of populations. This entry only exists if a four-population model was selected. Each entry is the standard deviation of the corresponding D-statistic in the Dstat entry.

Examples

```
# simulate Pool-seq data and compute summary statistics for a model with two populations
poolSim(model="2pops", nDip=400, nPops=2, nLoci=10, nSites=2000, mutrate=1.5e-8,
size=rep(list(rep(5, 20)), 2),mean=c(85, 65), variance=c(1400, 900), minimum=25,
maximum=165, min.minor=2, Nref=c(25000, 25000), ratio=c(0.1, 3), pool=c(5, 250),
seq=c(0.0001, 0.001), split=c(0, 3), CW=c(1e-13, 1e-3), WC=c(1e-13, 1e-3), bT=c(0, 0.5))

# simulate Pool-seq data and compute summary statistics for a model with four populations
poolSim(model="Single", nDip=400, nPops=4, nLoci=10, nSites=2000, mutrate=2e-8,
size=rep(list(rep(5, 20)), 4), mean=c(85, 65, 65, 70), variance=c(1400, 900, 850, 1000),
minimum=25, maximum=165, min.minor=2, Nref=c(25000, 25000), ratio=c(0.1, 3), pool=c(5, 250),
seq=c(0.0001, 0.001), split=c(0, 3), CW=c(1e-13, 1e-3), WC=c(1e-13, 1e-3), CC=c(1e-13, 1e-3),
WW=c(1e-13, 1e-3), ANC=c(1e-13, 1e-3), bT=c(0, 0.2), bCW=c(0, 0.5), bWC=c(0, 0.5))
```

---

poolStats *Compute summary statistics from Pooled DNA sequencing*

Description

This function combines all the necessary steps to simulate pooled sequencing data and compute summary statistics from that data.

Usage

```
poolStats(
  parameters,
  model,
  nDip,
  size,
  nLoci,
  nSites,
  mutrate,
  mean,
  variance,
  minimum,
  maximum,
  min.minor = NA,
  force = FALSE
)
```

Arguments

parameters	a vector of parameters used to create the command line for the scrm package. Each entry of the vector is a different parameter. Note that each vector entry should be named with the name of the corresponding parameter. The output of the `CreateParameters` function is the intended input.
model	a character, either 2pops , Single or Parallel indicating which model should be simulated.
nDip	an integer representing the total number of diploid individuals to simulate. Note that scrm actually simulates haplotypes, so the number of simulated haplotypes is double of this. Also note that this is the total number of diploid individuals and this function will distribute the individuals equally by the simulated populations.
size	a list with one entry per population. Each entry should be a vector containing the size (in number of diploid individuals) of each pool. Thus, if a population was sequenced using a single pool, the vector should contain only one entry. If a population was sequenced using two pools, each with 10 individuals, this vector should contain two entries and both will be 10.
nLoci	an integer that represents how many independent loci should be simulated.
nSites	is an integer that specifies how many base pairs should scrm simulate, i.e. how many sites per locus to simulate.
mutrate	an integer representing the mutation rate assumed for the simulations.
mean	an integer or a vector defining the mean value of the negative binomial distribution from which different number of reads are drawn. It represents the mean coverage across all sites. If a vector is supplied, the function assumes that each entry of the vector is the mean for a different population.
variance	an integer or a vector defining the variance of the negative binomial distribution from which different number of reads are drawn. It represents the variance of the total coverage across all sites. If a vector is supplied, the function assumes that each entry of the vector is the variance for a different population.

`minimum` an integer representing the minimum coverage allowed. Sites where any population has a depth of coverage below this threshold are removed from the data.

`maximum` an integer representing the maximum coverage allowed. Sites where any population has a depth of coverage above this threshold are removed from the data.

`min.minor` is an integer representing the minimum allowed number of minor-allele reads. Sites that, across all populations, have less minor-allele reads than this threshold will be removed from the data.

`force` is a logical value indicating whether the required number of loci should be enforced. The default is FALSE but, if set to TRUE, then additional loci will be simulated. These additional loci are simulated to try to have sufficient loci to keep the required number of loci after filtering.

Details

The sampled parameter values are incorporated into a command line for the scrm package. Then, genetic data is simulated according to a model of ecotype formation and the sampled parameters. Finally, various summary statistics are calculated from the simulated data.

Value

a list with several named entries. The number of entries depends of the chosen model.

`nPoly` numeric, mean number of polymorphic sites across all simulated locus.

`nFilter` numeric, mean number of polymorphic sites retained after filtering across all simulated locus.

`nLoci` numeric, total number of loci retained after filtering. Summary statistics are calculated for these loci.

`Sf` numeric, fraction of sites fixed between populations. For the model with two populations, this is a single value. For the four-population models, this includes three values: the first is the fraction of fixed sites between the two populations in the first location, the second value is between the populations in the second location and the third value is the overall fraction of fixed sites, obtained by comparing each population against the other three.

`Sx` numeric, fraction of exclusive sites per population. When running the model with two populations, this entry has two values - one per population. For the four-population models, there is also one value per population, followed by a fifth value representing the fraction of sites that are segregating in only one of the populations.

`SS` numeric values representing the fraction of sites shared between populations. For the model with two populations, this is a single value. When running one of the four-population models, this entry has three values. The first is the fraction of shared sites between the two populations in the first location, the second value is between the populations in the second location and the third value is the fraction of shared sites across all four populations.

`Mean_Het` numeric, expected heterozygosity within each population. This entry has two values when using a two populations model and four when running one of the four-populations model.

SD_Het — numeric, standard deviation of the expected heterozygosity for each population. This entry has two values when using a two populations model and four when running one of the four-populations model.

Mean_HetBet — numeric, mean heterozygosity between all pairs of populations. For the two populations model, this is a single value representing the heterozygosity between the two populations. For the four-population models, this entry includes six values. The first value is the heterozygosity between the first and the second population, the second value is between the first and the third population, the third value is between the first and fourth population, the fourth value is between the second and third populations, the fifth value is between the second and fourth population and the sixth value is between the third and fourth populations.

SD_HetBet — numeric, standard deviation of the mean heterozygosity between all pairs of populations. For the two populations model, this is a single value representing the standard deviation of heterozygosity between the two populations. When running one of the four-population models, this entry includes six values. The order of those entries is the same as for Mean_HetBet.

Mean_FST — numeric, mean pairwise FST between populations. For the two populations model, this is a single value representing the mean FST between the two populations. For the four-population models, this entry includes six values. The first value is the mean FST between the first and second populations, the second is between the first and third population, the third is between the second and third populations, the fourth is between the first and fourth populations, the fifth value is between the second and fourth populations and the sixth is between the third and fourth populations.

SD_FST — numeric, standard deviation of the mean pairwise FST between populations. For the two populations model, this is a single value representing the standard deviation of the FST between the two populations. When running one of the four-population models, this entry includes six values. The order of those entries is the same as for Mean_FST.

FSTQ1 — numeric, it is the 5% quantile of the mean pairwise FST distribution. For the two populations model, this is a single value representing the 5% quantile of the FST between the two populations. When running one of the four-population models, this entry includes six values. The order of those entries is the same as for Mean_FST.

FSTQ2 — numeric, it is the 95% quantile of the mean pairwise FST distribution. For the two populations model, this is a single value representing the 95% quantile of the FST between the two populations. For the four-population models, this entry includes six values. The order of those entries is the same as for Mean_FST.

Dstat — numeric, value of D-statistic for various combinations of populations. This entry only exists if a four-population model was selected. It includes three different values. For the first value, P1 was the W ecotype in the first location P2 was the W ecotype in the second location and P3 was the C ecotype at the first location. For the second value P1 was again the W ecotype in the first location but P2 was the C ecotype in the second ecotype and P3 was the C ecotype at the first location. For the third value, P1 was also the W ecotype at the first location, P2 was the C ecotype at the first location and P3 was the W ecotype at the second

location. For all combinations, P4 was assumed to be an outgroup fixed, at all sites, for the major allele.

`SD_dstat` numeric, standard deviation of D-statistic for various combinations of populations. This entry only exists if a four-population model was selected. Each entry is the standard deviation of the corresponding D-statistic in the `Dstat` entry.

Examples

```
# create a vector of parameters for a model with two populations
parameters <- createParams(Nref = c(25000, 25000), ratio = c(0.1, 3), pool = c(5, 250),
seq = c(0.0001, 0.001), split = c(0, 3), CW = c(1e-13, 1e-3), WC = c(1e-13, 1e-3),
bT = c(0, 0.2), model = "2pops")

# simulate a two populations model:
# note that we are using two pools for each population, each with 50 individuals
poolStats(parameters = parameters, model = "2pops", nDip = 200, size = rep(list(rep(50, 2)), 2),
nLoci = 100, nSites = 2000, mutrate = 2e-8, mean = c(100, 80), variance = c(200, 180), minimum = 10,
maximum = 150, min.minor = 1)
```

---

poststat *Calculate point estimates from the posterior distribution*

Description

Given a set of samples from the posterior distribution, computes the mean, median and mode of the posterior.

Usage

```
poststat(posterior, limits, method, wtreg = NULL)
```

Arguments

`posterior` is a matrix or a vector with samples from the posterior distribution obtained from ABC parameter estimation. If this input is a matrix, then each row should correspond to an accepted simulation (size S) and each column to a different parameter.

`limits` is a vector if there is only one parameter or a matrix if there are multiple parameters. In this latter instance, each row should correspond to a different parameter. In either instance, and considering matrix rows as vectors, then the first entry of the vector should be the minimum value of the prior and the second entry should be the maximum value (for any given parameter).

`method` either `rejection` or `regression` indicating whether a rejection sampling algorithm or a local linear regression algorithm were used during ABC parameter estimation.

`wtreg` is a required numeric vector if the method is `regression`. It should contain the weights for each accepted simulation (size S).

Details

If method is regression , the regression weights must also be made available. These will be used to compute the weighted mean, weighted median and weighted mode of the posterior.

Value

a matrix with the mode, median and mean of the posterior distribution for each parameter. Each point estimate is a different row and each parameter a different column.

Examples

```
# load the matrix with parameter values
data(params)
# load the matrix with simulated parameter values
data(sumstats)
# load the matrix with the prior limits
data(limits)

# select a random simulation to act as target just to test the function
target <- sumstats[10 ,]
# we should remove the random simulation from the sumstats and params matrices
sumstats <- sumstats[-10, ]; params <- params[-10, ]

# parameter estimation for a single target
myabc <- singleABC(target = target, params = params, sumstats = sumstats,
limits = limits, tol = 0.01, method = "regression")

# compute point estimates from the posterior distribution
poststat(posterior = myabc$adjusted, limits = limits, method = "regression", wtreg = myabc$wt)
```

---

prepareData *Organize information by contig - for multiple data files*

Description

Organize the information of multiple rc files into different entries for each contig.

Usage

```
prepareData(data, nPops, filter = FALSE, threshold = NA)
```

Arguments

data	is a list with four different entries. The entries should be named as rMajor , rMinor , coverage and info . The rMajor entry should be a matrix containing the number of observed major-allele reads. The rMinor entry should be a matrix containing the number of observed minor-allele reads. The coverage entry should be a matrix containing the total depth of coverage. The info entry

should be a matrix or a data frame containing the remaining relevant information, such as the contig name and the position of each SNP. Each row of these matrices should be a different site and each column should be a different population.

`nPops` is an integer indicating the total number of different populations in the dataset.

`filter` is a logical switch, either TRUE or FALSE. If TRUE, then the data is filtered by the frequency of the minor allele and if FALSE, that filter is not applied.

`threshold` is the minimum allowed frequency for the minor allele. Sites where the allelic frequency is below this threshold are removed from the data.

Details

This function removes all monomorphic sites from the dataset. Monomorphic sites are those where the frequency for all populations is 1 or 0. Then, the name of each contig is used to organize the information in a per contig basis. Thus, each output will be organized by contig. For example, the list with the number of minor-allele reads will contain several entries and each of those entries is a different contig.

If the filter input is set to TRUE, this function also filters the data by the frequency of the minor-allele. If a threshold is supplied, the computed frequency is compared to that threshold and sites where the frequency is below the threshold are removed from the dataset. If no threshold is supplied, the threshold is assumed to be `1/total coverage`, meaning that a site should have, at least, one minor-allele read.

Value

a list with six named entries:

`freqs` a list with the allele frequencies, computed by dividing the number of minor-allele reads by the total coverage. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.

`positions` a list with the positions of each SNP. Each entry of this list is a vector corresponding to a different contig.

`range` a list with the minimum and maximum SNP position of each contig. Each entry of this list is a vector corresponding to a different contig.

`rMajor` a list with the number of major-allele reads. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.

`rMinor` a list with the number of minor-allele reads. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.

`coverage` a list with the total coverage. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.

Examples

```
# load the data from two rc files
data(rc1, rc2)
# combine both files into a single list
mydata <- list(rc1, rc2)

# clean and organize the data for both files
mydata <- lapply(mydata, function(i) cleanData(file = i, pops = 7:10))

# organize the information by contigs
prepareData(data = mydata, nPops = 4)
```

---

prepareFile *Organize information by contigs - for a single data file*

Description

Organize the information of a single rc file into different entries for each contig.

Usage

```
prepareFile(data, nPops, filter = FALSE, threshold = NA)
```

Arguments

data	is a list with four different entries. The entries should be named as rMajor , rMinor , coverage and info . The rMajor entry should be a matrix containing the number of observed major-allele reads. The rMinor entry should be a matrix containing the number of observed minor-allele reads. The coverage entry should be a matrix containing the total depth of coverage. The info entry should be a matrix or a data frame containing the remaining relevant information, such as the contig name and the position of each SNP. Each row of these matrices should be a different site and each column should be a different population.
nPops	is an integer indicating the total number of different populations in the dataset.
filter	is a logical switch, either TRUE or FALSE. If TRUE, then the data is filtered by the frequency of the minor allele and if FALSE, that filter is not applied.
threshold	is the minimum allowed frequency for the minor allele. Sites where the allelic frequency is below this threshold are removed from the data.

Details

This function removes all monomorphic sites from the dataset. Monomorphic sites are those where the frequency for all populations is 1 or 0. Then, the name of each contig is used to organize the information in a per contig basis. Thus, each output will be organized by contig. For example, the

list with the number of minor-allele reads will contain several entries and each of those entries is a different contig.

If the filter input is set to TRUE, this function also filters the data by the frequency of the minor-allele. If a threshold is supplied, the computed frequency is compared to that threshold and sites where the frequency is below the threshold are removed from the dataset. If no threshold is supplied, the threshold is assumed to be `1/total coverage`, meaning that a site should have, at least, one minor-allele read.

Value

a list with six named entries:

`freqs`	a list with the allele frequencies, computed by dividing the number of minor-allele reads by the total coverage. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.
`positions`	a list with the positions of each SNP. Each entry of this list is a vector corresponding to a different contig.
`range`	a list with the minimum and maximum SNP position of each contig. Each entry of this list is a vector corresponding to a different contig.
`rMajor`	a list with the number of major-allele reads. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.
`rMinor`	a list with the number of minor-allele reads. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.
`coverage`	a list with the total coverage. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.

Examples

```
# load the data from one rc file
data(rc1)

# clean and organize the data in this single file
mydata <- cleanData(file = rc1, pops = 7:10)

# organize the information by contigs
prepareFile(data = mydata, nPops = 4)
```

priorsMatrix *Construct matrix of prior limits*

Description

Takes as input the minimum and maximum values of the prior distribution for all relevant parameters and constructs a matrix of prior limits.

Usage

```
priorsMatrix(model, inputParams)
```

Arguments

model	a character, either 2pops , Single or Parallel indicating which model was simulated.
inputParams	A vector containing the minimum and maximum values of the prior distribution for each parameter in the model. The input of the CreateParameters function can be converted into a vector and used here.

Details

The output matrix contains all parameters of a given model and, for each parameter, it contains the minimum and maximum value of the prior.

Value

a matrix where each row is a different parameter. Note also that each row is named after the corresponding parameter. For each row, the first column contains the minimum value of that parameter and the second column contains the maximum value.

Examples

```
# create a vector of input parameters for a model with two populations
inputs <- c(Nref = c(25000, 25000), ratio = c(0.1, 3), pool = c(5, 250), seq = c(0.0001, 0.001),
split = c(0, 3), CW = c(1e-13, 1e-3), WC = c(1e-13, 1e-3), bT = c(0, 0.2))

# construct a matrix with the limits of the prior distribution
priorsMatrix(model = "2pops", inputParams = inputs)
```

rc1 *Data frame with an example of observed data*

Description

Data frame with data in the _rc format for 25 populations Each row of the data frame is a different site. The first 9 columns contain general information about the site, while the remaining contain the number of reads observed at that site for each of the 25 populations.

Usage

```
rc1
```

Format

a data frame with 5000 rows and 59 columns. Each of the columns corresponds to :

col1 reference chromosome (contig).

col2 reference position.

col3 reference character.

col4 number of alleles found in all populations.

col5 allele characters in all populations (sorted by counts in all populations).

col6 sum of deletions in all populations (should be zero, if not the position may not be reliable).

col7 SNP type: `[pop, rc, rc|pop]`; pop: a SNP within or between the populations; rc: a SNP between the reference sequence character and the consensus of at least one populaton; rc pop: both.

col8 most frequent allele in all populations `[12345..]`.

col9 second most frequent allele in all populations `[12345..]`.

col10 - col34 frequencies of the most frequent allele (major) in the form allele-count/coverage.

col35 - col59 frequencies of the second most frequent allele (minor) in the form allele-count/coverage .

Source

Hern n E. Morales et al., Genomic architecture of parallel ecological divergence: Beyond a single environmental contrast. Sci. Adv.5, eaav9963(2019). DOI:10.1126/sciadv.aav9963

rc2 *Data frame with an example of observed data*

Description

Data frame with data in the _rc format for 25 populations Each row of the data frame is a different site. The first 9 columns contain general information about the site, while the remaining contain the number of reads observed at that site for each of the 25 populations.

Usage

```
rc2
```

Format

a data frame with 5000 rows and 59 columns. Each of the columns corresponds to :

col1 reference chromosome (contig).

col2 reference position.

col3 reference character.

col4 number of alleles found in all populations.

col5 allele characters in all populations (sorted by counts in all populations).

col6 sum of deletions in all populations (should be zero, if not the position may not be reliable).

col7 SNP type: [pop, rc, rc|pop]; pop: a SNP within or between the populations; rc: a SNP between the reference sequence character and the consensus of at least one populaton; rc pop: both.

col8 most frequent allele in all populations [12345..].

col9 second most frequent allele in all populations [12345..].

col10 - col34 frequencies of the most frequent allele (major) in the form allele-count/coverage.

col35 - col59 frequencies of the second most frequent allele (minor) in the form allele-count/coverage .

Source

Hern n E. Morales et al., Genomic architecture of parallel ecological divergence: Beyond a single environmental contrast. Sci. Adv.5, eaav9963(2019). DOI:10.1126/sciadv.aav9963

`regABC` *Parameter estimation with Approximate Bayesian Computation using local linear regression*

Description

This function performs multivariate parameter estimation based on summary statistics using an Approximate Bayesian Computation (ABC) algorithm. The algorithm used here is the local linear regression algorithm.

Usage

```
regABC(rej, parameter, tol = 1, simple = FALSE)
```

Arguments

`rej`	is a list with the results of the rejection sampling algorithm. The output of the `rejABC()` function is the ideal input here.
`parameter`	is a parameter vector (long vector of numbers from the simulations). Each vector entry should correspond to a different simulation. This is the dependent variable for the regression.
`tol`	is the tolerance rate, indicating the required proportion of points accepted nearest the target values. Note that the default value here is 1 because all points accepted in the rejection step should be used for the regression.
`simple`	logical, if TRUE a simplified output with only the essential information will be produced. If FALSE (default) the output will contain more information.

Details

Note that to use this function, the usual steps of ABC parameter estimation have to be performed. Briefly, data should have been simulated based on random draws from the prior distributions of the parameters of interest and a set of summary statistics should have been calculated from that data. The same set of summary statistics should have been calculated from the observed data to be used as the target for parameter inference. A previous rejection sampling step should also have been performed, where parameter values were accepted if the Euclidean distance between the set of summary statistics computed from the simulated data and the set of summary statistics computed from the observed data was sufficiently small. Then, the output of the rejection step is used as the input for this function and a local linear regression method is used to correct for the imperfect match between the summary statistics computed from the simulated data and the summary statistics computed from the observed data.

The parameter values accepted in the rejection step are weighted by a smooth function (kernel) of the distance between the simulated and observed summary statistics and corrected according to a linear transformation. This function calls the function `stats::lm()` to accomplish this.

Value

a list with the results from the regression correction

`adjusted`	regression adjusted parameter values.
`unadjusted`	parameter estimates obtained with the rejection sampling.
`wt`	regression weights.
`ss`	set of accepted summary statistics from the simulations.
`predmean`	estimates of the posterior mean for each parameter.
`fv`	fitted value from the regression.

Examples

```
# load the matrix with parameter values
data(params)
# load the matrix with simulated parameter values
data(sumstats)

# select a random simulation to act as target just to test the function
target <- sumstats[10 ,]

# parameter estimation using rejection sampling
rej <- rejABC(target = target, params = params, sumstats = sumstats[-10, ],
tol = 0.01, regression = TRUE)

# parameter estimation using local linear regression
# note that you should select a parameter from the unadjusted matrix
regABC(rej = rej, parameter = rej$unadjusted[, 1])
```

`rejABC` *Parameter estimation with Approximate Bayesian Computation using rejection sampling*

Description

This function performs multivariate parameter estimation based on summary statistics using an Approximate Bayesian Computation (ABC) algorithm. The algorithm used here is the rejection sampling algorithm. The output of this function can be tailored towards a posterior local linear regression method correction.

Usage

```
rejABC(target, params, sumstats, tol, regression = FALSE)
```

Arguments

`target`	a vector with the target summary statistics. These are usually the set of observed summary statistics.
`params`	is a vector or matrix of simulated parameter values i.e. numbers from the simulations. Each row or vector entry should be a different simulation and each column of a matrix should be a different parameter.
`sumstats`	is a vector or matrix of simulated summary statistics. Each row or vector entry should be a different simulation and each column of a matrix should be a different statistic.
`tol`	is the tolerance rate, indicating the required proportion of points accepted nearest the target values.
`regression`	logical, indicating whether the user intends to perform a local linear regression correction after the rejection step. If set to FALSE (default) the output of this function will contain just the results of the rejection step. If set to TRUE, the output will contain more details required for the regression step.

Details

The rejection sampling algorithm generates random samples from the posterior distributions of the parameters of interest. Note that to use this function, the usual steps of ABC parameter estimation have to be performed. Briefly, data should have been simulated based on random draws from the prior distributions of the parameters of interest and a set of summary statistics should have been calculated from that data. The same set of summary statistics should have been calculated from the observed data to be used as the `target` input in this function. Parameter values are accepted if the Euclidean distance between the set of summary statistics computed from the simulated data and the set of summary statistics computed from the observed data is sufficiently small. The percentage of accepted simulations is determined by `tol`.

Value

a list with the results of the rejection sampling algorithm. The elements of the list depend of the logical value of `regression`.

`s.target`	a scaled vector of the observed summary statistics. This element only exists if regression is TRUE.
`unadjusted`	parameter estimates obtained with the rejection sampling.
`ss`	set of accepted summary statistics from the simulations.
`s.sumstat`	set of scaled accepted summary statistics from the simulations. This element only exists if regression is TRUE.
`dst`	euclidean distances in the region of interest.

Examples

```
# load the matrix with parameter values
data(params)
# load the matrix with simulated parameter values
data(sumstats)
```

```
# select a random simulation to act as target just to test the function
target <- sumstats[10 ,]

# Parameter estimation using rejection sampling
rejABC(target = target, params = params, sumstats = sumstats[-10, ], tol = 0.01)
```

`remove_quantileReads` *Remove sites using quantiles of the depth of coverage*

Description

Removes sites that have too many or too few reads from the dataset.

Usage

```
remove_quantileReads(nPops, data)
```

Arguments

`nPops`	is an integer representing the total number of populations in the dataset.
`data`	is a dataset containing information about real populations. This dataset should have lists with the allelic frequencies, the position of the SNPs, the range of the contig, the number of major allele reads, the number of minor allele reads and the depth of coverage.

Details

The 25% and the 75% quantiles of the coverage distribution is computed for each population in the dataset. Then, the lowest 25% quantile across all populations is considered the minimum depth of coverage allowed. Similarly, the highest 75% quantile across all populations is considered the maximum depth of coverage allowed. The coverage of each population at each site is compared with those threshold values and any site, where the coverage of at least one population is below or above that threshold, is completely removed from the dataset.

Value

a list with the following elements:

`freqs`	a list with the allele frequencies, computed by dividing the number of minor-allele reads by the total coverage. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.
`positions`	a list with the positions of each SNP. Each entry of this list is a vector corresponding to a different contig.
`range`	a list with the minimum and maximum SNP position of each contig. Each entry of this list is a vector corresponding to a different contig.

rMajor	a list with the number of major-allele reads. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.
rMinor	a list with the number of minor-allele reads. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.
coverage	a list with the total coverage. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.

This output is identical to the data input, the only difference being the removal of sites with too many or too few reads.

Examples

```
# load the data from one rc file
data(rc1)

# clean and organize the data in this single file
mydata <- cleanData(file = rc1, pops = 7:10)

# organize the information by contigs
mydata <- prepareFile(data = mydata, nPops = 4)

# remove sites according to the coverage quantile
remove_quantileReads(nPops = 4, data = mydata)
```

---

remove_realReads *Remove sites, according to their coverage, from real data*

Description

Removes sites that have too many or too few reads from the dataset.

Usage

```
remove_realReads(nPops, data, minimum, maximum)
```

Arguments

nPops	is an integer representing the total number of populations in the dataset.
data	is a dataset containing information about real populations. This dataset should have lists with the allelic frequencies, the position of the SNPs, the range of the contig, the number of major allele reads, the number of minor allele reads and the depth of coverage.
minimum	the minimum depth of coverage allowed i.e. sites where the depth of coverage of any population is below this threshold are removed.

`maximum` he maximum depth of coverage allowed i.e. sites where the depth of coverage of any population is above this threshold are removed.

Details

The `minimum` and `maximum` inputs define, respectively, the minimum and maximum allowed coverage for the dataset. The coverage of each population at each site is compared with those threshold values and any site, where the coverage of at least one population is below or above the user defined threshold, is completely removed from the dataset.

Value

a list with the following elements:

`freqs` a list with the allele frequencies, computed by dividing the number of minor-allele reads by the total coverage. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.

`positions` a list with the positions of each SNP. Each entry of this list is a vector corresponding to a different contig.

`range` a list with the minimum and maximum SNP position of each contig. Each entry of this list is a vector corresponding to a different contig.

`rMajor` a list with the number of major-allele reads. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.

`rMinor` a list with the number of minor-allele reads. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.

`coverage` a list with the total coverage. Each entry of this list corresponds to a different contig. Each entry is a matrix where each row is a different site and each column is a different population.

This output is identical to the `data` input, the only difference being the removal of sites with too many or too few reads.

Examples

```
# load the data from one rc file
data(rc1)

# clean and organize the data in this single file
mydata <- cleanData(file = rc1, pops = 7:10)

# organize the information by contigs
mydata <- prepareFile(data = mydata, nPops = 4)

# remove sites with less than 10 reads or more than 180
remove_realReads(nPops = 4, data = mydata, minimum = 10, maximum = 180)
```

`runSCRM` *Run scrm and obtain genotypes*

Description

This function will run the scrm package, according to the command line supplied as input. It will also combine haplotypes into genotypes and re-organize the output if the simulations were performed under a single origin scenario. This is to ensure that the output of the four-population models will always follow the same order: the two divergent ecotypes in the first location, followed by the two divergent ecotypes in the second location.

Usage

```
runSCRM(commands, nDip, nPops, model)
```

Arguments

`commands`	A character string containing the commands for the scrm package. This string can be created using the `cmd2pops`, the `cmdSingle` or the `cmdParallel` functions.
`nDip`	An integer representing the total number of diploid individuals to simulate. Note that scrm actually simulates haplotypes, so the number of simulated haplotypes is double of this.
`nPops`	An integer that informs of how many populations exist on the model you are trying to run.
`model`	Either `2pops` , `Single` or `Parallel` indicating which model should be simulated.

Value

a list with the simulated genotypes. Each entry is a different locus and, for each locus, different rows represent different individuals and each column is a different site.

Examples

```
# create a vector with parameter values for a two populations model
params <- createParams(Nref = c(25000, 25000), ratio = c(0.1, 3), pool = c(5, 250),
seq = c(0.0001, 0.001), split = c(0, 3), CW = c(1e-13, 1e-3), WC = c(1e-13, 1e-3),
bT = c(0, 0.2), model = "2pops")

# create the command line for the scrm package
cmds <- cmd2pops(parameters = params, nSites = 2000, nLoci = 10, nDip = 100, mutrate = 2e-8)

# run SCRM and obtain the genotypes
runSCRM(commands = cmds, nDip = 100, nPops = 2, model = "2pops")
```

`scaled.migration` *Compute scaled migration rates*

Description

Computes and adds scaled migration rates to a matrix of simulated parameter values.

Usage

```
scaled.migration(parameters, model, Nref = NA)
```

Arguments

`parameters`	is a matrix of simulated parameter values i.e. numbers from the simulations. Each row or vector entry should be a different simulation and each column of a matrix should be a different parameter.
`model`	a character, either 2pops , Single or Parallel indicating which model was simulated.
`Nref`	a numeric value indicating the effective population size of the reference population.

Details

Migration rates are scaled according to the size of the population receiving the migrants and added to a matrix with the simulated parameter values. This is performed for the three available models and according to the specific model conformation.

Value

a matrix of simulated parameter values with added columns containing the scaled migration rates.

Examples

```
# compute scaled migration for a two-population model
scaled.migration(parameters = myparams, model = "2pops", Nref = 10000)
```

scaledPrior *Compute scaled migration rate limits*

Description

Computes and adds scaled migration rates to a matrix with the limits of the prior distributions.

Usage

```
scaledPrior(limits, model, Nref = NA)
```

Arguments

limits	is a matrix with two columns and as many rows as there are parameters. Each row should contain the minimum value of the prior for a given parameter in the first column and the maximum value in the second column.
model	a character, either 2pops , Single or Parallel indicating which model was simulated.
Nref	a numeric value indicating the effective population size of the reference population.

Details

Migration rates are scaled according to the size of the population receiving the migrants and added to a matrix with the prior limits. The minimum and maximum possible size of the population and of the migration rate are used to compute the minimum and maximum possible values of the scaled migration rates. This is performed for the three available models and according to the specific model conformation.

Value

a matrix where each row is a different parameter. This matrix is similar to the input argument limits but with added rows containing the scaled migration rates.

Examples

```
# create a vector of input parameters for a model with two populations
inputs <- c(Nref = c(25000, 25000), ratio = c(0.1, 3), pool = c(5, 250), seq = c(0.0001, 0.001),
split = c(0, 3), CW = c(1e-13, 1e-3), WC = c(1e-13, 1e-3), bT = c(0, 0.2))

# construct a matrix with the limits of the prior distribution
limits <- priorsMatrix(model = "2pops", inputParams = inputs)

# compute and add the prior limits of the scaled migration
scaledPrior(limits = limits, model = "2pops")
```

`simulationABC` *Perform an Approximate Bayesian Computation simulation study*

Description

Perform a leave-one-out cross validation for ABC via subsequent calls to the `singleABC()` function.

Usage

```
simulationABC(
  params,
  sumstats,
  limits,
  nval,
  tol,
  method,
  parallel = FALSE,
  ncores = NA
)
```

Arguments

`params`	is a vector or matrix of simulated parameter values i.e. numbers from the simulations. Each row or vector entry should be a different simulation and each column of a matrix should be a different parameter. This is the dependent variable for the regression, if a regression step is performed.
`sumstats`	is a vector or matrix of simulated summary statistics. Each row or vector entry should be a different simulation and each column of a matrix should be a different statistic. These act as the independent variables if a regression step is performed.
`limits`	is a matrix with two columns and as many rows as there are parameters. Each row should contain the minimum value of the prior for a given parameter in the first column and the maximum value in the second column.
`nval`	size of the cross-validation sample i.e. how many different evaluations should be performed. Each evaluation corresponds to a different target for the parameter estimation.
`tol`	is the tolerance rate, indicating the required proportion of points accepted nearest the target values.
`method`	either rejection or regression indicating whether a regression step should be performed during ABC parameter estimation.
`parallel`	logical, indicating whether this function should be run using parallel execution. The default setting is FALSE, meaning that this function will utilize a single core.
`ncores`	a non-negative integer that is required when `parallel` is TRUE. It specifies the number of cores to use for parallel execution.

Details

This function allows users to evaluate the impact of different tolerance rate on the quality of the estimation with ABC and whether a local linear regression algorithm improves the estimates. In subsequent steps, different point estimates of the posterior estimates can be compared with the true values, allowing the users to select the point estimate that leads to lower errors. Thus, performing a leave-one-out cross validation aids in selecting which point estimate is best - the mean, median or mode.

Value

a list with the following elements:

`true`	The parameter values of the simulations that served as validation.
`rej`	a list with the estimated parameter values under the rejection algorithm and using three different point estimates: mode, median and mean. The final entry of the list is the prediction error for each parameter, considering each of those point estimates as the estimated value.
`reg`	if method is regression then this is a list with the estimated parameter values under the regression algorithm and using three different point estimates: mode, median and mean. The final entry of the list is the prediction error for each parameter, considering each of those point estimates as the estimated value.

Examples

```
# load the matrix with parameter values
data(params)
# load the matrix with simulated parameter values
data(sumstats)
# load the matrix with the prior limits
data(limits)

# perform a leave-one-out cross validation for ABC
simulationABC(params = params, sumstats = sumstats, limits, nval = 10,
tol = 0.01, method = "regression")
```

`sim_modelSel` *Leave-one-out cross validation of model selection*

Description

This function performs a simulation study to assess the quality of model selection with ABC. This is done by performing a leave-one-out cross validation via subsequent calls to the function `modelSelect()`.

Usage

```
sim_modelSel(index, sumstats, nval, tol, warning = FALSE)
```

Arguments

index	is a vector of model indices. This can be a a character vector of model names, repeated as many times as there are simulations for each model. This vector will be coerced to factor and it must have the same length as nrow(sumstats) to indicate which row of the sumstats matrix belongs to which model.
sumstats	is a vector or matrix containing the simulated summary statistics for all the models. Each row or vector entry should be a different simulation and each column of a matrix should be a different statistic. The order must be the same as the order of the models in the index vector.
nval	a numerical value defining the the size of the cross-validation sample for each model.
tol	is a numerical value, indicating the required proportion of points nearest the target values (tolerance).
warning	logical, if FALSE (default) warnings produced while running this function, mainly related with accepting simulations for just one of the models, will not be displayed.

Details

One simulation is randomly selected from each model to be a validation simulation, while all the other simulations are used as training simulations. This random simulation is used as the target of the modelSelect() function and posterior model probabilities are estimated.

Please note that the actual size of the cross-validation sample is nval*the number of models. This is because nval cross-validation estimation steps are performed for each model.

Value

a list with the following elements:

cvsamples	is a vector of length nval*the number of models indicating which rows of the sumstat input were used as validation values for each model.
true	a character vector of the true models.
estimated	a character vector of the estimated models.
model.probs	a matrix with the estimated model probabilities. Each row of the matrix represents a different cross-validation trial.
models	a character vector with the designation of the models.

Examples

```
# load the matrix with simulated parameter values
data(sumstats)

# select a random simulation to act as target just to test the function
target <- sumstats[10 ,]

# create a "fake" vector of model indices
# this assumes that half the simulations were from one model and the other half from other model
```

```
# this is not true but serves as an example of how to use this function
index <- c(rep("model1", nrow(sumstats)/2), rep("model2", nrow(sumstats)/2))

# perform a leave-one-out cross validation of model selection
sim_modelSel(index = index, sumstats = sumstats, nval = 10, tol = 0.1)
```

`singleABC` *Parameter estimation with Approximate Bayesian Computation for a single target*

Description

Perform multivariate parameter estimation based on summary statistics using an Approximate Bayesian Computation (ABC) algorithm. This function always uses a rejection sampling algorithm while a local linear regression algorithm might or might not be used.

Usage

```
singleABC(target, params, sumstats, limits, tol, method)
```

Arguments

`target`	a vector with the target summary statistics. These are usually computed from observed data or selected from a random simulation when performing cross-validation.
`params`	is a vector or matrix of simulated parameter values i.e. numbers from the simulations. Each row or vector entry should be a different simulation and each column of a matrix should be a different parameter. This is the dependent variable for the regression, if a regression step is performed.
`sumstats`	is a vector or matrix of simulated summary statistics. Each row or vector entry should be a different simulation and each column of a matrix should be a different statistic. These act as the independent variables if a regression step is performed.
`limits`	is a matrix with two columns and as many rows as there are parameters. Each row should contain the minimum value of the prior for a given parameter in the first column and the maximum value in the second column.
`tol`	is the tolerance rate, indicating the required proportion of points accepted nearest the target values.
`method`	either rejection or regression indicating whether a regression step should be performed during ABC parameter estimation.

Details

To use this function, the usual steps of ABC parameter estimation have to be performed. Briefly, data should have been simulated based on random draws from the prior distributions of the parameters of interest and a set of summary statistics should have been calculated from that data. The same set of summary statistics should have been calculated from the observed data to be used as the `target` input in this function. Parameter values are accepted if the Euclidean distance between the set of summary statistics computed from the simulated data and the set of summary statistics computed from the observed data is sufficiently small. The percentage of accepted simulations is determined by `tol`. This function performs a simple rejection by calling the `rejABC()` function.

When `method` is `regression`, a local linear regression method is used to correct for the imperfect match between the summary statistics computed from the simulated data and the summary statistics computed from the observed data. The output of the `rejABC()` function is used as the input of the `regABC()` function to apply this correction. The parameter values accepted in the rejection step are weighted by a smooth function (kernel) of the distance between the simulated and observed summary statistics and corrected according to a linear transformation.

Value

the returned object is a list containing the following components:

`unadjusted`	parameter estimates obtained with the rejection sampling.
`rej.prediction`	point estimates of the posterior obtained with the rejection sampling.
`adjusted`	regression adjusted parameter values.
`loc.prediction`	point estimates of the regression adjusted posterior.
`ss`	set of accepted summary statistics from the simulations.
`wt`	regression weights.

Examples

```
# load the matrix with parameter values
data(params)
# load the matrix with simulated parameter values
data(sumstats)
# load the matrix with the prior limits
data(limits)

# select a random simulation to act as target just to test the function
target <- sumstats[10 ,]
# we should remove the random simulation from the sumstats and params matrices
sumstats <- sumstats[-10, ]; params <- params[-10, ]

# parameter estimation for a single target
singleABC(target = target, params = params, sumstats = sumstats, limits = limits,
tol = 0.01, method = "regression")
```

summary_modelSelect *Posterior model probabilities*

Description

Extract the posterior model probabilities and obtain a summary of model selection performed with Approximate Bayesian Computation.

Usage

```
summary_modelSelect(object, print = TRUE)
```

Arguments

object	a list created by the modelSelect() function, containing results of model selection with Approximate Bayesian Computation.
print	logical, if TRUE (default), then this function prints the mean models probabilities.

Details

This function produces an easy-to-read output of the model selection step. It also computes the Bayes factors.

Value

a list with two main elements if model selection used the regression algorithm or a single element if only the rejection step was used:

rejection	results of model selection based on the rejection method. This element contains two entries, the first is an object of class numeric with the posterior model probabilities and the second are the Bayes factors between pairs of models.
mnlogistic	results of model selection based on the regression method. This element contains two entries, the first is an object of class numeric with the posterior model probabilities and the second are the Bayes factors between pairs of models.

Examples

```
# load the matrix with simulated parameter values
data(sumstats)

# select a random simulation to act as target just to test the function
target <- sumstats[10 ,]

# create a "fake" vector of model indices
# this assumes that half the simulations were from one model and the other half from other model
# this is not true but serves as an example of how to use this function
index <- c(rep("model1", nrow(sumstats)/2), rep("model2", nrow(sumstats)/2))
```

```
# perform model selection with ABC
mysel <- modelSelect(target = target, index = index, sumstats = sumstats,
tol = 0.01, method = "regression")

# compute posterior model probabilities
summary_modelSelect(object = mysel)
```

`sumstats` *Matrix of summary statistics computed from simulated data*

Description

This data set contains a set of 14 summary statistics computed from data simulated under an isolation with migration model of two populations.

Usage

```
sumstats
```

Format

a matrix with 10000 rows and 14 columns:

Sf fraction of sites fixed between populations.

Sx1 fraction of exclusive sites for the first population.

Sx2 fraction of exclusive sites for the second population.

SS fraction of sites shared between the two populations.

Mean Het1 mean expected heterozygosity of the first population.

Mean Het2 mean expected heterozygosity of the second population.

SD Het1 standard deviation across loci of the mean expected heterozygosity of the first population.

SD Het2 standard deviation across loci of the mean expected heterozygosity of the second population.

Mean HetBet mean heterozygosity between the two populations.

SD HetBet standard deviation across loci of the mean heterozygosity between the two populations.

Mean FST mean pairwise FST between the two populations.

SD FST standard deviation across loci of the mean pairwise FST between the two populations.

FSTQ1 5% quantile of the mean pairwise FST distribution.

FSTQ2 95% quantile of the mean pairwise FST distribution.

Source

simulations performed

CHAPTER 5

Parallel evolution of local adaptation in *Littorina saxatilis* inferred with whole-genome pool-seq data

5.1 INTRODUCTION

Understanding the processes involved in the adaptation of populations to different environments is one of the key goals in evolutionary biology (Seehausen et al., 2014). Divergent natural selection can drive local adaption of populations to different ecological conditions. Importantly, the action of divergent natural selection in distinct environments is a common driving force behind population divergence and the evolution of reproductive isolation, which can ultimately result in new species (Nosil, 2012; Schluter, 2009). By uncovering the evolutionary drivers of adaptive divergence, we can improve our understanding of how biodiversity is generated and maintained, both within and between species. Gene flow between locally adapted populations may oppose local adaptation and the ensuing divergence process (Akerman & Bürger, 2014; Lenormand, 2002). Thus, the spatial context of population divergence and the extent (or possibility) of gene flow play a crucial role in determining the outcome of the speciation process (Smadja & Butlin, 2011). Therefore, a complete understanding of local adaptation and the evolution of reproductive isolation can only be obtained by considering the biogeographical and demographic history of populations (He et al., 2019; Hewitt, 2011). Genomic data and coalescent-modeling approaches have been used to infer such historical sequences of events (Sousa & Hey, 2013). For instance, a recent study by Portinha et al. (2022) used genomic data from wood ant species and coalescent-based modeling to reconstruct their demographic history, uncovering a pattern of divergence with continuous asymmetrical gene flow. Similarly, demographic analysis using whole-genome sequence data of two sympatric stickleback species showed that a long period of allopatry might have promoted the evolution of intrinsic incompatibilities (Yamasaki et al., 2020).

The action of divergent natural selection occurring in multiple populations facing similar environmental conditions can lead to similar phenotypic traits, a phenomenon known as parallel evolution (Bolnick, Barrett, Oke, Rennison, & Stuart, 2018; Elmer & Meyer, 2011). This phenotypic parallelism can be explained by different processes, ranging from convergent evolution acting on *de novo* mutations arising in different populations, to selection based on standing genetic variation leading to both phenotypic and genomic parallelism. Given the opposing effect of gene flow and the randomness of other processes such as genetic drift, parallel evolution stands as one of the most convincing forms of evidence for the significant and predictable role of natural selection in driving adaptive evolution (Faria et al., 2014; Nosil, 2012). Moreover, when adaptation to similarly divergent pairs of environments leads to the repeated evolution of reproductive isolation (i.e., parallel speciation), it underscores the potential of natural selection in driving the speciation process (Johannesson, 2001). However, environments can differ along multiple axes, potentially leading replicate populations to evolve along trajectories that deviate from the parallel expectation

and vary not only in the magnitude but even in the direction of phenotypic divergence (Langerhans, 2018; Oke, Rolshausen, LeBlond, & Hendry, 2017). Additionally, parallelism at the phenotypic level does not equate to parallelism at the genomic level given that different genetic changes can underlie the same phenotypic solution (Bainbridge et al., 2020; Poore et al., 2022).

Examples of parallelism at the phenotypic level associated with varying degrees of parallelism at the genomic level have been found in populations of the beach mouse, *Peromyscus polionotus* (Steiner, Römpler, Boettger, Schöneberg, & Hoekstra, 2009) and the American lake whitefish, *Coregonus clupeaformis* (Laporte et al., 2015). Another noteworthy example is the marine threespine stickleback, *Gasterosteus aculeatus*, that has independently colonized many freshwater sites. At each of those sites, the same pattern of reduced armour evolution has been observed (Colosimo et al., 2005; Jones et al., 2012). This pattern is accompanied by parallelism at the underlying major quantitative trait locus (EDA; Colosimo et al. 2005), representing a textbook example of phenotypic and genomic parallelism. Despite this, there are exceptions to this parallelism (Fang, Kemppainen, Momigliano, Feng, & Merilä, 2020), with different levels of lateral plating observed (Hansson, Fischer, Mazzarella, Voje, & Vøllestad, 2016) and the absence of genomic parallelism in the reduction of pelvic armour in replicate freshwater populations (Chan et al., 2010). Such deviations can be explained by non-adaptive processes such as genetic drift or differences in gene flow between habitats (Ferchaud & Hansen, 2016; Hendry & Taylor, 2004) or they might reflect adaptive responses to habitat-specific differences among similar habitats (DeFaveri, Shikano, Shimada, Goto, & Merilä, 2011). Interestingly, results from a meta-analysis suggest that the probability of genomic parallelism depends on the age of the populations and their access to the same pool of standing genetic variation (Conte, Arnegard, Peichel, & Schluter, 2012). Indeed, genomic parallelism is expected to be more prevalent in populations that share a common ancestry and are in close geographic proximity (Bohutínská et al., 2021; Rennison, Delmore, Samuk, Owens, & Miller, 2020). This is not surprising because populations with a recent shared evolutionary history and higher connectivity have access to a common pool of standing genetic variation. Conversely, habitat specific changes in the direction of selection or in the selective pressure itself can lead to nonparallel patterns of phenotypic divergence (Kaeuffer, Peichel, Bolnick, & Hendry, 2012; Thurman et al., 2023).

Interpreting the pattern of parallel local adaptation in the presence of contemporary gene flow is not straightforward since it can arise as a consequence of markedly different historical sequences of events (Johannesson et al., 2010). In particular, two broadly contrasting scenarios can give rise to this parallel pattern. In one scenario, the initial adaptive divergence evolves once (in sympatry or allopatry), with subsequent colonization of similar pairs of environments by differentially adapted forms. Alternatively, divergence occurs repeatedly and independently in multiple localities (Butlin et al., 2014; Johannesson et al., 2010). Interestingly, even if these repeated origins of phenotypic

divergence are independent, there is a possibility that the same adaptive alleles are used, possibly originating from standing genetic variation or gene flow across localities. The first step towards distinguishing between these alternatives requires the use of putatively neutral genetic markers to establish the demographic history of the populations. This can be achieved by using genomic data to inform coalescent-based models of parallel local adaptation integrated in an model-based inference framework such as Approximate Bayesian Computation (ABC). However, this would require sampling and sequencing multiple populations across a wide geographical range, which might quickly become cost prohibitive. A cost-effective approach is to pool DNA from multiple individuals and sequence them together (pool-seq), which makes it possible to quantify and characterize the genetic variation within and between populations (Schlötterer, Tobler, Kofler, & Nolte, 2014). However, estimates of allele frequency obtained with pool-seq can be affected by variations in DNA concentration of the pooled individuals and amplification/sequencing efficiency, on top of more common sources of errors such as depth of coverage variation across sites and sequencing errors. Nevertheless, pool-seq allows for the analysis of more individuals with similar or more precise allele frequency estimates compared to individual-based sequencing, making it suitable for inferring population genetic structure and estimating effective population sizes and divergence times (Collin et al., 2021; Futschik & Schlötterer, 2010).

Despite a general awareness that different sequences of events can lead to the same pattern of parallel divergence, relatively few studies have explicitly tested the parallel origin scenario against alternative hypotheses. Here we apply a recently developed approach (Carvalho, Morales, Faria, Butlin, & Sousa, 2023) to a large pool-sequencing dataset of the rough periwinkle, *Littorina saxatilis*, to contrast alternative scenarios for the origin of locally adapted ecotypes in this species. This rocky-shore gastropod has a wide distribution, covering most of the North Atlantic despite having a low lifetime dispersal (Reid, 1996). Across the species distribution, two ecotypes can be found in close proximity: a smaller morph with a thin shell and a larger aperture that allows it to withstand heavier wave exposure and a larger morph with a thicker shell and a small aperture that confers more protection against crab predation (Johannesson et al., 2010). These ecotypes have been the focus of numerous studies (e.g. Conde-Padín, Cruz, Hollander, and Rolan-Alvarez 2008; Grahame, Wilding, and Butlin 2006) and evidence suggests that sexual selection and assortative mating are also involved in reproductive isolation (Perini, Rafajlović, Westram, Johannesson, & Butlin, 2020). Recently, multiple chromosomal rearrangements associated with the ecotypes (Faria et al., 2019) have been connected to variation linked with adaptive traits (Koch et al., 2021). However, some studies indicate that loci outside inversions may also contribute to the divergence (Kess & Boulding, 2019; Koch, Ravinet, Westram, Johannesson, & Butlin, 2022; Westram, Faria, Johannesson, & Butlin, 2021). A common thread of most *L. saxatilis* studies is their focus on only one of three

European regions: Galicia in northwest Spain, the west coast of Sweden and the northeast coast of England. An independent and parallel origin of these ecotypes has often been invoked (Quesada, Posada, Caballero, Morán, & Rolán-Alvarez, 2007). However, some of these arguments were questioned (Butlin, Galindo, & Grahame, 2008) and it has been found that genomic parallelism is limited (Koch et al., 2022; Ravinet et al., 2016). Furthermore, the Iberian populations share a demographic history that is clearly different from the remaining European populations (Blakeslee et al., 2021; Morales et al., 2019; Panova et al., 2011), while the Swedish and British populations have probably shared a post-glacial colonization history. A single study (Butlin et al., 2014) has examined and compared different models of the demographic history of the ecotypes across a wide geographical range. However, this study relied on a limited set of genetic data, specifically mitochondrial and nuclear DNA sequences combined with amplified fragment length polymorphism data. Building upon previous research, we use whole-genome data to compare and contrast two explicit scenarios of ecotype formation. Our aim is to determine whether *L. saxatilis* ecotype formation on a large scale can be attributed to a single event or a series of parallel events.

5.2 MATERIAL AND METHODS

5.2.1 MODELS OF ECOTYPE FORMATION: SINGLE VS. PARALLEL

We compared two contrasting scenarios of ecotype formation (Figure 5.1). We represented these scenarios using a four-population model, wherein the four extant populations correspond to the two distinct ecotypes observed at two separate geographical locations. We considered 32 parameters of interest for both scenarios. These include the size of the four present-day populations, denoted as N_1 to N_4, as well as the two ancestral populations, NA_1 and NA_2. Additionally, the time of the recent split events between populations in the first location (RS_1) and between populations in the second location (RS_2), as well as the ancient split event (T_{As}) are all measured in generations. We also considered migration rates between the divergent ecotypes at each location, with m_{12} and m_{21} representing the migration from N_1 to N_2 and from N_2 to N_1, respectively. Similarly, the migration rates m_{34} and m_{43} represent the migration (forward in time) from N_3 to N_4, and from N_4 and N_3, respectively. We also considered migration between similar ecotypes inhabiting different locations and between ancestral populations. Note that to avoid increasing the number of parameters under consideration, the migration rates between similar ecotypes and between ancestral populations were assumed to be identical regardless of the direction. However, the scaled migration rate $4Nm$ can be different. In order to estimate the timing of events, we considered as a parameter the time

interval between the older recent split event and the ancient split ($\Delta_s = T_{As} - RS$). We incorporated the existence of bi-directional or uni-directional barrier loci in our model. For the bi-directional barrier, we included a parameter (P_{no}) to control the proportion of loci where there was no migration in either direction between the divergent ecotypes. As for the uni-directional barrier, we included parameters P_{cw} and P_{wc} representing the proportion of loci where no migration occurred from Crab to Wave or from Wave to Crab, respectively. We assumed that these parameters (P_{no}, P_{cw}, and P_{wc}) were the same across both localities. Depending on the topology, the four-population model can represent either a single origin scenario where ecotypes originate in distinct locations before dispersing and colonizing two geographic areas (Figure 5.1A), or a parallel origin scenario where each location is colonized independently, leading to the independent divergence of the distinct ecotypes (Figure 5.1B).

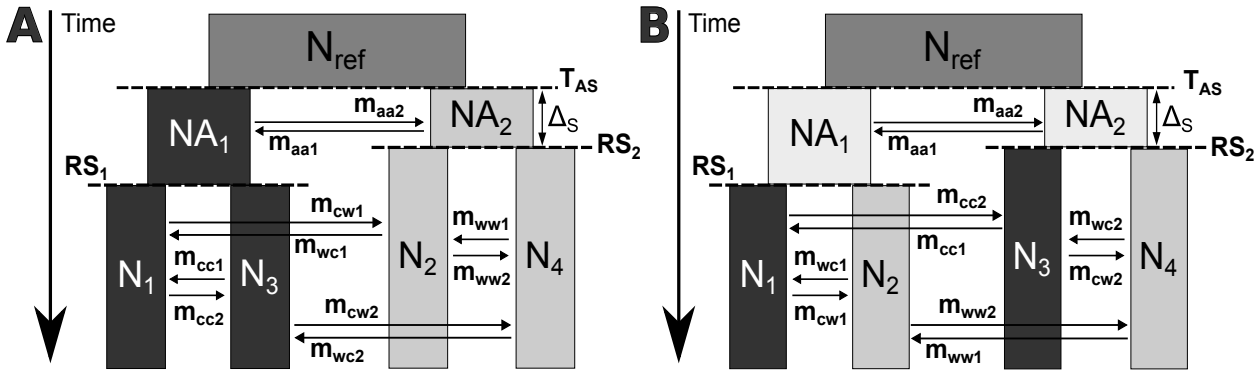


Figure 5.1: Demographic models for the single (A) and parallel (B) ecotype formation scenarios. Dark shading indicates one of the ecotypes, light shading the other ecotype. Parameters used were: N_{ref} - effective size of the ancestral population, NA_1 and NA_2 - size of the two ancestral populations, N_1 - N_4 - sizes of the present-day populations, RS_1 - time of separation of the present-day populations at the first location (in generations), RS_2 - time of separation of the present-day populations at the second location (in generations), T_{As} - time of the ancient split event (in generations), Δ_s - time interval between the two split events (in generations), m_{cw1} - probability per generation that an individual migrates from Crab N_1 to Wave N_2 (forward in time), m_{wc1} - probability per generation that an individual migrates from Wave N_2 to Crab N_1 (forward in time), m_{cw2} - probability per generation that an individual migrates from Crab N_3 to Wave N_4 (forward in time), m_{wc2} - probability per generation that an individual migrates from Wave N_4 to Crab N_3 (forward in time), m_{cc1} - probability per generation that an individual migrates from Crab N_3 to Crab N_1 (forward in time), m_{cc2} - probability per generation that an individual migrates from Crab N_1 to Crab N_3 (forward in time), m_{ww1} - probability per generation that an individual migrates from Wave N_4 to Wave N_2 (forward in time), m_{ww2} - probability per generation that an individual migrates from Wave N_2 to Wave N_4 (forward in time), m_{aa1} - probability per generation that an individual migrates from NA_2 to NA_1 (forward in time) and $maa2$ - probability per generation that an individual migrates from NA_1 to NA_2 (forward in time).

5.2.2 SIMULATION OF POOL-SEQ DATA

We simulated gene trees using coalescent theory and the *scrm* simulator (Staab, Zhu, Metzler, & Lunter, 2015). Mutations followed the infinite sites model, with a mutation rate (μ) per site per generation. Gene trees were simulated for each locus with the same sample size as the number of individuals in the pool, which corresponds to 100 diploid individuals (200 haplotypes) for each population. During gene tree simulations, the actual haplotypes of all individuals in the pool were assumed to be known, and the effect of pooling was simulated later. To simulate genotypes, random mating was assumed within each population, pairing haplotypes randomly at each locus to obtain genotypes for each biallelic SNP.

To model allele frequencies at biallelic SNPs obtained with pool-seq we followed a series of steps detailed in Carvalho et al. (2023). We simulated pool-seq data using parameters obtained from real *L. saxatilis* pool-seq data (see below). Briefly, we modelled the depth of coverage at each SNP (number of reads per site) using a negative binomial distribution (Sampson, Jacobs, Yeager, Chanock, & Chatterjee, 2011). This distribution is defined by the mean and variance of the coverage of the real data. DNA extraction for the *L. saxatilis* pool-seq data was performed for batches of five individuals by combining foot muscle tissue from five snails in one tube (Morales et al., 2019). The DNA from each batch was then pooled to create the final pools for sequencing. Thus, for each SNP, we modelled the possibility of uneven contributions between those batches of five individuals. This uneven contribution was modeled as a multinomial-Dirichlet distribution, where the number of reads from each batch is given by a multinomial with expected proportion of reads from each batch following a Dirichlet distribution.

We then modelled the heterogeneity in the contribution of each individual within each batch, following the same approach: number of reads from each individual as a multinomial distribution, where the expected proportion of reads from each individual followed a Dirichlet distribution. In both cases, the dispersion of the Dirichlet distribution is determined by explicit pool-seq error parameters (Carvalho et al., 2023; Gautier et al., 2013). These parameters influence the variance of the proportion of reads contributed by each batch or individual, representing the random variation in the batch or individual contribution to the pool (Carvalho et al., 2023; Gautier et al., 2013). Higher values of the pool-seq error parameter lead to increased variance, resulting in greater heterogeneity in the contributions of both individual and batches. This model assumes an equal expected contribution of reads from all individuals, with errors arising from unequal contributions accounted for through dispersion parameters that affect the variance. To model the potential impact of sequencing and mapping errors, we considered the possibility that the allele observed in the pool's reads might not corresponded to the true genotype of the sampled individuals. For each individual, the

count of reads with the alternative allele was modeled using a binomial distribution. This distribution assumed that, with a given error rate, the reference allele could be incorrectly identified as the alternative allele, or vice versa. We then defined which of the two alleles (i.e., reference or alternative) corresponded to the minor-allele and applied a minor-allele filter, discarding all SNPs with less than two minor-allele reads. We did not consider sequencing errors occurring at invariant sites, as the simulated pool-seq data only included polymorphic sites. Furthermore, any such errors would probably be eliminated by the minor-allele filter. To mimic real *L. saxatilis* pool-seq data (see below), we applied a coverage-based filter, removing all sites with a coverage below 14x or above 204x. Finally, we estimated the minor-allele frequencies as the proportion of reads with the minor-allele.

5.2.3 ABC IMPLEMENTATION

Our ABC implementation, using a rejection algorithm, followed several steps (Carvalho et al., 2023). In short, we started by sampling demographic and pool-seq parameters from prior distributions (Table 5.1). Subsequently, we simulated genotypes for each individual at L loci using coalescent theory and the sampled demographic history parameters. We then simulated pool-seq data for each biallelic SNP and implemented filters based on depth of coverage and minor-allele counts. Afterwards, we computed summary statistics for both the observed and simulated data and calculated the Euclidean distance between the observed and simulated summary statistics, ensuring standardization so that that all summary statistics have the same mean and variance. Next, we rejected parameters that led to summary statistics with distances exceeding a specified tolerance threshold (i.e., rejection step). The accepted parameters from the rejection step are used to approximate the posterior distribution of the parameters. Finally, we applied a post-processing regression to adjust the accepted parameter values (Beaumont, Zhang, & Balding, 2002).

We chose a set of statistics (Table A5.1) to capture the patterns of relative diversity and differentiation among populations (Fraïsse et al., 2021; Jay, Boitard, & Austerlitz, 2019). These statistics were computed exclusively for polymorphic sites across all populations. Specifically, we considered the following statistics: (i) expected heterozygosity, both within each population and between all pairs of populations (Nei & Roychoudhury, 1974); (ii) Pairwise F_{ST} values, computed for all possible population pairs (Bhatia, Patterson, Sankararaman, & Price, 2013); (iii) proportion of SNPs exhibiting fixed differences between populations (Fraïsse et al., 2021); (iv) proportion of SNPs exclusive to each individual population (Fraïsse et al., 2021) and (V) various D-statistics using different combinations of the P1, P2, and P3 populations (adapted from Malinsky, Matschiner, & Svardal, 2021). To capture the variation across loci, we considered the mean and standard de-

viation of the aforementioned statistics. We also considered the 5% and the 95% quantiles of F_{ST} because they are expected to provide insights into the impact of barriers to gene flow. In total, we considered a set of 61 summary statistics for both scenarios of ecotype formation (Table A5.1). Notably, all of these summary statistics are relative measures of diversity and differentiation (e.g., F_{ST}) and their values are contingent upon the relative branch lengths of coalescent trees.

Table 5.1: **Prior distributions and their ranges for each parameter.** Parameters are presented for the single and parallel origin scenarios. n_i - relative sizes of the extant populations (n_1, n_2, n_3, n_4); na_i - relative sizes of the ancestral populations (na_1, na_2); rs - relative time of the recent split event; δ_s - relative time interval between rs and the ancient split event (t_{As}); ε_{pool} - experimental error introduced by the pooling procedures; ε_{seq} - error associated with sequencing and mapping errors; m_{cw} - probability per generation that an individual migrates from the the Crab population to the Wave population (forward in time), m_{wc} - probability per generation that an individual migrates from the Wave population to the Crab population (forward in time); m_{cc} - probability per generation that an individual migrates between Crab populations from different geographic locations (forward in time); m_{ww} - probability per generation that an individual migrates between Wave populations from different geographic locations (forward in time); m_{aa} - probability per generation that an individual migrates between ancestral populations (forward in time); P_{cw} - proportion of the simulated loci where no migration occurs from the Crab to the Wave population; P_{wc} - proportion of the simulated loci where no migration occurs from the Wave to the Crab population and P_{no} - proportion of the simulated loci where no migration occurs between ecotypes.

		Single origin		Parallel origin	
parameter	distribution	minimum	maximum	minimum	maximum
n_i	log-uniform	0.1	3	0.1	3
na_i	log-uniform	0.1	3	0.1	3
rs	uniform	0	3	0	3
δ_s	uniform	0	3	0	3
ε_{pool}	uniform	5	250	5	250
ε_{seq}	uniform	0.0001	0.01	0.0001	0.01
m_{cw}	uniform	10^{-13}	10^{-3}	10^{-13}	10^{-3}
m_{wc}	uniform	10^{-13}	10^{-3}	10^{-13}	10^{-3}
m_{cc}	uniform	10^{-13}	10^{-3}	10^{-16}	10^{-4}
m_{ww}	uniform	10^{-13}	10^{-3}	10^{-16}	10^{-4}
m_{aa}	uniform	0	10^{-8}	10^{-16}	10^{-4}
P_{cw}	beta	0	0.2	0	0.2
P_{wc}	beta	0	0.2	0	0.2
P_{no}	beta	0	0.1	0	0.1

Given that all summary statistics depend on the relative branch lengths of coalescent trees, we could increase the efficiency of our simulations by inferring relative demographic parameters scaled by the ancestral effective population size N_{ref}. This approach allowed us to estimate relative effective sizes, such as $n_1 = N_1/N_{ref}$, relative divergence times, such as $\delta_s = \Delta_s/4N_{ref}$, and scaled migration rates, such as $4N_1m_{21}$. It should be noted that relative parameters are denoted with

lowercase letters (e.g., n_1), while absolute parameters are represented with uppercase letters (e.g., N_1). Scaled migration rates (forward in time) specify which population is receiving immigrants, indicated by the subscript next to N. We estimated relative parameters by performing coalescent simulations, with the ancestral effective population size set at $N_{ref} = 25000$ and the mutation rate at $\mu = 1.5 \times 10^{-8}$ per site per generation. These parameter values were chosen based on previous *L. saxatilis* studies (Butlin et al., 2014). In order to obtain absolute parameter estimates, we used a re-scaling factor defined as $f = obs[S]/E[S]$. This factor depends on the observed number of SNPs ($obs[S]$) and on the expected number of SNPs calculated using the parameter estimates of a given scenario ($E[S]$). Under the assumptions of the infinite sites mutation model, the expected number of segregating sites ($E[S]$) was determined by considering the expected total branch length ($E[T]$), the mutation rate per site (μ), and the number of sites (S). This relationship can be expressed as $E[S] = E[T] \cdot \mu \cdot S$ (Hudson, 1990). To obtain $E[T]$ we simulated 100,000 gene trees, using the parameter estimates from the ecotype formation scenario that received the highest posterior support. The absolute effective population sizes and times of events in generations were then obtained by multiplying the rescaling factor with the corresponding relative values, i.e., $N_e = f \times n_e$ and $T_s = f \times t_s$, respectively.

5.2.4 SIMULATION STUDY

To mitigate the computational load associated with simulating entire genomes, we instead simulated sets of L loci. We assumed complete independence and allowed for free recombination between all pairs of loci ($r_b = 0.5$). Additionally, within each locus, we assumed no recombination ($r_w = 0.0$). We performed 5×10^5 simulations for each scenario of ecotype formation. In each simulation, we generated $L = 300$ independent loci, each consisting of $b = 2000$ base pairs, sampling 100 diploid individuals from each population. Pool-seq data were generated for each population by assuming 20 pools, with each pool containing 5 individuals. We assumed that all loci within a subset (i.e., each set of $L = 300$ independent loci) share a common demographic history. However, we modeled a migration rate of zero for a proportion of loci, P_{no}, P_{cw} and P_{wc}, to account for the influence of selection effects resulting from barrier loci. Most of the parameter values were randomly sampled from uniform or log-uniform prior distributions, as outlined in Table 5.1. The proportions of the genome without migration (P_{no}, P_{cw} and P_{wc}) were sampled from a Beta distribution specifically designed to reflect a low proportion of loci without migration as a prior assumption. We truncated the distribution of these parameters, replacing values below the minimum prior range or above the maximum prior range with the corresponding minimum or maximum values (Table 5.1).

We performed a leave-one-out cross-validation (Csilléry, François, & Blum, 2012) to assess the accuracy of parameter estimation and model choice. Briefly, we randomly selected a simulation and used its summary statistics as pseudo-observed data. The remaining simulations were then utilized to infer the parameters associated with the selected simulation. This process was repeated for a total of n pseudo-observed datasets. In this context, we define accuracy as the proximity of a specific point estimate to the true parameter value. We computed the prediction error for parameter inferences as:

$$\varepsilon_{pred} = \frac{1}{n} \cdot \frac{\sum_{i=1}^{n} (\hat{\Theta}_i - \Theta_i)^2}{var(\Theta)} \tag{5.1}$$

where Θ_i represents the true parameter value of the i^{th} pseudo-observed dataset, $\hat{\Theta}_i$ denotes the estimated parameter value, and $var(\Theta)$ corresponds to the variance of the true parameter values. We evaluated the prediction error for parameter inference with $n = 5000$ pseudo-observed datasets and considering three different point estimates (mode, median and mean of the posterior distribution, at two tolerance values (0.005 or 0.01). To evaluate the prediction error for model choice, we used $n = 1000$ pseudo-observed datasets. We used two posterior probability thresholds to define the estimated model for each pseudo-observed dataset. The first threshold was set at 0.5, assigning a dataset to the model with a posterior probability higher than 0.5. For a more stringent criterion, we used a threshold of 0.9, only assigning a dataset to a model if its posterior probability exceeded this threshold. If the posterior probability did not meet the 0.9 threshold, it was classified as "unclear".

5.2.5 POOL-SEQ DATA FROM *LITTORINA SAXATILIS*

We contrasted scenarios of ecotype formation using previously published pool-seq data (Morales et al., 2018) from *L. saxatilis* populations. We compared populations sampled from Sweden (Arsklovet) and Spain (Burela). At each of those countries, 100 females from the Crab ecotype and another 100 females from the Wave ecotype were sequenced in separate pools (Morales et al., 2019). DNA was extracted from groups of individuals by combining foot muscle tissue samples from five snails in one tube. Subsequently, the reads obtained were trimmed using Trimmomatic v0.36 (Bolger, Lohse, & Usadel, 2014) and mapped against the reference genome of *L. saxatilis*, which was generated from one individual of the Crab ecotype (Westram et al., 2018). The mapping process was performed using CLC v5.0.3 from Qiagen Bioinformatics (`www.qiagenbioinformatics.com`). After mapping the reads, only those with a mapping score higher than Q20 were retained for further analysis. The resulting BAM files were processed using SAMtools v1.3.1 (Danecek et al., 2021),

BEDtools v2.25.0 (Quinlan & Hall, 2010), and Picard tools v2.7.1. For each set of BAM files, reads with a base quality lower than 30 that mapped to contigs shorter than 500 base pairs were filtered out. To minimize potential artifacts, sites with a coverage lower than 14x or higher than 204x were excluded. This allowed us to discard low-coverage sites that lacked reads for most individuals ($<$14x), as well as sites in potentially repetitive or duplicated regions that led to unusually high coverage ($>$204x). Additionally, we excluded sites that had fewer than two minor-allele reads observed across all populations.

Recently, the significance of chromosomal inversions in the adaptive divergence of *L. saxatilis* has been highlighted (Faria et al., 2019; Koch et al., 2021; Morales et al., 2019). Each inversion is likely to have its own unique evolutionary history, which can be influenced by a range of demographic and selective processes, including divergent and balancing selection. It is important to note that the evolutionary dynamics of inversions may differ from the overall population history. Consequently, to ensure unbiased estimates, it would be necessary to use inversion-specific inference methods that take into account specific features, such as variable recombination rates between homozygotes and heterozygotes. As our primary goal was to infer the demographic history rather than the specific dynamics of inversions, we opted to remove regions that could potentially be associated or linked with the reported inversions (Westram et al., 2021). The list of retained and excluded contigs can be found in the Supplementary information. By removing these regions, we focused the analysis on genomic regions less likely to be influenced by inversion-specific processes, thus ensuring a more conservative inference of the neutral demographic history of those *L. saxatilis* populations. Due to uncertainty about the precise positions of breakpoints for many inversions we made the decision to exclude a total of 3671 contigs located within inversions or buffer regions. This accounts for approximately 3.3% of the entire genome pool-seq dataset. These excluded contigs were distributed across the genome, with roughly one-third of them located in chromosomes 10 and 12. Furthermore, we removed all contigs that did not map to known collinear regions (Westram et al., 2018). Consequently, around 80% of the remaining contigs were excluded from our analysis, leaving only the known collinear regions.

To reduce computational burden, we saved parameter and summary statistic tables from the simulation study, which were reused to perform model choice and estimate parameters for the *L. saxatilis* populations. We chose the scenario with the highest posterior probability as the preferred model and performed parameter inference for the selected scenario. Following our strategy of using subsets of loci, we treated each contig in the *L. saxatilis* dataset as an independent locus. To obtain posterior probabilities, we combined 1000 subsets, each consisting of 300 randomly selected loci ($L = 300$). The selection process for each subset involved randomly choosing 300 contigs without replacement. From these selected contigs, we then randomly extracted a window of

$b = 2000$ base pairs. We computed summary statistics for each subset consisting of the 300 selected windows. Given the reduced number of mapped collinear contigs available ($\sim$ 8000), contigs were reused in different subsets. Nevertheless, due to the high probability of selecting different 2000 base pairs windows, each subset likely represents a distinct combination of loci. The independent posterior samples obtained from the 1000 subsets of loci were merged, taking into account the distance between the mean summary statistics of each subset and the overall mean across all loci in the genome. To achieve this, we used the Epanechnikov kernel, which assigns greater weight to subsets of loci with means that are closer to the overall mean. This approach was designed to minimize the influence of outlier subsets of loci on the posterior estimates, given that demographic history is expected to affect all loci similarly across the genome. All steps were performed using the R package *poolABC* (Carvalho et al., 2023). Here we report the mean of the merged posterior distribution as a point estimate and the 95% credible intervals using weighted quantiles. To re-scale the relative parameters, we calculated the number of SNPs per window, considering that all remaining sites were monomorphic. We converted the time of events from generations to years by assuming a generation time of 0.5 years (Butlin et al., 2014).

5.3 RESULTS

5.3.1 ACCURACY OF ABC POINT ESTIMATES

Results from the simulation study showed that prediction errors were lower, for all parameters, when using the mean or median with the regression-based adjustment (supplementary Tables A5.2 and A5.3). Thus, unless specified, hereafter we summarize results obtained using the regression-based adjustment and the mean as a point estimate, with a tolerance of 0.01.

For the relative effective sizes of present-day populations (Figure 5.1), the prediction errors ranged between 0.190 and 0.193 for the single origin scenario, and between 0.119 and 0.135 for the parallel origin scenario. These results indicate that the means of the posterior distributions provide accurate point estimates of these parameters. When considering the relative sizes of the ancestral populations (absolute values indicated by NA_1 and NA_2 in Figure 5.1), the prediction errors were similar in the two scenarios of ecotype formation (Table 5.2). In both models, the prediction error for the relative sizes of ancestral populations, na_1 and na_2, was higher (ranging from 0.863 to 0.911) than the error observed for the present-day populations.

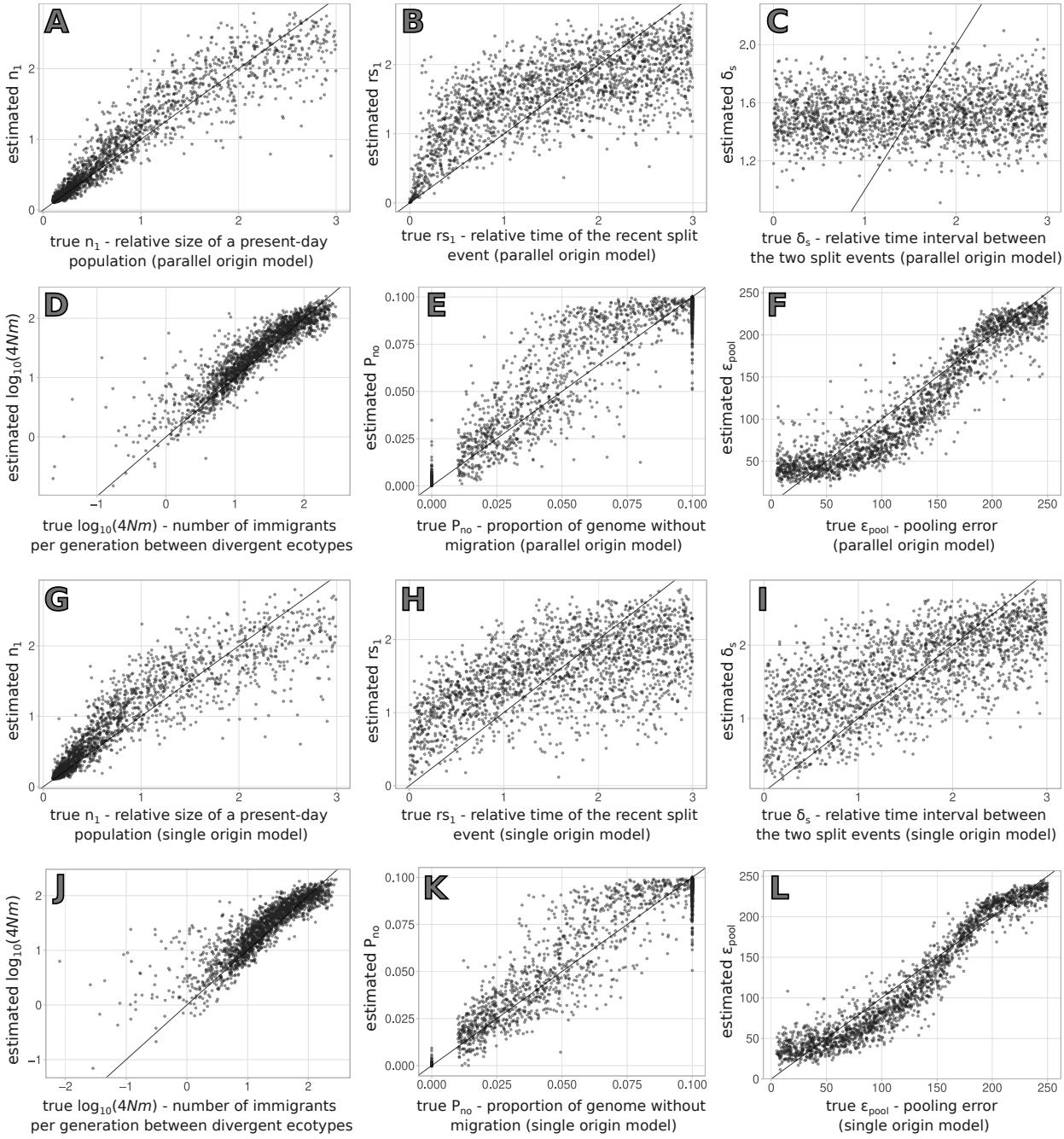


Figure 5.2: Results of the cross-validation for parameter estimation. The y-axis displays the estimated values, plotted against the true parameter values on the x-axis. Estimates correspond to the mean of the posterior obtained with a tolerance rate of 0.01. The top 6 panels (A-F) show the results for the parallel origin scenario, while the bottom 6 panels (G-L) correspond to the single origin scenario. Parameters shown here are: A and G - relative size of a present-day population (n_1); B and H - relative time of the recent split event in the first location (rs_1); C and I - relative time interval between the two split events (δ_s); D and J - average number of immigrants per generation in log_{10} scale; E and K - proportion of the simulated loci where no migration occurs between ecotypes (P_{no}); F and L - pooling error.

This suggests that point estimates are much less accurate for the ancestral effective sizes compared to the present-day populations (Figure 5.2 and supplementary Figures A5.1 and A5.2). The

prediction errors for the relative time of the recent split event between populations was lower in the parallel scenario than in the single scenario. Specifically, at the first location (rs_1), the prediction error was 0.505 for the parallel scenario and 0.616 for the single scenario. Likewise, at the second location (rs_2), the prediction error was 0.513 for the parallel scenario and 0.631 for the single scenario. In contrast, the prediction error for the relative time interval between split events (δ_s) was lower for the single origin scenario (0.547) compared to the parallel scenario (0.992).

Table 5.2: **Prediction errors for relative parameters estimation.** Prediction errors were computed using the mean of the posterior distribution, obtained after the regression adjustment and a tolerance of 0.01. n_1 to n_4 - relative population sizes of the extant populations; na_1 and na_2 - relative population sizes of the ancestral populations; rs_1 and rs_2 - relative time of the recent split event; δ_s - relative time interval between rs and the ancient split event (t_{As}); ε_{pool} - experimental error introduced by the pooling procedures; ε_{seq} - error associated with sequencing and mapping errors; m_{cw1}, m_{cw2} - probability per generation that an individual migrates from the N_1 or N_3 (Crab) population to the N_2 or N_4 (Wave) population (forward in time), m_{wc1}, m_{wc2} - probability per generation that an individual migrates from the N_2 or N_4 (Wave) population to the N_1 or N_3 (Crab) population (forward in time); m_{cc} - probability per generation that an individual migrates from one Crab population to the other (forward in time); m_{ww} - probability per generation that an individual migrates from one Wave population to the other (forward in time); m_{aa} - probability per generation that an individual migrates from one ancestral population to the other (forward in time); $4N_i m_{cw1}$ and $4N_i m_{wc1}$ - average number of immigrants per generation ($4Nm$) from Crab to Wave and from Wave to Crab (respectively) in the first location; $4N_4 m_{cw2}$ and $4N_3 m_{wc2}$ - equivalent immigration rates at the second site; $4N_i m_{cc1}$ and $4N_i m_{cc2}$ - average number of immigrants per generation from the Crab population in the second location to the first and vice-versa (respectively); $4N_i m_{ww1}$ and $4N_i m_{ww2}$ - average number of immigrants per generation from the Wave population in the second location to the first and vice-versa (respectively); $4NA_1 m_{aa1}$ and $4NA_2 m_{aa2}$ - average number of immigrants per generation from na_2 to na_1 and vice-versa (respectively); P_{cw} - proportion of the simulated loci where no migration occurs from the Crab to the Wave population; P_{wc} - proportion of the simulated loci where no migration occurs from the Wave to the Crab population and P_{no} - proportion of the simulated loci where no migration occurs between ecotypes.

parameter	single	parallel	parameter	single	parallel
n_1	0.182	0.119	m_{ww}	0.196	0.148
n_2	0.182	0.132	m_{aa}	1.021	0.977
n_3	0.193	0.135	$4N_i m_{cw1}$	0.330	0.287
n_4	0.190	0.129	$4N_i m_{wc1}$	0.319	0.266
na_1	0.863	0.883	$4N_i m_{cw2}$	0.333	0.308
na_2	0.911	0.907	$4N_i m_{wc2}$	0.332	0.326
rs_1	0.616	0.505	$4N_i m_{cc1}$	0.254	0.143
rs_2	0.631	0.513	$4N_i m_{cc2}$	0.263	0.146
δ_s	0.547	0.992	$4N_i m_{ww1}$	0.266	0.133
ε_{pool}	0.082	0.138	$4N_i m_{ww2}$	0.252	0.140
ε_{seq}	0.021	0.022	$4NA_1 m_{aa1}$	0.934	0.979
m_{cw1}	0.469	0.449	$4NA_2 m_{aa2}$	0.952	0.981
m_{cw2}	0.472	0.465	P_{cw}	0.800	0.827
m_{wc1}	0.451	0.427	P_{wc}	0.753	0.814
m_{wc2}	0.468	0.480	P_{no}	0.094	0.140
m_{cc}	0.211	0.154			

Although we parameterized our models using prior immigration rates m_{ij}, which represent the probability of a lineage migrating from population i to j in each generation, we focus here on the average number of immigrants per generation denoted as $4N_j m_{ij}$. In this context, N_j represents the effective size of the population that receives immigrants. This measure takes into consideration both migration, which is proportional to m_{ij}, and genetic drift, which is proportional to N_j. Notably, when $4N_j m_{ij} > 1$, migration occurs at a higher rate than drift. Prediction errors for average number of immigrants per generation between the divergent ecotypes were similar for the two scenarios of ecotype formation, ranging from 0.266 to 0.333 (Table 5.2).

Regarding the number of immigrants between populations of the same ecotype inhabiting different locations, the prediction errors were slightly lower in the parallel scenario ($\sim$ 0.150) compared to the single origin scenario ($\sim$ 0.200). The prediction errors for the number of immigrants between ancestral populations were high in both scenarios, ranging from 0.934 to 0.981. In contrast to the high prediction error observed for the proportion of loci without migration from the Crab to the Wave ecotype (P_{cw}) and vice versa (P_{wc}), which was approximately 0.8 in both scenarios, the proportion of loci without migration in either direction (P_{no}) was accurately estimated. The prediction error for P_{no} was 0.140 in the parallel scenario and 0.094 in the single origin scenario (Table 5.2).

Finally, and although our primary goal was to infer demographic parameters while explicitly modeling pool-seq data and treating pooling and sequencing errors as nuisance parameters, we also provide the prediction error for these parameters. The accuracy of the inference of the pooling error was similar to that of other parameters, with errors ranging from 0.082 to 0.138 (Table 5.2). This parameter was reasonably well estimated by the posterior mean when simulations were done with pooling errors above 150% (Figure 5.2). For the sequencing error, prediction error was low ($\sim$ 0.02, Table 5.2 and supplementary Figures A5.1 and A5.1), probably because of the number of individuals considered here.

5.3.2 ACCURACY OF MODEL CHOICE

The results of our simulation study demonstrate that the combination of ABC with pool-seq data effectively enables to distinguish between the two scenarios of ecotype formation considered here. Using a 50% posterior probability threshold, the correct model was successfully inferred for 988 out of 1,000 pseudo-observed datasets simulated under the parallel origin scenario, with a mean posterior probability of 0.976. Similarly, for the single origin scenario, the correct model was inferred for 989 pseudo-observed datasets, with a mean posterior probability of 0.995 (Figure 5.3A).

In cases where the model with the highest posterior probability was incorrect, its posterior probability was lower. Specifically, when the parallel origin scenario was incorrectly inferred as single origin, the posterior probability was 0.671 and, when the single origin scenario was mistakenly inferred as parallel origin, the posterior probability was 0.899. Even with a more stringent threshold of 90% posterior probability, ABC analysis continued to successfully differentiate between the two scenarios. Out of the pseudo-observed datasets analyzed, the correct model was inferred for 913 datasets of parallel origin (with 87 classified as unclear), and for 973 datasets simulated under single origin, with 6 incorrectly assigned to parallel and 21 classified as unclear (Figure 5.3B).

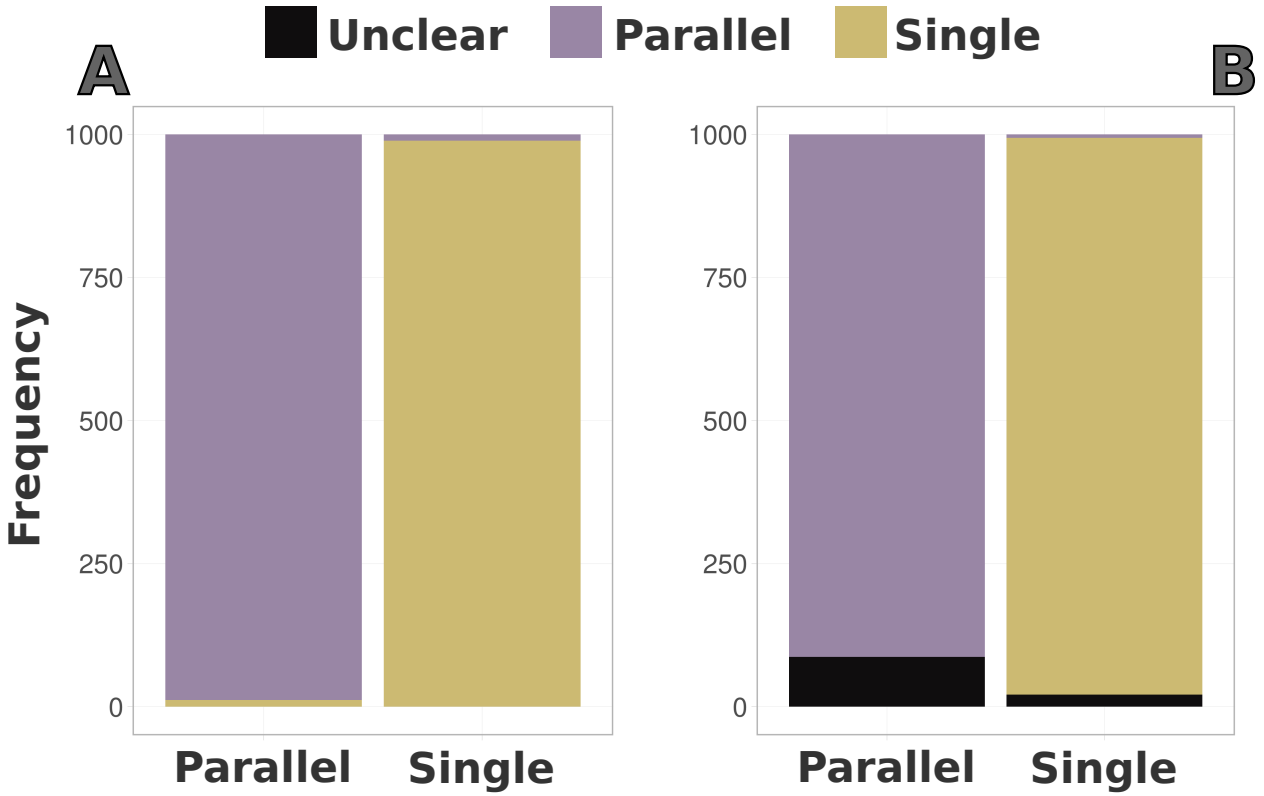


Figure 5.3: Model misclassification for the parallel and single origin models. Misclassification is based on the confusion matrix obtained using two different thresholds: (A) Simulations assigned to a model if posterior probability was above 0.5 or (B) Simulations assigned to a model only if posterior probability was above 0.9 (B).

5.3.3 MODEL CHOICE AND PARAMETER INFERENCE OF *LITTORINA SAXATILIS*

Our analysis of the Crab and Wave ecotypes sampled from two geographically distant locations (Arsklovet in Sweden and Burela in Spain) provides strong support for the parallel origin model. The

posterior probability obtained using the rejection algorithm was 0.993, while the posterior probability obtained using logistic regression was 1.000. Thus, we report here the parameter estimates obtained using the parallel origin scenario. To simplify the presentation of results, we re-scaled relative parameters to absolute effective sizes and time of events in years, using *k* to indicate thousands (Table 5.3, but refer to Table A5.4 for the relative estimates). The re-scaling process was conducted after combining the posterior distributions from 1000 subsets of loci, assigning more weight to subsets of loci with summary statistics closer to the overall mean of the entire genome.

Table 5.3: **Absolute parameter estimates for *Littorina saxatilis* populations.** Results are shown for the parallel origin scenario using the Arsklovet and Burela populations. For this model N_1 and N_2 correspond, respectively, to the absolute size of the Arsklovet Crab and Wave populations, while N_3 and N_4 correspond to the absolute size of the Burela Crab and Wave populations, respectively. For each parameter, the value outside brackets corresponds to the re-scaled mean of the posterior distribution and in-between brackets is the 95% credible interval. RS_1, RS_2 and Δ_s are presented in years. Parameters indicated here are the same as in table 5.2, except for P_{no}, which is converted to the percentage of the genome where no migration occurs between ecotypes.

parameter	absolute estimate
N_1	8842 (6434 - 16442)
N_2	11562 (7404 - 23233)
N_3	36816 (17565 - 88184)
N_4	25573 (12204 - 59078)
NA_1	42654 (7098 - 138326)
NA_2	59906 (8068 - 142922)
RS_1	14675 (2655 - 115400)
RS_2	57296 (10723 - 241318)
Δ_s	220941 (31604 - 298711)
$4N_2m_{cw1}$	15.8 (4.2 - 35.4)
$4N_1m_{wc1}$	10.4 (2.8 - 23.3)
$4N_4m_{cw2}$	8.3 (1.8 - 20.4)
$4N_3m_{wc2}$	23.4 (4.7 - 62.4)
$4N_1m_{cc1}$	0.5 (0.2 - 1.0)
$4N_3m_{cc2}$	2.3 (0.8 - 4.7)
$4N_2m_{ww1}$	0.3 (0.1 - 0.6)
$4N_4m_{ww2}$	0.7 (0.2 - 1.4)
$4NA_1m_{aa1}$	3.3 (0.2 - 12.9)
$4NA_2m_{aa2}$	4.2 (0.2 - 15.5)
P_{no}	3.7 (0.5 - 8.8)

Estimates based on the parallel origin scenario show that, in Spain, the present-day Crab population has a larger effective size of approximately 37k (95% CI: 18k - 89k) compared to the Wave population, which has an effective size of approximately 26k (95% CI: 12k - 59k). In contrast, the present-day populations in Sweden exhibit smaller effective sizes relative to the Spanish populations (Table 5.3), with the Wave population having a slightly larger effective size of approximately 12k (95% CI: 7k - 23k) compared to the Crab population, which has an effective size of $\sim$ 9k (95% CI: 6k - 16k). Our parameter estimates suggest that the two ecotypes diverged approximately 15,000 years ago (95% CI: 3k to 115k years) in Sweden (RS_1), with a much older split between Crab and Wave ecotype populations in Spain (RS_2), occurring $\sim$ 57,000 years ago (95% CI: 11k to 241k years). This divergence process was accompanied by gene flow in both countries. However, there were differences in the pattern, with higher scaled immigration ($4Nm$) from the Crab into the Wave ecotype in Sweden and from the Wave into the Crab ecotype in Spain (Table 5.3). Parameter estimates suggest that the separation between Spanish and Swedish populations (T_{As}) occurred 278k years ago (95% CI: 42k to 540k years). The point estimates also supported a larger ancestral effective size of the Spanish population (mean $\sim$ 60k, 95% CI: 8k - 142k) compared with the Swedish population (mean $\sim$ 43k, 95% CI: 7k - 138k). Lastly, we inferred a mean proportion of loci without migration in either direction (P_{no}) of approximately 4% with an upper CI close to 9% (Table 5.3). Furthermore, the proportion of loci without migration from Crab to Wave was approximately 2% (95% CI: 0% - 17.2%), and from Wave to Crab, it was around 3% (95% CI: 0% - 18.6%).

5.4 DISCUSSION

We contrasted two possible scenarios of ecotype formation in *L. saxatilis* by using whole-genome data obtained with pool-sequencing. We used a recently developed model-based method that is specifically designed for analysing pooled-sequencing data (Carvalho et al., 2023). This method combines an ABC (Approximate Bayesian Computation) inference framework with the explicit modeling of the various sources of error associated with pool-sequencing. By incorporating those sources of error into the analysis, we aimed to avoid biases in demographic estimates due to pool-seq errors.

5.4.1 POOL-SEQ DIFFERENTIATES BETWEEN COMPLEX SCENARIOS OF ECOTYPE FORMATION

The prediction errors obtained with the simulation study show that, for the datasets analysed here, the means of the posterior distributions provide accurate point estimates for most demographic history parameters of both scenarios of ecotype formation. The clear exceptions were the parameters related with the ancestral populations, such as the relative sizes of those populations and the migration between them. The high uncertainty associated with these parameters suggests that the summary statistics used do not contain information about older events. This observation is supported by the posterior distributions of these parameters in *L. saxatilis*, as they show minimal deviation from their prior distributions (Figure 5.4). Most of the remaining demographic history parameters exhibited similar prediction errors for both scenarios (Table 5.2). However, the prediction error for the relative time interval between split events (δ_s) is higher for the parallel origin than the single origin scenario. This disparity can be explained by the underlying topology of the parallel origin scenario. In this scenario, the diverging ecotypes found at each location have a shared evolutionary history and experience ongoing gene flow between them. Consequently, accurately inferring past events in such a complex context can prove challenging. Another possible explanation is the existence of a lower migration rate between ancestral populations in the single origin scenario (Table 5.2). Furthermore, there is also a slightly larger prediction error for migration rates between the same ecotypes in the single origin scenario. This can be attributed to the challenge of distinguishing between gene flow and incomplete lineage sorting within populations that share a common evolutionary history in the single origin model.

Even though we discarded the majority of available contigs, results from our simulation study confirm that pool-seq provides sufficient information to differentiate between complex scenarios of ecotype formation. Remarkably, this distinction remains possible even when setting a high threshold for posterior probability, as indicated by the proportion of correctly assigned simulations with 90% posterior probability above 0.9 for both models (Figure 5.3B). This result is surprising considering the high connectivity between populations in our model, where virtually all pairs of populations have the potential for migration (Figure 5.1). Nevertheless, the single and parallel origin scenarios considered here exhibit different mean values for various summary statistics (Figure A5.3), which accounts for our ability to distinguish between them (Marin, Pillai, Robert, & Rousseau, 2014). Using this approach, we found evidence that supports a parallel origin of the Crab and Wave ecotypes, without allopatric separation, and after colonization of the different regions by an ancestral population, which is consistent with the findings of Butlin et al. (2014). However, we did not specifically include periods of allopatry (i.e. isolation without gene flow) in our models.

Thus, it it remains a possibility that periods of allopatry played a role in the evolution of *L. saxatilis* ecotypes. Additionally, it is worth noting that the support for our findings is based on neutral loci and thus, it is still uncertain whether the alleles responsible for adaptive traits, potentially associated with inversions, have evolved in parallel.

5.4.2 RECENT PARALLEL ORIGIN OF *LITTORINA SAXATILIS* ECOTYPES

Our results support a relatively recent divergence of the ecotypes in both countries. Notably, the divergence in Sweden appears to be more recent, around 15,000 years ago, while in Spain, it took place approximately 57,000 years ago. These results are aligned with the hypothesis of a recent postglacial colonization of Swedish islands (Panova et al., 2011) and match previous estimates for Swedish populations (Carvalho et al., 2023). The estimates obtained by Butlin et al. (2014) date the divergence of the ecotypes to approximately 19k or 30k years ago, depending on the populations pairs analysed.

These estimates fall in-between the range of our estimates, which could be attributed to the fact that Butlin et al. (2014) included a single ecotype formation split event, thereby not accounting for potential variations between regions. The separation of ecotypes occurred relatively recently compared to the separation of populations in different regions, which we estimated to have taken place around 278,000 years ago. This suggests that the time interval between ancestral geographic structuring in Europe and the formation of the ecotypes was roughly 221k years in Spain and 263k years in Sweden. Nevertheless, it is likely that the split between different regions inferred here reflects an older split between Iberia and a northern refuge located outside of Sweden. Indeed, the formation of the Crab ecotype in Sweden required the arrival of predators selecting for the Crab ecotype, which occurred progressively in warming regions after the last glacial maximum. Thus, it is likely that *L. saxatilis* populations did not exist in Sweden for more than 200k years without ecotype formation. It is possible that the ecotypes were repeatedly formed and lost due to habitats changes associated with glacial fluctuations. This phenomenon could have had a more pronounced impact in Sweden, where such oscillations are expected to be more intense and frequent. Therefore, our estimates of approximately 15k years ago in Sweden and 57k years ago in Spain, may just reflect the most recent split between the two ecotypes.

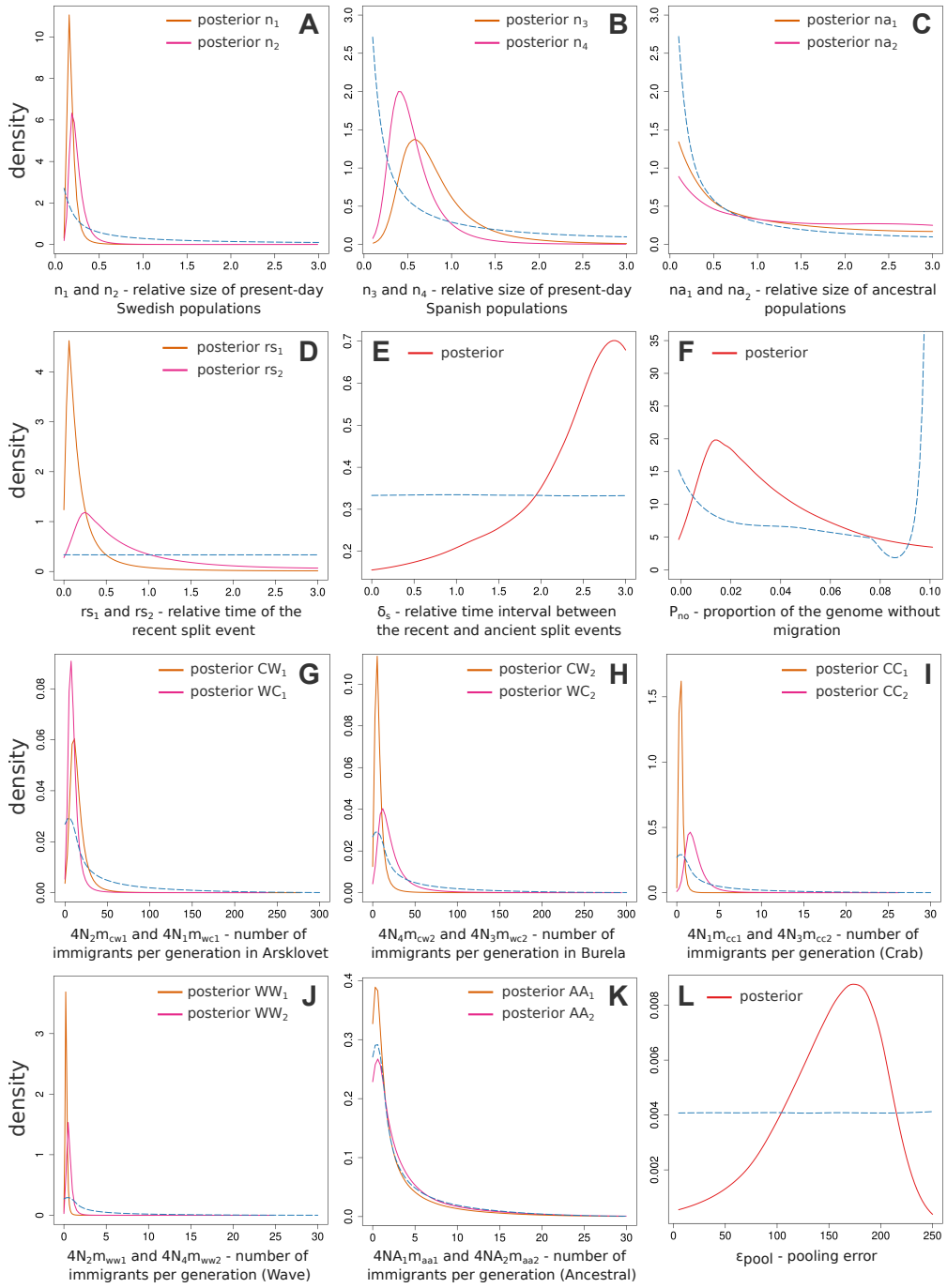


Figure 5.4: Posterior distributions of relative *L. saxatilis* parameters with regression adjustment and a tolerance of 0.01. Prior distributions are shown as a dotted blue line for reference. A - relative size of Arsklovet Crab (n_1) and Wave populations (n_2), B - relative size of Burela Crab (n_3) and Wave populations (n_4), C - relative size of ancestral populations (na_1 and na_2), D - relative time of the recent split events (rs_1 and rs_2), E - relative time interval between the two split events (δ_s), F - proportion of the genome without migration (P_{no}), G - average number of immigrants per generation ($4N_2m_{cw1}$ and $4N_1m_{wc1}$) in Arsklovet, H - average number of immigrants per generation ($4N_4m_{cw2}$ and $4N_3m_{wc2}$) in Burela, I - average number of immigrants per generation ($4N_1m_{cc1}$ and $4N_3m_{cc2}$) between Crab populations, J - average number of immigrants per generation ($4N_2m_{ww1}$ and $4N_4m_{ww2}$) between Wave populations, K - average number of immigrants per generation ($4NA_1m_{aa1}$ and $4NA_2m_{aa2}$) between ancestral populations and L - pooling error.

Additionally, the time period until ecotype formation could have been influenced by local extinctions triggered by factors such as toxic algal blooms (Johannesson & Johannesson, 1995), which could lead to population reestablishment through individuals carrying alleles from source populations of both ecotypes (Butlin et al., 2014). We found that the effective size of the Crab ecotype population in Spain was larger than that of the Wave ecotype population, whereas the Wave population in Sweden displayed a slightly larger effective size compared to the Crab population. Our estimates also support a lower effective size for current populations compared to ancestral populations, confirming a lack of support for past population expansions (Butlin et al., 2014).

The larger effective sizes of both present-day and ancestral populations in Spain could be attributed to various factors. One possibility is that these differences could arise from variations in carrying capacities or population densities between the two geographical locations. Alternatively, this difference might reflect historical demographic dynamics, such as population bottlenecks caused by toxic algal blooms in Sweden (Johannesson & Johannesson, 1995), potentially leading to more pronounced founder events in Sweden compared to Spain. Furthermore, the observed differences could be attributed to the proximity of Spanish populations to glacial refugia (Blakeslee et al., 2021; Bosso et al., 2022), which could have acted as safe havens during past glacial periods, allowing populations in Spain to retain larger sizes. Additionally, the larger effective size of the Spanish Crab population could be due to patterns of gene flow, given that we estimated a higher migration rate from the Wave to the Crab population in Spain (Table 5.3). On the whole, our estimates suggest that *L. saxatilis* effective population sizes remained consistently large, possibly due to gene flow and a high degree of cold-tolerance.

As mentioned, we did not specifically include periods without gene flow in our models and thus, caution is needed when considering the extent of gene flow between the ecotypes during the divergence process. Nevertheless, our results indicate that the divergence process between ecotypes in both Spain and Sweden was accompanied by gene flow. Specifically, we observed high migration rates between the divergent ecotypes, with values exceeding $4Nm > 10$ in most of the comparisons examined. Interestingly, we inferred a slightly higher migration rate from Crab to Wave ecotype in Sweden. This is consistent with the suggested higher net dispersal from the Crab to Wave ecotype as an explanation for the observed shift in cline centers towards the Wave habitat on Swedish islands (Westram et al., 2021).

In summary, our results argue in favor of a demographic history characterized by the spatial division of an ancestral population into a Spanish ancestral population and another ancestral population from which the ecotypes in Spain and Sweden, respectively, originated. Over time, distinct habitat-associated populations evolved at each of those geographic locations, while still experi-

encing gene flow between the diverging ecotypes. Thus, the available evidence strongly supports the conclusion that the Crab and Wave ecotypes observed in *L. saxatilis* arose as a result of divergent selection, despite the ongoing exchange of genetic material between them. This interplay between divergent selection and gene flow likely played a crucial role in shaping the evolutionary history of the *L. saxatilis* ecotypes, influencing the evolution of chromosomal rearrangements such as inversions (Ortiz-Barrientos, Engelstädter, & Rieseberg, 2016). Understanding the evolutionary processes that led to the emergence of these ecotypes not only sheds light on the population dynamics of *L. saxatilis*, but it also provides valuable insights into the mechanisms of parallel ecological adaptation and speciation. Despite some limitations, our findings indicate that combining pool-seq with ABC is an effective approach for investigating parallel evolution across a wide geographical range. The approach detailed here should be applied to other *L. saxatilis* populations, including additional sampling locations in Sweden, Spain and the United Kingdom.

5.5 REFERENCES

Akerman, A., & Bürger, R. (2014). The consequences of gene flow for local adaptation and differentiation: a two-locus two-deme model. *Journal of mathematical biology*, *68*(5), 1135-1198. doi: 10.1007/s00285-013-0660-z

Bainbridge, H. E., Brien, M. N., Morochz, C., Salazar, P. A., Rastas, P., & Nadeau, N. J. (2020). Limited genetic parallels underlie convergent evolution of quantitative pattern variation in mimetic butterflies. *Journal of Evolutionary Biology*, *33*(11), 1516-1529. doi: 10.1111/jeb.13704

Beaumont, M. A., Zhang, W., & Balding, D. J. (2002). Approximate bayesian computation in population genetics. *Genetics*, *162*(4), 2025-2035. doi: 10.1111/j.1937-2817.2010.tb01236.x

Bhatia, G., Patterson, N., Sankararaman, S., & Price, A. L. (2013). Estimating and interpreting fst: the impact of rare variants. *Genome research*, *23*(9), 1514-1521. doi: 10.1101/gr.154831.113

Blakeslee, A. M., Miller, A. W., Ruiz, G. M., Johannesson, K., André, C., & Panova, M. (2021). Population structure and phylogeography of two north atlantic littorina species with contrasting larval development. *Marine Biology*, *168*, 1-16. doi: 10.1007/s00227-021-03918-8

Bohutínská, M., Vlček, J., Yair, S., Laenen, B., Konečná, V., Fracassetti, M., ... Kolář, F. (2021). Genomic basis of parallel adaptation varies with divergence in arabidopsis and its relatives. *Proceedings of the National Academy of Sciences*, *118*(21), e2022713118. doi: 10.1073/pnas.2022713118

Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: a flexible trimmer for illumina sequence data. *Bioinformatics*, *30*(15), 2114-2120. doi: 10.1093/bioinformatics/btu170

Bolnick, D. I., Barrett, R. D., Oke, K. B., Rennison, D. J., & Stuart, Y. E. (2018). (non) parallel

evolution. *Annual Review of Ecology, Evolution, and Systematics*, *49*(1), 303-330. doi: 10.1146/annurev-ecolsys-110617-062240

Bosso, L., Smeraldo, S., Russo, D., Chiusano, M. L., Bertorelle, G., Johannesson, K., ... Raffini, F. (2022). The rise and fall of an alien: Why the successful colonizer littorina saxatilis failed to invade the mediterranean sea. *Biological Invasions*, *24*(10), 3169-3187. doi: 10.1007/s10530-022-02838-y

Butlin, R. K., Galindo, J., & Grahame, J. W. (2008). Sympatric, parapatric or allopatric: the most important way to classify speciation? *Philosophical Transactions of the Royal Society B: Biological Sciences*, *363*(1506), 2997-3007. doi: 10.1098/rstb.2008.0076

Butlin, R. K., Saura, M., Charrier, G., Jackson, B., André, C., Caballero, A., ... Rolán-Alvarez, E. (2014). Parallel evolution of local adaptation and reproductive isolation in the face of gene flow. *Evolution*, *68*(4), 935-949. doi: 10.1111/evo.12329

Carvalho, J., Morales, H. E., Faria, R., Butlin, R. K., & Sousa, V. C. (2023). Integrating pool-seq uncertainties into demographic inference. *Molecular Ecology Resources*, *23*(7), 1737-1755. doi: 10.1111/1755-0998.13834

Chan, Y. F., Marks, M. E., Jones, F. C., Villarreal Jr, G., Shapiro, M. D., Brady, S. D., ... others (2010). Adaptive evolution of pelvic reduction in sticklebacks by recurrent deletion of a pitx1 enhancer. *Science*, *327*(5963), 302-305. doi: 10.1126/science.1182213

Collin, F.-d., Durif, G., Raynal, L., Lombaert, E., Gautier, M., Vitalis, R., ... Estoup, A. (2021). Extending approximate bayesian computation with supervised machine learning to infer demographic history from genetic polymorphisms using diyabc random forest. *Molecular Ecology Resources*, *21*(8), 2598-2613. doi: 10.1111/1755-0998.13413

Colosimo, P. F., Hosemann, K. E., Balabhadra, S., Villarreal, G., Dickson, M., Grimwood, J., ... Kingsley, D. M. (2005). Widespread parallel evolution in sticklebacks by repeated fixation of ectodysplasin alleles. *Science*, *307*(5717), 1928-1933. doi: 10.1126/science.1107239

Conde-Padín, P., Cruz, R., Hollander, J., & Rolan-Alvarez, E. (2008). Revealing the mechanisms of sexual isolation in a case of sympatric and parallel ecological divergence. *Biological Journal of the Linnean Society*, *94*(3), 513-526. doi: 10.1111/j.1095-8312.2008.00998.x

Conte, G. L., Arnegard, M. E., Peichel, C. L., & Schluter, D. (2012). The probability of genetic parallelism and convergence in natural populations. *Proceedings of the Royal Society B: Biological Sciences*, *279*(1749), 5039-5047. doi: 10.1098/rspb.2012.2146

Csilléry, K., François, O., & Blum, M. G. (2012). Abc: An r package for approximate bayesian computation (abc). *Methods in Ecology and Evolution*, *3*(3), 475-479. doi: 10.1111/j.2041-210X.2011.00179.x

Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., ... Li, H. (2021). Twelve years of samtools and bcftools. *Gigascience*, *10*(2), giab008. doi: 10.1093/gigascience/giab008

DeFaveri, J., Shikano, T., Shimada, Y., Goto, A., & Merilä, J. (2011). Global analysis of genes involved in freshwater adaptation in threespine sticklebacks (*Gasterosteus aculeatus*). *Evo-*

lution: International Journal of Organic Evolution, *65*(6), 1800-1807. doi: 10.1111/j.1558-5646.2011.01247.x

Elmer, K. R., & Meyer, A. (2011). Adaptation in the age of ecological genomics: insights from parallelism and convergence. *Trends in ecology & evolution*, *26*(6), 298-306. doi: 10.1016/j.tree.2011.02.008

Fang, B., Kemppainen, P., Momigliano, P., Feng, X., & Merilä, J. (2020). On the causes of geographically heterogeneous parallel evolution in sticklebacks. *Nature ecology & evolution*, *4*(8), 1105-1115. doi: 10.1038/s41559-020-1222-6

Faria, R., Chaube, P., Morales, H. E., Larsson, T., Lemmon, A. R., Lemmon, E. M., ... Butlin, R. K. (2019). Multiple chromosomal rearrangements in a hybrid zone between *Littorina saxatilis* ecotypes. *Molecular Ecology*, *28*(6), 1375-1393. doi: 10.1111/mec.14972

Faria, R., Renaut, S., Galindo, J., Pinho, C., Melo-Ferreira, J., Melo, M., ... Butlin, R. K. (2014). Advances in ecological speciation: an integrative approach. *Molecular Ecology*, *23*(3), 513-521. doi: 10.1111/mec.12616

Ferchaud, A.-L., & Hansen, M. M. (2016). The impact of selection, gene flow and demographic history on heterogeneous genomic divergence: Three-spine sticklebacks in divergent environments. *Molecular Ecology*, *25*(1), 238-259. doi: 10.1111/mec.13399

Fraïsse, C., Popovic, I., Mazoyer, C., Spataro, B., Delmotte, S., Romiguier, J., ... Roux, C. (2021). Dils: Demographic inferences with linked selection by using abc. *Molecular Ecology Resources*, *21*(8), 2629-2644. doi: 10.1111/1755-0998.13323

Futschik, A., & Schlötterer, C. (2010). The next generation of molecular markers from massively parallel sequencing of pooled dna samples. *Genetics*, *186*(1), 207-218. doi: 10.1534/genetics.110.114397

Gautier, M., Foucaud, J., Gharbi, K., Cézard, T., Galan, M., Loiseau, A., ... Estoup, A. (2013). Estimation of population allele frequencies from next-generation sequencing data: pool-versus individual-based genotyping. *Molecular Ecology*, *22*(14), 3766-3779. doi: 10.1111/mec.12360

Grahame, J. W., Wilding, C. S., & Butlin, R. K. (2006). Adaptation to a steep environmental gradient and an associated barrier to gene exchange in *Littorina saxatilis*. *Evolution*, *60*(2), 268-278. doi: 10.1111/j.0014-3820.2006.tb01105.x

Hansson, T. H., Fischer, B., Mazzarella, A. B., Voje, K. L., & Vøllestad, L. A. (2016). Lateral plate number in low-plated threespine stickleback: a study of plasticity and heritability. *Ecology and Evolution*, *6*(10), 3154-3160. doi: 10.1002/ece3.2020

He, Z., Li, X., Yang, M., Wang, X., Zhong, C., Duke, N. C., ... Shi, S. (2019). Speciation with gene flow via cycles of isolation and migration: insights from multiple mangrove taxa. *National Science Review*, *6*(2), 275-288. doi: 10.1093/nsr/nwy078

Hendry, A. P., & Taylor, E. B. (2004). How much of the variation in adaptive divergence can be explained by gene flow? an evaluation using lake-stream stickleback pairs. *Evolution*, *58*(10), 2319-2331. doi: 10.1111/j.0014-3820.2004.tb01606.x

Hewitt, G. M. (2011). Quaternary phylogeography: the roots of hybrid zones. *Genetica*, *139*, 617-638. doi: 10.1007/s10709-011-9547-3

Hudson, R. R. (1990). Gene genealogies and the coalescent process. *Oxford surveys in evolutionary biology*, *7*(1), 1-44.

Jay, F., Boitard, S., & Austerlitz, F. (2019). An abc method for whole-genome sequence data: inferring paleolithic and neolithic human expansions. *Molecular biology and evolution*, *36*(7), 1565-1579. doi: 10.1093/molbev/msz038

Johannesson, K. (2001). Parallel speciation: a key to sympatric divergence. *Trends in Ecology & Evolution*, *16*(3), 148-153. doi: 10.1016/S0169-5347(00)02078-4

Johannesson, K., & Johannesson, B. (1995). Dispersal and population expansion in a direct developing marine snail (*Littorina saxatilis*) following a severe population bottleneck. *Hydrobiologia*, *309*, 173–180. doi: 10.1007/BF00014485

Johannesson, K., Panova, M., Kemppainen, P., André, C., Rolán-Alvarez, E., & Butlin, R. K. (2010). Repeated evolution of reproductive isolation in a marine snail: unveiling mechanisms of speciation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *365*(1547), 1735-1747. doi: 10.1098/rstb.2009.0256

Jones, F. C., Grabherr, M. G., Chan, Y. F., Russell, P., Mauceli, E., Johnson, J., ... Team, B. I. G. S. P. . W. G. A. (2012). The genomic basis of adaptive evolution in threespine sticklebacks. *Nature*, *484*(7392), 55-61. doi: 10.1038/nature10944

Kaeuffer, R., Peichel, C. L., Bolnick, D. I., & Hendry, A. P. (2012). Parallel and nonparallel aspects of ecological, phenotypic, and genetic divergence across replicate population pairs of lake and stream stickleback. *Evolution*, *66*(2), 402-418. doi: 10.1111/j.1558-5646.2011.01440.x

Kess, T., & Boulding, E. G. (2019). Genome-wide association analyses reveal polygenic genomic architecture underlying divergent shell morphology in spanish *Littorina saxatilis* ecotypes. *Ecology and evolution*, *9*(17), 9427-9441. doi: 10.1002/ece3.5378

Koch, E. L., Morales, H. E., Larsson, J., Westram, A. M., Faria, R., Lemmon, A. R., ... Butlin, R. K. (2021). Genetic variation for adaptive traits is associated with polymorphic inversions in *Littorina saxatilis*. *Evolution Letters*, *5*(3), 196-213. doi: 10.1002/evl3.227

Koch, E. L., Ravinet, M., Westram, A. M., Johannesson, K., & Butlin, R. K. (2022). Genetic architecture of repeated phenotypic divergence in *Littorina saxatilis* ecotype evolution. *Evolution*, *76*(10), 2332-2346. doi: 10.1111/evo.14602

Langerhans, R. B. (2018). Predictability and parallelism of multitrait adaptation. *Journal of Heredity*, *109*(1), 59-70. doi: 10.1093/jhered/esx043

Laporte, M., Rogers, S. M., Dion-Côté, A.-M., Normandeau, E., Gagnaire, P.-A., Dalziel, A. C., ... Bernatchez, L. (2015). Rad-qtl mapping reveals both genome-level parallelism and different genetic architecture underlying the evolution of body shape in lake whitefish (*Coregonus clupeaformis*) species pairs. *G3: Genes, Genomes, Genetics*, *5*(7), 1481-1491. doi: 10.1534/g3.115.019067

Lenormand, T. (2002). Gene flow and the limits to natural selection. *Trends in ecology & evolution*, *17*(4), 183-189. doi: 10.1016/S0169-5347(02)02497-7

Malinsky, M., Matschiner, M., & Svardal, H. (2021). Dsuite-fast d-statistics and related admixture evidence from vcf files. *Molecular ecology resources*, *21*(2), 584-595. doi: 10.1111/1755-0998.13265

Marin, J.-M., Pillai, N. S., Robert, C. P., & Rousseau, J. (2014). Relevant statistics for bayesian model choice. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *76*(5), 833-859. doi: 10.1111/rssb.12056

Morales, H. E., Faria, R., Johannesson, K., Larsson, T., Panova, M., Westram, A. M., & Butlin, R. K. (2018). *Littorina saxatilis genome sequencing and population re-sequencing.* Retrieved from `https://www.ncbi.nlm.nih.gov/bioproject/PRJNA494650`

Morales, H. E., Faria, R., Johannesson, K., Larsson, T., Panova, M., Westram, A. M., & Butlin, R. K. (2019). Genomic architecture of parallel ecological divergence: beyond a single environmental contrast. *Science advances*, *5*(12), eaav9963. doi: 10.1126/sciadv.aav9963

Nei, M., & Roychoudhury, A. K. (1974). Sampling variances of heterozygosity and genetic distance. *Genetics*, *76*(2), 379-390. doi: 10.1093/genetics/76.2.379

Nosil, P. (2012). *Ecological speciation*. Oxford University Press.

Oke, K. B., Rolshausen, G., LeBlond, C., & Hendry, A. P. (2017). How parallel is parallel evolution? a comparative analysis in fishes. *The American Naturalist*, *190*(1), 1-16. doi: 10.1086/691989

Ortiz-Barrientos, D., Engelstädter, J., & Rieseberg, L. H. (2016). Recombination rate evolution and the origin of species. *Trends in ecology & evolution*, *31*(3), 226-236. doi: 10.1016/j.tree.2015.12.016

Panova, M., Blakeslee, A. M., Miller, A. W., Mäkinen, T., Ruiz, G. M., Johannesson, K., & André, C. (2011). Glacial history of the north atlantic marine snail, *Littorina saxatilis*, inferred from distribution of mitochondrial dna lineages. *PLoS One*, *6*(3), e17511. doi: 10.1371/journal.pone.0017511

Perini, S., Rafajlović, M., Westram, A. M., Johannesson, K., & Butlin, R. K. (2020). Assortative mating, sexual selection, and their consequences for gene flow in littorina. *Evolution*, *74*(7), 1482-1497. doi: 10.1111/evo.14027

Poore, H. A., Stuart, Y. E., Rennison, D. J., Roesti, M., Hendry, A. P., Bolnick, D. I., & Peichel, C. L. (2022). Repeated genetic divergence plays a minor role in repeated phenotypic divergence of lake-stream stickleback. *Evolution*. doi: 10.1093/evolut/qpac025

Portinha, B., Avril, A., Bernasconi, C., Helanterä, H., Monaghan, J., Seifert, B., ... Nouhaud, P. (2022). Whole-genome analysis of multiple wood ant population pairs supports similar speciation histories, but different degrees of gene flow, across their european ranges. *Molecular Ecology*, *31*(12), 3416-3431. doi: 10.1111/mec.16481

Quesada, H., Posada, D., Caballero, A., Morán, P., & Rolán-Alvarez, E. (2007). Phyloge-

netic evidence for multiple sympatric ecological diversification in a marine snail. *Evolution: International Journal of Organic Evolution*, *61*(7), 1600-1612. doi: 10.1111/j.1558-5646.2007.00135.x

Quinlan, A. R., & Hall, I. M. (2010). Bedtools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, *26*(6), 841-842. doi: 10.1093/bioinformatics/btq033

Ravinet, M., Westram, A., Johannesson, K., Butlin, R., André, C., & Panova, M. (2016). Shared and nonshared genomic divergence in parallel ecotypes of *Littorina saxatilis* at a local scale. *Molecular ecology*, *25*(1), 287-305. doi: 10.1111/mec.13332

Reid, D. G. (1996). *Systematics and evolution of Littorina*. London: Ray Society.

Rennison, D. J., Delmore, K. E., Samuk, K., Owens, G. L., & Miller, S. E. (2020). Shared patterns of genome-wide differentiation are more strongly predicted by geography than by ecology. *The American Naturalist*, *195*(2), 192-200. doi: 10.1086/706476

Sampson, J., Jacobs, K., Yeager, M., Chanock, S., & Chatterjee, N. (2011). Efficient study design for next generation sequencing. *Genetic epidemiology*, *35*(4), 269-277. doi: 10.1002/gepi.20575

Schlötterer, C., Tobler, R., Kofler, R., & Nolte, V. (2014). Sequencing pools of individuals — mining genome-wide polymorphism data without big funding. *Nature Reviews Genetics*, *15*(11), 749-763. doi: 10.1038/nrg3803

Schluter, D. (2009). Evidence for ecological speciation and its alternative. *Science*, *323*(5915), 737-741. doi: 10.1126/science.1160006

Seehausen, O., Butlin, R. K., Keller, I., Wagner, C. E., Boughman, J. W., Hohenlohe, P. A., ... Widmer, A. (2014). Genomics and the origin of species. *Nature Reviews Genetics*, *15*(3), 176-192. doi: 10.1038/nrg3644

Smadja, C. M., & Butlin, R. K. (2011). A framework for comparing processes of speciation in the presence of gene flow. *Molecular Ecology*, *20*(24), 5123-5140. doi: 10.1111/j.1365-294X.2011.05350.x

Sousa, V. C., & Hey, J. (2013). Understanding the origin of species with genome-scale data: modelling gene flow. *Nature Reviews Genetics*, *14*(6), 404-414. doi: 10.1038/nrg3446

Staab, P. R., Zhu, S., Metzler, D., & Lunter, G. (2015). scrm: efficiently simulating long sequences using the approximated coalescent with recombination. *Bioinformatics*, *31*(10), 1680-1682. doi: 10.1093/bioinformatics/btu861

Steiner, C. C., Römpler, H., Boettger, L. M., Schöneberg, T., & Hoekstra, H. E. (2009). The genetic basis of phenotypic convergence in beach mice: similar pigment patterns but different genes. *Molecular Biology and Evolution*, *26*(1), 35-45. doi: 10.1093/molbev/msn218

Thurman, T. J., Palmer, T. M., Kolbe, J. J., Askary, A. M., Gotanda, K. M., Lapiedra, O., ... Barrett, R. D. H. (2023). The difficulty of predicting evolutionary change in response to novel ecological interactions: a field experiment with anolis lizards. *The American Naturalist*, *201*(4). doi: https://doi.org/10.1086/723209

Westram, A. M., Faria, R., Johannesson, K., & Butlin, R. K. (2021). Using replicate hybrid zones to understand the genomic basis of adaptive divergence. *Molecular ecology*, *30*(15), 3797-3814. doi: 10.1111/mec.15861

Westram, A. M., Rafajlović, M., Chaube, P., Faria, R., Larsson, T., Panova, M., ... Butlin, R. K. (2018). Clines on the seashore: The genomic architecture underlying rapid divergence in the face of gene flow. *Evolution letters*, *2*(4), 297-309. doi: 10.1002/evl3.74

Yamasaki, Y. Y., Kakioka, R., Takahashi, H., Toyoda, A., Nagano, A. J., Machida, Y., ... Kitano, J. (2020). Genome-wide patterns of divergence and introgression after secondary contact between *Pungitius* sticklebacks. *Philosophical Transactions of the Royal Society B*, *375*(1806). doi: 10.1098/rstb.2019.0548

5.6 APPENDIX

Table A5.1: **Set of summary statistics considered.** Different combinations of D-statistics were employed to examine the level of introgression occurring between the distinct ecotypes. For D-statistic 1, P1 represented the Wave population in the first location (N_2), P2 stood for the Wave population in the second location (N_4), and P3 denoted the Crab population at the first location (N_1). For D-statistic 2, P1 again referred to the Wave population at the first location (N_2), while P2 now represented the Crab population in the second location (N_3), and P3 was the Crab population at the first location (N_1). D-statistic 3 retained P1 as the Wave population at the first location (N_2), designated P2 as the Crab population at the first location (N_1), and P3 as the Wave population in the second location (N_4). In all these combinations, P4 was assumed to be an outgroup fixed, at all sites, for the major allele. For the proportion of exclusive SNPs, we computed this per location i.e. checking if each site was segregating in one population but not in the other population inhabiting the same location and globally by computing the proportion of sites that were segregating in only one population and not in the other three. For the proportion of shared or fixed differences SNPs, we computed this for the two populations that inhabit the same location and globally by comparing each population with the other three.

summary statistic	four-populations
mean heterozygosity	4 values (1 per population)
SD heterozygosity	4 values (1 per population)
mean heterozygosity between populations	6 pairwise values
SD heterozygosity between populations	6 pairwise values
pairwise F_{ST}	6 pairwise values
SD F_{ST}	6 pairwise values
5%F_{ST}	6 pairwise values
95%F_{ST}	6 pairwise values
proportion of fixed differences	3 values
proportion of exclusive SNPs	5 values
proportion of shared SNPs	3 values
mean D-statistic 1	1 value
mean D-statistic 2	1 value
mean D-statistic 3	1 value
SD D-statistic 1	1 value
SD D-statistic 2	1 value
SD D-statistic 3	1 value
total	61

Table A5.2: **Prediction errors for the parallel origin parameters.** Parameter inference was performed using a simple rejection or a regression adjustment using a local linear regression. For each method, values are presented for two different tolerance rates. n_1 to n_4 - relative population sizes of the extant populations, na_1 and na_2 - relative population sizes of the ancestral populations, rs_1 and rs_2 - relative time of the recent split events, δ_s - relative time interval between *rs* and the ancient split event (t_{As}), ε_{pool} - experimental error introduced by the pooling procedures, ε_{seq} - error associated with sequencing and mapping errors, m_{cw1} and m_{cw2} - probability per generation that an individual migrates from the Crab population to the Wave population in the first and second locations, respectively (forward in time), m_{wc1} and m_{wc2} - probability per generation that an individual migrates from the Wave population to the Crab population in the first and second locations, respectively (forward in time), m_{cc} - probability per generation that an individual migrates from one Crab population to the other (forward in time), m_{ww} - probability per generation that an individual migrates from one Wave population to the other (forward in time), m_{aa} - probability per generation that an individual migrates from one ancestral population to the other (forward in time), P_{cw} - proportion of the simulated loci where no migration occurs from the Crab to the Wave population; P_{wc} - proportion of the simulated loci where no migration occurs from the Wave to the Crab population, P_{no} - proportion of the simulated loci where no migration occurs between ecotypes, $4N_2m_{cw1}$ and $4N_1m_{wc1}$ - average number of immigrants per generation from Crab to Wave and from Wave to Crab (respectively) in the first location, $4N_4m_{cw2}$ and $4N_3m_{wc2}$ - average number of immigrants per generation from Crab to Wave and from Wave to Crab (respectively) in the second location, $4N_1m_{cc1}$ and $4N_3m_{cc2}$ - average number of immigrants per generation from the Crab population in the second location to the first and vice-versa (respectively), $4N_2m_{ww1}$ and $4N_4m_{ww2}$ - average number of immigrants per generation from the Wave population in the second location to the first and vice-versa (respectively), $4NA_1m_{aa1}$ and $4NA_2m_{aa2}$ - average number of immigrants per generation from na_2 to na_1 and vice-versa (respectively).

	REJECTION						**REGRESSION**					
	tolerance of 0.005			*tolerance of 0.01*			*tolerance of 0.005*			*tolerance of 0.01*		
parameter	mode	median	mean	mode	median	mean	mode	median	mean	mode	median	mean
n_1	0.94	0.49	0.44	1.04	0.55	0.49	0.12	0.12	0.13	0.13	0.12	0.13
n_2	0.92	0.49	0.44	1.03	0.55	0.48	0.13	0.12	0.13	0.13	0.13	0.13
n_3	0.97	0.50	0.46	1.05	0.53	0.48	0.14	0.14	0.14	0.13	0.13	0.13
n_4	1.01	0.53	0.47	1.05	0.53	0.49	0.12	0.12	0.13	0.12	0.12	0.13
na_1	2.00	1.13	0.96	1.97	1.13	0.98	1.96	0.97	0.90	1.91	0.99	0.93
na_2	1.99	1.14	0.98	1.95	1.12	0.97	1.93	1.01	0.93	1.92	0.96	0.91
rs_1	1.49	0.79	0.80	1.37	0.79	0.80	0.74	0.51	0.50	0.82	0.52	0.51
rs_2	1.37	0.73	0.77	1.42	0.78	0.80	0.72	0.49	0.48	0.81	0.53	0.52
δ_s	2.81	0.99	0.99	3.04	1.00	1.00	2.66	1.05	1.00	2.79	1.02	0.99
ε_{pool}	0.44	0.34	0.39	0.50	0.39	0.44	0.13	0.13	0.13	0.14	0.14	0.14
ε_{seq}	0.25	0.21	0.26	0.27	0.27	0.33	0.03	0.02	0.02	0.03	0.03	0.02
m_{cw1}	1.61	0.80	0.85	1.80	0.84	0.87	0.66	0.49	0.47	0.61	0.45	0.45
m_{cw2}	1.68	0.81	0.85	1.70	0.84	0.88	0.61	0.44	0.44	0.63	0.47	0.46
m_{wc1}	1.55	0.81	0.85	1.55	0.84	0.88	0.62	0.46	0.45	0.54	0.42	0.43
m_{wc2}	1.74	0.81	0.85	1.83	0.86	0.89	0.55	0.41	0.41	0.65	0.48	0.48
m_{cc}	0.86	0.54	0.57	0.95	0.54	0.59	0.16	0.15	0.15	0.17	0.16	0.15
m_{ww}	0.83	0.51	0.55	0.92	0.56	0.60	0.15	0.14	0.14	0.16	0.15	0.15
m_{aa}	2.73	0.99	0.99	2.79	0.98	0.98	2.50	1.05	0.99	2.66	1.02	0.98
P_{cw}	2.81	1.01	0.97	2.85	1.04	0.98	2.18	0.86	0.78	2.28	0.92	0.83
P_{wc}	2.75	1.05	0.97	2.81	1.04	0.98	2.12	0.87	0.78	2.33	0.91	0.81
P_{no}	0.75	0.30	0.30	0.79	0.31	0.32	0.28	0.17	0.14	0.28	0.18	0.14
$4N_2m_{cw1}$	1.16	0.76	0.65	1.24	0.82	0.68	0.33	0.30	0.29	0.34	0.30	0.29
$4N_1m_{wc1}$	1.17	0.78	0.67	1.21	0.82	0.70	0.34	0.29	0.28	0.33	0.28	0.27
$4N_4m_{cw2}$	1.21	0.80	0.68	1.26	0.83	0.71	0.32	0.28	0.28	0.36	0.31	0.31
$4N_3m_{wc2}$	1.18	0.78	0.67	1.25	0.83	0.70	0.34	0.29	0.29	0.39	0.34	0.33
$4N_1m_{cc1}$	0.93	0.62	0.52	0.99	0.67	0.54	0.14	0.14	0.14	0.15	0.14	0.14
$4N_3m_{cc2}$	0.94	0.62	0.52	0.99	0.66	0.55	0.17	0.16	0.16	0.15	0.14	0.15
$4N_2m_{ww1}$	0.91	0.61	0.51	0.98	0.66	0.54	0.15	0.14	0.14	0.14	0.13	0.13
$4N_4m_{ww2}$	0.94	0.62	0.51	1.00	0.67	0.56	0.14	0.13	0.13	0.15	0.14	0.14
$4NA_1m_{aa1}$	1.68	1.16	0.98	1.67	1.16	0.99	1.65	1.12	0.98	1.65	1.12	0.98
$4NA_2m_{aa2}$	1.65	1.17	0.99	1.68	1.16	0.99	1.61	1.12	0.98	1.66	1.12	0.98

Table A5.3: **Prediction errors for the single origin parameters.** Parameter inference was performed using a simple rejection or a regression adjustment using a local linear regression. For each method, values are presented for two different tolerance rates. n_1 to n_4 - relative population sizes of the extant populations, na_1 and na_2 - relative population sizes of the ancestral populations, rs_1 and rs_2 - relative time of the recent split events, δ_s - relative time interval between rs and the ancient split event (t_{As}), ε_{pool} - experimental error introduced by the pooling procedures, ε_{seq} - error associated with sequencing and mapping errors, m_{cw1} and m_{cw2} - probability per generation that an individual migrates from the Crab population to the Wave population in the first and second locations, respectively (forward in time), m_{wc1} and m_{wc2} - probability per generation that an individual migrates from the Wave population to the Crab population in the first and second locations, respectively (forward in time), m_{cc} - probability per generation that an individual migrates from one Crab population to the other (forward in time), m_{ww} - probability per generation that an individual migrates from one Wave population to the other (forward in time), m_{aa} - probability per generation that an individual migrates from one ancestral population to the other (forward in time), P_{cw} - proportion of the simulated loci where no migration occurs from the Crab to the Wave population; P_{wc} - proportion of the simulated loci where no migration occurs from the Wave to the Crab population, P_{no} - proportion of the simulated loci where no migration occurs between ecotypes, $4N_2m_{cw1}$ and $4N_1m_{wc1}$ - average number of immigrants per generation from Crab to Wave and from Wave to Crab (respectively) in the first location, $4N_4m_{cw2}$ and $4N_3m_{wc2}$ - average number of immigrants per generation from Crab to Wave and from Wave to Crab (respectively) in the second location, $4N_1m_{cc1}$ and $4N_3m_{cc2}$ - average number of immigrants per generation from the Crab population in the second location to the first and vice-versa (respectively), $4N_2m_{ww1}$ and $4N_4m_{ww2}$ - average number of immigrants per generation from the Wave population in the second location to the first and vice-versa (respectively), $4NA_1m_{aa1}$ and $4NA_2m_{aa2}$ - average number of immigrants per generation from na_2 to na_1 and vice-versa (respectively).

	REJECTION						**REGRESSION**					
	tolerance of 0.005			*tolerance of 0.01*			*tolerance of 0.005*			*tolerance of 0.01*		
parameter	mode	median	mean	mode	median	mean	mode	median	mean	mode	median	mean
n_1	1.26	0.66	0.58	1.39	0.70	0.62	0.18	0.17	0.18	0.18	0.17	0.18
n_2	1.25	0.67	0.60	1.35	0.69	0.61	0.20	0.19	0.20	0.19	0.17	0.18
n_3	1.20	0.64	0.58	1.29	0.66	0.60	0.19	0.18	0.19	0.19	0.18	0.19
n_4	1.26	0.68	0.60	1.40	0.71	0.63	0.20	0.19	0.20	0.21	0.20	0.21
na_1	1.99	1.13	0.98	1.97	1.16	0.99	1.91	0.99	0.89	1.92	1.02	0.90
na_2	1.92	1.10	0.98	1.97	1.13	0.99	1.86	1.00	0.91	1.91	1.03	0.92
rs_1	1.23	0.86	0.83	1.20	0.85	0.84	1.01	0.69	0.63	1.06	0.68	0.64
rs_2	1.21	0.85	0.84	1.28	0.88	0.86	1.07	0.73	0.68	1.09	0.72	0.66
δ_s	1.66	0.77	0.79	1.69	0.78	0.80	0.85	0.62	0.57	0.82	0.59	0.55
ε_{pool}	0.57	0.34	0.34	0.72	0.39	0.39	0.09	0.09	0.09	0.10	0.09	0.09
ε_{seq}	0.22	0.18	0.23	0.27	0.24	0.29	0.02	0.02	0.02	0.02	0.02	0.02
m_{cw1}	1.89	0.88	0.91	2.07	0.90	0.92	0.66	0.50	0.49	0.61	0.47	0.47
m_{cw2}	2.01	0.90	0.92	2.23	0.90	0.93	0.62	0.46	0.45	0.63	0.48	0.47
m_{wc1}	2.04	0.88	0.91	2.02	0.90	0.92	0.64	0.48	0.47	0.60	0.45	0.45
m_{wc2}	2.05	0.88	0.91	2.16	0.91	0.93	0.68	0.50	0.49	0.60	0.47	0.47
m_{cc}	1.14	0.60	0.65	1.08	0.62	0.67	0.21	0.21	0.22	0.21	0.20	0.21
m_{ww}	1.06	0.60	0.65	1.20	0.63	0.68	0.23	0.21	0.22	0.22	0.21	0.22
m_{aa}	3.05	1.00	1.00	2.98	1.00	1.00	3.27	1.11	1.05	3.17	1.05	1.02
P_{cw}	2.79	1.00	0.94	3.04	0.99	0.96	2.16	0.90	0.79	2.30	0.91	0.81
P_{wc}	2.82	0.98	0.94	3.02	1.01	0.97	2.15	0.89	0.78	2.36	0.89	0.80
P_{no}	0.34	0.20	0.16	0.33	0.20	0.16	0.25	0.13	0.11	0.22	0.11	0.10
$4N_2m_{cw1}$	1.35	0.87	0.75	1.38	0.87	0.76	0.37	0.33	0.32	0.34	0.31	0.31
$4N_1m_{wc1}$	1.36	0.87	0.75	1.39	0.90	0.77	0.36	0.33	0.33	0.35	0.31	0.31
$4N_4m_{cw2}$	1.34	0.86	0.74	1.43	0.90	0.77	0.36	0.33	0.33	0.37	0.33	0.33
$4N_3m_{wc2}$	1.31	0.85	0.74	1.36	0.88	0.76	0.34	0.31	0.31	0.34	0.31	0.31
$4N_1m_{cc1}$	1.08	0.78	0.70	1.15	0.81	0.70	0.30	0.26	0.25	0.34	0.27	0.25
$4N_3m_{cc2}$	1.09	0.78	0.67	1.15	0.80	0.69	0.30	0.26	0.25	0.34	0.28	0.27
$4N_2m_{ww1}$	1.06	0.77	0.67	1.08	0.78	0.69	0.32	0.27	0.26	0.32	0.27	0.26
$4N_4m_{ww2}$	1.09	0.78	0.68	1.14	0.81	0.70	0.35	0.29	0.28	0.32	0.27	0.27
$4NA_1m_{aa1}$	1.72	1.17	0.99	1.69	1.17	1.00	1.72	1.10	0.96	1.69	1.09	0.94
$4NA_2m_{aa2}$	1.70	1.15	0.99	1.72	1.16	0.99	1.69	1.07	0.95	1.71	1.09	0.96

Table A5.4: **Estimates for relative parameters of *Littorina saxatilis* populations.** Parameter inference was performed with a regression adjustment using a local linear regression and a tolerance rate of 0.01. Results are shown for the Arsklovet and Burela populations using the parallel origin scenario. For this model, n_1 and n_2 correspond to the Arsklovet Crab and Wave population respectively, while n_3 and n_4 correspond to the Burela Crab and Wave population respectively. For each parameter, the value outside brackets corresponds to the mean of the posterior distribution and in-between brackets is the 95% credible interval. Parameters indicated here are the same as in table A5.2.

parameter	relative estimate
n_1	0.200 (0.126 - 0.322)
n_2	0.259 (0.145 - 0.455)
n_3	0.830 (0.344 - 1.727)
n_4	0.574 (0.239 - 1.157)
na_1	1.079 (0.139 - 2.709)
na_2	1.307 (0.158 - 2.799)
rs_1	0.288 (0.026 - 1.130)
rs_2	0.797 (0.105 - 2.363)
δ_s	1.934 (0.309 - 2.925)
m_{cw1}	0.00057 (0.00014 - 0.00096)
m_{wc1}	0.00065 (0.00019 - 0.00097)
m_{cw2}	0.00020 (0.00002 - 0.00070)
m_{wc2}	0.00042 (0.00007 - 0.00089)
m_{cc}	0.00004 (0.00002 - 0.00007)
m_{ww}	0.00001 (0.00000 - 0.00003)
m_{aa}	0.00003 (0.00000 - 0.00009)
P_{cw}	0.024 (0.000 - 0.172)
P_{wc}	0.028 (0.000 - 0.186)
P_{no}	0.037 (0.005 - 0.088)

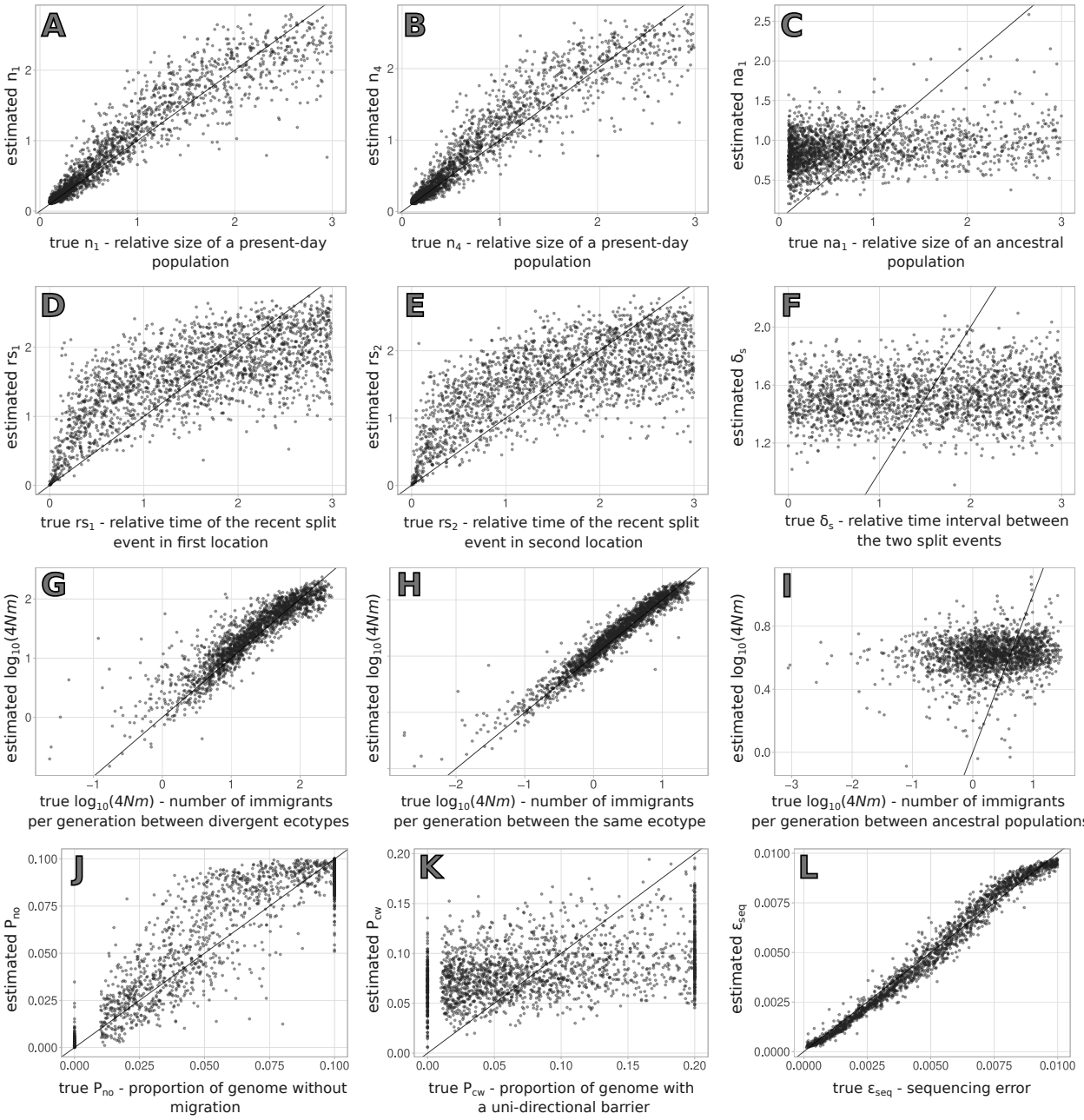


Figure A5.1: Results of the cross-validation for parameter estimation using the parallel origin scenario. The y-axis displays the estimated values, plotted against the true parameter values on the x-axis. Estimates correspond to the mean of the posterior obtained with a tolerance rate of 0.01. Parameters shown here are: A and B - relative size of a present-day population (n_1 and n_2); C - relative size of an ancestral population (na_1); D and E - relative time of the recent split event in the first (rs_1) and second locations (rs_2) respectively; F - relative time interval between the two split events (δ_s); G - average number of immigrants per generation in log_{10} scale between populations of the divergent ecotypes; H - average number of immigrants per generation in log_{10} scale between populations of the same ecotype; I - average number of immigrants per generation in log_{10} scale between ancestral populations; J - proportion of the simulated loci where no migration occurs between ecotypes (P_{no}); K - proportion of the simulated loci where no migration occurs from one of the ecotypes to the other (P_{cw}) and L - sequencing error.

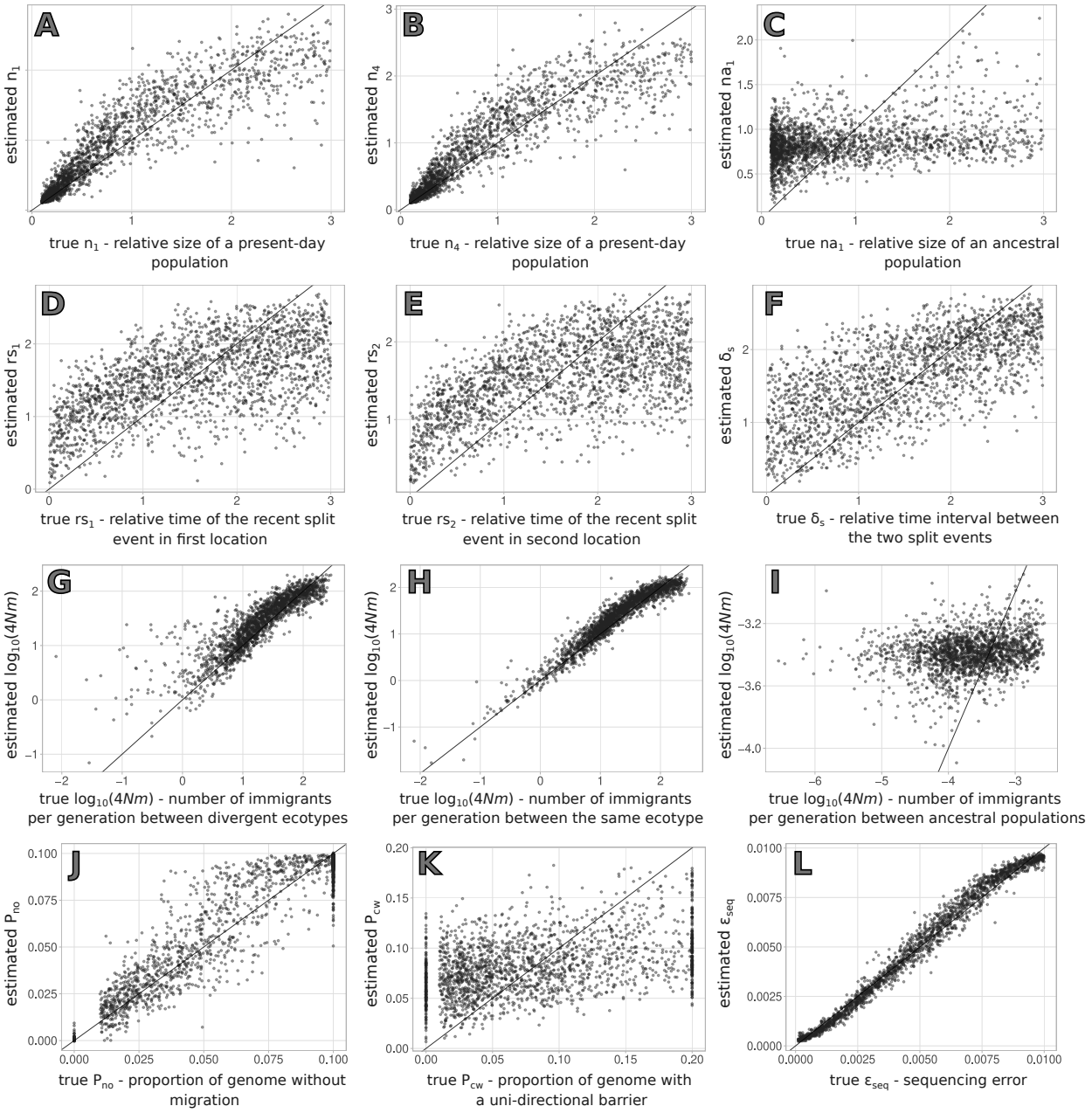


Figure A5.2: Results of the cross-validation for parameter estimation using the single origin scenario. The y-axis displays the estimated values, plotted against the true parameter values on the x-axis. Estimates correspond to the mean of the posterior obtained with a tolerance rate of 0.01. Parameters shown here are: A and B - relative size of a present-day population (n_1 and n_2); C - relative size of an ancestral population (na_1); D and E - relative time of the recent split event in the first (rs_1) and second locations (rs_2) respectively; F - relative time interval between the two split events (δ_s); G - average number of immigrants per generation in log_{10} scale between populations of the divergent ecotypes; H - average number of immigrants per generation in log_{10} scale between populations of the same ecotype; I - average number of immigrants per generation in log_{10} scale between ancestral populations; J - proportion of the simulated loci where no migration occurs between ecotypes (P_{no}); K - proportion of the simulated loci where no migration occurs from one of the ecotypes to the other (P_{cw}) and L - sequencing error.

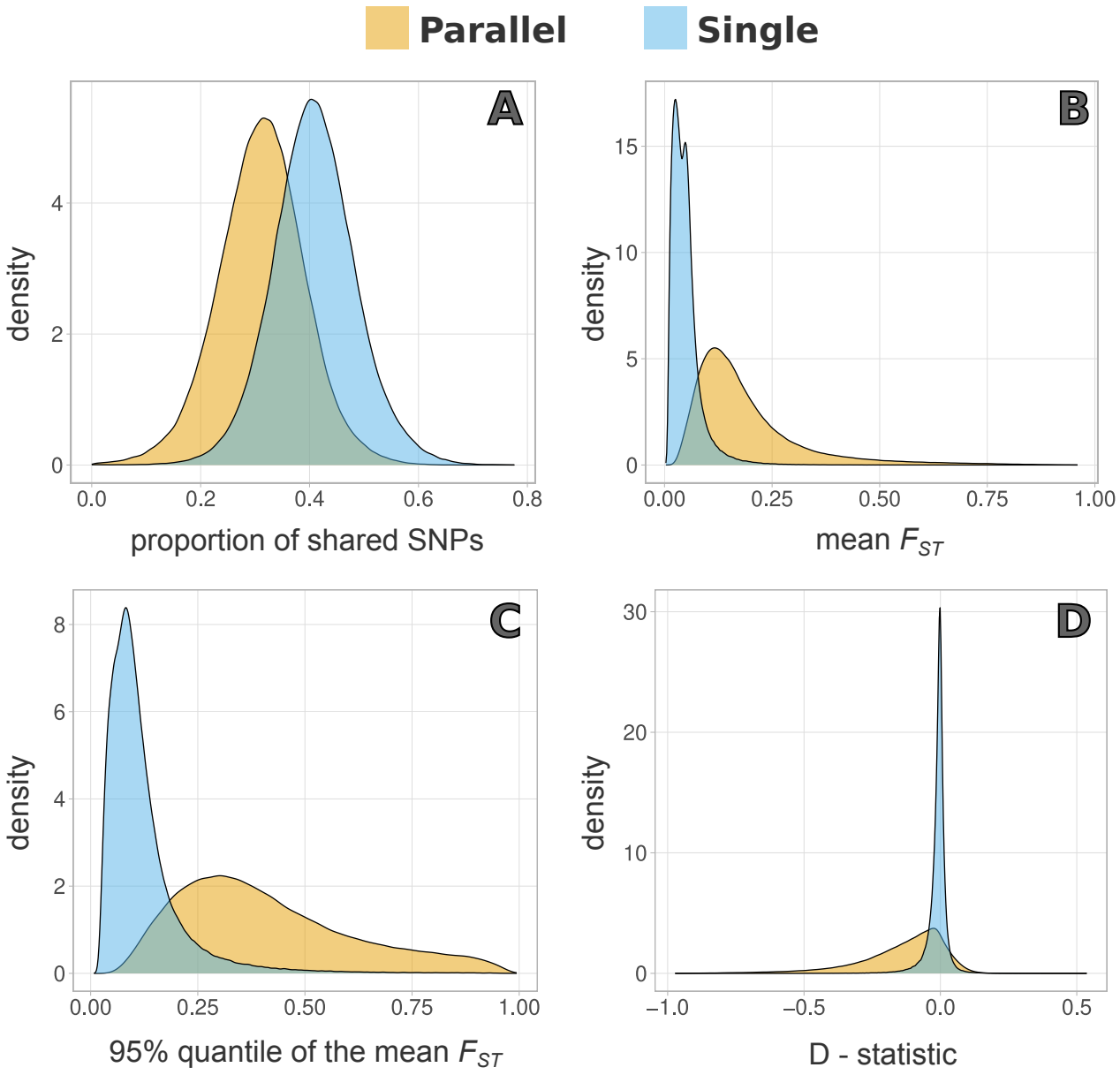


Figure A5.3: Distribution of summary statistics obtained for the single and parallel origin scenarios. Summary statistics are: A - proportion of shared SNPs between all four present-day populations, B - mean pairwise F_{ST} between Wave populations at different locations, C - 95% quantile of the mean pairwise F_{ST} between Crab populations at different locations and D - D-statistic.

CHAPTER 6

General Discussion

GENERAL DISCUSSION

There are many possible routes towards speciation, which among other factors is defined by the main driver behind the evolution of reproductive isolation. These drivers include isolation by distance (i.e. allopatric speciation Price 2008), sexual selection (Seehausen et al., 2008), genetic incompatibilities leading to hybrid speciation (Chapman & Burke, 2007) or divergent natural selection, leading to local adaptation and to progressively more isolated populations (i.e. ecological speciation, Schluter, 2009). A practical way of studying the speciation process is to treat it as a continuum, depicting a series of genetically-driven changes that unfold as two lineages diverge on the path towards complete reproductive isolation (Stankowski & Ravinet, 2021).

Within this framework, the emergence of ecotypes is considered one of the early steps of the route towards ecological speciation (Funk, 2012; Lowry, 2012; Via, 2009). One seldom mentioned aspect of the definition of ecotypes is that they must be observed across multiple locations. Otherwise, if differently adapted forms are only found in a single location, this represents simply an event of local adaptation and ecotype formation does not need to be invoked. Thus, it is not surprising that the study of ecotype formation is often associated with research questions focused on parallel evolution and the repeatability of the evolutionary process. However, the occurrence of similar ecotypes at multiple locations does not necessarily imply parallel evolution (Faria et al., 2014; Johannesson et al., 2010). In fact, different scenarios can lead to the existence of divergently adapted ecotypes at multiple locations. Thus, it is necessary to distinguish between those possible scenarios, which is possible by using whole-genome data and a model-based inference approach.

Ecotype formation hinges on the interplay between the strength of selection and the ability to disperse, making it more likely to take place when strong selective pressures vary within a relatively confined geographic area. Therefore, the intertidal zone represents a unique setting in which to study local adaptation and ecotype formation as it represents one of the sharpest environmental gradients on earth (Little & Kitching, 1996; Raffaelli & Hawkins, 1996; Tomanek & Helmuth, 2002). As a result, divergent ecotypes can be found in several species inhabiting the intertidal, which has lead to many studies focused on their evolution (e.g. Butlin et al., 2014; Kemppainen, Lindskog, Butlin, & Johannesson, 2011; Marshall, Taha, Brahim, & Abdelhady, 2021; Ruesink, Ortiz, Mawson, & Boardman, 2022). Nevertheless, species inhabiting the intertidal zone have rarely been studied in the context of the speciation continuum i.e. studying the process across population pairs that are located at different stages along the continuum. This might partly be due to the challenge in obtaining whole-genome data from multiple populations of different species. One of the best possibilities for obtaining population-level genomic data from multiple locations,

known as pooled sequencing, is associated with specific sources of error that, until now, have impeded its use as a source of genetic polymorphisms for demographic inference. Consequently, even though pooled sequencing holds considerable promise as a means of accessing genome-wide data to investigate whether ecotype formation resulted from a sequence of parallel events across multiple species, this potential has largely remained untapped.

This thesis is focused on two distinct species within the *Littorina* genus that potentially represent different points along the speciation continuum. While measuring reproductive isolation is challenging (Westram, Stankowski, Surendranadh, & Barton, 2022), making it difficult to precisely place population pairs along the speciation continuum, several lines of evidence within each species indicate that *L. fabalis* and *L. saxatilis* ecotypes may represent distinct stages in the process of diversification. Firstly, studies of male mate choice in both species suggest that size-related mating barriers are stronger in *L. saxatilis* than in *L. fabalis* (Perini, Rafajlović, Westram, Johannesson, & Butlin, 2020; Saltin, Schade, & Johannesson, 2013). Secondly, F_{ST} estimates from northern Europe suggest a higher degree of differentiation between *L. fabalis* ecotypes (ranging from 0.02 to 0.08; Galindo et al., 2021) compared to *L. saxatilis* ecotypes in the same region (F_{ST} = 0.04 in Sweden; Westram, Faria, Johannesson, & Butlin, 2021).

However, in Iberia, the situation is less clear, with F_{ST} estimates ranging between 0.12 to 0.24 for *L. saxatilis* ecotypes (Kess, Galindo, & Boulding, 2018) and between 0.10 to 0.23 for *L. fabalis* (Carvalho, Sotelo, Galindo, & Faria, 2016). While this might suggest a slightly higher isolation between *L. saxatilis* ecotypes in Iberia, mean F_{ST} computed using whole-genome Pool-seq data results in genome-wide values of genetic differentiation ranging from 0.09 to 0.12 for Spanish *L. saxatilis* ecotypes (Morales et al., 2019). The F_{ST} estimates reported here were obtained with a variety of genetic markers, ranging from microsatellite loci (Carvalho et al., 2016), RAD loci (Kess et al., 2018) to Pool-seq data (Morales et al., 2019). This highlights the need for more standardized measures of divergence between the ecotypes of the two species. It is also important to note that *L. fabalis* ecotypes are commonly found in association with different algae (Williams, 1990), which might represent an additional axis of divergence that is not present in *L. saxatilis*.

The main question throughout this study was whether the phenotypic divergence observed in these species is the outcome of parallel evolution occurring in multiple locations or a singular occurrence. Throughout this thesis, I have developed and implemented methodologies tailored to model and simulate pooled sequencing data, integrating it into demographic inferences designed to test different scenarios of ecotype formation. Using different lines of evidence, this thesis establishes that the observed phenotypic divergence within the two *Littorina* species is likely the result of parallel events. The findings and methods developed in this thesis have the potential to contribute

to a broader investigation of speciation and adaptation in other taxa that exhibit analogous patterns of phenotypic divergence across multiple geographical locations.

6.1 EVIDENCE SUPPORTING THE HYPOTHESIS OF PARALLEL EVOLUTION IN *L. FABALIS*

In **Chapter 2**, we extended the knowledge about local adaptation in marine intertidal species by studying the genetic and morphological divergence between *Littorina fabalis* ecotypes. The work developed in **Chapter 2** constituted the first attempt to identify genetic variation linked to divergent ecotypes in terms of wave-exposure and/or related ecological factors. It was also the first evaluation of the degree of sharing of outlier loci among ecotypes across *L. fabalis* populations from the UK, Sweden and Norway.

The morphological analysis revealed consistent differences in shell morphology between *L. fabalis* ecotypes. Besides these habitat-related differences in shell morphology, our results from **Chapter 2** also highlighted the existence of important location-specific effects on shell morphology. The genetic analysis revealed heterogeneous differentiation between ecotypes, with a relatively small fraction of loci displaying high levels of differentiation. These loci (i.e. outlier loci) appear to resist the substantial gene flow that contributes to eroding differentiation to very low levels in the rest of the genome. The results obtained here support the idea that natural selection plays a significant role in driving divergence between *L. fabalis* ecotypes in the presence of gene flow.

The level of outlier sharing identified in this chapter supports the hypothesis of evolution in concert, suggesting that certain adaptive alleles spread across populations through gene flow while others might have originated locally (Johannesson et al., 2010; Kemppainen et al., 2011). Another possible explanation is a significant contribution from shared standing genetic variation inherited from a common ancestral population (Westram, Panova, Galindo, & Butlin, 2016). However, due to the limited number of loci genotyped, the resolution to precisely assess the number of outliers and the extent of sharing across locations is relatively low. The results from **Chapter 2** also suggest that ecotype formation could have occurred in parallel in *L. fabalis*. Indeed, the clustering of populations from both ecotypes by geography, when considering nonoutlier loci, supports the hypothesis of parallel evolution. Interestingly, the results from this chapter also revealed that nearly half of all outliers are unique to one location, suggesting that some aspects of divergent evolution are site-specific. Therefore, the hypothesis of parallel evolution in this system requires further examination. In the future, implementing a modelling approach using whole-genome data, similar to

the methodology developed in other chapters of this thesis, would be highly valuable. However, Pool-seq data are not currently available for this system. Subsequent chapters of this thesis show that it would be important to generate such data for this species, as it would allow the inference of the demographic history of *L. fabalis* ecotypes, with a specific focus on investigating the possibility of parallel evolution within this system.

The *L. fabalis* system has several similarities with the *Littorina saxatilis* system but it has received comparatively less attention in research. The work conducted in **Chapter 2** could help in establishing *L. fabalis* as a valuable comparison with *L. saxatilis*, allowing us to gather information from multiple instances of divergence across similar environmental transitions in different species. Indeed, investigating local adaptation in intertidal habitats of rocky shores is crucial for quantifying the contribution of ecological speciation to marine biodiversity (Faria, Johannesson, & Stankowski, 2021; Sanford & Kelly, 2011). In conclusion, the *L. fabalis* system represents an interesting system from the marine environment where multiple instances of divergence occur across an environmental transition. Further studies within this system, along with comparisons to *Littorina saxatilis*, are likely to enhance our comprehension of how populations adapt to environmental changes and how distinct reproductive barriers accumulate during the process of speciation.

6.2 POOL-SEQ ALLOWS DISTINGUISHING BETWEEN SCENARIOS OF ECOTYPE FORMATION

In **Chapter 3**, we developed a new approach to model and simulate Pool-seq data and incorporated it into an R package named *poolHelper*. The work developed in this chapter takes into consideration known Pool-seq sources of errors. In particular, our method models the unequal contribution due to variations in DNA concentration and amplification efficiency during DNA extraction and library preparation, fluctuations in sequencing depth across different SNPs, as well as errors arising from sequencing and read alignment. Importantly, these sources of errors are modelled by parameters that can be directly adjusted by the user. Our approach differs from previous efforts that did not explicitly model variation introduced by unequal individual contribution and ignored sequencing errors (e.g. Collin et al., 2021; Taus, Futschik, & Schlötterer, 2017). By incorporating these parameters, our method offers a more comprehensive and detailed simulation of Pool-seq data, leading to a more accurate representation of the experimental conditions. As a consequence, the R package, *poolHelper*, enables the simulation of more realistic data that can be effectively used with existing downstream methods and analysis pipelines, providing a valuable resource for

investigating the performance of different methodologies and optimizing Pool-seq experiments for specific research objectives. Additionally, the implementation of this method into an R package contributes to a more reproducible and user friendly research.

The results we obtained in this chapter illustrate that selecting the optimal Pool-seq study sampling design is a complex task, as various combinations of pool sizes and average depth of coverage can yield similar errors in allele frequencies in the sample. This highlights the importance of conducting *a priori* simulation study to evaluate the most efficient sampling scheme for obtaining precise allele frequencies while minimising the sequencing effort. To address this need, the *poolHelper* package provides functions for simulating Pool-seq data under a range of user-defined conditions.

The *poolHelper* package is freely available to the scientific community, providing researchers with the ability to simulate pooled sequencing data based on a model of a single panmictic population. It allows users to compute the error in sample allele frequencies and expected heterozygosity obtained with different Pool-seq experimental designs and commonly used filters. Researchers can conveniently apply filters such as those on minimum and maximum depth of coverage and minimum number of minor-allele reads to explore different scenarios and analyse the impact on the mean error. By providing users with the ability to modify parameters such as the average coverage and the error associated with unequal individual contribution, and by offering immediate visualisation of how these adjustments impact allele frequency estimates, the package facilitates a clearer comprehension of the uncertainties associated with Pool-seq. To contribute to reproducibility, we also created a package manual and a detailed vignette. The manual provides comprehensive documentation of the package's functions, while the vignette offers a thorough explanation of how to use the package, facilitating a deeper understanding of the package's functionalities and promoting its implementation.

Chapter 4 integrates the work developed in the previous chapter, along with the framework established by Gautier et al. (2013) into an Approximate Bayesian Computation (ABC) inference framework. In this chapter, we present a novel model-based method to analyse pooled-sequencing data, allowing the joint-inference of various Pool-seq sources of error (e.g., variation in depth of coverage, unequal individual contribution, merging multiple pools) and demographic parameters. The method we developed in **Chapter 4** is computationally efficient as it relies on simulating subsets of loci, estimating relative parameters and using relative summary statistics. Furthermore, to circumvent the difficulties in implementing ABC for whole-genome data (Smith & Flaxman, 2020), the method relies on the simulation of multiple subsets of independent loci. The posterior distributions obtained for each of these subsets are subsequently combined, using a weighting scheme

based on their proximity to the genome-wide mean summary statistics. Given that demographic history is expected to uniformly affect all loci throughout the genome, this approach not only overcomes computational barriers, but it also mitigates the influence of outlier subsets of loci caused by non-neutral processes (e.g., background selection) on the posterior estimates. We implemented this method as a freely available R package, named *poolABC*, enabling users to conduct model choice and perform parameter inference of demographic history using Pool-seq data from natural populations. The package provides a user-friendly tool to take advantage of Pool-seq data in the context of demographic inference, facilitating the investigation of evolutionary processes in a wide range of organisms.

Results from the simulation study conducted in **Chapter 4** show that the means of the posterior distributions provide accurate point estimates for most demographic history parameters of the two- and four-population models considered. Furthermore, the prediction errors obtained in this simulation study using Pool-seq data were comparable to those of recent ABC methods based on individual genotypes (Fraïsse et al., 2021). Interestingly, results from the simulation study also demonstrated the feasibility of estimating the proportion of loci without migration, indicating the potential for estimating the number of barrier loci under selection. Additionally, the model choice results from the simulation study show that Pool-seq data provides sufficient information to differentiate between scenarios of ecotype formation with high posterior probabilities. The results of **Chapter 4** also clearly highlight the importance of considering Pool-Seq errors, as neglecting them can lead to inaccurate estimates. On the other hand, explicitly modeling Pool-Seq errors allows for accurate estimation of demographic parameters. This finding emphasizes the critical role of accounting for sequencing errors, pooling errors, and other sources of uncertainty in Pool-seq data analysis, as it significantly impacts parameter estimates and hence the reliability of demographic inference.

While the models considered in **Chapter 4** were relatively simple, we also assessed the performance of our method using more complex models in **Chapter 5**. These models include the possibility of migration between ancestral populations and between populations of the same ecotype inhabiting different locations. Additionally, the models considered in **Chapter 5** also account for the possibility of uni-directional barriers to gene flow between the ecotypes, instead of only considering a total barrier between both ecotypes. The results we obtained in this chapter demonstrated that the means of the posterior distributions offer accurate point estimates for most demographic history parameters. However, the parameters related to the ancestral populations, such as their relative sizes and migration between them, showed high uncertainty. This suggests that the summary statistics used may lack information about older events or that Pool-seq may not be the optimal choice for inferring those specific parameters when working with complex models. Alternatively,

this might be a result of the limited ammount of information used in this study, as we discarded roughly 80% of the available contigs. Nevertheless, results from the simulation study in **Chapter 5** confirmed that Pool-seq provides adequate information to distinguish between scenarios of ecotype formation, even for the more complex models considered here. This distinction remained possible even at high posterior probabilities, reinforcing the potential of combining Pool-seq with ABC for investigating parallel evolution.

Although the costs of sequencing continue to decrease, our results clearly show that Pool-seq is still a viable sequencing alternative. This holds true, especially when studying small organisms like some of the *Littorina* ecotypes, or when conducting Evolve-and-Resequence experiments and investigating parallel evolution. These studies often involve a substantial number of populations or numerous sampling points spanning spatial gradients or temporal sequences. The work conducted in **Chapter 4** successfully bridged the gap between Pool-seq data and demographic inference, resulting in an easy-to-use method that harnesses whole-genome data to distinguish between various scenarios of ecotype formation. Our findings highlight that Pool-seq can be a cost-effective approach to demographic inference, particularly when the research question implies sampling and comparing multiple populations.

The resulting R package allows users to seamlessly integrate our package with other tools at various stages of their analyses (e.g., abc R package, Csilléry, François, and Blum 2012). This user-friendly tool facilitates the exploration and interpretation of complex demographic processes using Pool-seq data and serves as a valuable resource for researchers working in the field of population genetics and evolutionary biology. In this context, the implementation of this method into an R package provides a user friendly tool to encourage the use of Pool-seq data in demographic inference. As in the previous chapter, we also elaborated a comprehensive package manual and a detailed vignette that provides a thorough explanation of how to use the package. Importantly, our R package includes functions to compute prediction errors and to assess the fit of the models to the data, allowing users to perform simulation studies based on their specific set of models, prior distributions, sample sizes, depths of coverage and numbers of pools. Despite some limitations, the results from **Chapter 4** and **Chapter 5** clearly show that combining Pool-seq with ABC is an effective approach for investigating parallel evolution in taxa where similar ecotypes exist at multiple locations. The demographic history models explored in this chapter serve as appropriate null models that could be used to gain a better understanding of the genetic basis of divergent adaptation across various taxa.

6.3 DETECTING PARALLEL ECOTYPE FORMATION AT DIFFERENT GEOGRAPHIC SCALES

The methods developed in **Chapter 3** and **Chapter 4** were implemented in the analysis of Pool-Seq data. As previously mentioned, this method combines an Approximate Bayesian Computation (ABC) inference framework with explicit modeling of various sources of error associated with pooled sequencing. In **Chapter 4**, we demonstrated the application of these methods by analysing pools of Swedish populations of *L. saxatilis* ecotypes. Subsequently, in **Chapter 5**, the same methods were applied to analyse pools of both Swedish and Spanish populations of *L. saxatilis* ecotypes.

In **Chapter 4**, we found that Pool-seq data from Crab and Wave ecotypes in Sweden were compatible with a single origin of the ecotypes. The point estimates indicate that ecotype divergence occurred relatively recently, approximately 15,000 years ago (95% CI: 5000 to 43000 years), followed by a population split in different locations around 1,000 generations ago (roughly 500 years ago; 95% CI: 300 to 800 years), with high gene flow between ecotypes. These results are consistent with the hypothesis of a recent postglacial colonization of Swedish islands (Panova et al., 2011). As mentioned, the goal of **Chapter 4** was to develop and showcase the implementation of an ABC method using Pool-seq data, assessing its performance under generic divergence models involving two or four populations. Thus, the models examined in this chapter were relatively simple and may not fully capture the intricate process of ecotype formation in the geographically restricted *L. saxatilis* Crab and Wave ecotypes. As a result, it remained uncertain whether Pool-seq data combined with an ABC framework could effectively distinguish between more complex models of ecotype formation.

To adress this, in **Chapter 5**, we considered two possible scenarios of ecotype formation in *L. saxatilis* that are considerably more complex than those analysed in the previous chapter, as they encompass a broader range of migration possibilities and uni-directional barriers to gene flow (see above). We contrasted those two scenarios using whole-genome data obtained through pool-sequencing and the model-based method developed in **Chapter 3** and **Chapter 4**.

Our findings from **Chapter 5** support the parallel origin of the Crab and Wave ecotypes in *L. saxatilis*, without allopatric separation, and after colonization of the different regions. The estimated divergence times suggest a relatively recent divergence of the ecotypes in both countries. In Sweden, the divergence appears to be more recent, occurring around 15,000 years ago (95% CI: 3k to 115k years), while in Spain, it took place approximately 57,000 years ago (95% CI: 11k to 241k

years). These estimates are consistent with the idea of a recent postglacial colonization of Swedish islands (Panova et al., 2011) and are within the range of previous estimates (Butlin et al., 2014). Additionally, the results indicate that the divergence process between ecotypes in both Spain and Sweden was accompanied by gene flow, with high migration rates estimated between the divergent ecotypes. In the majority of the comparisons examined, these rates exceeded $4Nm > 10$, indicating substantial ongoing genetic exchange despite the divergence in ecological traits between the two ecotypes. Nevertheless, and using only known collinear regions, we inferred that there is no migration between divergent ecotypes at roughly 4% of analysed loci (CI: 0.5% to 8.8%). This suggests that, while migration rates may be high overall, this does not influence all loci uniformly.

Overall, our results in **Chapter 5** strongly suggest a demographic history characterized by the spatial division of an ancestral population into two distinct groups: a Spanish ancestral population and another ancestral population from which the ecotypes in Sweden originated. Over time, *L. saxatilis* populations evolved independently into distinct ecotypes associated with different habitats, while maintaining gene flow between them. The divergence between ecotypes is much older in Spain and may potentially represent a further step along the speciation continuum than the divergence in Sweden, where the ecotypes diverged more recently.

Our findings confirm that comparisons between *L. saxatilis* populations in Spain and Sweden represent a true example of parallel evolution, providing evidence of the impact of natural selection on the divergence process (Lenormand, Roze, & Rousset, 2009; Schluter & Nagel, 1995). Thus, the evidence strongly supports the role of divergent selection in the formation of Crab and Wave ecotypes, despite ongoing genetic exchange, potentially influencing the evolution of chromosomal rearrangements such as inversions (Kirkpatrick & Barton, 2006; Ortiz-Barrientos, Engelstädter, & Rieseberg, 2016). However, the models considered in **Chapter 5** did not specifically include periods of allopatry, leaving open the possibility that such periods might have influenced the evolution of *L. saxatilis* ecotypes. Furthermore, the support for our results depends on the impact that the barriers to gene flow between populations in the contrasting environments have on neutral loci. It is still uncertain whether some alleles responsible for adaptive traits, potentially linked to inversions, have evolved in parallel. For instance, chromosomal inversions between two *Drosophila* species were already present in their ancestral population long before the species split (Fuller, Leonard, Young, Schaeffer, & Phadnis, 2018). Thus, it is possible that *L. saxatilis* chromosomal inversions were already present in an ancestral population before the ecotypes were formed and that the origin of those inversions is not due to recurrent evolution of the same inversions in multiple locations. Interestingly, recent evidence suggests that some inversions are widespread (Westram, Morales, Johannesson, Butlin, & Faria, 2023) and thus might represent ancestral polymorphism.

Interestingly, the results from **Chapter 4** support a single origin of *L. saxatilis* Crab and Wave ecotypes in Sweden, while the results from **Chapter 5** support a parallel origin of the ecotypes when contrasting populations from Spain and Sweden. This implies that the geographical scale considered might influence the results obtained. Model-based inference studies focused on ecotypes are mostly used to contrast between two-population models representing scenarios of strict isolation, isolation with migration or secondary contact (e.g. Le Moan, Gagnaire, & Bonhomme, 2016; Rougeux, Bernatchez, & Gagnaire, 2017), without explicitly contrasting between parallel or a single origin scenarios. Thus, it is not clear if model support is influenced by the geographical scale in other taxa. Here, by considering models with four populations it was possible to distinguish such scenarios.

Nevertheless, several studies have shown the existence of some extent of genomic-level parallelism. This parallelism can be explained by standing genetic variation shared among lineages due to pre- or post-divergence gene flow, as is the case with stick insects (Soria-Carrasco et al., 2014), the saltmarsh beetle *Pogonus chalceus* (Van Belleghem et al., 2018) and bottlenose dolphins (Louis et al., 2021). Alternatively, it can arise by recurrent de novo mutations with large phenotypic effects, as observed in beach mice (Hoekstra, Hirschmann, Bundey, Insel, & Crossland, 2006) and *Antirrhinum species* (Tan et al., 2020). Conversely, there is also compelling evidence of phenotypic convergence resulting from nonparallel signatures of adaptation in beach mice (Steiner, Römpler, Boettger, Schöneberg, & Hoekstra, 2009), cichlid fishes (Elmer et al., 2014) and guppies (Fischer, Song, Hughes, Zhou, & Hoke, 2021).

These and other studies have made it clear that evolution of phenotypic similarity can involve highly heterogeneous routes contingent upon factors such as gene flow variation, effective population size, strength of selection and demographic history, leading to different degrees of parallelism (Yeaman, Gerstein, Hodgins, & Whitlock, 2018). Although some studies have pointed out that the geographical scale considered can impact the level of parallelism observed (Fang, Kemppainen, Momigliano, Feng, & Merilä, 2020; Ravinet et al., 2016), there is a clear need for more studies that explicitly contrast scenarios of parallel origin with alternative scenarios across different geographical scales. Appropriately, while **Chapter 4** showed that combining Pool-seq with ABC was an effective approach for investigating parallel evolution at a limited geographical range, **Chapter 5** showed that it is also useful across a wide geographical range and using more complex models. Thus, this approach can be used to investigate scenarios of (non)parallel evolution, not only in other *L. saxatilis* populations sampled from Sweden, Spain, and the United Kingdom, but also in a range of other species, including *L. fabalis*.

6.4 LIMITATIONS AND FUTURE WORK

The conclusions drawn in this thesis are based on the results obtained herein and are subject to the limitations of the research conducted. To better understand if the evolution of phenotypic divergence observed in *L. fabalis* and *L. saxatilis* is the outcome of a series of parallel events or a singular event requires further research. Indeed, for *L. fabalis*, a more comprehensive and representative whole-genome dataset encompassing multiple populations is needed. This expanded dataset should then be analysed using model-based inference methods to compare and contrast various scenarios of ecotype formation specific to this species. The work conducted in Chapter 2 was centered on *L. fabalis* populations from northern Europe. As evidenced by our results for *L. saxatilis*, the inference of parallel or single origin scenarios might depend on the geographical scale considered, and hence it will be necessary to study the demographic history of *L. fabalis* across different geographical scales. Using whole-genome data, future research can compare populations within individual countries and contrast populations from the Iberian Peninsula, where three different ecotypes can be found (Carvalho et al., 2016), with those in northern Europe. This broader perspective will allow us to contrast between different scenarios of ecotype formation, both at a local scale and across a wide geographical range. This, in turn, will aid us in determining whether the *L. fabalis* system is a true example of parallel evolution or not. Furthermore, while the relative placement of *L. fabalis* and *L. saxatilis* ecotypes on the speciation continuum might differ for northern European or Iberian ecotypes, a more comprehensive understanding of the level of reproductive isolation and the relative significance of barriers such as assortative mating is required.

For *L. saxatilis*, the main limitation of this thesis is the absence of additional pairwise comparisons among different Spanish populations and between populations in Sweden, Spain, and the United Kingdom. Conducting these comparisons will be crucial in unraveling whether ecotype formation occurred in parallel across various geographical scales. By including a broader range of populations in the analyses, future research can confirm if a single origin scenario is always supported at a local scale or if Sweden stands as an exception. Additionally, future work should also explore different models with varying strengths of selection at barrier loci or the possibility of one ecotype serving as a reservoir of standing genetic variation, similar to the stickleback system (Jones et al., 2012; Liu, Ferchaud, Grønkjær, Nygaard, & Hansen, 2018). In this system, the influx of alleles, including chromosomal inversions, adapted to freshwater environments occurs through gene flow from established freshwater populations into marine gene pools. These alleles are subsequently available for establishing new freshwater populations, via a scenario compatible with the transporter hypothesis (Schluter & Conte, 2009). Given the importance of chromosomal inversions in the ecological divergence of *L. saxatilis* (Faria et al., 2019; Koch et al., 2021; Morales et al.,

2019), even across a wide geographical range (Morales et al., 2019), it would be interesting to include those genomic features in our models. Specifically, it remains unclear if the relatively recent divergence of *L. saxatilis* ecotypes could be modelled by including chromosomal inversions as a source of shared variation. This may facilitate rapid parallel adaptation to diverse environments, akin to what has been demonstrated in *Drosophila* species (Lohse, Clarke, Ritchie, & Etges, 2015).

At a more technical level, the R packages developed in the context of this thesis and presented in Chapters 3 and 4 would also benefit from further work. Currently, the *poolHelper* package is primarily designed for a single population, and simulations are carried out under this assumption, assuming a constant effective population size. While it is possible to use the package with multiple populations, this is not straightforward. Further work is required to streamline the process and allow users to work with more than one population. For instance, the package could be extended to accommodate populations sequenced at different depths of coverage. Similarly, the *poolABC* package could be improved by incorporating more complex scenarios of ecotype formation, akin to those explored in Chapter 5. To enhance its functionality, the package could be extended to include more sophisticated algorithms, such as neural networks or random forest ABC (as included for instance in the abc R package - Csilléry et al., 2012 or the DIYABC-RF software - Collin et al., 2021), and provide additional options for summary statistics. These improvements would improve the accuracy of the inferences made and enable researchers to investigate more intricate evolutionary scenarios using Pool-seq data.

Further work is necessary to establish flat (*L. fabalis*) and rough periwinkles (*L. saxatilis*) as model systems for studying local adaptation and the early stages of speciation. While the idea of comparing mechanisms and processes across two distinct species facing similar selective pressures is intriguing, the disparity in research effort between the two species currently limits this comparison. As mentioned, a more thorough understanding of the demographic history of *L. fabalis* ecotypes and the genomic regions involved in their divergence would enable more meaningful comparisons between the two systems. Specifically, a compelling avenue of investigation would involve comparing the demographic histories of the ecotypes in both species and contrasting the genomic regions under divergent selection that are potentially involved in phenotypic differences. Considering that *L. fabalis* and *L. saxatilis* represent examples of local adaptation along comparable intertidal ecological gradients, it would be interesting to assess whether adaptation to similar selective pressures involves the same genomic regions in two different species. Furthermore, given the association of *L. fabalis* ecotypes with different types of algae (Carvalho et al., 2016; Williams, 1990), which could influence habitat preference, it would be valuable to assess and compare the relative importance of pre- and post-zygotic barriers to gene flow in the two species and whether any differences in relative importance are related with their placement in the speciation continuum.

Notably, the recent discovery of chromosomal inversions involved in the adaptive divergence of *L. fabalis* ecotypes (Le Moan et al., 2023) opens up exciting possibilities for future research. Given their potential role in driving local adaptation to similar selective pressures, it would be interesting to investigate whether the same inversions also play a role in the divergence process of both *Littorina* species. This would require a comparative genomic study of *L. fabalis* and *L. saxatilis*, analysing the distribution and frequency of the chromosomal inversions in different populations of both species. Furthermore, population genetic analyses could be used to trace the evolutionary history of these inversions by examining patterns of genetic variation and linkage disequilibrium around the inversion breakpoints. In conclusion, comparative genomic studies of both species, with a specific focus on chromosomal inversions or other genomic regions, hold great promise for advancing our understanding of the genetic basis of adaptation and speciation in marine organisms. Understanding how marine organisms adapt and evolve is crucial for predicting and managing the impact of environmental changes and human activities on marine ecosystems.

6.5 FINAL REMARKS

This thesis presents compelling evidence that the observed phenotypic divergence in two marine gastropod species from the *Littorina* genus likely resulted from independent and parallel evolution in different geographic locations. Furthermore, it emphasizes that this divergence was probably accompanied by gene flow between the divergent ecotypes. The work developed here, particularly the two R packages and associated methodologies, are likely applicable to any taxa characterized by the existence of similar ecotypes across a wide geographical range. These tools allow researchers to simulate Pool-seq data tailored to their specific research questions and integrate this simulated data into a model-based inference framework. As a result, this thesis is expected to promote the wider adoption of Pool-seq data for investigating questions related to parallel evolution and ecotype formation. Moreover, this study highlights that explicitly contrasting demographic models is essential to disentangle between scenarios of ecotype formation and establish which species constitute true examples of parallel evolution.

This thesis represents an example of research on speciation and local adaptation, encompassing ecotypes of different species that potentially represent different stages of the speciation continuum. It underscores the significance of comparative studies, in advancing our comprehension of parallel evolution, local adaptation, ecotype formation, and the reproducibility of evolutionary processes. In the future, such studies should be extended to identify patterns that extend beyond individual species or environments. That would provide valuable information about the relative importance

of pre- and post-zygotic barriers to gene flow at different stages of the speciation continuum, the evolution of the genetic architecture during the speciation process, for instance whether capture of genomic variation associated with divergence in chromosomal inversions is fundamental at any stage of the continuum, or whether adaptation to similar selective pressures in different species follows a predictable pattern. A critical first step in all these studies is to reconstruct the demographic history and determine whether ecotypes evolved in parallel with gene flow, which, as shown in this thesis, can be accomplished using Pool-seq data. I am convinced that combining population genomics datasets with modelling to examine multiple cases across the speciation continuum will allow us to understand the complex dynamics that shape biodiversity and the origin of species.

6.6 REFERENCES

Butlin, R. K., Saura, M., Charrier, G., Jackson, B., André, C., Caballero, A., … Rolán-Alvarez, E. (2014). Parallel evolution of local adaptation and reproductive isolation in the face of gene flow. *Evolution*, *68*(4), 935-949. doi: 10.1111/evo.12329

Carvalho, J., Sotelo, G., Galindo, J., & Faria, R. (2016). Genetic characterization of flat periwinkles (littorinidae) from the iberian peninsula reveals interspecific hybridization and different degrees of differentiation. *Biological Journal of the Linnean Society*, *118*(3), 503-519. doi: 10.1111/bij.12762

Chapman, M. A., & Burke, J. M. (2007). Genetic divergence and hybrid speciation. *Evolution*, *61*(7), 1773-1780. doi: 10.1111/j.1558-5646.2007.00134.x

Collin, F.-d., Durif, G., Raynal, L., Lombaert, E., Gautier, M., Vitalis, R., … Estoup, A. (2021). Extending approximate bayesian computation with supervised machine learning to infer demographic history from genetic polymorphisms using diyabc random forest. *Molecular Ecology Resources*, *21*(8), 2598-2613. doi: 10.1111/1755-0998.13413

Csilléry, K., François, O., & Blum, M. G. (2012). Abc: An r package for approximate bayesian computation (abc). *Methods in Ecology and Evolution*, *3*(3), 475-479. doi: 10.1111/j.2041-210X.2011.00179.x

Elmer, K. R., Fan, S., Kusche, H., Luise Spreitzer, M., Kautt, A. F., Franchini, P., & Meyer, A. (2014). Parallel evolution of nicaraguan crater lake cichlid fishes via non-parallel routes. *Nature communications*, *5*(1), 1-8. doi: 10.1038/ncomms6168

Fang, B., Kemppainen, P., Momigliano, P., Feng, X., & Merilä, J. (2020). On the causes of geographically heterogeneous parallel evolution in sticklebacks. *Nature ecology & evolution*, *4*(8), 1105-1115. doi: 10.1038/s41559-020-1222-6

Faria, R., Chaube, P., Morales, H. E., Larsson, T., Lemmon, A. R., Lemmon, E. M., … Butlin, R. K. (2019). Multiple chromosomal rearrangements in a hybrid zone between *Littorina saxatilis* ecotypes. *Molecular Ecology*, *28*(6), 1375-1393. doi: 10.1111/mec.14972

Faria, R., Johannesson, K., & Stankowski, S. (2021). Speciation in marine environments: Diving under the surface. *Journal of Evolutionary Biology*, *34*(1), 4-15. doi: 10.1111/jeb.13756

Faria, R., Renaut, S., Galindo, J., Pinho, C., Melo-Ferreira, J., Melo, M., ... Butlin, R. K. (2014). Advances in ecological speciation: an integrative approach. *Molecular Ecology*, *23*(3), 513-521. doi: 10.1111/mec.12616

Fischer, E. K., Song, Y., Hughes, K. A., Zhou, W., & Hoke, K. L. (2021). Nonparallel transcriptional divergence during parallel adaptation. *Molecular Ecology*, *30*(6), 1516-1530. doi: 10.1111/mec.15823

Fraïsse, C., Popovic, I., Mazoyer, C., Spataro, B., Delmotte, S., Romiguier, J., ... Roux, C. (2021). Dils: Demographic inferences with linked selection by using abc. *Molecular Ecology Resources*, *21*(8), 2629-2644. doi: 10.1111/1755-0998.13323

Fuller, Z. L., Leonard, C. J., Young, R. E., Schaeffer, S. W., & Phadnis, N. (2018). Ancestral polymorphisms explain the role of chromosomal inversions in speciation. *PLoS genetics*, *14*(7), e1007526. doi: 10.1371/journal.pgen.1007526

Funk, D. J. (2012). Of "host forms" and host races: Terminological issues in ecological speciation. *International Journal of Ecology*. doi: 10.1155/2012/506957

Galindo, J., Carvalho, J., Sotelo, G., Duvetorp, M., Costa, D., Kemppainen, P., ... Faria, R. (2021). Genetic and morphological divergence between littorina fabalis ecotypes in northern europe. *Journal of Evolutionary Biology*, *34*(1), 97-113. doi: 10.1111/jeb.13705

Gautier, M., Foucaud, J., Gharbi, K., Cézard, T., Galan, M., Loiseau, A., ... Estoup, A. (2013). Estimation of population allele frequencies from next-generation sequencing data: pool-versus individual-based genotyping. *Molecular Ecology*, *22*(14), 3766-3779. doi: 10.1111/mec.12360

Hoekstra, H. E., Hirschmann, R. J., Bundey, R. A., Insel, P. A., & Crossland, J. P. (2006). A single amino acid mutation contributes to adaptive beach mouse color pattern. *Science*, *313*(5783), 101-104. doi: 10.1126/science.1126121

Johannesson, K., Panova, M., Kemppainen, P., André, C., Rolán-Alvarez, E., & Butlin, R. K. (2010). Repeated evolution of reproductive isolation in a marine snail: unveiling mechanisms of speciation. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *365*(1547), 1735-1747. doi: 10.1098/rstb.2009.0256

Jones, F. C., Grabherr, M. G., Chan, Y. F., Russell, P., Mauceli, E., Johnson, J., ... Team, B. I. G. S. P. . W. G. A. (2012). The genomic basis of adaptive evolution in threespine sticklebacks. *Nature*, *484*(7392), 55-61. doi: 10.1038/nature10944

Kemppainen, P., Lindskog, T., Butlin, R. K., & Johannesson, K. (2011). Intron sequences of arginine kinase in an intertidal snail suggest an ecotype-specific selective sweep and a gene duplication. *Heredity*, *106*(5), 808-816. doi: 10.1038/hdy.2010.123

Kess, T., Galindo, J., & Boulding, E. G. (2018). Genomic divergence between spanish *Littorina saxatilis* ecotypes unravels limited admixture and extensive parallelism associated with population history. *Ecology and Evolution*, *8*(16), 8311-8327. doi: 10.1002/ece3.4304

Kirkpatrick, M., & Barton, N. (2006). Chromosome inversions, local adaptation and speciation. *Genetics*, *173*(1), 419-434. doi: 10.1534/genetics.105.047985

Koch, E. L., Morales, H. E., Larsson, J., Westram, A. M., Faria, R., Lemmon, A. R., ... Butlin, R. K. (2021). Genetic variation for adaptive traits is associated with polymorphic inversions in *Littorina saxatilis*. *Evolution Letters*, *5*(3), 196-213. doi: 10.1002/evl3.227

Le Moan, A., Gagnaire, P.-A., & Bonhomme, F. (2016). Parallel genetic divergence among coastal-marine ecotype pairs of european anchovy explained by differential introgression after secondary contact. *Molecular Ecology*, *25*(13), 3187-3202. doi: 10.1111/mec.13627

Le Moan, A., Panova, M., De Jode, A., Ortega-Martinez, O., Duvetorp, M., Faria, R., ... Johannesson, K. (2023). An allozyme polymorphism is associated with a large chromosomal inversion in the marine snail littorina fabalis. *Evolutionary Applications*, *16*(2), 279-292. doi: 10.1111/eva.13427

Lenormand, T., Roze, D., & Rousset, F. (2009). Stochasticity in evolution. *Trends in ecology & evolution*, *24*(3), 157-165. doi: 10.1016/j.tree.2008.09.014

Little, C., & Kitching, J. A. (1996). *The biology of rocky shores*. Oxford University Press, USA.

Liu, S., Ferchaud, A.-L., Grønkjær, P., Nygaard, R., & Hansen, M. M. (2018). Genomic parallelism and lack thereof in contrasting systems of three-spined sticklebacks. *Molecular ecology*, *27*(23), 4725-4743. doi: 10.1111/mec.14782

Lohse, K., Clarke, M., Ritchie, M. G., & Etges, W. J. (2015). Genome-wide tests for introgression between cactophilic drosophila implicate a role of inversions during speciation. *Evolution*, *69*(5), 1178-1190. doi: 10.1111/evo.12650

Louis, M., Galimberti, M., Archer, F., Berrow, S., Brownlow, A., Fallon, R., ... Gaggiotti, O. E. (2021). Selection on ancestral genetic variation fuels repeated ecotype formation in bottlenose dolphins. *Science Advances*, *7*(44), eabg1245. doi: 10.1126/sciadv.abg1245

Lowry, D. B. (2012). Ecotypes and the controversy over stages in the formation of new species. *Biological Journal of the Linnean Society*, *106*(2), 241-257. doi: 10.1111/j.1095-8312.2012.01867.x

Marshall, D. J., Taha, H., Brahim, A., & Abdelhady, A. A. (2021). Supratidal existence drives phenotypic divergence, but not speciation, in tropical rocky-shore snails. *Biological Journal of the Linnean Society*, *132*(1), 1-16. doi: 10.1093/biolinnean/blaa164

Morales, H. E., Faria, R., Johannesson, K., Larsson, T., Panova, M., Westram, A. M., & Butlin, R. K. (2019). Genomic architecture of parallel ecological divergence: beyond a single environmental contrast. *Science advances*, *5*(12), eaav9963. doi: 10.1126/sciadv.aav9963

Ortiz-Barrientos, D., Engelstädter, J., & Rieseberg, L. H. (2016). Recombination rate evolution and the origin of species. *Trends in ecology & evolution*, *31*(3), 226-236. doi: 10.1016/j.tree.2015.12.016

Panova, M., Blakeslee, A. M., Miller, A. W., Mäkinen, T., Ruiz, G. M., Johannesson, K., & André, C. (2011). Glacial history of the north atlantic marine snail, *Littorina saxatilis*, inferred

from distribution of mitochondrial dna lineages. *PLoS One*, *6*(3), e17511. doi: 10.1371/journal.pone.0017511

Perini, S., Rafajlović, M., Westram, A. M., Johannesson, K., & Butlin, R. K. (2020). Assortative mating, sexual selection, and their consequences for gene flow in littorina. *Evolution*, *74*(7), 1482-1497. doi: 10.1111/evo.14027

Price, T. (2008). *Speciation in birds*. Woodbury, NY: Roberts and Company.

Raffaelli, D., & Hawkins, S. J. (1996). *Intertidal ecology*. Springer Science & Business Media.

Ravinet, M., Westram, A., Johannesson, K., Butlin, R., André, C., & Panova, M. (2016). Shared and nonshared genomic divergence in parallel ecotypes of *Littorina saxatilis* at a local scale. *Molecular ecology*, *25*(1), 287-305. doi: 10.1111/mec.13332

Rougeux, C., Bernatchez, L., & Gagnaire, P.-A. (2017). Modeling the multiple facets of speciation-with-gene-flow toward inferring the divergence history of lake whitefish species pairs (coregonus clupeaformis). *Genome biology and evolution*, *9*(8), 2057-2074. doi: 10.1093/gbe/evx150

Ruesink, J., Ortiz, B. B., Mawson, C., & Boardman, F. (2022). Tradeoffs in life history investment of eelgrass zostera marina across estuarine intertidal conditions. *Marine Ecology Progress Series*, *686*, 61-70. doi: 10.3354/meps14000

Saltin, S. H., Schade, H., & Johannesson, K. (2013). Preference of males for large females causes a partial mating barrier between a large and a small ecotype of littorina fabalis (w. turton, 1825). *Journal of Molluscan Studies*, *79*(2), 128-132. doi: 10.1093/mollus/eyt003

Sanford, E., & Kelly, M. W. (2011). Local adaptation in marine invertebrates. *Annual review of marine science*, *3*, 509-535. doi: 10.1146/annurev-marine-120709-142756

Schluter, D. (2009). Evidence for ecological speciation and its alternative. *Science*, *323*(5915), 737-741. doi: 10.1126/science.1160006

Schluter, D., & Conte, G. L. (2009). Genetics and ecological speciation. *Proceedings of the National Academy of Sciences*, *106*, 9955-9962. doi: 10.1073/pnas.0901264106

Schluter, D., & Nagel, L. M. (1995). Parallel speciation by natural selection. *The American Naturalist*, *146*(2), 292-301. doi: 10.1086/285799

Seehausen, O., Terai, Y., Magalhaes, I. S., Carleton, K. L., Mrosso, H. D., Miyagi, R., ... Okada, N. (2008). Speciation through sensory drive in cichlid fish. *Nature*, *455*(7213), 620-626. doi: 10.1038/nature07285

Smith, C. C., & Flaxman, S. M. (2020). Leveraging whole genome sequencing data for demographic inference with approximate bayesian computation. *Molecular ecology resources*, *20*(1), 125-139. doi: 10.1111/1755-0998.13092

Soria-Carrasco, V., Gompert, Z., Comeault, A. A., Farkas, T. E., Parchman, T. L., Johnston, J. S., ... others (2014). Stick insect genomes reveal natural selection's role in parallel speciation. *Science*, *344*(6185), 738-742. doi: 10.1126/science.1252136

Stankowski, S., & Ravinet, M. (2021). Defining the speciation continuum. *Evolution*, *75*(6), 1256-1273. doi: 10.1111/evo.14215

Steiner, C. C., Römpler, H., Boettger, L. M., Schöneberg, T., & Hoekstra, H. E. (2009). The genetic basis of phenotypic convergence in beach mice: similar pigment patterns but different genes. *Molecular Biology and Evolution*, *26*(1), 35-45. doi: 10.1093/molbev/msn218

Tan, Y., Barnbrook, M., Wilson, Y., Molnar, A., Bukys, A., & Hudson, A. (2020). Shared mutations in a novel glutaredoxin repressor of multicellular trichome fate underlie parallel evolution of antirrhinum species. *Current Biology*, *30*(8), 1357-1366. doi: 10.1016/j.cub.2020.01.060

Taus, T., Futschik, A., & Schlötterer, C. (2017). Quantifying selection with pool-seq time series data. *Molecular biology and evolution*, *34*(11), 3023-3034. doi: 10.1093/molbev/msx225

Tomanek, L., & Helmuth, B. (2002). Physiological ecology of rocky intertidal organisms: a synergy of concepts. *Integrative and Comparative Biology*, *42*(4), 771-775. doi: 10.1093/icb/42.4.771

Van Belleghem, S. M., Vangestel, C., De Wolf, K., De Corte, Z., Möst, M., Rastas, P., ... Hendrickx, F. (2018). Evolution at two time frames: polymorphisms from an ancient singular divergence event fuel contemporary parallel evolution. *PLoS genetics*, *14*(11), e1007796. doi: 10.1371/journal.pgen.1007796

Via, S. (2009). Natural selection in action during speciation. *Proceedings of the National Academy of Sciences*, *106*, 9939-9946. doi: 10.1073/pnas.0901397106

Westram, A. M., Faria, R., Johannesson, K., & Butlin, R. K. (2021). Using replicate hybrid zones to understand the genomic basis of adaptive divergence. *Molecular ecology*, *30*(15), 3797-3814. doi: 10.1111/mec.15861

Westram, A. M., Morales, H. E., Johannesson, K., Butlin, R., & Faria, R. (2023). Understanding the adaptive role of chromosomal inversions across large geographical scales: The potential of pool-seq data. *bioRxiv*. doi: 10.1101/2023.08.20.553987

Westram, A. M., Panova, M., Galindo, J., & Butlin, R. K. (2016). Targeted resequencing reveals geographical patterns of differentiation for loci implicated in parallel evolution. *Molecular ecology*, *25*(13), 3169-3186. doi: 10.1111/mec.13640

Westram, A. M., Stankowski, S., Surendranadh, P., & Barton, N. (2022). What is reproductive isolation? *Journal of evolutionary biology*, *35*(9), 1143-1164. doi: 10.1111/jeb.14005

Williams, G. A. (1990). Periwinkles, *Littorina obtusata* (l.) and *L. mariae Sacchi et Rastelli*. *Field Studies*, *7*, 469-482.

Yeaman, S., Gerstein, A. C., Hodgins, K. A., & Whitlock, M. C. (2018). Quantifying how constraints limit the diversity of viable routes to adaptation. *PLoS genetics*, *14*(10), e1007717. doi: 10.1371/journal.pgen.1007717