UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE BIOLOGIA ANIMAL

Analysis of RAD sequencing data from species of Mediterranean cicadas

Inês Vieira Oliveira

Mestrado em Bioinformática e Biologia Computacional
Especialização em Biologia Computacional

Dissertação orientada por:
Profa Doutora Paula Simões

2019

"You can do irrefutably impossible things
with the right amount of planning
and support from intelligent and hardworking people
and pizza".

Scott M. Gimple

Agradecimentos

Este espaço é dedicado a todos os que ao longo deste tempo deram a sua contribuição para que esta dissertação fosse realizada. A todos eles deixo aqui o meu agradecimento sincero.

Os principais "prejudicados" com esta minha dissertação foram, sem dúvida, o Francisco Pina Martins e a Raquel Mendes. A eles particularmente, antes de mais ninguém, devo os meus profundos agradecimentos pelo modo como me aturaram, apoiaram e acompanharam ao longo desta árdua caminhada. Sendo assim, obrigada ao Francisco que se dispôs desde logo do início para me ensinar logo o "1+1=2" da linha de comandos e a ser a minha salvação na instalação de todos os programas que foram necessários e a resolver os problemas que iam aparecendo, mostrando-se sempre disposto sempre que eu, cheia de medo e receosa, mais uma vez o tinha que ir chatear. À Raquel, que para além de uma amiga que se tornou para mim, devo um enorme obrigada porque foi o meu braço direito e um motor para a concretização desta dissertação. Apoiou-me desde logo o início e durante toda a caminhada, dando-me todo o material para começar a estar a par do que se tratava o meu *data set* e como iniciaria o tratamento com o ipyrad; comentando os meus resultados que iam aparecendo e melhorias a fazer; resolver de erros e problemas que iam surgindo, assim como todas as palavras de encorajamento e de amizade que me deu quando as coisas não corriam bem e que me deram força para continuar e não desistir. Assim, devo a ela, sem dúvida, o concretizar desta dissertação.

Quero agradecer particularmente à minha orientadora, Paula Simões, pelo apoio e total disponibilidade para me receber no seu gabinete, sempre com simpatia e com conselhos fulcrais para o sucesso da realização desta dissertação. Já tinha tido a professora como docente de uma das cadeiras na licenciatura em Biologia, e como desde logo estabeleci ótima ligação com a mesma, ser orientadora desta tese foi sem dúvida uma razão que me a fez escolher. Sem os seus conselhos, as suas palavras amigas de encorajamento e orientação na análise dos resultados cruciais, nada disto teria sido possível.

Ao professor Octávio Paulo, que teve o poder de incutir em mim a paixão pelas *Next-Generation Sequencing Technologies* (NGS), muito obrigada. A forma brilhante como deu a cadeira Biologia Computacional e Genómica, fez-me querer escolher uma tese em que tivesse que trabalhar com dados gerados com estas tecnologias, porque de facto, foi a melhor escolha que fiz. Sendo assim, muito obrigada não só por me ter tornado apaixonada por este universo das NGS, mas também pelos conselhos e apoio ao longo de toda esta dissertação.

Quero agradecer também à Sara Silva, que me deu força quando, numa etapa final do período do ano letivo, tive que voltar a fazer tudo de novo. E cujo apoio na análise de dados com o Maverick foi crucial demonstrando-se sempre disponível para me ajudar e resolver todos os problemas que surgiram e interpretação dos resultados. A ela devo também um enorme obrigada mesmo estando lotada de trabalho pela sua disponibilidade para comigo que foi crucial pois sem ela também não seria possível concretizar este trabalho.

À Sofia Seabra, ao Gonçalo Costa e ao Eduardo Marabuto e aos restantes membros do grupo Cobig2 um imenso obrigada não só pelas opiniões que me deram para melhoria do tratamento dos meus dados que foram também cruciais durante a caminhada, e dando as suas interpretações do que os meus resultados iam "dizendo", mas também pela forma calorosa e a simpatia como me acolheram no grupo mostrando-se sempre disponíveis para tudo.

À Vera Nunes pela sua ajuda, numa fase mais inicial deste trabalho, com a contextualização do estado de arte do que já se tinham feito com as cigarras do género *Tettigettalna*.

Ao professor Vítor Sousa a quem agradeço pelo apoio que me deu já numa fase final de análise dos dados com o teste ABBA/BABA.

Quero agradecer também ao Diogo Silva, que mesmo nunca o tendo visto, teve também a sua contribuição essencial pela autoria de alguns *scripts* usados. Portanto, a ele devo também um imenso obrigada, porque sem a sua genialidade na "confeção" daqueles maravilhosos *scripts*, este trabalho também não teria sido possível.

Por último, mas não menos importante, quero agradecer de forma especial à minha família. À minha mãe Odília, ao meu pai José David, à minha irmã Diana e à minha avó Lúcia, que mesmo não percebendo patavina do andei para aqui a fazer e para que serve o meu curso, deram-se sempre total apoio e força para não baixar os braços mesmo quando estava prestes a desistir ou mais desanimada. Sem eles também a concretização desta dissertação não teria sido possível. Obrigada a vocês, de coração.

Inês Oliveira

Aos meus pais, irmã e avó.

Resumo

Compreender a divergência e especiação entre espécies próximas sempre foi um tema desafiador no âmbito da biologia evolutiva. Os marcadores de DNA citoplasmáticos, os quais muitas vezes são usados em investigações no contexto de marcadores moleculares, nem sempre deram resultados bem-sucedidos que conseguissem resolver as respetivas filogenias e outras questões.

Nos últimos anos, com o surgimento da Nova Geração de Tecnologias de Sequenciação e técnicas associadas que tiram partido de uma reduzida representação do genoma, é agora possível responder a questões relacionadas com a divergência populações e especiação.

Aqui retratamos o potencial de uma dessas técnicas – Restriction-site Associated DNA (RAD) Sequencing -, para contribuir para a resolução de algumas questões no âmbito da especiação de um grupo particular de insetos, as cigarras mediterrânicas do género *Tettigettalna*.

A técnica RAD sequencing tira partido da Illumina, uma das Tecnologias da Nova Geração de Sequenciação, para gerar dados genómicos de zonas adjacentes a locais de corte de restrição por enzimas (RAD tags). Isto permite simultaneamente identificar e marcar milhares de SNPs espalhados por todo o genoma, de qualquer tamanho, em centenas de indivíduos e para organismos modelo ou não. Como a RAD-Seq é uma técnica de sequenciação de reduzida representação do genoma, é claro que o seu uso tem muitas mais vantagens em comparação com técnicas de sequenciação de todo o genoma. Isto permitiu que a RAD-Seq se tenha tornado a metodologia genómica mais usada para a descoberta de SNPs em estudos filogenéticos e de evolução de organismos não-modelo como é o caso das espécies de cicadas do género *Tettigettalna*.

Este género constitui um complexo de espécies de cigarras intimamente relacionadas que divergiram recentemente. Elas são morfologicamente semelhantes o que as torna um desafiante grupo taxonómico. Adicionalmente, o canto de chamamento produzido pelos machos é a principal característica que permite a distinção entre as espécies.

Na Península Ibérica, a diversidade das cigarras foi amplamente subestimada até à recente descrição e revisão taxonómica de nove espécies de cicadas de pequeno porte pertencentes ao género *Tettigettalna*: *Tettigettalna mariae*, *Tettigettalna argentata*, *Tettigettalna aneabi*, *Tettigettalna josei*, *Tettigettalna defauti*, *Tettigettalna armandi*, *Tettigettalna helianthemi*, *Tettigettalna boulardi* e *Tettigettalna estrellae*.
Algumas das espécies mencionadas são restritas a Espanha, sendo que apenas uma delas, *Tettigettalna estrellae*, é restrita a Portugal. *Tettigettalna argentata* é a única que para além da Península Ibérica se estende para mais países Europeus.

Alguns estudos focados nas espécies da zona do Mediterrâneo pertencentes a este género evidenciaram a ocorrência de simpatria entre algumas espécies de *Tettigettalna* do sudoeste da Península Ibérica. As populações de *Tettigettalna argentata* têm uma distribuição que faz com que por vezes se sobreponham com outras populações de outras espécies. No Algarve (Portugal), as populações de *Tettigettalna mariae* e *Tettigettalna argentata* podem ser encontradas em simpatria ou parapatria. Estas duas espécies são consideradas um complexo de espécies gémeas, sendo morfologicamente muito semelhantes e apenas se distinguindo pelo seu canto de chamamento.

Trabalhos baseados na análise de sequências mitocondriais (COI) permitiram a separação de populações de *Tettigettalna argentata* em clade do norte e clade do Sul. Adicionalmente, este clade do Sul revelou não ser geneticamente distinto dos espécimenes de *Tettigettalna mariae*, com o qual partilha a maior parte dos haplótipos. Assim, é muitas vezes impossível discriminar os espécimenes de *T. mariae* dos espécimenes de *T. argentata* (clade do Sul) com base apenas na análise de sequências COI.

Como referido, as espécies de *Tettigettalna* podem ser distinguidas através dos sons produzidos pelos machos, pelo que se pensa que estes sinais acústicos possam ter um papel preponderante no isolamento reprodutivo das espécies. Na verdade, estudos baseados em dados de acústica revelam que diferentes espécies têm diferentes padrões acústicos. Porém, outros trabalhos com dados genéticos não esclarecem várias questões. Nomeadamente, se a partilha de haplótipos entre o clade Sul de *Tettigettalna argentata* e as *Tettigettalna mariae* será devida a introgressão (existência de fluxo genético entre populações) ou "Incomplete Lineage Sorting", (segregação imperfeita de alelos em linhagens bem definidas).

Os trabalhos realizados apontam assim para a necessidade de uma metodologia multilocus que possa ser uma melhor abordagem a adotar, por forma a responder às questoes acima mencionadas.

Neste trabalho, utilizámos então uma abordagem multilocus, ou seja, dados de RAD-Seq das cigarras do género *Tettigettalna*. Com este tipo de dados e utilizando ferramentas de limpeza e filtragem dos dados, como o Ipyrad, VCFtools e outros scripts, foi assim possível gerar resultados que permitiram responder melhor a questões que até agora não tinham sido respondidas à luz de abordagens single locus e/ou com dados de outras naturezas.

Com esta nova abordagem mostrámos que os dados RAD-Seq tornam evidentes os padrões de distribuição geográficos das espécies/populações das cigarras do género *Tettigettalna*, bem como parecem indicar que a partilha de haplótipos entre *Tettigettalna argentata* e *Tettigettalna mariae* de populações simpátricas na região Algarvia, é explicada pelo fenómeno de introgressão.

Palavras-chave: RAD-Sequencing, *Tettigettalna*, Península Ibérica, introgressão e incomplete lineage sorting.

Abstract

Understanding population divergence and speciation among closely related species has long been a challenge in evolutionary biology. Cytoplasmic DNA markers, which have been widely used in the context of molecular barcoding, have not always proved successful in resolving phylogenies and other related questions.

With the advent of Next-Generation Sequencing technologies and associated techniques of reduced genome representation, not only the phylogenies of closely related species are now being resolved at a much greater detail, but are also allpwing a much better understanding on divergence and speciation patterns and processes.

Here we examine the potential of one of such techniques - Restriction-site Associated DNA (RAD) sequencing -, in disentangling questions related to the divergence and speciation of a particular group of insects, the meditteranean cicadas from the *Tettigettalna* genus. This genus constitutes a complex of closely related and recently diverged species. They are morphologically similar what makes them a taxonomical challenging group. The calling songs are the main character used for their identification.

Work focused on the Mediterranean species of this genus revealed the accurance of sympatric populations among some of the southern Iberian *Tettigettalna* species. In fact, *Tettigettalna mariae* and *Tettigettalna argentata* populations can be found in sympatry or close parapatry. As already referred, these two species are morphologically very similar and only distinguishable by their calling songs. However, mitochondrial COI studies also showed that these species share haplotypes but the results couldn't reveal if this sharing was due to introgression (existence of gene flow between populations) or incomplete lineage sorting (defective segregation of alleles into well-defined lineages).

The present multilocus approach with RAD-Seq data, not only revealed a better understanding of the geographical patterns of distribution of the *Tettigettalna* species and populations, but also gave evidence that it is the phenomenom of introgression that explains the sharing of haplotypes between *Tettigettalna argentata* and *Tettigettalna mariae*, when in sympatry.

Therefore, the use of the Next-Generation sequencing data, in particular RAD-seq data, in this thesis has reinforced the utility of the methodology applied to solve problems related to recent diverged complexes of species, such our study group of insects in which we were able to give a significant contribution to a better understanding of its divergence and speciation.

Keywords: RAD-Sequencing, *Tettigettalna*, Iberian Peninsula, introgression and incomplete lineage sorting.

Table of Contents

List of Figures

List of Tables

List of Files

Chapter 1

Introduction

1.1 Sequencing Technologies: from the early DNA discovery to present day research tools

Determining the sequence of nucleic acids residues in biological samples is an integral component of a large variety of research applications since it gives the information for the hereditary and biochemical properties of terrestrial life. DNA sequencing plays such an important role as it allows measuring one of the major properties at which the life forms can be defined and differentiated from each other (Heather & Chain, 2016).

Over the last half century, a large number of researchers have invested a great deal of time and resources to the development and improvement of the sequencing technologies. Therefore, we have witnessed tremendous challenges over those years, moving from sequencing short sequences (gene scale) to millions of bases (whole genome scale) (Heather & Chain, 2016).

Hence, the evolution of DNA sequencing has a rich story full of several generations of sequencing technology that can be characterized in terms of their nature and the output generated by them (McGinn & Gut, 2013).

The history behind Next-Generation Sequencing (NGS), also known as high-throughput sequencing (HTS) techniques, goes back to the discovery of the double-helix DNA structure in 1953 (Watson & Crick, 1953). Some years later, Robert Holley and colleagues were able to produce the first whole nucleic acid sequence and structure, namely the 77 ribonucleotides of alanine tRNA from *Saccharomyces cerevisiae*, opening the door for others to determine the sequence of not only other RNAs but also DNA (Holley, Madison, & Zamir, 1964; Holley et al., 1965).

In 1977, Frederick Sanger was the first to sequence a complete DNA genome of bacteriophage ΦX 174 and was the pioneer of the Sanger method, the known first-generation technique of sequencing (Sanger et al., 1977a; Sanger, Nicklen, & Coulson, 1977b). In fact, he was the first to realise the importance of genome sequencing on the biological studies, like he said:

"…[A] knowledge of sequences could contribute much to our understanding of living matter." [Frederick Sanger] (Heather & Chain, 2016)

Next-Generation Sequencing refers to the deep, high-throughput DNA sequencing technologies developed after the Sanger sequencing method first emerged in 1977 (Sanger et al., 1977a).

Also referred as the Sanger DNA chain-termination method or dideoxy method, it has remained the most frequently used DNA sequencing technology for 30 years (McGinn & Gut, 2013).

1.1.1 The Sanger Sequencing: the first sequencing method

The chain-termination method (see Figure ***1.1***) is based on the DNA polymerase-dependent synthesis of a complementary DNA strand in the presence of natural deoxynucleotide triphosphates (dNTPs) and dideoxynucleotide triphosphates (ddNTPs) (Sanger et al., 1977a).

The ddNTPs are modified nucleotides that lack the 3'- OH group that is required for the establishment of the phosphodiester bonds between nucleotides (Chidgeavadze et al., 1984) during strand elongation. Hence, the lack of it makes them, during the elongation process, being responsible for the DNA synthesis termination, once one is incorporated by the DNA polymerase (Morozova & Marra, 2008).

By performing four parallel reactions, the products are then separated by size using a polyacrylamide gel electrophoresis and the DNA sequence of the template strand is revealed with the use/principle of autoradiography (Heather & Chain, 2016; Morozova & Marra, 2008).

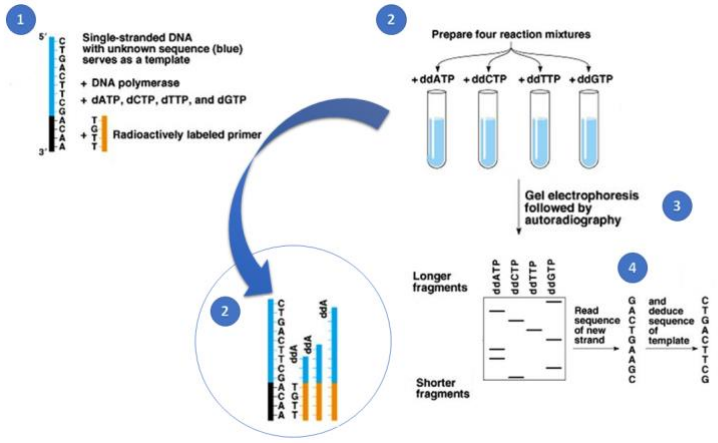


Figure 1.1 Sanger Sequencing
(1) Single-stranded DNA with an unknown sequence (blue) serves as a template. (2) Four reaction mixtures are prepared, one for each nucleotide, in the presence of a DNA polymerase, a radioactively labelled primer, dNTPs ("normal" nucleotides) and ddNTPs (modified nucleotides), which allows the DNA synthesis in vitro of the complementary strand. (3) After the DNA synthesis, a gel electrophoresis is performed followed by autoradiography. (4) The new strand sequence is read from the gel electrophoresis, and the sequence of the template is deduced from the previous one. Adapted from https://www.onlinebiologynotes.com/sangers-method-gene-sequencing/ in March 2019.

A number of improvements throughout the years were made to the classical Sanger sequencing technique. The most relevant improvements made to the classical Sanger sequencing method were incrementing the read length of the sequences, lower error rates of sequencing (Heather & Chain, 2016;

Morozova & Marra, 2008) and improving detection with the use of capillary gel electrophoresis and the use of fluorescent dyes of different colours (Smith et al., 1986). Hence, those improvements made the accuracy, the simplicity and robustness of the classical Sanger's method made it the pioneer technology for DNA sequencing (Heather & Chain, 2016).

1.1.2 Human Genome Project

The Sanger sequencing method was used to complete Human Genome Sequencing initiatives led by the International Human Genome Sequencing Consortium and Celera Genomics (Consortium, 2004; Venter et al., 2001).

The Human Genome Project (HGP) was initiated in 1990 and required 13 years until the sequence was published. (Consortium, 2001). This project was the proof that sequencing an entire genome is achievable but at in high cost level and with limitations in the throughput (McGinn & Gut, 2013).

Sanger's method limitations and the associated Human Genome Project have aroused the need of better sequencing technologies not only for sequencing human genomes but also other genomes (Barba, Czosnek, & Hadidi, 2013). Following the publication in 2004 of the human genome sequence (Consortium, 2004), the National Human Genome Research Institute have invested 70 million in a DNA sequencing initiative with the goal of having a human genome sequence in 10 years possible at a most reducible cost (Reuter, Spacek, & Snyder, 2015). This triggered in 2005 the rise of the beginning of the revolutionary Next-Generation Sequencing technologies (McGinn & Gut, 2013).

1.1.3 Next-Generation Sequencing technologies

The Next-Generation Sequencing technologies refer to all the technologies that followed the first-generation of sequencing – the Sanger Sequencing technique.

Hence, both 2nd generation of sequencing and 3rd generation of sequencing technologies refers to technologies that belong to the next-generation technologies. In the 2nd generation of sequencing, the main technologies are 454 pyrosequencing technology, Illumina sequencing technology, SOLiD sequencing technology, Ion Torrent sequencing technology and Single Molecule Real Time (SMRT) sequencing technology. For the 3rd generation of sequencing, the most notable technology of this generation is the Oxford Nanopore sequencing technology.

Here, it will be described not only an overall review of the chemistry of the sequencing method for each technology, the improvements that were made to the methods, but also their advantages/disadvantages.

The aim of these sequencing methods of the Next-Generation of Sequencing technologies was producing big volume of data (large amounts of DNA reads) and deliver fast and accurate genome information at a low cost (Barba et al., 2013; McGinn & Gut, 2013) which would have a dramatic impact on genomic research. Each technology associated with the platforms available determines the quality, quantity and

biases of the output generated which is essential to be able to know which platform to use depending with the type of output data pretended (Reuter et al., 2015). Since, the platform applications will be later discussed, after understanding, first of all, the sequencing principles of each technology.

2nd Generation Sequencing

There are several High-Throughput Sequencing platforms available in the market that used second generation sequencing methods (McGinn & Gut, 2013). This generation methods' workflow are similar: library preparation, amplification and several rounds of enormously parallel sequencing (Reuter et al., 2015).
However, the main characteristics of this generation of sequencing is the use of many clonal templates in parallel and the use of enzymatic replication system to determine the sequence (McGinn & Gut, 2013).

Roche/454 pyrosequencing technology

The previous presented Sanger sequencing method had the limitation of requiring *in vivo* amplification of the DNA fragments to be sequenced, which were performed in bacterial hosts. However, this cloning process is labour intensive, lengthy and have error biases associated (Hall, 2007).

In the 454-pyrosequencing technology, the first of the next-generation sequencing technologies released at the market, there is no need of cloning amplification in a host like on Sanger's sequencing (Tawfik & Griffiths, 1998). Furthermore, the nucleotide inference is in real-time and there is no use of modified dNTPs (Nyrén, 1987; Ronaghi et al., 1996; Ronaghi, Uhlén, & Nyrén, 1998).
Despite the differences, both Sanger and pyrosequencing methods are sequence-by-synthesis techniques because both of them require action of DNA polymerase (Heather & Chain, 2016). Furthermore, pyrosequencing provides intermediate read lengths and price per base compared to Sanger sequencing in one hand, and Illumina and SOLid platforms on the other hand (Barba et al., 2013).

In pyrosequencing sequencing (see

Figure ***1.2***) technologies, after the library preparation, the DNA amplification is done *in vitro* in a method called Emulsion PCR (EmPCR) (Tawfik & Griffiths, 1998). In EmPCR, a huge number of copies of a unique template DNA per bead is obtained (Morozova & Marra, 2008). The clonal templates are then sequenced using the pyrosequencing method, at which dNTPs are added one at time. Whenever a nucleotide is incorporated, the release of pyrophosphate (PPi) occurs and it's detectable by light produced by a chemiluminescent enzyme present in the reaction. The sequence of the DNA template is then taken (Morozova & Marra, 2008).

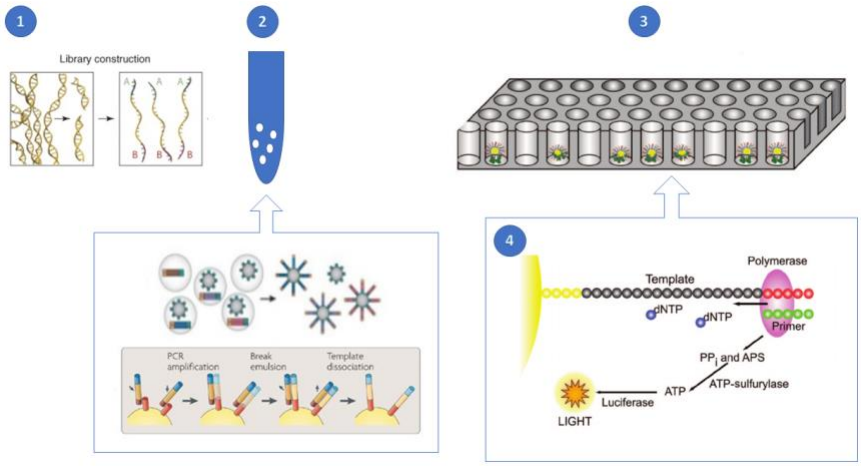


Figure 1.2 454 pyrosequencing
(1) Preparation of Adapter ligated single-stranded DNA Library (A-[insert]-B). (2) *In vitro* Emulsion PCR (EmPCR) in beads in water-in-oil microreactors that allows DNA amplification. Each bead contains clonal sequences of a unique template. (3) Beads with clonally amplified template DNAs are deposited in a picotiter plate with sequencing enzymes. (4) The pyrosequencing starts with the sequential addition of dNTPs. The release of PPi detectable by light, once a dNTP is incorporated, allows the determination of the template sequence. Adapted from (Leong, Skinner, & Love, 2014; Margulies et al., 2005; Metzker, 2009; Voelkerding, Dames, & Durtschi, 2009).

The first high-throughput sequencing device to be available in the market was from 454 Corporation, was in 2005-2006. It was called the GS 20 and was capable of producing 20 Mbp. The 454 Corporation was later bought by Roche and the technology evolved in 2007 to the Roche GS FLX, which offered greater number of reads because of the increment of the number of wells in the picotiter plate as well as better quality data. In 2008 the upgraded 454 GS-FLX+ Titanium was available which was capable of producing over 600 Mbp of sequence data in a single run (Barba et al., 2013).

The Roche technology have then demonstrated that mass parallelisation of reactions to sequence data is possible in only one run (Margulies et al., 2005).

Illumina/Sequencing by synthesis with reversible terminators technology

Following the success of 454 pyrosequencing, a several number of parallel sequencing techniques sprung up (Voelkerding et al., 2009). The most important among them was the Solexa method of sequencing that released the Genome Analyzer in 2005, and which was later, in 2007, acquired by Illumina (Barba et al., 2013).
In Illumina sequencing method (see Figure ***1.3***), instead of parallelising by performing bead-based emulsion PCR, adapter linked DNA molecules are passed over a lawn of complementary oligonucleotides bound to a flow cell. Following a solid phase PCR, it produces neighbouring clusters of clonal populations from each of the individual original flow cell binding DNA strands (Bentley et al., 2008; Fedurco et al., 2006), a process called bridge amplification. This name is due to the replicating DNA strands have to arch over to prime the next round of polymerisation of neighbouring surface-bound oligonucleotides (Voelkerding et al., 2009).

The sequencing is achieved using a sequencing-by-synthesis (SBS) approach using fluorescent reversible-terminator dNTPs, which can not immediately bind further nucleotides as the fluorophore occupies the 3' hydroxyl position; this must be cleaved away before polymerisation can continue, which

allows the sequencing to occur in a synchronous way (Turcatti et al., 2008). The modified dNTPs and the DNA polymerase are washed over the primed, single stranded flow-cell bound clusters in cycles. At each cycle, the identity of the incorporating nucleotide can be monitored with a charge-coupled device (CCD) camera by exciting the fluorophores with appropriate lasers, before enzymatic removal of the blocking fluorescent moieties and continuation to the next position (Heather & Chain, 2016).

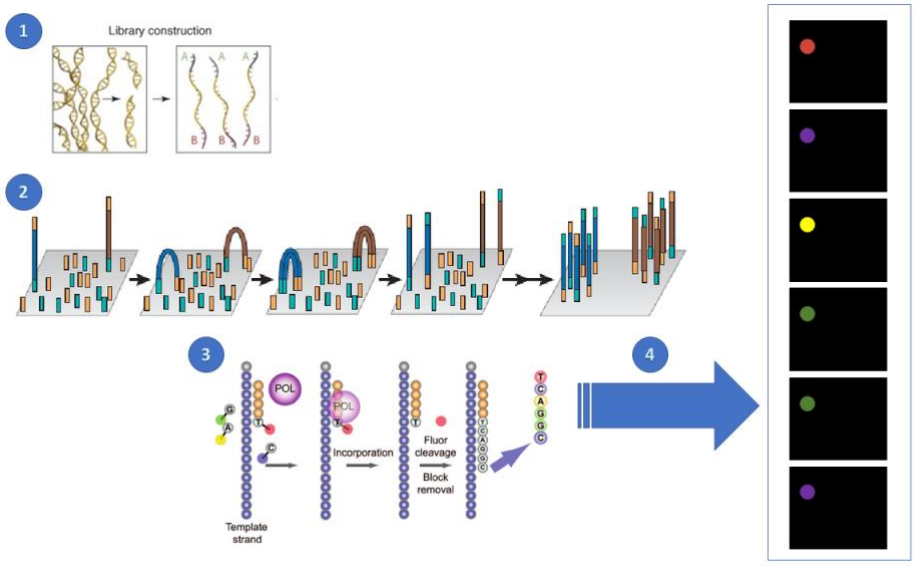


Figure 1.3 Illumina sequencing
(1) DNA fragmentation and *in vitro* adaptor ligation. (2) Solid-phase bridge amplification. (3) Sequencing obtained with the use of fluorescent reversible-terminators nucleotides (labelled nucleotides), primers and DNA polymerase. (4) After laser excitation, an image per cycle is obtained by a CCD camera allowing the identification of the base incorporated by the emitted fluorescence. The sequence cycles are repeated to determine the sequence of the templates, one base at time. Adapted from (Leong et al., 2014; Margulies et al., 2005; Voelkerding et al., 2009).

The first Genome Analyzer machines were initially only capable of producing very short reads (up to 35 bp long) but they had an advantage in that they could produce paired-end data, in which the sequence at both ends of each DNA cluster is recorded. Having paired-end data, it provides a greater amount of information, which improves the accuracy when mapping reads to reference genomes, especially across repetitive sequences and aids detection of spliced exons and rearranged DNA or fused genes (Heather & Chain, 2016).

The standard Genome Analyzer was latter followed by the HiSeq, a machine capable of even greater read length and depth, and then the MiSeq, which was a lower-throughput but lower cost machine with faster turnaround and longer read lengths (Balasubramanian, 2011; Quail et al., 2012). HiSeq 2500 series platform has a particular capacity that allow sequencing a human genome in 24 hours, the namely "Genome in a day" (Barba et al., 2013).

Across all Illumina models the overall error rates are bellow 1% and the most common type of error is substitution (Dohm et al., 2008), however they currently dominated the High-Throughput Sequencing market (Reuter et al., 2015).

ABI/SOLiD sequencing by ligation technology

The SOLiD technology was released by Applied Biosystems, which became Life Technologies following a merger with Invitrogen (McKernan et al., 2009) .
In 2007, the first SOLiD sequencing system was released, followed by the SOLiD 5500w and 5500 xlw sequencing systems in 2010. The last has read lengths of 85 bps with 99,99% of accuracy and a data output of 30 Gb per run. Applications of SOLid include analysis of whole genome clusters with runs being finished in a week (Barba et al., 2013). The results of sequencing in terms of quantities and length are comparable to Illumina sequencing described before (Barba et al., 2013).

In the Sequencing by Oligonucleotide Ligation and detection (SOLiD) system, as it name suggests the sequencing process does not occurs by synthesis (with a polymerase) but by ligation, using a DNA ligase (Shendure et al., 2005).
Libraries preparation begin with an emulsion PCR, the amplification step similar to that one used in the 454 technology. The PCR products are then sequenced in sequential rounds of hybridization and ligation of 16 dinucleotide combinations labelled by four fluorescent dyes (each dye is used to label 4 dinucleotides) on a glass surface. Using the four-dye encoding scheme, each position is probed twice and the identification of the nucleotide is determined by analysing the colour that results from two successive ligation reactions (Morozova & Marra, 2008).

Due to this two-base encoding system, an inherent accuracy check is built into the technology which allows 99,99% of accuracy. The chemistry of the system also means that it is not hindered by homopolymers unlike the Roche 454 FLX system, then large and difficult homopolymer repeats are no longer a problem to sequence. The technology is then used to sequence DNA but the particular feature of high parallelisation nature makes it also applicable in transcriptomics and epigenomics (Morozova & Marra, 2008).

Life Technologies/Ion Torrent sequencing technology

Ion Torrent is the first so-called “post-light” sequencing method as it uses neither fluoresce nor luminescence, which speeds sequencing and reduces costs (Rothberg et al., 2011).

The template preparation and sequencing are similar to the 454-pyrosequencing method (Mellmann et al., 2011). An emulsion PCR is used to clonally amplify adaptor-ligated DNA fragments on beads. The beads are distributed on wells where sequencing-by-synthesis occurs. The difference is that, instead of base incorporation with production of light, the Ion torrent sequencing measures pH changes induced by the release of H^+ during DNA extension. Those pH changes, with the sequential addition of individual nucleotides during each sequencing cycle, are detected by a sensor and converted into a voltage signal. Since it is proportional to the number of bases incorporated it allows fast base discrimination (Reuter et al., 2015; Rothberg et al., 2011).

In 2010, Life Technologies released the Ion Torrent sequencing technology in the form of a benchtop Ion Personal Genome Machine (PGM) sequencer. A second machine was released in 2012, the Ion Proton with its increments of the output generated (10 Gb) over the PGM (1 Gb). However, Ion Proton

sequencer features a maximum of 200 bp read lengths as opposed to 400 bp for the PGM (Reuter et al., 2015).

In what refers to their applications, Proton is more useful for exome sequencing and whole-transcriptome analysis, and PGM for targeted resequencing projects and small genome analysis (Reuter et al., 2015). The speed of sequencing, two to eight hours depending of the machine and chip used, make these sequencers particularly useful for clinical applications (Mellmann et al., 2011). Error rates, instead, emerge by the presence of homopolymer repeats longer than six bp (Rothberg et al., 2011), insertions and deletions (Liu et al., 2012) on the data to be sequenced.

PacBio/SMRT sequencing technology

The Single Molecule Real Time (SMRT) sequencing technology commercialized by Pacific Biosciences is sometimes referred as a 3rd generation technology because it allows sequencing single molecules in real-time with no need of previous amplification (Reuter et al., 2015), however it satisfies the 2nd generation characteristic of using an enzymatic replication system to sequence the data (Schadt, Turner, & Kasarskis, 2010). Because of this duality, it was stablished that this sequencing technology is an intermediate state of the 2nd and 3rd generations, because in fact, it goes beyond the 2nd generation characteristics (McGinn & Gut, 2013).

In PacBio technology, the SMRT sequencing is a parallelized sequencing method that utilizes SMRT cells with a lot of zero-mode waveguides. Each zero mode waveguides (ZMW) have a single DNA polymerase that is affixed at its bottom with a single molecule of DNA as the template (Levene et al., 2003). Throughout this complex, light can penetrate and create a visualization chamber that allows monitoring the activity in real-time of the polymerase enzyme at a single molecule level (Eid et al., 2008). Each of the four bases are labelled with fluorescent dyes and added simultaneously, and whenever a nucleotide is incorporated into the growing strand, the fluorescent tag is cleaved off providing detectable fluorescence signals by a sensor which results in DNA sequencing in real-time. The speed of sequencing is much faster compared to the technologies where individual nucleotides are flushed sequentially (Kulski, 2016; McGinn & Gut, 2013).

The commercially available device from Pacific Biosciences, the PacBio RS II, was released in 2010, which evolved through the years with improvements in the number of zero mode waveguides with a performance of an average of more than 14 kb of read lengths but it can be until 60 kb, generating then 1 Gb of data in 4 hours (McGinn & Gut, 2013). There are errors also associated with the method and the fact of avoiding the amplification step turns it is less sensitive to GC content, comparing to other platforms (Loomis et al., 2013).
However, SMRT sequencing characteristics makes it particularly useful for producing kinetic data and applied in projects involving *de novo* assemblies (English et al., 2012).

3rd Generation Sequencing

The third-generation sequencing technologies have emerged in a continuously effort to reduce the price of sequencing and became preparatory procedures and sequencing methods even more simpler (Metzker, 2009; Schadt et al., 2010).

There is a discussion about what is the "line" that separates second-generation sequencing methods and the third-generation sequencing methods. However, there are arguments that allowed a division between them, namely the fact that the third-generation is characterized by the possibility of sequencing single molecules (single molecule sequencing – SMS), sequencing in real-time and the most remarkable characteristic, the absence of a replication enzymatic system which means there is no requirement for DNA amplification in those methods (Gut, 2013; Niedringhaus et al., 2011; Schadt et al., 2010).

Oxford Nanopore Technology and its sequencing method

Nanopore sequencing method is a typical example of a third-generation sequencing method that was led by the Oxford Nanopore Technologies, the first company that released in the market the first commercially nanopore based platforms like GridION and MinION. The last one is an USB portable sequencer device that was available at the market in 2014 (Clarke et al., 2009; Eisenstein, 2012; Loman & Quinlan, 2014).

The principle of nanopore sequencing is simple and essentially consists of a synthetic or biological bilayer membrane immersed in salt solution and perforated by biologic nanopores (Bayley, 2006; Reuter et al., 2015).

The library preparation of the biomolecules to be sequenced is reduced only with DNA fragmentation and with ligation by enzyme proteins of adapters being then unwinded and guided to the nanopore with the help of a protein on the top of the nanopore. This library preparation doesn't require PCR amplification, and it's designed to allow sequencing of both strands of the DNA which improves accuracy (Ashton et al., 2014; Quick, Quinlan, & Loman, 2014; Reuter et al., 2015).
An application of an electrical current to this system drives ions through the nanopore, and the arrival of a biomolecule such as DNA base create resistance in the flow of the ions which causes changes in the electrical current that can be measured. The distinct four DNA bases when passing through the nanopore promote different changes in the electrical current and, because of that, DNA sequencing is accomplished. (Bayley, 2006)

Although this method has error rates quite high (Jain et al., 2015), it makes possible, comparing to the previous next-sequencing methods, sequencing at faster rates and at low costs of long DNA reads bigger than two kilobases (kb), even RNAs and allows also parallelization (Branton et al., 2008; Derrington et al., 2010).

The MinIon device which technology of sequencing is the nanopore sequencing (deals with long reads length) was, in one of its first applications, used in combination with the short-read sequencing methods (high read depth and accuracy) to generate position and structure of a bacterial genome sequence. This

proved that combination of sequencing methods is plausible and very effective to get the best characteristics of both methods used (Loman, Quick, & Simpson, 2015; Quick et al., 2014).

1.2 Platform selection and RAD-Seq applications

There are important factors to be considered before choosing an adequate sequencing platform. The choice depends on the size or expected size of the genome being studied, its complexity like GC content and the depth of coverage and accuracy needed (Barba et al., 2013).

For *de novo* genome sequencing which is our case of the data to be treated and analysed, longer read length may be appropriate; for fast turnover times and limited throughput, smaller laboratory bench top platforms may offer greater flexibility (Loman et al., 2012); for amplicon sequencing, Roche 454 platform is suitable because of its longer reads, however its currently expensive. Recently, platforms like Illumina MiSeq and Ion PGM platforms are suitable for sequencing amplicons; for RNA-Sequencing and projects that require high depths of coverage, Illumina and SOLiD platforms offer the best cost, accuracy and throughput (Radford et al., 2012).
Summarizing, the Roche 454 has the longest read length, Illumina HiSeq 2500 features the biggest output and lowest sequencing cost and SOLiD 5500 xlw the highest accuracy (Liu et al., 2012).

Biologists have always dreamed of a day when perfect genetic knowledge would be available for almost any organism (Etter et al., 2011a).

The current Next-Generation Sequencing (NGS) technologies are fulfilling that promise and revolutionizing the fields of evolutionary biology (Rokas & Abbot, 2009) and biomedical sciences (Asmann, Wallace, & Thompson, 2008; Marguerat, Wilhelm, & Bähler, 2008; Mortazavi et al., 2008), opening the possibility for genetic analysis at scales not previously possible. Furthermore, researches related to population genomics (Hohenlohe et al., 2010), quantitative trait mapping (Baird et al., 2008), comparative genomics and phylogeography (Emerson et al., 2010; Gompert et al., 2010) that were unthinkable even a few years ago, are now possible. Perhaps the most critical aspect of these breakthroughs is the unshackling of genetic analysis from traditional model organisms, allowing genomic studies to be performed in organisms for which few genomic resources presently exist (Mardis, 2008; Van Tassell et al., 2008).

1.2.1 Restriction-site Associated DNA (RAD) sequencing

We present here one sequencing technique approach, the Restriction-site Associated DNA (RAD) sequencing, a focused reduced-representation methodology that makes use of the Next-Generation Sequencing technologies and that speed up this revolution in evolutionary biology.

1.2.1.1 Methodology of RAD-Seq

RAD sequencing uses Illumina next-generation sequencing method to generate sequence data adjacent to restriction cut sites (RAD tags), which allows simultaneously discovering and scoring of tens to

hundreds of thousands of single-nucleotide polymorphism (SNP) markers (RAD markers) spread throughout the genome (of any size) in hundreds of individuals and for any model or non-model organism of choice (Baird et al., 2008; Davey et al., 2011; Luikart et al., 2008).
In the original RAD-seq method (Baird et al., 2008), genomic DNA is digested with one restriction enzyme and numerous modifications can be made to the protocol (similar methods) to suit a diversity of evolutionary genetic questions, like inferring marker *loci* and genotypes *de novo* from RAD tags sequences for an organism with no sequenced genome, distinguish SNPs from error and inferring heterozygoty in the face of sampling variance (Etter et al., 2011a; Marguerat et al., 2008).

Although the choice of the appropriate method is important because it can severally influence all steps of a genomic study (Andrews et al., 2016), in general, all the RAD-Seq techniques share basic steps. All of them need a high-molecular-weight genomic DNA, which is digested with one or more restriction enzymes depending of the method chosen. As seen on Figure **1.4**, an adaptor (P1) is ligated to the fragment's overhanging ends. This adapter contains forward amplification and Illumina sequencing primer sites, as well as a nucleotide barcode 4 or 5 base pair (bp) long for sample identification (Baird et al., 2008). These barcodes are used to identify individual samples that are sequenced together (multiplexed) in a single library (Andrews et al., 2016), and differ by at least two nucleotides in order to reduce erroneous sample assignment (Baird et al., 2008). As soon as barcoded adaptors are ligated to each sample, the samples can be multiplexed, which can greatly reduce the expense and time of the subsequent steps in studies with large numbers of samples (Andrews et al., 2016). The adapter-ligated fragments are then pooled, randomly sheared and size-selected. A second adapter (P2) with divergent ends is then ligated. The reverse amplification primer is unable to P2 unless the complementary sequence is filled in during the first round of forward elongation originated from the P1 amplification primer. The structure of this adapter ensures that only P1 adapter-ligated RAD tags are amplified during the final PCR amplification step (Baird et al., 2008). In the original RAD-Seq protocol each RAD tag has one end defined by the restriction enzyme recognition site and the other end defined by random shearing.

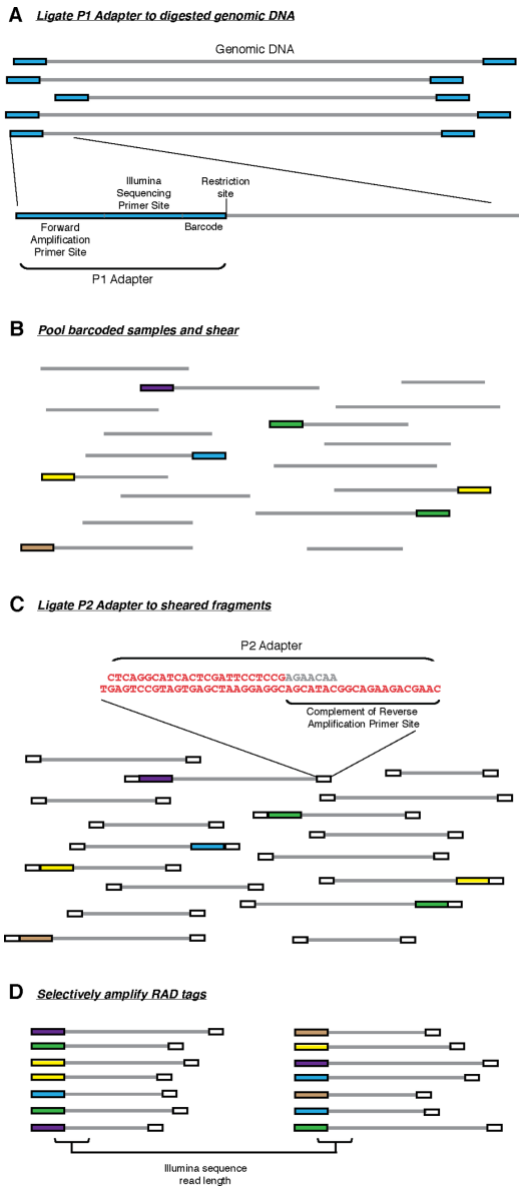


Figure 1.4 RAD marker library generation

(A) Genomic DNA was digested with a restriction enzyme and the P1 adapter was ligated to the fragments. The P1 adapter contains a forward amplification primer site, an Illumina sequencing primer site, and a barcode (coloured boxes represent P1 adapters with different barcodes). (B) Adapter-ligated fragments were combined (if multiplexing), sheared and (C) ligated to a second adapter (P2, white boxes). The P2 adapter is a divergent "Y" adapter, containing the reverse complement of the reverse amplification primer site preventing amplification of genomic fragments lacking a P1 adapter. (D) RAD tags, which have a P1 adapter, will be selectively and robustly enriched. Reproduced from (Baird et al., 2008).

Thereafter, Etter et al., (2011b) adapted the original protocol to used paired-end reads, in which the two ends of a DNA fragment are sequenced and are known to belong to the same fragment. In order to do so, the authors altered two keys aspects of the RAD-seq protocol: 1) a wider size range of fragments (300-800 bp) was isolated after shearing; 2) a longer, divergent P2 adapter that contains the reverse sequencing primer sequence was ligated to the variable end of the RAD tags before amplification, allowing the randomly sheared end of the RAD fragments to be sequenced by the second read. This small set of sheared-end sequences can be assembled into a larger contig, hence the quality of the entire data set improves with a paired-end approach (Etter et al., 2011b).

As RAD-seq is a reduced-representation sequencing approach, it targets a subset of the genome and, because of that, it is clear that it provides advantages over whole-genome sequencing. The advantages are including a greater depth of cover per locus, which improves confidence in genotype calls; and sequencing of greater numbers of samples for a given budget.
Consequently, because of these advantages RAD-seq has become the most widely used genomic approach for high-throughput SNP discovery and genotyping in evolutionary and phylogenetic studies of non-model organisms (Andrews et al., 2016).

Despite the applicability and ease of use of RAD and other NGS protocols, a significant continuously challenge facing biologists is developing the appropriate analytical and bioinformatics tools for these type of data (Etter et al., 2011a).

1.2.1.2 Applications of RAD-Seq

There are various examples of studies in different groups of organisms like birds, plants, molluscs and insects where the RAD-seq genomic approach was conducted. They aimed to resolve, not always in a straightforward way, some evolutionary questions and phylogenetic relationships, some of them the same of ours.

For example, in birds Ng et al., (2017) were worried about understanding the impact of trade on the population genomic patterns of connectivity and differentiation under the study organism, a threatened songbird – the white-rumped shama *Copsychus malabaricus*. Previous studies done based on a single locus approach (a mitochondrial gene) of the songbird (Lim et al., 2010, 2011), have obtained results that were not conclusive and a lot of questions still unresolved. For that purpose, a reduced-representation library of the bird genome with a reference genome was prepared using a double digest restriction-site associated DNA sequencing (ddRAD-seq) protocol (multilocus approach). This successful methodology fully solved the questions that were previously not answered yet.

In plants, Deng et al. (2018) have used the *de novo* RAD-sequencing genomic approach to resolve the phylogenetic relationships' questions that remained under East Asian evergreen oaks under the genus, section *Cyclobalanopsis*. This genus is phylogenetically challenging due to high intraspecific genetic variation, low interspecific differentiation and frequent interspecific gene flow (Denk & Grimm, 2010; Hipp, 2015; Kremer et al., 2012; Simeone et al., 2013). The reason why this genomic approach was chosen to resolve the genus was because previous studies of application of RAD-sequencing, have demonstrated that this genomic approach robustly estimated phylogenetic relationships among older oak

clades (Hipp et al., 2014), contrarily to the use of plastid DNA (Manos, Cannon, & Oh, 2008; Pham et al., 2017; Simeone et al., 2013; Xing et al., 2014) and ribosomal nuclear markers (Deng, Zhou, & Li, 2013) where the resolution of the phylogenetic relationships among the genus was not fully resolved. The RAD-seq approach results obtained have successfully corroborated previous results by supporting the separation the Eurasian and American clades and also provided a valuable framework and the best-resolved topology to date for understanding the phylogeny of the East Asian evergreen oaks in Eurasian clade.

Besides, Curto et al., (2018) performed a study interested in obtaining a more comprehensive picture of the evolutionary history of a plant genus *Micromeria*. The species from this genus are morphologically similar but ecologically diverse on each Canary Island, constituting a great model to investigate niche shifts and adaptation within the Canary Archipelago. Previous attempts to reconstruct the phylogenetic relationships among the genus did not led to robust phylogenies, presumably due to introgression and/or incomplete lineage sorting (Curto, Puppo, Kratschmer, & Meimberg, 2017; Puppo, Curto, & Meimberg, 2016). Because no genomic information is available for *Micromeria* to date, the most common reduced-representation sequencing technique, the RAD-sequencing method (Baird et al., 2008; Cronn et al., 2012; Elshire et al., 2011) was used. The results obtained corroborated the current reclassification of *Micromeria* and suggested that introgression have played a role in the evolution of the genus as suggested previously (Curto et al., 2017; Puppo et al., 2016). However, they recommended, for a more detailed understanding the history of the genus, more studies to be performed, since there was a lack of outgroups which led to misinterpretations.

In molluscs, Razkin et al., (2016) applied the *de novo* RAD-Seq approach to assess the phylogenetic relationships, interspecific hybridization and species delimitation in the cryptic, non-model land snail complex of the genus *Pyramidula.* Previous phylogenies based on mitochondrial COI and 16S and nuclear markers showed several unsupported branches and incongruent topologies (Razkin et al., 2016). This latter observation was tentatively interpreted as the result of incomplete lineage sorting. Hence, further work was needed involving more molecular markers to assess whether, and to what extent, processes like interspecific gene flow or incomplete lineage sorting have shaped the phylogenetic relationships among *Pyramidula* species. The RAD-Seq results obtained helped to fully resolved the phylogenetic relationships between species. In fact, the tests for intraspecific hybridization have successfully revealed that, the incongruences between the mtDNA and nDNA gene trees regarding two of the genus' species, were due to incomplete lineage sorting, which have been also documented in other molluscs (Wilding, Grahame, & Mill, 2000). Furthermore, as the two species didn't live in sympatry, the hypothesis of the incongruences being caused by gene flow were less plausible (Ballard & Whitlock, 2004).

In what refers to insects, Suchan et al., (2017) were interested in disentangling the phylogeny of the fly genus *Chiastocheta* at which there is no reference genome. Previous studies relying on a single, non-recombinant marker and ignoring potential incongruences between mitochondrial and nuclear loci provided an incomplete account of the lineage history of the genus (Després et al., 2002; Espíndola, Buerki, & Alvarez, 2012). Suchan et al., (2017) has proven that despite the higher performance of RAD-seq in terms of species trees resolution compared to cytoplasmic markers, reconstruction of inter-specific relationships among recent diverged lineages may lie beyond the possibilities offered by large data sets of RAD-sequencing markers in cases of strong gene tree incongruence due to incomplete lineage sorting.

1.2.1.3 A bioinformatics tool for RAD datasets analysis: Ipyrad

RAD-sequencing technique generates thousands of reads per individual, hence the process of analysing the data requires high computational tools and takes much more time than the time needed to generate the RAD sequence reads (Andrews et al., 2016).

There are several programs designed to analyse RAD-seq data like Stacks (Catchen et al., 2011) and Rainbow (Chong, Ruan, & Wu, 2012), however, most of the studies based on assembly of *de novo* RAD-seq *loci* (without reference genome) like (Curto et al., 2018; Deng et al., 2018; Razkin et al., 2016; Suchan et al., 2017) for phylogenetic analysis and population genetics used the pyRAD or ipyrad program tools.

Ipyrad (Eaton, 2015) is a toolkit specially used for assembly and analysis of genomic RAD-Seq data sets. It offers powerful methods to generate output files (assembled data) for forward downstream genomic analysis for both population genetics and phylogenetic studies.
Any type of data generated with restriction digest methods like RAD (Baird et al., 2008), double-digest RAD (Peterson, Weber, Kay, Fisher, & Hoekstra, 2012) and Genotyping-By-Sequencing; or amplification-based processes like Next-RAD and RAPture can be so assembled with this tool in four modes of assembly. Both approaches yield data that is anchored on at least one side, so that, reads are expected to align fairly close. However, it's important to have in mind that ipyrad is not intended for constructing long contigs from partially overlapping sequences but can accommodate paired-end reads and has particular methods for detecting and merging overlaps. For last, it can combine reads of various lengths what makes possible the combination of older data with newer data with different lengths.

This tool is very similar to pyRAD (D. Eaton, 2014), being a complete re-write of it but with a different approach. Besides ipyrad retains an easy-to-use command-line interface, the notorious power of ipyrad comes from its implementation through a Python API, which allows users to write scripts that detail complex assemblies able to construct multiple data sets under multiple parameters settings (D. Eaton, 2015).

Other improvements include:

- *de novo*, reference alignments and hybrid modes of assembly (four assembly methods);
- Parallel implementation using ipyparallel which utilizes MPI allowing use of HPC clusters;
- Possibility of restarting a script from the point of a job at which an interruption occurred;
- Faster code and no external installations;
- Writhe highly reproducible documented code with Jupyter Notebooks.

The typical workflow to move from Fastq formatted input data (dataset) to assembled output files, in ipyrad pipeline involves seven sequential steps (assembly steps) under a single set of parameters defined in a .txt params file (D. Eaton, 2015).

- ❖ The RAD sequence data as our dataset, can be received as one giant file or in many smaller files. The files may contain data from all of our individuals mixed up together, or as separate files for each sample. When mixed up together, the case that the data is not sorted among

individuals/samples (raw sequence files), then our data need to be first of all demultiplexed based on barcodes or indices. The .txt barcodes file, is then very important in the demultiplexing step (D. Eaton, 2014). This file is a simple table linking barcodes to samples. Each line of this file should have one name of the sample and then the respective barcode sequence, separated by a whitespace. The barcodes can be of varying lengths.

- The parameters input file needs to be created on ipyrad and its name includes the prefix "params-*assembly_name*.txt". It lists all of the 29 parameters settings (assembly parameters) that could be and should be modified to create assemblies under different combinations of parameter settings, on the basis of the recommendations (D. Eaton, 2015). It is crucial to vary the parameters used in all steps of the analysis to critically evaluate the sensitivity of the results and to optimize the analysis (Andrews et al., 2016), since several publications (Ilut, Nydam, & Hare, 2014; Mastretta-Yanes et al., 2015) emphasized the fact that the analytical results can be considerably affected by the parameters settings used in *de novo* assemblies.

The Seven Sequential Steps

The assembly process is separated in steps which is very advantageous, because it allows the process to be restarted at any point if interrupted and can be easily branched at different points to create assemblies under the different combinations of the parameter settings in the params file (D. Eaton, 2014, 2015).

The seven steps are:
1. Demultiplexing (separate by barcodes);
2. Quality filtering and removal of barcodes, cut sites and adapters;
3. Clustering within samples and alignment;
4. Joint estimation of heterozygosity and error rate;
5. Consensus base calling and filtering;
6. Clustering across samples and alignment;
7. Filtering and formatting output files.

1.3 The cicada group model in evolutionary studies

Cicadas are a group of insects, belonging to the family Cicadidae, super-family Cicadoidea and the order Hemiptera, with more than 2500 species described around the world (Sun et al., 2009). Cicadas have long larval stages underground which can be seen on Figure **1.5**, which is the case of the periodical cicadas, *Magicicada* spp. that have up to 13 or 17 years synchronized nymph cycles (Cooley et al., 2001b; Yoshimura, 1997). In the case of the European cicadas, they mostly have a life cycle of two to six years. The advantage of having long larval stages underground is that it reduces losses by starving their predators and allows their emergence in huge numbers that overwhelms any predators (Williams & Simon, 1995).

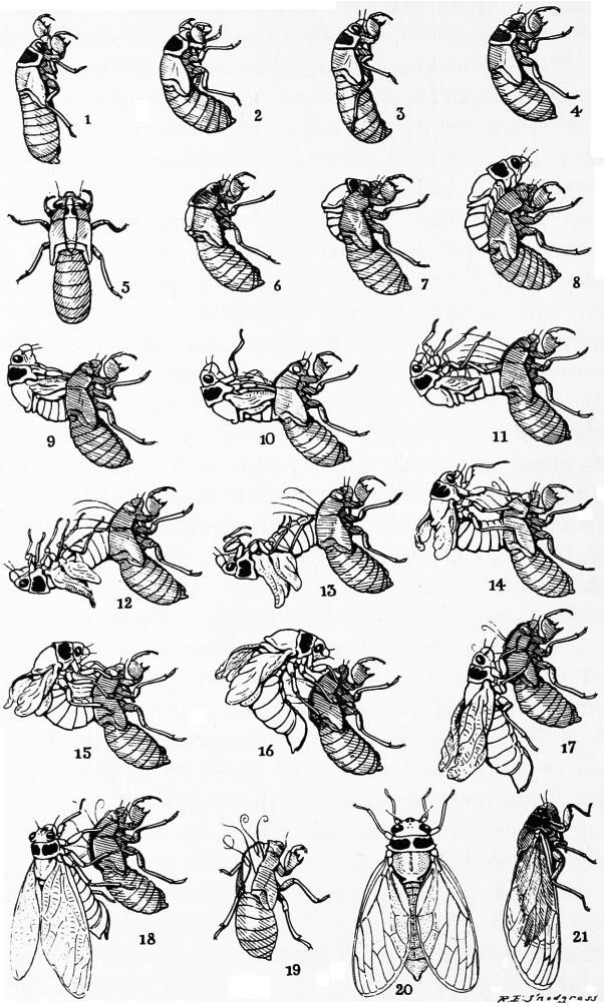


Figure 1.5 Maturation stages of a periodical cicada

Transformation of the periodical cicada from the mature nymph to the adult. Figure 118. from "Insects, their way and means of living", R. E. Snodgrass (1930).

They are well-known for the male ability to produce loud sounds during summer time by means of a tymbal mechanism (Claridge, 1985; Quartau & Boulard, 1995). The tymbal is the sound-specialized organ responsible for the production of a variety of calls. As presented on Figure **1.6**, the cicada tymbal consists on an abdominal membrane attached to the tymbalic muscles which, with each round of contraction and relaxation, are able to vibrate on frequencies between 35 and 100 Hz producing sounds with frequencies up to 25 kHz, past the human audible frequency (Wessel et al., 2014).

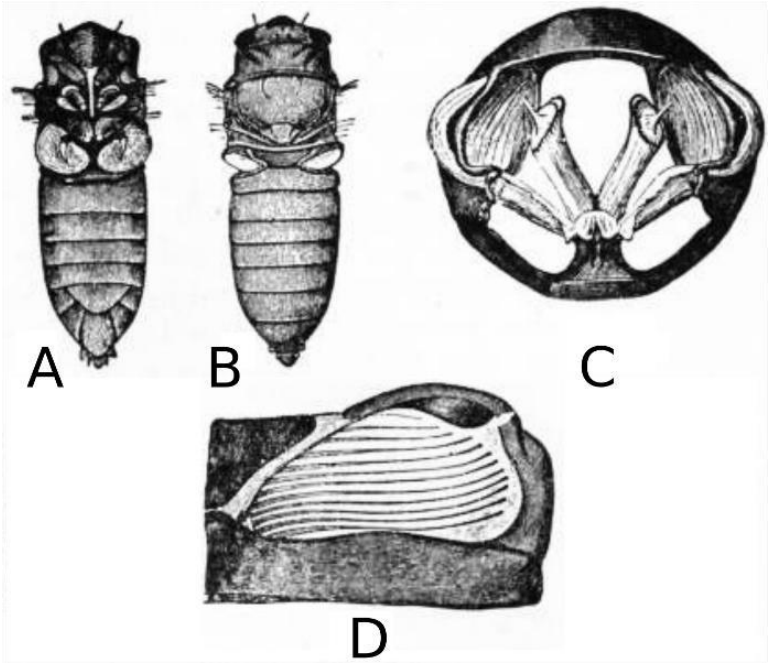


Figure 1.6 Morphology of a cicada tymbal
(A) Ventral view of a male cicada showing the fan-shaped opercula covering the tympana, which the cicada uses to hear. (B) Dorsal view of a male cicada showing the tymbals (drums) between the thorax and abdomen. (C) Abdominal cross-section of the male cicada showing the tymbalic muscles connected to the tymbals and anchored to the sternal cuticle. (D) Tymbal membrane, a convex layer of the cuticle, and often possessing several thin and resilin-coated portions intercalated by thickened ribs, as shown. Adapted from Carpenter (1911).

Different types of sounds can be produced, and each has a different function (Boulard & Mondon 1996):

- Calling song: the most common call of male cicadas, is used to call the females for pair formation and courtship;
- Courtship call: is emitted when the singing male is approached by an interested female;
- Alarm call: produced when a cicada senses something unusual in its environment;
- Protest calls: can be subdivided in:
    - Opposition calls: when multiple males of same species are present in the same area or tree;
    - Distress calls: when a male cicada is caught.

The calling songs are species-specific (Boulard, 2006; Cooley & Marshall, 2001a). Being a species-specific character, it can be used to distinguish different species even between closely-related ones. This makes it a taxonomical valuable character (Boulard, 1982, 2006; Claridge, 1985; Sueur, 2006) for cicadas' species discrimination.

The Iberian Peninsula have been identified as an area of high diversity and endemism (García-Barros et al., 2002; Jong, 1998), not only for plants (García-Barros et al., 2002; Medail & Quezel, 1997; Moreno Saiz et al., 1998; Mota et al., 2002), amphibians and fishes (Vargas et al., 1998) but also for a lot of insects including the cicadas (Jong, 1998; Ribera, 2000). Among the Mediterranean countries, Portugal

is placed at the western most point of Europe and is close to the North Africa being affected by the clime of both Mediterranean and Atlantic, one of the reasons that allows Portugal to constitute a real hotspot for cicada's diversity (Sueur et al., 2004). The *Tettigettalna* is the most biodiverse genus of cicads in o Portugal giving raise to the particular interest in studying evolutionary questions and speciation of this genus.

1.3.1 The *Tettigettalna* genus case study

Tettigettalna is a genus of cicadas from south-western Europe that occur in typical Mediterranean landscapes. This genus constitutes a complex of closely related and recently diverged species. They are morphologically similar what makes them a taxonomical challenging group. Fortunately, the calling songs are the main characters for their identification which were confirmed by Mendes et al., (2014) study.

In the Iberian Peninsula, the diversity of cicadas was undervalued until the recent description and taxonomic revision of nine small-sized cicadas species under the genus *Tettigettalna* (Boulard, 1982; Puissant & Sueur, 2010; Quartau & Boulard, 1995). The distribution ranges of eight cicada species under the *Tettigettalna* genus can be seen on Figure **1.7**. *Tettigettalna argentata* distribution was not shown because its widespread distribution along other countries.

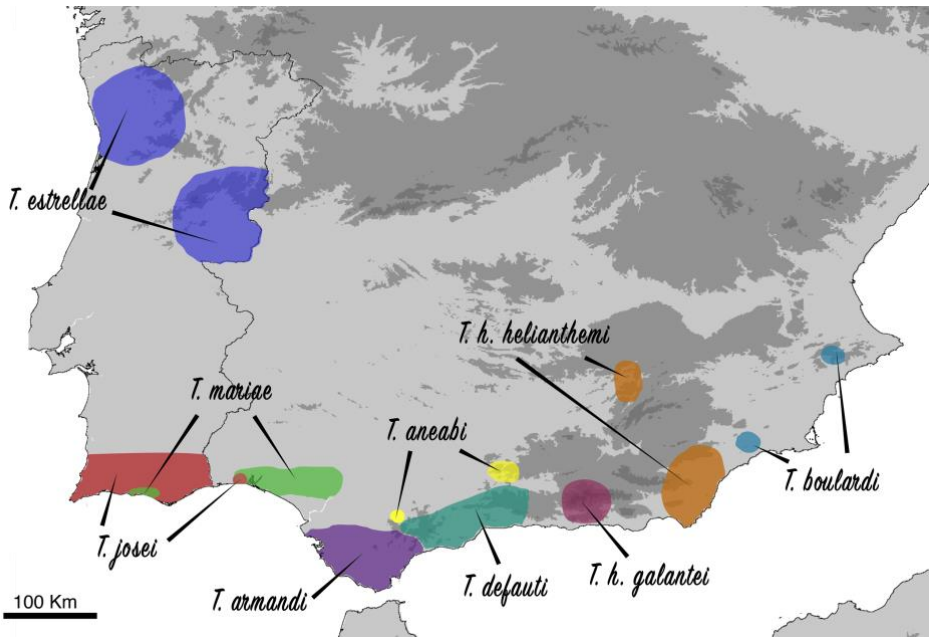


Figure 1.7 Iberian Peninsula map with approximated distributions of the genus *Tettigettalna*
The distribution areas were based on (Nunes et al., 2014; Puissant & Sueur, 2010; Simões et al., 2013; Simões et al., 2014). Scale bar indicates 100 km. Figure from (Costa, 2017).

Eight of these nine *Tettigettalna* species are endemic to the Iberian Peninsula: *T. mariae* (Quartau & Boulard, 1995), *T. aneabi* (Boulard, 2000), *T. josei* (Boulard, 1982), *T. defauti* (Puissant & Sueur, 2010), *T. armandi* (Puissant & Sueur, 2010), *T. helianthemi* (Rambur, 1840) and *T. boulardi* (Puissant & Sueur, 2010) occur only in the southern part and *T. estrellae* restricted to the northwest part (Puissant & Sueur, 2010; Simões et al., 2013; Sueur et al., 2004).

Five of these *Tettigettalna's* species - *Tettigettalna armandi*, *Tettigettalna aneabi*, *Tettigettalna defauti*, *Tettigettalna heliathemi* and *Tettigettalna boulardi* - are restricted to the south of Spain (endemism of Spain). *Tettigettalna estrellae* is the only one restricted to Portugal (endemism of Portugal). The last one mentioned can be found, more specifically in the North of Portugal (Nunes et al., 2014; Puissant & Sueur, 2010; Simões et al., 2013), in the regions of Serra da Estrela, Póvoa do Lanhoso e Régua (Sueur et al., 2004).

Tettigettalna josei (Figure **1.8**) was also considered a species restricted to Portugal with a quite widespread distribution in Algarve in open habitats covered with low vegetation and well exposed to sunlight (Sueur et al., 2004). However, the study performed by Simões et al., (2014) in the Southern Iberian Peninsula (regions of Algarve and Andalusia) during the summers of 2011-2013 have clarified the currently distribution range of the *Tettigettalna josei* species. The authors found occurrences of *T. josei* species in small numbers in Cartaya (Huelva, Spain) which extended their distribution range to Spain and, therefore, became an Iberian endemism.

Figure 1.8 *Tettigettalna josei* copulation
Tettigettalna josei male and female coupling during copulation observed in July 2013 near Sesmarias (37º04' 38.6"N, 8º18' 28.9"W), in Algarve, Portugal. Photo taken by Vera Nunes.

In what refers to *Tettigettalna mariae* species (Figure **1.9**), it was thought to be endemic to Portugal (occurring in Algarve) (Sueur et al., 2004). A study performed by (P. C. Simões et al., 2013) between 2011-2013 have allowed, the acquisition of the first records of *Tettigettalna mariae*'s acoustic signals in the province of Huelva (Southern Spain). Therefore, this species is know considered an Iberian endemism.

A further study about the current distribution and habitat preferences of *T. mariae* was performed by (Nunes et al, (2014a), confirming the previous results and notiing the restricted and fragmented distribution of this species to the coast of central Algarve, in Portugal, and Huelva province, in Spain. The latter study (Nunes et al., 2014a) has demonstrated that this species is habitat-specific, with a particular preference for habitats with stone pine and at a close distance from the sea (

Figure 1.10). The authors also refer that throughout the years the decline of stone pines' population resulted in the fragmented distribution of *T. mariae* between central Algarve and Huelva.

Figure 1.9 Specimen of *Tettigettalna mariae*
Tettigettalna mariae male photo taken in Quinta do Lago (Algarve, Portugal) (Lat. 37º 03' 31.2", Long. 8º 01' 16.0") by Vera Nunes and Raquel Mendes.

Figure 1.10 Stone pine wood from Cartaya
Stone pine (*Pinus pinea*) wood photo taken in July 2013 (Lat. 37º15'38.44"N, Long. 7º07'43.52"W) near Cartaya (Huelva, Spain). Cartaya corresponds to the location where the largest population of *T. mariae* was found so far. The photo was taken by Vera Nunes and Raquel Mendes.

T. argentata (Olivier, 1790) (Figure **1.11**) is the only species that extends its range distribution behond the Iberian Peninsula to other European countries like Italy, south border of Switzerland, south of France and karstic region of Slovenia. Therefore this is the species with the largest distribution range in its genus (Gogala & Gogala, 1999; Hertach, 2008; Nast, 1972; Puissant & Sueur, 2010; Sueur et al., 2004). In Portugal, this species is widely distributed in areas of Algarve, Baixo Alentejo, Alto Alentejo, Estremadura, Beira Alta, Trás-os-Montes and Minho (Sueur et al., 2004).

Figure 1.11 Specimen of *Tettigettalna argentata*
Tettigettalna argentata male photo taken in Quinta do Lago (Algarve, Portugal) (Lat. 37º 03' 31.2", Long. 8º 01' 16.0") by Vera Nunes and Raquel Mendes.

1.3.2 *Tettigettalna mariae* and *Tettigettalna argentata:* the sibling species

Previous studies have revealed the existance of sympatry among some of the southern Iberian *Tettigettalna* species. In fact, *Tettigettalna argentata* populations have overlapping distributions with populations of other species (Boulard, 1982; Quartau & Boulard, 1995; Sueur et al., 2004): *Tettigettalna estrellae* (Boulard, 1982), *Tettigettalna mariae* (Quartau & Boulard, 1995) and *Tettigettalna josei* (Boulard, 1982). *Tettigettalna mariae* and *Tettigettalna josei* may also be found in Algarve (Simões et al., 2013).

In Algarve, *Tettigettalna mariae* and *Tettigettalna argentata* populations can be found in sympatry or close parapatry (Simões et al., 2013). These two species are morphologically very similar and only distinguishable by their calling songs (Mendes et al., 2014).

A mitochondrial COI sequences analysis performed by Nunes et al., (2014b) allowed the separation of *Tettigettalna argentata* in northern (*T. argentata* from Italy, France, and two localities of Iberian Peninsula – Braga and Sesimbra) and southern (*T. argentata* from São Bartolomeu de Messines, Portel, Espiel and Ayamonte) clades. Besides, the southern clade was genetically inseparable from *Tettigettalna mariae*, sharing with it its most common haplotype. This makes it impossible to unambiguously discriminate *T. mariae* specimens from *T. argentata* (clade South) on the basis of COI sequences analysis alone.

Furthermore, as *Tettigettalna* species can be distinguished by the acoustic signals, what were thought to might play an important role on their reproductive isolation, an acoustic and morphological study was

performed by Mendes et al., (2014). The authors were interested in understanding if there were patterns in *Tettigettalna* populations that provide evidence of species recognition and reproductive isolation. In fact, acoustic results were conclusive that the different species have different acoustic patterns but genetic ones make it unclear whether the sharing of haplotypes between clade South of *T. argentata* and *T. mariae* specimens were due to introgression (existence of gene flow between populations) or incomplete lineage sorting (defective segregation of alleles into well-defined lineages) (Mendes et al., 2014).

Both studies (Mendes et al., 2014; Nunes et al., 2014b) revealed the need for more research, namely that instead of a single locus approach, a multilocus approach should be performed to confront the results with the previously obtained. Therefore, the use of the Next-Generation sequencing data can greatly speed up the research effort to investigate the complex relationships of this group.

1.4 Objectives of the Thesis

In the *argentata* complex, *Tettigettalna mariae* has a restricted and fragmented distribution along the southern coast of the Iberian Peninsula and occurs in sympatry or close parapatry with *Tettigettalna argentata* in several locations, showing little or no genetic divergence from *Tettigettalna argentata*.

This genetic variation shared between closely related species may be due to retention of ancestral polymorphisms because of incomplete lineage sorting (ILS) and/or introgression following secondary contact. It is challenging to distinguish ILS and introgression because they generate similar patterns of shared genetic diversity, but this is nonetheless essential for inferring accurately the history of species with overlapping distributions.

Hence, this thesis entitled by "Analysis of RAD sequencing data from Mediterranean cicadas" emerges as an investigation focused on the treatment and analysis of RAD-seq data from Mediterranean Cicadas. This was possible with the help of various bioinformatics tools such as Ipyrad, VCFtools and other scripts designed to clean and filter this type of data; and for the analysis of the data, programs like Maverick and other scripts to perform data tests were used as an attempt to enlighten some unanswered questions about Mediterranean cicadas.

This thesis has two main goals: it is an effort to understand if RAD-seq data, in particular, gives support or not the geographical distribution patterns of the *Tettigettalna's* complex under study; and if this type of data is able to demystify the big question that if the haplotype sharing between the pair of sibling species, *Tettigettalna mariae* and *Tettigettalna argentata* from the *argentata* complex, found in sympatry or close parapatry on Algarve, is due to a phenomenon of introgression or incomplete lineage sorting.

Chapter 2

Material and Methods

Material used was collected by hand or using a sweeping net during the summer surveys from 2011 to 2013. It contains forty *Tettigettalna* specimens with individuals belonging to the *Tettigettalna argentata* species complex (*T. argentata* (allopatric and sympatric), *T. mariae* (allopatric and sympatric) and *T. aneabi*) and also a basal species, one *T. josei* specimen (see Table ***6.1*** in Supporting Information). In the field, each specimen was assigned to a tracking number and to a species according to the male calling song.

Each of the species of the *Tettigettalna* genus under study were collected on the surveys from the regions that can be seen on Figure 2.1.

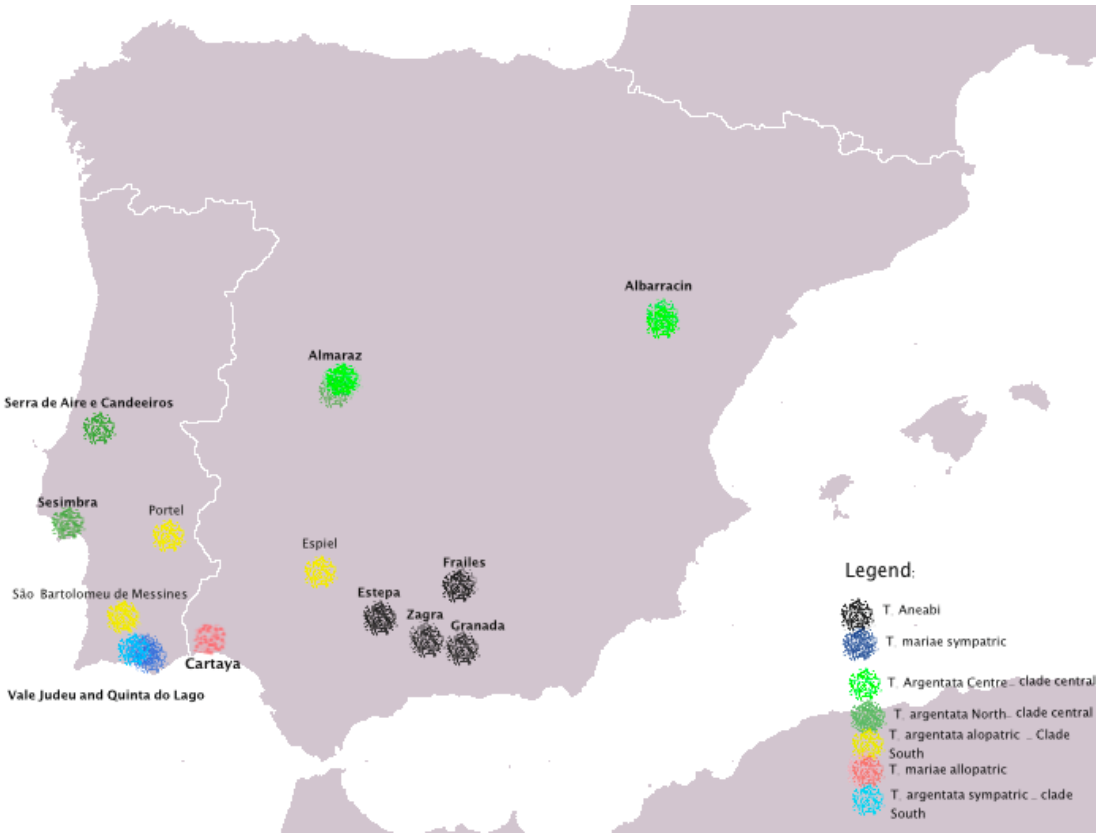


Figure 2.1 Species locations collected during the surveys 2011-2013 on Iberian Peninsula

A front leg from each specimen was removed and preserved in 100% ethanol and used for DNA isolation with DNeasy Blood & Tissue extraction kit (Qiagen).

After DNA extraction, a RAD-seq protocol (Baird et al., 2008) was used and adapted to use paired-end DNA fragments (Etter et al., 2011b). DNA digestion for the library preparation was performed with *SbfI*, a rare cutter enzyme suitable for a reasonable coverage in large size genomes, as is the case of cicadas. The paired-end sequencing was performed in Illumina HiSeq 2000/2005 platform at Edinburgh Genomics, Ashworth Laboratories (https://genomics.ed.ac.uk/).

A folder with the raw sequence data, with 197972629 raw reads, was successfully generated from the sequencing process was received for the beginning of the bioinformatics treatment, presented in the present thesis.

2.1 The RAD-Seq data treatment

The data was first treated with the ipyrad pipeline (Eaton, 2014, 2015) since studies with same type of data and problematic gave support and recommended the use of this bioinformatics tool (Curto et al., 2018; Deng et al., 2018; Razkin et al., 2016; Suchan et al., 2017).

2.1.1 The sequence treatment with Ipyrad

A barcodes file is essential for running the first two steps of Ipyrad assembly. Three barcodes files were considered: "barcodes_aneabi.txt", "barcodes_sem_aneabi.txt" and "barcodes_com_josei.txt", the first containing *Tettigettalna aneabi* individuals, the second one excluding them (see File ***6.1*** and File ***6.2*** in the Supporting information) and the third one, that included *T. josei*, was created to perform three new assemblies runs with the individuals of only those four populations for the ABBA/BABA test (see File ***6.3*** in the Supporting Information).. Those two barcodes files excluded the *Tettigettalna josei* individuals since its genomic data could bring error and bias to the assemblies, because they are an outgoup (they are genetically distant from the rest of the complex). Hence, the total of individuals in the barcodes file taken into analysis "barcodes_aneabi.txt" (File ***6.2***) was 37, in the "barcodes_sem_aneabi.txt" (File ***6.1***) was 31 and "barcodes_com_josei.txt" (File ***6.3***) was 21.

A parameters file was also essential and was created with a given name (the initial assembly). The first four assembly parameters from that parameters file were crucial to run step one and step two of the seven steps of a complete assembly run on Ipyrad.

The command presented below generated a "params-params.txt" parameters file (File ***6.6***) with the same name that was given to the assembly. In this case, "params" was the name given to the assembly. Hence, the parameters file created, puts always the prefix "params-" and joins the name given to the assembly, which results in "params-params.txt"

>>> **ipyrad -n params**

Besides params assembly, 11 more assemblies were created because different combination values' of the principal parameters for the data treatment needed to be tested. Then, each new assembly was created for a specific value combination for those principal parameters that should be tested.

From params to params9 assemblies, the barcodes file used were the one that included the *T. aneabi* individuals, the "barcodes_aneabi.txt" (File ***6.2***).
For the assemblies params10 and params11 the barcodes file used were the one that excludes the *Tettigettalna aneabi* individuals, the "barcodes_sem_aneabi.txt" (File ***6.1***).
For the assemblies params12, params13 and params14, the barcodes file used were the one that included *T. josei* (File ***6.3***).

>>> **ipyrad -n params1**
>>> **ipyrad -n params2**
>>> **ipyrad -n params3**
>>> **ipyrad -n params4**
>>> **ipyrad -n params5**
>>> **ipyrad -n params6**
>>> **ipyrad -n params7**
>>> **ipyrad -n params8**
>>> **ipyrad -n params9**

Assemblies without *Tettigettalna aneabi* individuals:

>>> **ipyrad -n params10**
>>> **ipyrad -n params11**

Assemblies with*Tettigettalna josei* individual:

>>> **ipyrad -n params12**
>>> **ipyrad -n params13**
>>> **ipyrad -n params14**

- **STEPS 1-2**

After creating the assemblies params and params10, the establishment of the first four parameters of the parameters file were crucial for step one and two. They were responsible for loading the raw data, demultiplexing - with the help of the barcodes file - and filtering the data during steps 1 and 2.

------- ipyrad params file (v.0.7.15)----------
params ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final ## [1] [project_dir]: Project dir (made in curdir if not present)
/home/ioliveira/lane2/raw/*.sanfastq.gz ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_aneabi.txt ## [3] [barcodes_path]: Location of barcodes file

------- ipyrad params file (v.0.7.15)----------
params10 ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final ## [1] [project_dir]: Project dir (made in curdir if not present)
/home/ioliveira/lane2/raw/*.sanfastq.gz ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_sem_aneabi.txt ## [3] [barcodes_path]: Location of barcodes file

params12 ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final_josei ## [1] [project_dir]: Project dir (made in curdir if not present)
/home/ioliveira/lane2/raw/*.sanfastq.gz ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/**home/ioliveira/barcodes_com_josei.txt** ## [3] [barcodes_path]: Location of barcodes file

The command lines for running step one and two after the establishment of the crucial parameters for those steps were:

>>> **ipyrad -p params-params.txt -s 12**
>>> **ipyrad -p params-params10.txt -s 12**
>>> **ipyrad -p params-params12.txt -s 12**

- **STEPS 3-7**

From step three to seven, all the parameters of the parameters.txt file for all the assemblies have to be fulfilled with a value. The values could be set by default when recommended or altered respecting the range values that are recommended and depending of the dataset.

Various studies (Eaton & Ree, 2013; Escudero, Eaton, Hahn, & Hipp, 2014; Razkin et al., 2016; Suchan et al., 2017; Takahashi & Moreno, 2015; Takahashi, Nagata, & Sota, 2014) with the same data type (RAD-Seq data) and the same problematic as ours have used the pyRAD tool that is similar to ipyrad. They have defined the **min_samples_locus (#21 parameter)** and the **clust_threshold (#14 parameter)** as the main parameters of the parameters file that should be altered and the range of the values to fulfill them.

In our study, all the parameters values in the parameters.txt file for each of the assemblies can be consulted from File ***6.6*** to File ***6.20***. Furthermore, resumes of the information, in terms of values for those two main parameters, of the assemblies, are presented in Table ***6.2*** in Supporting Information.

Before running the rest of the steps, an alteration of the path of the data to be loaded should be done. As the data have been already demultiplexed and filtered by step one and two, the data are sorted. Hence, the parameter two should be left blanked and the parameter four should be fulfilled with the path to the sorted fastq files (see below).

------- ipyrad params file (v.0.7.15)----------

params ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final ## [1] [project_dir]: Project dir (made in curdir if not present)
[2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/**home/ioliveira/barcodes_aneabi.txt** ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final/params_fastqs/*.gz ## [4] [sorted_fastq_path]: Location of demultiplexed/ sorted fastq files

------- ipyrad params file (v.0.7.15)----------
params10 ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final ## [1] [project_dir]: Project dir (made in curdir if not present)
[2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/**home/ioliveira/barcodes_aneabi.txt** ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final/params_fastqs/*.gz ## [4] [sorted_fastq_path]: Location of demultiplexed/ sorted fastq files

------- ipyrad params file (v.0.7.15)----------
params12 ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final_josei ## [1] [project_dir]: Project dir (made in curdir if not present)
[2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_com_josei.txt ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final_josei/params12_fastqs/*.gz ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files

After the modification of the path, step three to seven were run fot both params, params10 and params12 assemblies:

>>> **ipyrad -p params-params.txt -s 34567**
>>> **ipyrad -p params-params10.txt -s 34567**
>>> **ipyrad -p params-params12.txt -s 34567**

For the assemblies from params1 to params9, their .txt parameters file had the parameter four fulfilled with the path of the sorted fastq files from the "params_fastqs" folder and the parameter two blanked. The same occurred for params11 assembly that didn't have *Tettigettalna aneabi* specimens. The parameters file of the params11 assembly had the parameter four fulfilled with the path to the sorted fastq files from the "params10_fastqs" folder and the parameter two blanked. The parameters file of the params13 to params14 assemblies had the parameter four fulfilled with the path to the sorted fastq files from the "params12_fastqs" folder and the parameter two blanked.

This way, in a single run all the steps were done (see below):

>>> **ipyrad -p params-params1.txt -s 1234567**
>>> **ipyrad -p params-params2.txt -s 1234567**

>>> ipyrad -p params-params3.txt -s 1234567
>>> ipyrad -p params-params4.txt -s 1234567
>>> ipyrad -p params-params5.txt -s 1234567
>>> ipyrad -p params-params6.txt -s 1234567
>>> ipyrad -p params-params7.txt -s 1234567
>>> ipyrad -p params-params8.txt -s 1234567
>>> ipyrad -p params-params9.txt -s 1234567

For the assembly params11 without *Tettigettalna aneabi* individuals:

>>> ipyrad -p params-params11.txt -s 1234567

For the assembly params13 and params14 with *Tettigettalna josei* individuals:

>>> ipyrad -p params-params13.txt -s 1234567
>>> ipyrad -p params-params14.txt -s 1234567

2.1.2 The filtering step with VCFtools program

The VCFtools, as the name suggests is a program package designed for working with Variant Call Format (VCF) files, such as those generated with the 1000 Genomes Project. The aim of this tool is to provide easily accessible methods for working with complex genetic variation data in the form of VCF files. This tool was used in this work to filter our VCF files, obtained from the assemblies performed on Ipyrad tool, by two filtering criteria that needed to be taken in consideration on the RAD-Seq data. Hence, the result of the filtering step with this tool is VCF files with SNPs filtered in terms of the two criteria that would be mentioned below.

The SNPs of the .vcf file of each assembly saved in the outfiles' folder named "*assembly_name*_outfile" were filtered with VCFtools program in terms of minor allele frequency (maf) and max-missing (see below). This filtering procedure was performed for all the 15 assemblies (from params.vcf to params14.vcf). The command bellow was used for the "params" assembly filtering but serves as an example of what was done for all the assemblies.

>>> vcftools --vcf params.vcf --maf 0.05 --max-missing 0.80 --recode

A maf filtering of 0.05 means that SNPs were filter out with a minor allele frequency less than 5% in the VCF file.

A max-missing filtering of 0.80 means that the SNPs were filtered on the basis of the proportion of missing data, in this case 0.80 means specifically that SNPs with more than 20% of missing data were excluded.

From this VCFtools filtering step, twelve *assembly_name*.recode.vcf files generated by this VCFtools filtering step were obtained.

2.1.3 The filtering step with vcf_parser.py code

A **vcf_parser.py** python code from https://github.com/CoBiG2/RAD_Tools, as of commit 9a5bc1bddc (see File ***6.21*** in Supporting Information) designed to perform filtering and transformations steps on a Variant Call Format (VCF) file that are not possible with VCFtools was then used. This code has five optional functions that filters by five different criteria the input VCF file, generating an output file for each of the filtering function chosen.

```
MacBook-Air-de-Ines:Desktop inesoliveira$ python vcf_parser.py --help
usage: vcf_parser.py [-h] -vcf VCF_INFILE [--remove-inv] [--remove-singletons]
                     [--one-snp] [--random-snp] [--center-snp]

Filtering of VCF files
optional arguments:
  -h, --help            show this help message and exit
  -vcf VCF_INFILE       Provide VCF file(s).
  --remove-inv          Filters invariable SNP sites from the VCF file (These sites may occurr when
individuals are removed from a VCF file).
  --remove-singletons   Filters singletons SNP sites from the VCF file.
  --one-snp             Filters the VCF file so that only one SNP per locus is retained - The first one
  --random-snp          Filters the VCF file so that only one random SNP per locus is retained.
  --center-snp          Filters the VCF file so that only the SNP closest to the locus center is retained.
```

The 12 *assembly_name*.recode.vcf files were all filtered by three of the five possible filtering options available by this script: SNPs were filtered in terms of invariable sites and singletons; from the other three options, the centre SNP function were chosen because the other two approaches (random SNP and the first SNP per locus) would turn our data biased. The command line used, as an example, for the assembly params was:

```
>>> python3 /.../ficheiros_recode_final/vcf_parser.py -vcf
/.../ficheiros_recode_final/params.recode.vcf --remove-inv --remove-singletons --center-snp
```

The line code used have generated three output VCF files for each assembly. The name of the output VCF files had always the prefix name being the assembly name of that VCF file, plus the suffix that is the filtering option applied: ***assembly_name*CenterSNP.vcf**, ***assembly_name*_NoSing.vcf** and ***assembly_name*_NoInv.vcf**. The three output VCF files generated for each of the assemblies were finally analysed with the >>> wc (word count) command followed by the name of the vcf file, to compare the files in terms of number of SNPs to the respective *assembly_name*.recode.vcf file that served as input.

There weren't no differences in terms of number of SNPs between each of the 15 input VCF files (*assembly_name*.recode.vcf files) and the respective VCF output files that filtered the SNPs in terms of invariable sites and singletons. Henceforth, our input files didn't have singletons neither invariable sites. However, for the VCF output file that filtered the SNPs by choosing the centre SNP for each *locus*, there was a considerable reduction of SNPs. Hence, for all the 12 assemblies, the VCF file chosen to proceed to the analysis were the ***assembly_name*CenterSNP.vcf** file.

2.2 The Principal Component Analysis of the RAD-Seq data

After the filtering process, a Principal Component Analysis (PCA) of the RAD-Seq data was performed. This analysis was performed to examine in two principal components (PC1 & PC2), how correlated are the individuals based on the analysis of all the SNP markers for all of them for each file. This PCA analysis were crucial to analyse if there are clustering groups of individuals and if it the results are in conformity with species/populations geographical distribution patterns.

A PCA was performed for the **12 *assembly_name*.recode.vcf** and **12 *assembly_name*CenterSNP.vcf** files, ecludig the assemblies with *T. josei*. We used an R script named **vcf2PCA.R** from https://github.com/CoBiG2/RAD_Tools, as of commit 9a5bc1bddc (see File ***6.22*** in Supporting Information) that needed as input the VCF file and one .txt file with the population name in the same order each individual appeared in the VCF file.

The two formatted .txt files used in this step are named "individuos_sem_aneabi_algarve.txt" (File ***6.5***) and "individuos_algarve.txt" (File ***6.4***). Those two optional .txt files were created because some of the VCF files (params10 and params11 assemblies) didn't have the *Tettigettalna aneabi* individuals on the data set. The two .txt files took in consideration the change of the population name of three individuals from Vale Judeu, Algarve to sympatric populations - Tma071_sym, Tma729_sym and Tma068_sym -, that a previous PCA of the genomic data appointed as sympatric, but that were registered incorrectly as allopatric based on the field surveys. It was recommended to make this change because although being appointed as allopatric, due to the absence of songs of other species cicadas nearby the location, maybe existed in fact other species nearby but not heard at that moment. Besides that, these files didn't contain the two individuals with a lot of missing data.

2.3 Analysis of the RAD-Seq data with MavericK program

After the Principal Component Analysis, MavericK program was used for inferring population structure on the basis of genetic information. The mixture modelling framework although identical to that used in the Structure program (Falush, Stephens, & Pritchard, 2003, 2007; Hubisz et al., 2009; Pritchard, Stephens, & Donnelly, 2000) has the ability to estimate the number of demes (denoted K) in a reliable way using a technique known as thermodynamic integration (TI). While this technique is more computationally intensive than standard MCMC based methods, it has been found to provide estimates of K that are more accurate and precise than those based on some heuristic estimators. The MavericK program also implements certain MCMC techniques that reduce the number of iterations needed to achieve convergence, and allows for some non-standard inputs and outputs (Nichols, 2017).

In terms of workflow, MavericK is very simple. The program reads in two text files – a parameters file and a data file. Then it carries out the analysis specified by the user in the parameters file. Outputs are produced in the form of multiple .csv and .txt files.

We started by performing a pilot run for each of the assemblies params1CenterSNP.vcf, params5CenterSNP.vcf, params8CenterSNP.vcf and params10CenterSNP.vcf that were converted to the STRUCTURE format with PGDSpider. The same values of MavericK's tutorial were adopted for

each of the parameters in the "parameters.txt" file (see **Error! Reference source not found.** in the Supporting Information) with the exception of the value of K that we set from one to ten. The command line presented serves as an example of the procedure to run Maverick pilot run of the assembly params8:

```
structure_threader run -i /home/ioliveira/MAVERICK/params8_final -o
/home/ioliveira/MAVERICK/params8_pilot_final –params
/home/ioliveira/MAVERICK/parameters.txt -mv ~/.local/bin/MavericK -K 10 -t 4
```

Each pilot run generated output files that were taken into analysis with a R script called "MavericK1.0_functions.R" (see **Error! Reference source not found.** Supporting Information). This script generated plots whose analysis was crucial to verify if the values stablished in the parameters.txt settings need to be optimized or if they are appropriate for the next step (the final run that estimate the number of demes (K) using the thermodynamic integration (TI)).

For the last Maverick run, the params8 and params10 assemblies were chosen. The individuals of the STRUCTURE formatted input files mentioned (params8_final and params10), on Geany editor, were ordered/clustered together in terms of population groups. The changes on the population names of that three individuals from Vale Judeu, Algarve (Tma071_allo, Tma729_allo and Tma068_allo) was also taken in consideration, in those two assembly files. This because a previous PCA of the genomic data appointed as sympatric, but that were registered incorrectly as allopatric based on the field surveys. It was recommended to make this change because of PCA evidence of being sympatric. Although appointed as allopatric, due to the absence of songs of other species cicadas nearby the location, maybe existed in fact other species nearby but not heard at that moment.

The parameters .txt files, files "parameters_final8_final.txt" (File ***6.25***) and "parameters_final10.txt" (File ***6.26***) were used for those final runs with the purpose of the estimation of the best K, between one to seven, with the Thermodynamic Integration. The command lines presented corresponds to the procedure to run Maverick final run of the assemblies params8 and params10:

```
~/.local/bin/structure_threader run -K 7 -i
/home/ioliveira/maverick_final/mavparams8/params8_final.txt -o
/home/ioliveira/maverick_final/mavparams8/results/ -t 8 -mv ~/.local/bin/MavericK --params
/home/ioliveira/maverick_final/mavparams8/parameters_final8_final.txt --log=1
```

```
~/.local/bin/structure_threader run -K 7 -i
/home/ioliveira/maverick_final/mavparams10/params10.txt -o
/home/ioliveira/maverick_final/mavparams10/results/ -t 8 -mv ~/.local/bin/MavericK --params
/home/ioliveira/maverick_final/mavparams10/parameters_final10.txt --log=1
```

2.4 ABBA/BABA test

ABBA/BABA statistics, also called D-statistics, was the final step of the analysis of our data in order to understand if the similar patterns of shared genetic diversity between *T. mariae* and *T. argentata* found in sympatry or close parapatry were due to introgression or incomplete lineage sorting. This statistic was then used because it provides a simple and powerful test for a deviation from a strict bifurcating evolutionary history, being frequently used to test for introgression using genomic SNP data (Zhou et al., 2017).

For the ABBA/BABA test we needed to be stablished four populations (P1, P2, P3 and P4 (Outgroup)). Our outgroup was the *Tettigettalna josei* because it is the most distant *Tettigettalna* from all the *Tettigettalna* under study; the other three populations were *Tettigettalna argentata* from South in allopatry from South clade (P1), *Tettigettalna argentata* from South in sympatry (P2) and *Tettigettalna mariae* in sympatry (P3) that share the haplotype with *Tettigettalna argentata* in sympatry (P2).

After the data treatment for the assemblies params12, params13 and params14 (params12.recode.vcf, params12.recode.vcf and params13.recode.vcf), the three assemblies' recode.vcf files were filtered with the VCFtools to remove indels and SNPs that were not bi-allelic. Hence, the command line below is the example for the assembly params12 but were done also for assemblies params13 and params14:

```
>>> vcftools --vcf params12.recode.vcf --remove-indels --min-alleles 2 --max-alleles 2 –recode
```

The command line generated an out.log and an out.recode.vcf file that was renamed to the respective assembly (for example: params12filtered.recode.vcf and params12filtered.log).

After that, the VCF files were converted with bcftools:

The bcftools v1.9 program was used to generate GenoType (GT) formatted files, which is the file format needed to do the introgression test. The command presented below is an example for the assembly params12, but the same were done for params13 and params14 assemblies.

```
>>> vcffile=params12filtered.recode
>>> bcftools query -f '[%GT\t]\n' ${vcffile}.vcf > ${vcffile}.GT
>>> sed -i "s/0\/0/0/g;s/0\/1/1/g;s/1\/0/1/g;s/1\/1/2/g;s/\.\/\./-1/g;s/\./-1/g" ${vcffile}.GT
```

The command line generated a GT file with the same name of the input file, in this case we obtained with the three runs performed a "params12filtered.recode.GT" file, a "params13filtered.recode.GT" file and a "params14filtered.recode.GT" file.

After having the GT formatted files, a script written in R named "Dstat.R" (Seabra et al., 2019) (see File ***6.28*** in Supporting Information) wase used to calculate the D-statistic. Besides the input .GT file the script required a .txt file that was named "IndsPops.txt" (see File ***6.27*** in the Supporting information) in which the first column corresponded to the individual ID and the second column the population name to which each individual belongs. The order of the individuals of that .txt file was the same of the .vcf files. *Tettigettalna*'s RAD-Seq data analysis was concluded with this ABBA/BABA test.

Chapter 3

Results

3.1 The RAD-Seq data treatment

3.1.1 The sequence treatment with Ipyrad

The RAD-Seq data was cleaned and filtered with the Ipyrad in order to obtain quality data to be analysed and assure that results from this subsequent analysis aren´t poor and difficult to interpret. Our filtered results were good, which was verified by the quality and unambiguous results of the analysis that will be presented below.

3.1.2 The filtering step with VCFtools program

The filtering step with the VCFtools command filtered the SNPs of the 12 assembly_name.vcf files in terms of two criteria: minimum allele frequency (maf) and maximum missing data. The outputs generated were VCF files named *assembly_name*.recode.vcf. On Table *3.1* it is possible to verify the number of SNPs before and after the filtering with the VCFtools in terms of those two criteria.

Table 3.1 Results obtained in terms of number of SNPs for each assembly VCF file, after filtering the original VCF file with the VCFtools filtering program in terms of minimum allele frequency and maximum missing data

Assembly files	**Nº of SNPs**
params.vcf	133489
params.recode.vcf	8318
params1.vcf	142787
params1.recode.vcf	8935
params2.vcf	135181
params2.recode.vcf	6495
params3.vcf	119601
params3.recode.vcf	7475
params4.vcf	112909
params4.recode.vcf	8342
params5.vcf	125182
params5.recode.vcf	8968
params6.vcf	122203
params6.recode.vcf	6499
params7.vcf	95462
params7.recode.vcf	8354
params8.vcf	108035
params8.recode.vcf	8980
params9.vcf	106648
params9.recode.vcf	6499
params10.vcf	98374
params10.recode.vcf	12959
params11.vcf	113618
params11.recode.vcf	12049

The results obtained with the filtering provided by the VCFtools, show that only 5-13% SNPs were kept (more than 90% of the original number of SNPs were excluded), when we only keep the SNPs that have a minor allele frequency bigger than 5% and with only a maximum of 20% of missing data. These results were surprising because we dind´t expect such a large reduction of the number of SNPs. However, these reductions have allowed the exclusion of irrelevant SNPs, therefore assuring the quality of the results that were obtained *à posteriori*.

3.1.3 The filtering step with vcf_parser.py code

This filtering step of the *assembly_name*.recode.vcf files with the python code vcf_parser.py from https://github.com/CoBiG2/RAD_Tools, as of commit 9a5bc1bddc, allowed filtering the invariable sites and singletons and choosing the center SNP for each locus.

For each type of filter, an output VCF file was generated, and further compared, in terms of number of SNPs, to the input VCF. The results were the following presented for the 12 assemblies, in the following Table *3.2*, from the assembly named **params** to the assembly named **params11**.

Table 3.2 Results obtained in terms of number of SNPs for each assembly VCF file, after filtering the recode VCF file with the vcf_parser.py filtering code

Assembly files	Nº of SNPs
params.recode.vcf	8318
paramsCenterSNP.vcf	**2817**
params_NoInv.vcf	8318
params_NoSing.vcf	8318
params1.recode.vcf	8935
params1CenterSNP.vcf	**3083**
params1_NoInv.vcf	8935
params1_NoSing.vcf	8935
params2.recode.vcf	6495
params2CenterSNP.vcf	**2786**
params2_NoInv.vcf	6495
params2_NoSing.vcf	6495
params3.recode.vcf	7475
params3CenterSNP.vcf	**2523**
params3_NoInv.vcf	7475
params3_NoSing.vcf	7475
params4.recode.vcf	8342
params4CenterSNP.vcf	**2821**
params4_NoInv.vcf	8342
params4_NoSing.vcf	8342
params5.recode.vcf	8968
params5CenterSNP.vcf	**3089**
params5_NoInv.vcf	8968
params5_NoSing.vcf	8968
params6.recode.vcf	6499
params6CenterSNP.vcf	**2787**
params6_NoInv.vcf	6499
params6_NoSing.vcf	6499
params7.recode.vcf	8354
params7CenterSNP.vcf	**2824**
params7_NoInv.vcf	8354
params7_NoSing.vcf	8354
params8.recode.vcf	8980
params8CenterSNP.vcf	**3091**
params8_NoInv.vcf	8980
params8_NoSing.vcf	8980
params9.recode.vcf	6499
params9CenterSNP.vcf	**2787**
params9_NoInv.vcf	6499
params9_NoSing.vcf	6499
params10.recode.vcf	12059
params10CenterSNP.vcf	**3763**
params10_NoInv.vcf	12059
params10_NoSing.vcf	12059
params11.recode.vcf	12049
params11CenterSNP.vcf	**3762**
params11_NoInv.vcf	12049
params11_NoSing.vcf	12049

After obtaining the number of SNPs in each document taking in consideration the filtering process on each one, two graphics Figure ***3.1*** and Figure 3.2 were produced to analyse, on an easier way, the number of SNPs on the assemblies and the average of missing data associated with the assemblies, respectively.

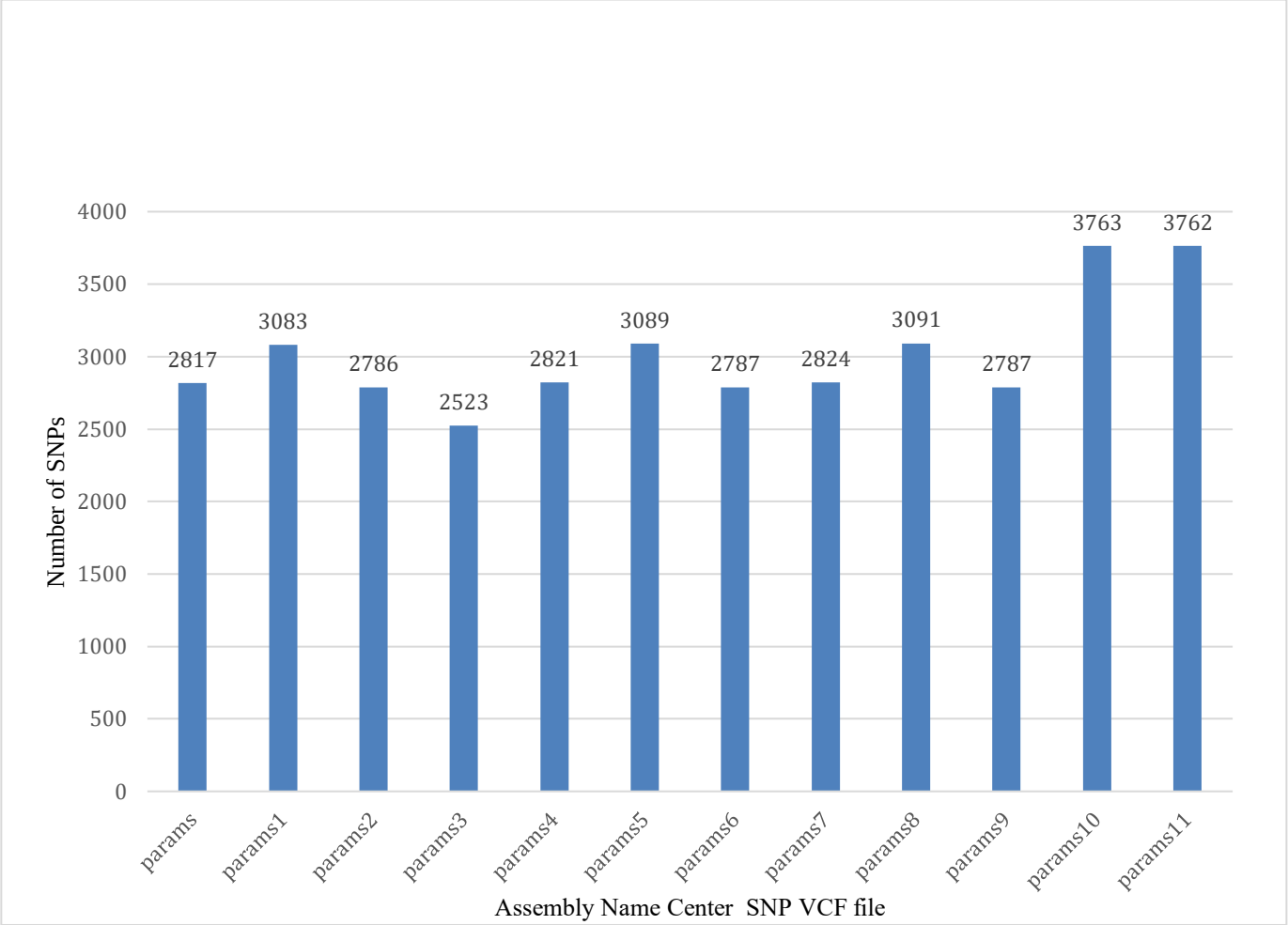


Figure 3.1 Number of SNPs in each Assembly Name Center VCF file

Graphic with the number of SNPs obtained for each assembly name VCF file where the Center SNP for each locus was chosen. Assemblies params to params9 correspond to the assemblies where *Tettigettalna aneabi* were included vs params10 and params11 where they were excluded.

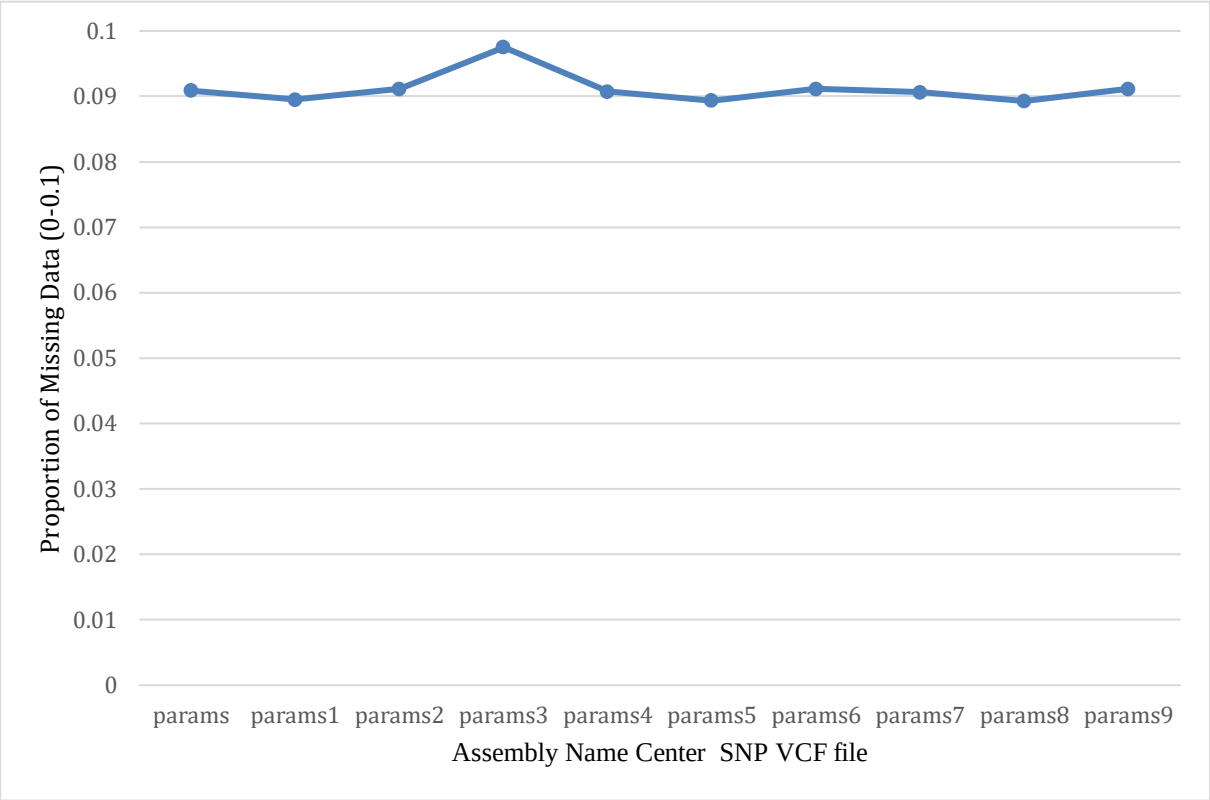


Figure 3.2 Average Missing Data

Graphic with the average of missing data for each assembly name VCF file where the Center SNP for each locus was chosen. The assemblies here are only the ones where *Tettigettalna aneabi* was included.

This filtering step with vcf_parser.py allowed filtering the invariable sites and singletons and choosing the center SNP for each locus. The results obtained (Table *3.2*) showed that there was no difference in terms of number of SNPs between the recode.vcf files, the VCF files where the singletons were removed and the VCF files where the invariable sites were removed, unless the VCF files where the center SNP for each locus were chosen. Once again, a reduction of the number of SNPs were seen in those Center SNP files, not in a big scale compared with the previous filtering step with the VCFtools. In this case, in the CenterSNP.vcf files we kept 30-40% of the number of SNPs from the recode.vcf files. These results showed that we didn´t have invariable sites neither singletons since the number of SNPs didn´t change. The case for the SNP Center VCF file, this reduction is high but expected, since for each *locus* we were chosing the Center SNP and this justifies the reduction of the number of SNPs.

Nevertheless, when comparing between files, the number of SNPs (Figure ***3.1***) and the average missing data (Figure 3.2), are very similar (between 2500 and 3700 SNPs). Moreover, the average missing data is low, since it is below 0.1, an accepted value for missing data. The assembly files that were chosen for the following analysis were the assembly params8CenterSNP.vcf and the assembly params10CenterSNP.vcf (without *Tettigettalna aneabi* specimens) where we have the biggest number of SNPs to improve the viability of the analysis of the RAD-seq data.

3.2 The Principal Component Analysis of the RAD-Seq data

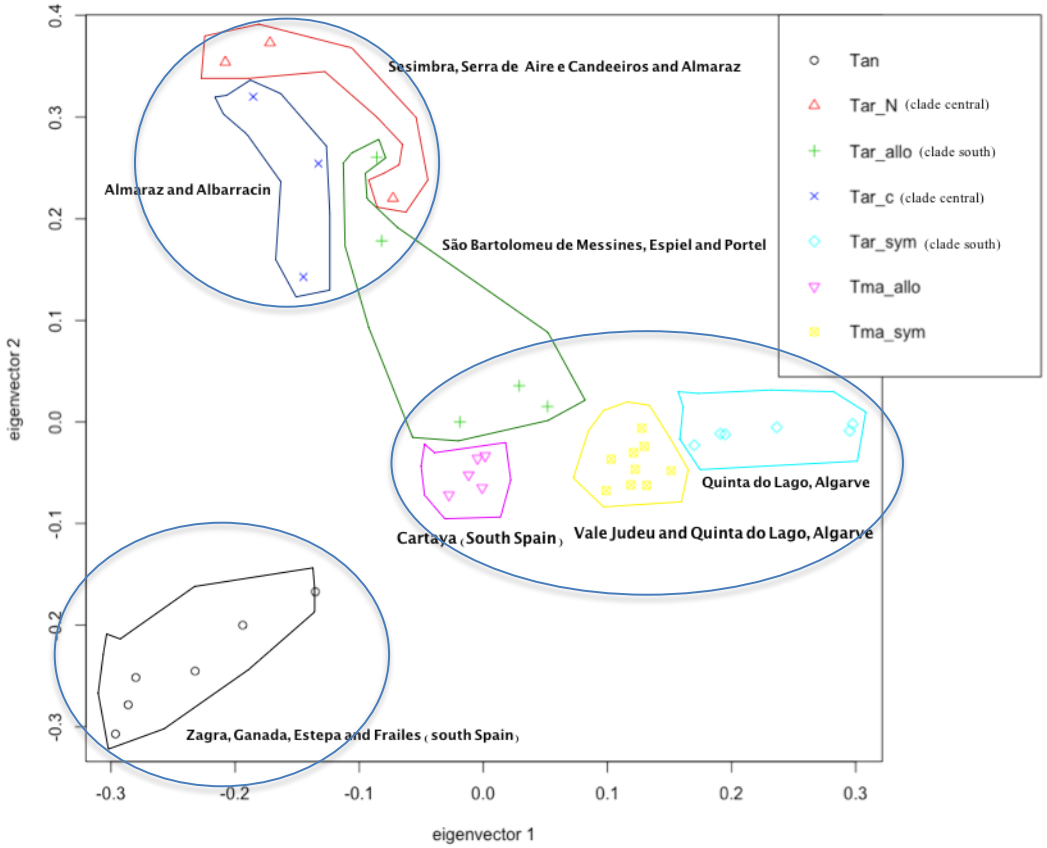
A Principal Component Analysis (PCA) was performed to understand if there were clustering groups of individuals, on the basis of the RAD-Seq genomic data, in conformity with the species/populations geographical distribution patterns.

The PCA results between the assemblies with *Tettigettalna aneabi* individuals were almost the same with little variances of the values of the positions of the individuals in the PCAs, so it was decided to present only the PCAs of one assembly with *Tettigettalna aneabi* and other without *Tettigettalna aneabi*.

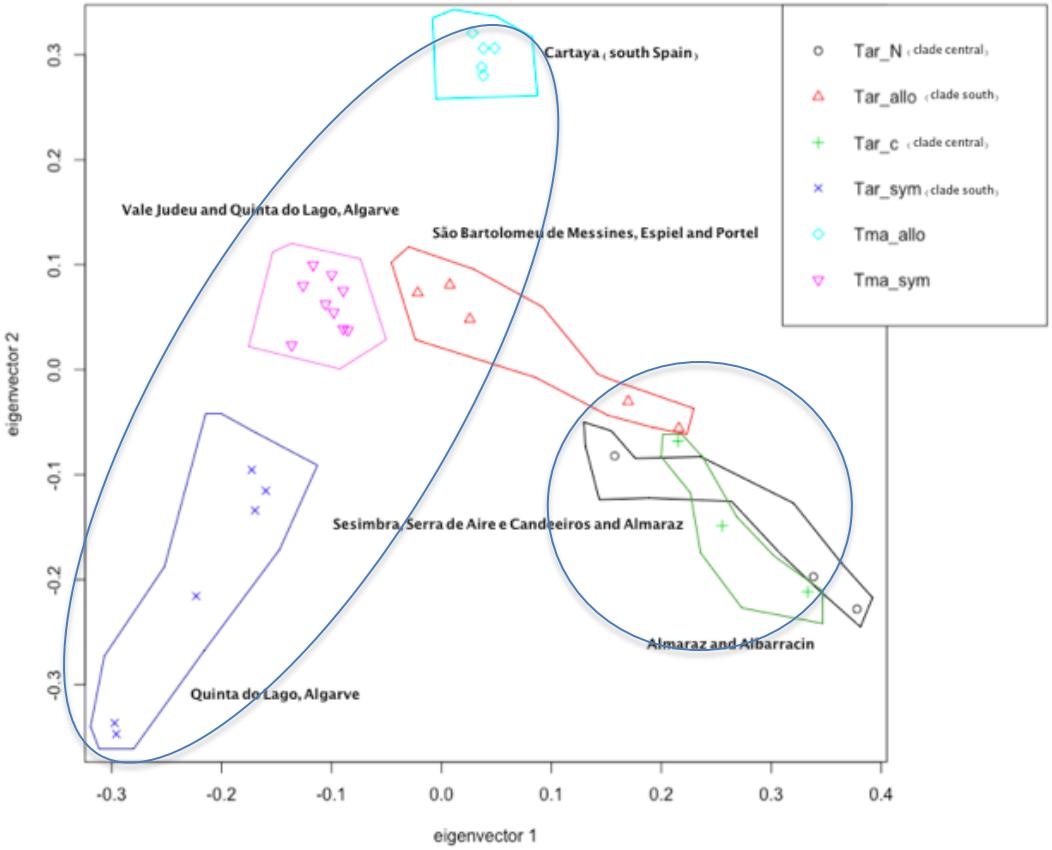
Here, we present four PCAs of the assemblies params8 (on Figure 3.3 (A and C)) and params10 (on Figure 3.3 (B and D)) using the input files params8.recode.vcf, params8CenterSNP.vcf, params10.recode.vcf and params10CenterSNP.vcf.

For a better understanding of the clustering results, we confront de results with the populations locations presented on Figure 2.1.

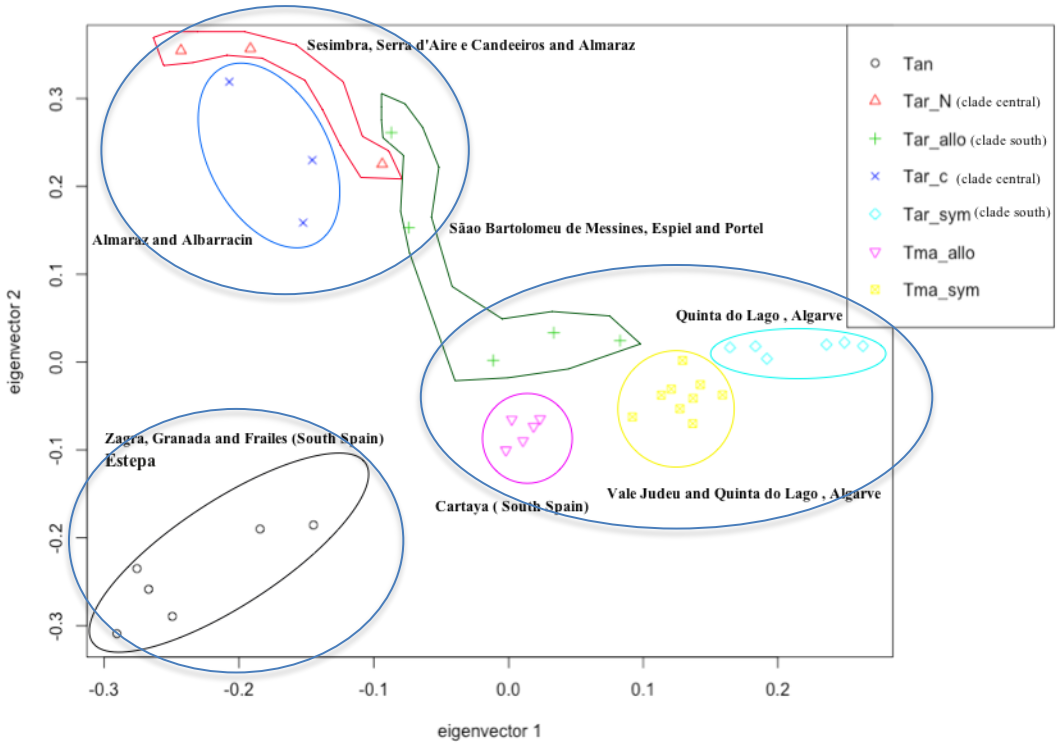
The main goal was to present the principal clustering groups that are formed with *Tettigettalna aneabi* specimens (on Figure 3.3 (A and C)) and without *Tettigettalna aneabi* specimens (on Figure 3.3 (B and D)) and to compare the results between the assemblies where the Centre SNP of each *locus* where chosen versus the ones where this selection step was not done.


Sesimbra, Serra de Aire e Candeeiros and Almaraz
Almaraz and Albarracin
São Bartolomeu de Messines, Espiel and Portel
Quinta do Lago, Algarve
Cartaya (South Spain)
Vale Judeu and Quinta do Lago, Algarve
Zagra, Ganada, Estepa and Frailes (south Spain)
Tan
Tar_N (clade central)
Tar_allo (clade south)
Tar_c (clade central)
Tar_sym (clade south)
Tma_allo
Tma_sym
eigenvector 1
eigenvector 2

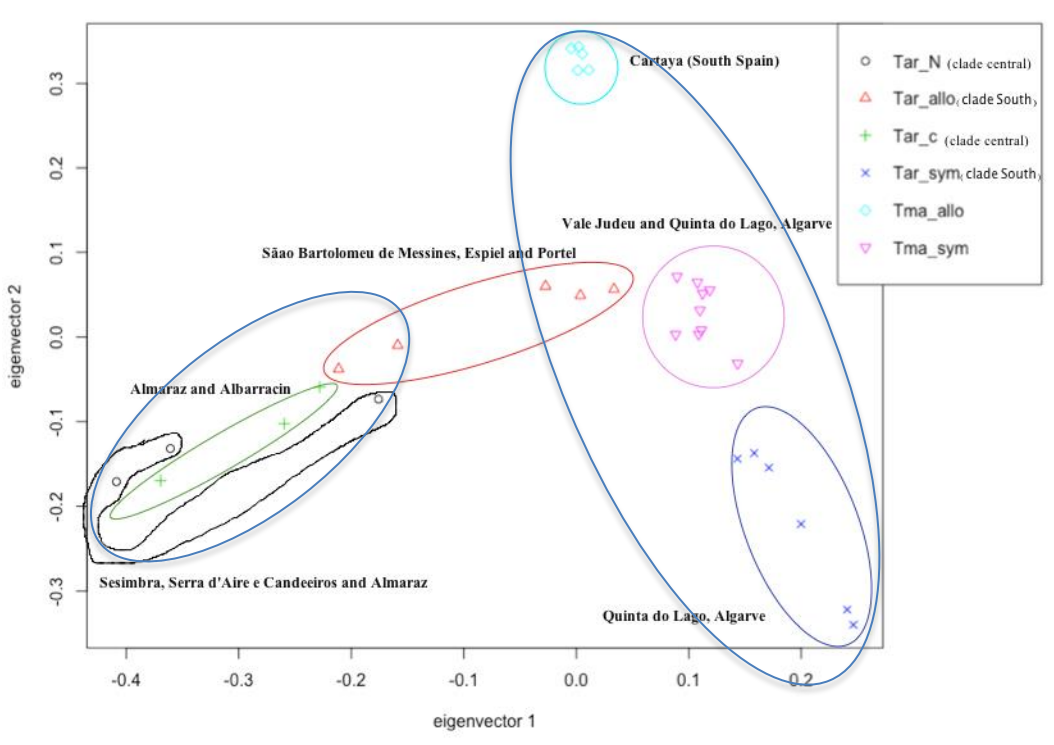
A


Cartaya (south Spain)
Vale Judeu and Quinta do Lago, Algarve
São Bartolomeu de Messines, Espiel and Portel
Sesimbra, Serra de Aire e Candeeiros and Almaraz
Almaraz and Albarracin
Quinta do Lago, Algarve
Tar_N (clade central)
Tar_allo (clade south)
Tar_c (clade central)
Tar_sym (clade south)
Tma_allo
Tma_sym
eigenvector 1
eigenvector 2

B



C



D

Figure 3.3 PCA results from RAD-Seq data treated after Ipyrad and filtering with vcf_parser.py code

Figure A and B correspond to the assemblies without choosing SNPs. Figure C and D correspond to the assemblies where the Center SNP was chosen for each locus. Figure A and C correspond to the assemblies where *Tettigettalna aneabi* was included (assemblies params8.recode.vcf and params8CenterSNP.vcf, respectively) and figure B and D correspond to the assemblies where *Tettigettalna aneabi* was excluded (assemblies params10.recode.vcf and params10CenterSNP.vcf, respectively). Tan - *Tettigettalna aneabi*; Tar_N (clade central) - *Tettigettalna argentata* North from clade central Iberia; Tar_allo (clade south) - *Tettigettalna argentata* in allopatry from clade south; Tar_c (clade central) - *Tettigettalna argentata* from clade central Iberia; Tar_sym (clade south)- *Tettigettalna argentata* in sympatry from clade south; Tma_allo - *Tettigettalna mariae* in allopatry; Tma_sym - *Tettigettalna mariae* in sympatry. For more information see Table ***6.1***

Results from PCAs explained variances for eigenvector 1 that maximizes the variance: for assembly 8 on Figure 3.3 (A and C) without and with the choose of Centre SNP and including *Tettigettalna aneabi*, were 7,98% and 8.9%, respectively; and for assembly 10 on Figure 3.3 (B and D), without and with the choose of Centre SNP and excluding *Tettigettalna aneabi*, were 8.11% and 9.16%, respectively. Those values are low, but expected, since as the variables are the SNPs and we have about thousands, the variation will be dispersed.

Results obtained on Figure 3.3 show that, although without high eigenvalues explaining the variance of the data sample by its eigenvector 1, there are well defined clustering groups of individuals and the relatedness between the cluster populations given by these data are generally in agreenment with the geographical distribution patterns of the *Tettigettalna* species and populations in analysis (see Figure 2.1).

Individuals from *T. argentata* North and *T. argentata* Centre, in fact are not two different clusters because they are a little overlapped on Figure 3.3. In fact, *T. argentata* North correspond to centre Portugal Figure 2.1), and *T. argentata* Centre correspond to centre Spain (Figure 2.1). They look like a unique cluster with two different populations (on Figure 3.3), what makes sense because they together correspond central group of Iberian Peninsula (Figure 2.1).

In terms of differences between the results using the Centre SNP for each locus Figure 3.3 (C and D) and the results where that selection of SNPs was not done (Figure 3.3 (A and B)), it is clearly evident that it makes little diferences in the relatedness distribution patterns of the clusters of individuals.

Comparing the outcomes when *Tettigettalna aneabi* were included (assembly 8) on Figure 3.3 (A and C) with those when *Tettigettalna aneabi* individuals were excluded (assembly 10) on Figure 3.3 (B and D), the results were almost the same. In fact, figures B and D, since those are excluding the *Tettigettalna aneabi* individuals, make a zoom-like of the clusters pattern of the *Tettigettalna argentata* and *Tettigettalna mariae* individuals giving a more detailed and clear vision of the genetic proximity between the *T. argentata* and *T. mariae*' cluster groups of individuals.

Analysing the PCA results with more detail, the following evidences were obtained:

On all PCAs from Figure 3.3, there is a well-defined separation of the clusters which are not being overlapped. There are two principal clusters (blue circles) where *T. aneabi* is excluded on Figure 3.3 (B and D); and three principal clusters (also defined by blue circles) on Figure 3.3 (A and C) where *T. aneabi* is included.
On Figure 3.3 (A and C), including the *Tettigettalna* aneabi, we can easily see the *Tettigettalna aneabi* individuals are the most distant from all the rest of the *Tettigettalna*.
On Figure 3.3, the separation of the clusters is consistent with the geographical pattern of the individuals (see also Figure 2.1). There is a separation between individuals from the south of Iberian Peninsula (*T. argentanta* sympatric clade south, *T. mariae* sympatric, *T. argentata* allopatric clade south) and Central Peninsula Iberia (*T. argentata* North and *T. argentata* Centre).

Moreover, on Figure 3.3, *Tettigettalna argentata* individuals in sympatry (*T. argentata* sympatric clade South), from Algarve, have the biggest distance to their conspecific ones: the *Tettigettalna argentata* from North/Centre (central Peninsula Iberia) and *Tettigettalna argentata* in allopatry *(T. argentata* clade South).

On Figure 3.3 (C and D) from the *Tettigettalna* specimens from complex of South Iberia (*Tettigettalna mariae* in sympatry, *Tettigettalna mariae* in allopatry and *Tettigettalna argentata* in sympatry clade south) there are more genetic proximity between the congeneric ones (*Tettigettalna mariae* in sympatry and *Tettigettalna argentata* in sympatry clade south) rather than with its conspecific specimens of *Tettigettalna mariae* in allopatry (Tma_allo) and *Tettigettalna argentata* (in allopatry and central Iberia) (Tar_allo (clade south), Tar_N (clade central), Tar_C (clade central)), respectively.

Hence, the results have shown that our RAD-Seq data have allowed an evident well definition of clustering groups of individuals, what showed us that this type of data are powerful.

3.3 Analysis of the RAD-Seq data with MavericK program

The two matrixes obtained for the two assemblies params8 and params10 are present on Figure 3.4 and Figure 3.5, respectively.

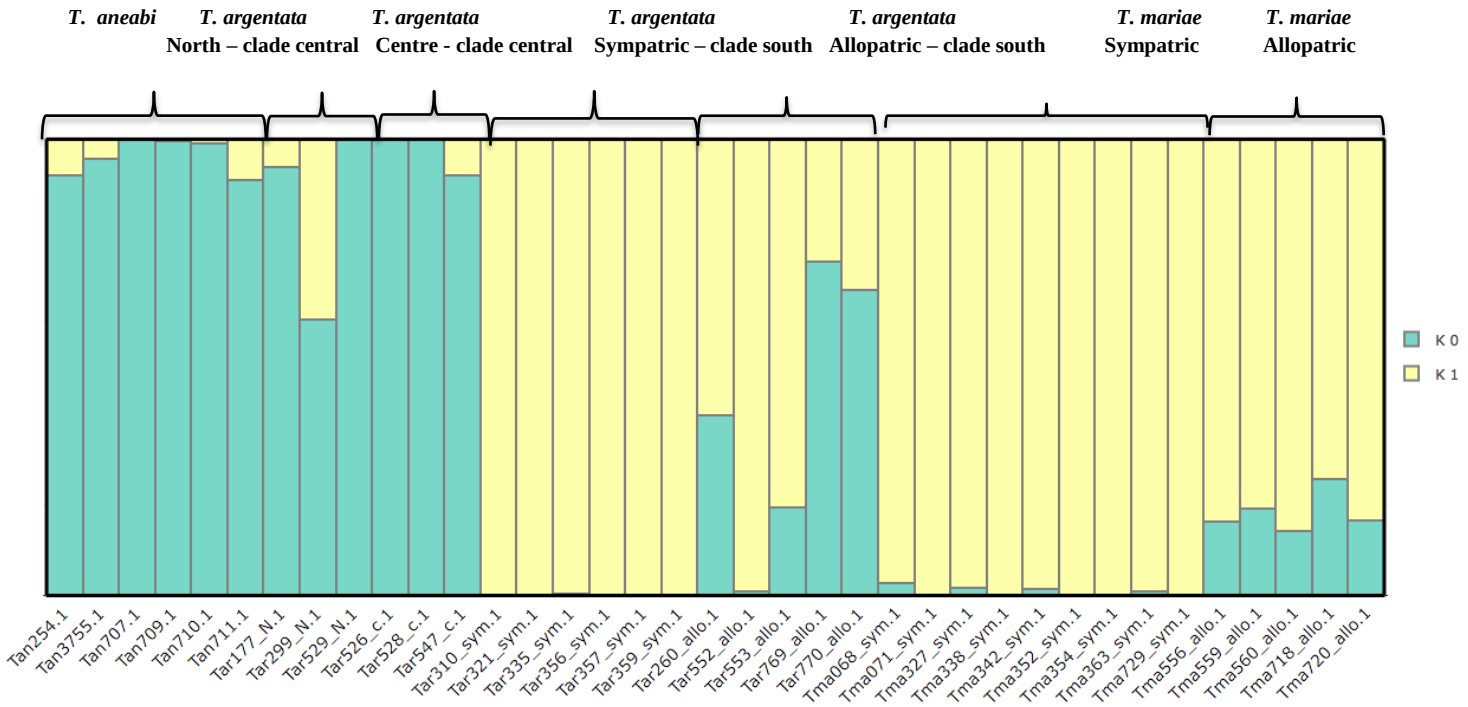


Figure 3.4 MavericK clustering plot result for assembly params 8 (including the *Tettigettalna* aneabi)

This result was generated by the MavericK program in the final run, where the best K estimated was a K = 2**.** The details of the individuals of *Tettigettalna* cicadas under analysis can be consulted on Table ***6.1***.

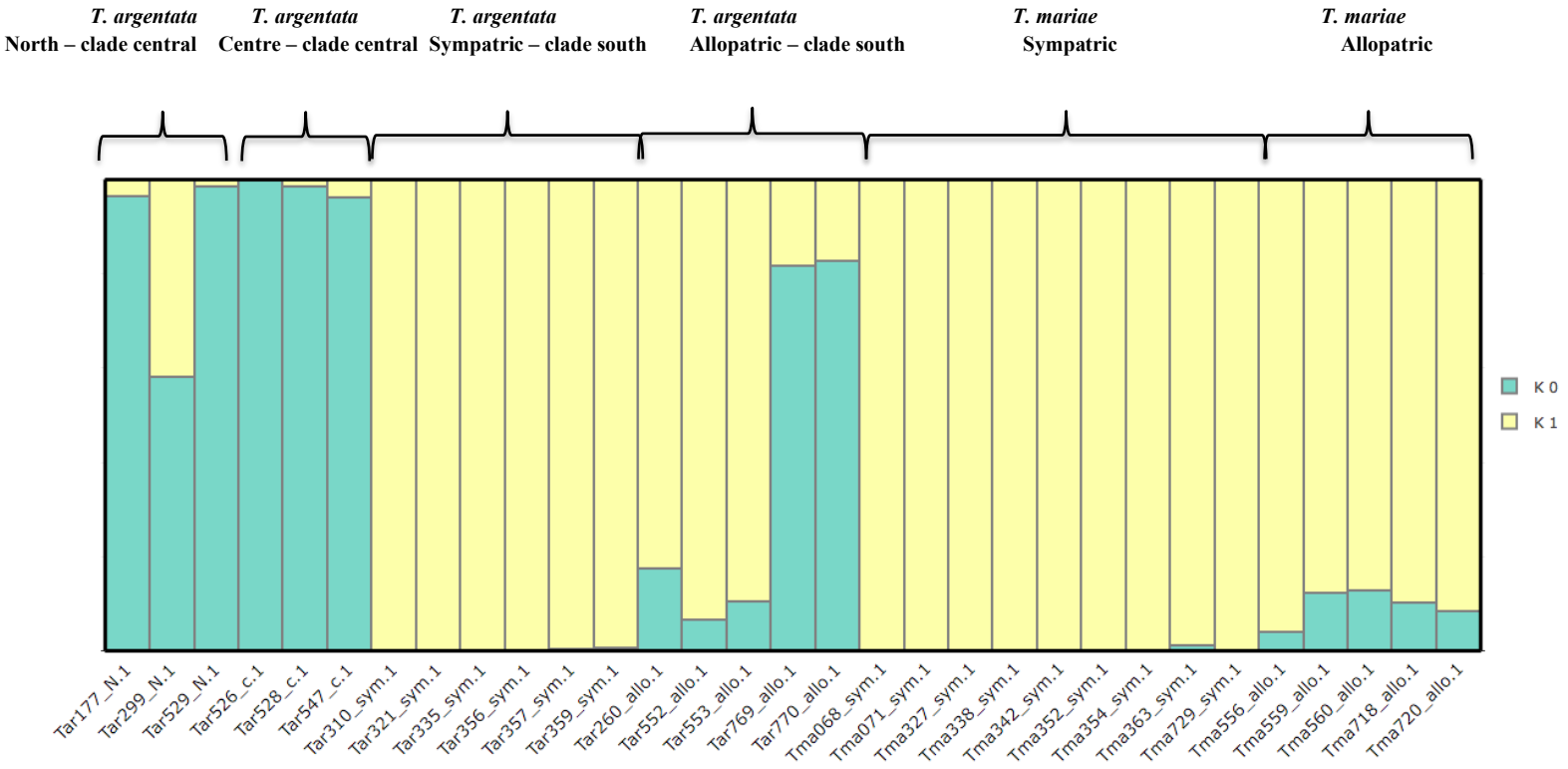


Figure 3.5 MavericK clustering plot result for assembly params 10 (excluding *Tettigettalna aneabi*)

This result was generated by the MavericK program in the final run, where the best K estimated was a K = 2. The details of the individuals of *Tettigettalna* cicadas under analysis can be consulted on Table ***6.1***.

As can be observed in Figure 3.4 and Figure 3.5, two results were obtained for the assemblies chosen for the final MavericK run - the params8 and params10 assemblies.

The inclusion or the absence of *Tettigettalna aneabi* individuals didn't affect the number of demes/clusters (K demes) that was estimated as K= 2 and considered the best K. This result assume that individuals belong to one deme 1 or to the other deme 2, or otherwise, have an admixture of genetic information from both demes.

Addicionally, when comparing the two results (Figure 3.4), the genetic patterns of attribution to the two clusters, estimated by the program as the best K, are very similar.

In both results the individuals belonging to *Tettigettalna mariae* in sympatry (individuals Tma***_sym) and *Tettigettalna argentata* in sympatry (*T. argentata* from clade South) (individuals Tar***_sym) have all a signal K1 (in yellow) (almost 100%), being these the ones from the South (Vale Judeu and Quinta do Lago in Algarve) (see Table ***6.1*** and Figure 2.1 to see the region, country and species name of the mentioned individuals), therefore according to the geographical proximity of these two species.

On the other hand, *Tettigettalna mariae* individuals in allopatry from Cartaya (Tma***_allo) (see Table ***6.1*** and also Figure 2.1 to see the region, country and species name of the mentioned individuals), in both Figure 3.4 and Figure 3.5, although having mostly (about 80%) a signal K1, all of them have a signal K0 what means they have some admixture of genetic information from the both demes K0 and K1 demes estimated.

In what refers to the *Tettigettalna argentata* in allopatry (*T. argentata* from clade south), in both on Figure 3.4 and Figure 3.5, from São Bartolmeu de Messines, Espiel and Portel (Tar***_allo) (see Table ***6.1*** to see the region, country and species name of the mentioned individuals and also Figure 2.1) have all of them have genetic information from both demes K0 and K1, although varying in some degree.

The *Tettigettalna argentata* from North and Centre, (Tar***_N and Tar***_c), the *Tettigettalna argentanta* from Central Iberia: *T. argentata* from centre of Portugal and centre of Spain, respectively, like the *Tettigettalna aneabi* individuals (Tan***) from South Spain, in Figure 3.4, have almost 100% signal K0 (in green) with only a small signal from the other K1 deme (see Table ***6.1*** and Figure 2.1 to see the region, country and species name of the mentioned individuals), giving evidence that they are close genetically. This is a result that was not expected because they are from different species and its is not according the geographic proximity of these two species that are really distant (see Figure 2.1).

3.4 ABBA/BABA test

When running "Dstat.R" R script (Seabra et al., 2019) to obtain the D-statistics, the results obtained for the three GT formatted input files "params12filtered.recode.GT", "params13filtered.recode.GT" and "params14filtered.recode.GT" are presented on Table *3.3* and Table *3.4*.

Table 3.3 Results obtained for D-statistic, abba, baba and p-value for the three input GT files

Input GT File	$D	$abba	$baba	pvals
params12filtered.recode.GT	0.147922	513.8528	381.422	2.874072e-18
params13filtered.recode.GT	0.147922	513.8528	381.422	2.874072e-18
params14filtered.recode.GT	0.1483996	514.6921	381.672	4.32408e-18

Table 3.4 Results obtained for D-statistic, deviation and z-value for the three input GT files

Input GT File	D	sd	z
params12filtered.recode.GT	0.147922	0.01712565	8.63745
params13filtered.recode.GT	0.147922	0.01712565	8.63745
params14filtered.recode.GT	0.1483996	0.01727456	8.590646

The values obtained for the introgession test - ABBA/BABA test - were crucial to answer, in a very clearly way, one of the main questions of this work. We were interested in understanding if haplotypes shared by *Tettigettalna argentata* and *Tettigettalna mariae* individuals from sympatric populations in Algarve, was due to incomplete lineage sorting or introgression. Therefore, we considered the null hypothesis (D-statistic equal to zero) as no gene flow (no introgression) giving evidence of incomplete lineage sorting.

We have used different three assemblies as input files, to see if the results were homogenous when using the parameters in the Ipyrad runs that maximize the number of SNPs and to give more support to the value obtained for this statistic, because we were taking in consideration three D-statistics from different files instead of only one.

The D-statistic obtained was in all cases different than zero, which made us exclude the null hypothesis that there was no introgression. Since the D-statistic was bigger than zero, this meant there was an excess of sites with the ABBA allele sharing pattern, resulting in a significant D-statistic value. Being bigger than zero, that positive value gave us the information that a phenomenom of introgression may have occurred between the two populations *Tettigettalna argentata* in sympatry from clade South (P2) with *Tettigettalna mariae* in sympatry (P3). The statistic was approximate for all the three input GT files, confirming that the results are homogenous and giving more support to the value obtained for D-statistic.

The z-value is much bigger than 1.96 (in a normal distribution), meaning our p-value is significant. This low p-value suggests our sample provided enough and strong evidence that we can reject the null hypothesis for the entire population. Even taking in consideration the deviation value (sd), the D-statistic

continue being strongly different than zero, which gave a lot of support to the rejection of the null hypothesis.

Chapter 4

Discussion and Conclusions

The *Tettigettalna* cicadas are a closely-related and recent diverged group of species that are morphologically very similar but whose acoustic signals generally allow us to distinguish them (Mendes et al., 2014).

Previous studies on the *Tettigettalna argentata* and *Tettigettalna mariae* known to occur in sympatry or close parapatry, in Algarve (Portugal), showed no consistent genetic differences since they share some haplotypes (Nunes et al., 2014b).
Therefore, this work emerged as an opportunity to better investigate the history of the Mediterranean *Tettigettalna* cicadas, on the basis of RAD-seq data. A particular effort was performed to understand if RAD-seq data, gives support the geographical distribution patterns of the *Tettigettalna's* complex under study and if this type of data is able to demystify the question if the haplotype sharing between the pair of sibling species is due to a phenomenon of introgression or incomplete lineage sorting.

To achieve our goals, we used various bioinformatics tools to better understand the geographical patterns of distribution of the closely related species.

We were able to show that RAD-Seq data was well-cleaned by the Ipyrad and the other scripts (used to erase error). Principal Component Analysis revealed well-defined clusters, whose dispersion/proximity reflects the geographical distribution of the *Tettigettalna* species/populations. Therefore, these results with data from Next Generation Sequencing methods support the use of NGS data for other researches focused on better understanding the genetic proximity between this group of closely-related species.

In fact, detailed cluster analyses revealed that there was a well-defined separation of *Tettigettalna aneabi* from the other species (*Tettigettalna mariae* and *Tettigettalna argentata*), which is contrarily to the evidence provided by the MavericK program. Furthemore, *T. argentata* in sympatry on Algarve (south clade) have the biggest distance from their conspecific ones: the *Tettigettalna argentata* from North/Centre (central Iberia) and *Tettigettalna argentata* in allopatry (south clade); on the other hand, this group revealed the smaller distance from the congeneric, *Tettigettalna mariae* in sympatry (Algarve) which was validated by the ABBA/BABA test that appointed for introgression between those.

Results from the MavericK program for inferring the population structure on the basis of the genetic information show that the *Tettigettalna argentata* from the North and Centre (Central Iberia) are genetically very close to the *Tettigettalna aneabi* specimens. This result is different from what Nunes et al., (2014b) have showed, on the basis of COI sequences. Their work revealed that *T. aneabi* represented the sister taxon of *T. mariae* and *T. argentata* clade south. Our new results also raise new questions, namely the possibility of introgression between both referred species.

The *Tettigettalna argentata* in sympatry (clade south) and *mariae* in sympatry have exactly the same percentage (100%) of genetic attribution to deme K1 (Figure 3.4 and Figure 3.5), in agreement with the previous knowledge of shared haplotype between them and the possibility of introgression.

In what refers to *Tettigettalna argentata* allopatric (clade south) and *mariae* individuals in allopatry, both have an admixture of atribution to both demes (Figure 3.4 and Figure 3.5). *Tettigettalna argentata* allopatric from clade South are between the Central Iberia *Tettigettalna argentata* and the *Tettigettalna argentata* sympatric (clade south) which explained the admixture of information (see Figure 2.1**Error! Reference source not found.**). The last-mentioned cicadas, the *Tettigettalna mariae* in allopatry, from Cartaya, have mostly signal to deme K1, being attribution to the same cluster as the sympatric *T. argentata* from clade south and *T. mariae*, which makes sense since they are geographically closer to each other (see Figure 2.1).

The previously study performed by Nunes et al., (2014b) on the basis of mitochondrial COI sequences intended to analyse the relatedness among *Tettigettalna* species. Their results allowed the separation of *Tettigettalna argentata* in northern and southern clades, and our study are in conformity with them. We have also that separation, but now we have also the populations of *T. argentata* from Central Spain (Tar_C)

Nunes et al., (2014b) noticed that the southern clade was genetically inseparable from *Tettigettalna mariae*, sharing with it its most common haplotype (forming a species complex). However, it was not possible for them to unambiguously genetically discriminate *T. mariae* in sympatry specimens from *T. argentata* in sympatry (clade South) on the basis of COI sequences analysis, a single locus approach. Our results, contrarily, were positively able to explain the share os haplotypes through a ABBA/BABA test.

Regarding the haplotype sharing between the pair of sibling species studied by Nunes et al., (2014b) with our ABBA/BABA test using the RAD-seq data (data produced by a multilocus approach) we obtained a D – statistic bigger than zero. The statistic was approximate for all the three input GT files, confirming that the results are homogenous and giving more support to the value obtained for D-statistic. That positive value indicates that it may have been a phenomenom of introgression and not incomplete lineage sorting that occurred between *Tettigettalna argentata* sympatric (clade south) and *Tettigettalna mariae* sympatric, on Algarve. Although we were not able to know in each way the gene flow occurred, our results reveal that the females from *T. argentata* and *T. mariae*, both sympatric (clade south), had past episodes (and will envetually have in the future) of reproduction with the males of the opposite species when in sympatry, leading to the occurrence of hybrids. The introgression phenomenom explains the genetic proximity obtained on the PCA results between the congeneric ones (*T. argentata* sympatric clade south and *T. mariae* sympatric) rather than its conspecific ones.

The ABBA/BABA test with RAD-Seq data as in fact been used with similar results for other organisms, such as bumblebees. Seabra et al., 2019 have also used this test and their results have detected the phenomenom of introgression between the populations, allowing the authors to understand the reason for the studied genetic proximities.

Therefore, our results not only give a new insight for the reason of the studied genetic proximity but are also a key piece to continue our understanding of the processes of divergence and speciation of this genus. Present results on the patterns of distribution of species and populations of the *Tettigettalna* under

study don´t exclude the scenario of dispersal of *Tettigettalna argentata* occuring from eastern Europe and going to North and Centre Iberia and migrating until de South of the peninsula. Afterwards, in South of Portugal, *T. argentata* specimens may have found the *Tettigettalna mariae* individuals and by mistake the reproduction between them occurred, leading to gene flow between the species. Previous studies, (Costa, 2017), performed with *T. argentata* also didn´t bring any final explanation.

In conclusion, present work has shown that the RAD-seq data (multilocus approach) produced by the RAD-Sequencing technologies (one of the Next Generation Sequencing technologies) allowed a better understanding on the closely related group of *Tettigettalna* cicadas. This reinforces the power of the Next-Generation Sequencing technologies and its viability to speed up the investigation of complex relationships between recently diverged species, contrarly to single locus approaches, that namy times are insufficient to determine even simple species limits.

However, new questions arised. As such, it would be interesting to study in each way the phenomenom of introgression could have occurred, we mean, if it was from *T. argentata* sympatric (clade south) to *T. mariae* sympatric; if it was from *T. mariae* sympatric to *T. argentata* sympatric (clade south), or if it was in both directions. Moreover, other approaches like testing for introgression using the ABBA/BABA test could give more information to explain the close relationship between the *Tettigettalna aneabi* from South Spain and *Tettigettalna argentata* North (Tar_N) and *Tettigettalna argentata* Centre (Tar_c) (the Central Iberia *T. argentata* populations).

Lastely, understanding if the dispersal of *Tettigettalna argentata* occurrred from Africa to all the rest of the Europe, or the opposite is a problem that remains to be answered and that would give new evidences of the evolutionary history of the *Tettigettalna* group of species. The history of the *Tettigettalna* cicadas is just in the beginning…

Chapter 5

References

Andrews, K. R., Good, J. M., Miller, M. R., Luikart, G., & Hohenlohe, P. A. (2016). Harnessing the power of RADseq for ecological and evolutionary genomics. *National Review of Genetics*, *17*(2), 81–92. https://doi.org/10.1038/nrg.2015.28.Harnessing

Ashton, P. M., Nair, S., Dallman, T., Rubino, S., Rabsch, W., Mwaigwisya, S., Wain, J., O'Grady, J. (2014). MinION nanopore sequencing identifies the position and structure of a bacterial antibiotic resistance island. *Nature Biotechnology*, *33*, 296. Retrieved from http://dx.doi.org/10.1038/nbt.3103

Asmann, Y. W., Wallace, M. B., & Thompson, E. A. (2008). Transcriptome Profiling Using Next-Generation Sequencing. *Gastroenterology*, *135*(5), 1466–1468. https://doi.org/https://doi.org/10.1053/j.gastro.2008.09.042

Baird, N. A., Etter, P. D., Atwood, T. S., Currey, M. C., Shiver, A. L., Lewis, Z. A., Selker, E. U., Cresko, W. A., Johnson, E. A. (2008). Rapid SNP discovery and genetic mapping using sequenced RAD markers. *PloS One*, *3*(10), e3376.

Balasubramanian, S. (2011). Sequencing nucleic acids: from chemistry to medicine. *Chemical Communications (Cambridge, England)*, *47*(26), 7281–7286. https://doi.org/10.1039/c1cc11078k

Ballard, J. W. O., & Whitlock, M. C. (2004). The incomplete natural history of mitochondria. *Molecular Ecology*, *13*(4), 729–744.

Barba, M., Czosnek, H., & Hadidi, A. (2013). Historical perspective, development and applications of next-generation sequencing in plant virology. *Viruses*, *6*(1), 106–136. https://doi.org/10.3390/v6010106

Bayley, H. (2006). Sequencing single molecules of DNA. *Current Opinion in Chemical Biology*, *10*(6), 628–637. https://doi.org/https://doi.org/10.1016/j.cbpa.2006.10.040

Bentley, D. R., Balasubramanian, S., Swerdlow, H. P., Smith, G. P., Milton, J., Brown, C. G., Hall, K. P., Evers, D. J., Barnes, C. L., Bignell, H. R., Boutell, J. M., Bryant, J., Carter, R. J., Cheetham, R. K., Cox, A. J., Ellis, D. J., Flatbush, M. R., Gormley, N. A., Humphray, S. J., Irving, L. J., Karbelashvili, M. S., Kirk, S. M., Li, H., Liu, X., Maisinger, K. S., Murray, L. J., Obradovic, B., Ost, T., Parkinson, M. L., Pratt, M. R., Rasolonjatovo, I. M., J., Reed, M. T., Rigatti, R., Rodighiero, C., Ross, M. T., Sabot, A., Sankar, S. V., Scally, A., Schroth, G. P., Smith, M. E., Smith, V. P., Spiridou, A., Torrance, P. E., Tzonev, S. S., Vermas, E. H., Walter, K., Wu, X., Zhang, L., Alam, M. D., Anastasi, C., Aniebo, I. C., Bailey, D. M. D., Bancarz, I. R., Banerjee, S., Barbour, S. G., Baybayan, P. A., Benoit, V. A., Benson, K. F., Bevis, C., Black, P. J., Bodhun, A., Brennan, J. S., Bridgham, J. A., Brown, R. C., Brown, A. A., Buermann, D. H., Bundu, A. A., Burrows, J. C., Carter, N. P., Castillo, N., Catenazzi, M. C. E., Chang, S., Cooley, R. N., Crake, N. R., Dada, O. O., Diakoumakos, K. D., Dominguez-Fernandez, B., Earnshaw, D. J., Egbujor, U. C., Elmore, D. W., Etchin, S. S., Ewan, M. R., Fedurco, M., Fraser, L. J., Fajardo, K. V. F., Furey, W. S., George, D., Gietzen, K. J., Goddard, C. P., Golda, G. S., Granieiri, P. A., Green, D. E., Gustafson, D. L., Hansen, N. F., Harnish, K., Haudenschild, C. D., Heyer, N. I., Hims, M. M., Ho, J. T., Horgan, A. M., Hoschler, K., Hurwitz, S., Ivanov, D. V., Johnson, M. Q., James, T., Jones, T. A. H., Kang, G., Kerelska, T. H., Kersey, A. D., Khrebtukova, I., Kindwall, A. P., Kingsbury,

Z., Kokko-Gonzales, P., Kumar, A., Laurent, M. A., Lawley, C. T., Lee, S. E., Lee, X., Liao, A. K., Loch, J. A., Lok, M., Luo, S., Mammen, R. M., Martin, J. W., McCauley, P. G., McNitt, P., Mehta, P., Moon, K. W., Mullens, J. W., Newington, T., Ning, Z., Ng, B. L., Novo, S. M., O'Neill, M. J., Osborne, M. A., Osnowski, A., Ostadan, O., Paraschos, L. L., Pickering, L., Pike, A. C., Pike, A. C., Pinkard, D. C., Pliskin, D. P., Podhasky, J., Quijano, V. J., Raczy, C., Rae, V. H., Rawlings, S. R., Rodriguez, A. C., Roe, P. M., Rogers, J., Rogert, M. C., Romanov, N., Romieu, A., Roth, R. K., Rourke, N. J., Ruediger, S. T., Rusman, E., Sanches-Kuiper, r. m., Schenker, M. R., Seoane, J. M., Shaw, R. J., Shiver, M. K., Short, S. W., Sizto, M. L., Sluis, J. P., Smith, M. A., Sohna, J. E., Spence, E. J., Stevens, K.,, Sutton, N., Szajkowski, L., Tregidgo, C. L., Turcatti, G., Vandevondele, S., Verhovsky, Y., Virk, S. M., Wakelin, S., Walcot, G. C., Wang, J., Worsley, G. J., Yan, J., Yau, L., Zuerlein, M., Rogers, J., Mullikin, J. C., Hurles, M. E., McCooke, N. J., West, J. S., Oaks, F. L., Lundberg, P. L., Klenerman, D., Durbin, R., Smith, A. J. (2008). Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, *456*, 53. Retrieved from http://dx.doi.org/10.1038/nature07517

Boulard, M. (1982). Boulard1982_les cigales du portugal.pdf.

Boulard, M. (2006). *Acoustic signals, diversity and behaviour of Cicadas (Cicadidae, Hemiptera) //Insect sounds and communication*. *Physiology, Behaviour, Ecology and Evolution*.

Branton, D., Deamer, D. W., Marziali, A., Bayley, H., Benner, S. A., Butler, T., Di Ventra, M., Garaj, S., Hibbs, A., Huang, X., Jovanovich, S. B., Krstic, P. S., Lindsay, S., Ling, X. S., Mastrangelo, C. H., Meller, A., Oliver, J. S., Pershin, Y. V., Ramsey, J. M., Riehn, R., Soni, G. V., Tabard-Cossa,V., Wanunu, M., Wiggin, M.,Schloss, J. A. (2008). The potential and challenges of nanopore sequencing. *Nature Biotechnology*, *26*, 1146. Retrieved from http://dx.doi.org/10.1038/nbt.1495

Catchen, J. M., Amores, A., Hohenlohe, P., Cresko, W., & Postlethwait, J. H. (2011). Stacks: building and genotyping loci de novo from short-read sequences. *G3: Genes, Genomes, Genetics*, *1*(3), 171–182.

Chidgeavadze, Z. G., Beabealashvilli, R. S., Atrazhev, A. M., Kukhanova, M. K., Azhayev, A. V, & Krayevsky, A. A. (1984). 2’,3’-Dideoxy-3’ aminonucleoside 5’-triphosphates are the terminators of DNA synthesis catalyzed by DNA polymerases. *Nucleic Acids Research*, *12*(3), 1671–1686. Retrieved from http://www.ncbi.nlm.nih.gov/pmc/articles/PMC318607/

Chong, Z., Ruan, J., & Wu, C.-I. (2012). Rainbow: an integrated tool for efficient clustering and assembling RAD-seq reads. *Bioinformatics*, *28*(21), 2732–2737.

Claridge, M. F. (1985). Acoustic Signals in the Homoptera: Behavior, Taxonomy, and Evolution. *Annual Review of Entomology*, *30*(1), 297–317. https://doi.org/10.1146/annurev.en.30.010185.001501

Clarke, J., Wu, H.-C., Jayasinghe, L., Patel, A., Reid, S., & Bayley, H. (2009). Continuous base identification for single-molecule nanopore DNA sequencing. *Nature Nanotechnology*, *4*, 265. Retrieved from http://dx.doi.org/10.1038/nnano.2009.12

Consortium, I. H. G. S. (2001). Initial sequencing and analysis of the human genome. *Nature*, *409*, 860. Retrieved from http://dx.doi.org/10.1038/35057062

Consortium, I. H. G. S. (2004). Finishing the euchromatic sequence of the human genome. *Nature*, *431*, 931. Retrieved from http://dx.doi.org/10.1038/nature03001

Cooley, J. R., & Marshall, D. C. (2001a). Sexual signaling in periodical cicadas, Magicicada spp.(Hemiptera: Cicadidae). *Behaviour*, *138*(7), 827–855.

Cooley, J. R., Simon, C., Marshall, D. C., Slon, K., & Ehrhardt, C. (2001b). Allochronic speciation, secondary contact, and reproductive character displacement in periodical cicadas (Hemiptera: Magicicada spp.): genetic, morphological, and behavioural evidence. *Molecular Ecology*, *10*(3), 661–671.

Costa, G. (2017). *Integrative approach unravels the evolutionary history of Western Mediterranean small cicadas (Hemiptera: Cicadettini)*.

Cronn, R., Knaus, B. J., Liston, A., Maughan, P. J., Parks, M., Syring, J. V, & Udall, J. (2012). Targeted enrichment strategies for next-generation plant biology. *American Journal of Botany*, *99*(2), 291–311.

Curto, M., Puppo, P., Kratschmer, S., & Meimberg, H. (2017). Genetic diversity and differentiation

patterns in Micromeria from the Canary Islands are congruent with multiple colonization dynamics and the establishment of species syngameons. *BMC Evolutionary Biology*, *17*(1), 198.

Curto, M., Schachtler, C., Puppo, P., & Meimberg, H. (2018). Using a new RAD-sequencing approach to study the evolution of Micromeria in the Canary islands. *Molecular Phylogenetics and Evolution*, *119*(November 2017), 160–169. https://doi.org/10.1016/j.ympev.2017.11.005

Davey, J. W., Hohenlohe, P. A., Etter, P. D., Boone, J. Q., Catchen, J. M., & Blaxter, M. L. (2011). Genome-wide genetic marker discovery and genotyping using next-generation sequencing. *Nature Reviews Genetics*, *12*(7), 499.

Deng, M., Zhou, Z.-K., & Li, Q. (2013). *Taxonomy and systematics of Quercus subgenus Cyclobalanopsis*. *International Oaks (the Journal of the International Oak Society)* (Vol. 24).

Deng, M., Jiang, X. L., Hipp, A. L., Manos, P. S., & Hahn, M. (2018). Phylogeny and biogeography of East Asian evergreen oaks (Quercus section Cyclobalanopsis; Fagaceae): Insights into the Cenozoic history of evergreen broad-leaved forests in subtropical Asia. *Molecular Phylogenetics and Evolution*, *119*(November 2017), 170–181. https://doi.org/10.1016/j.ympev.2017.11.003

Denk, T., & Grimm, G. W. (2010). The oaks of western Eurasia: traditional classifications and evidence from two nuclear markers. *Taxon*, *59*(2), 351–366.

Derrington, I. M., Butler, T. Z., Collins, M. D., Manrao, E., Pavlenok, M., Niederweis, M., & Gundlach, J. H. (2010). Nanopore DNA sequencing with MspA. *Proceedings of the National Academy of Sciences of the United States of America*, *107*(37), 16060–16065. https://doi.org/10.1073/pnas.1001831107

Després, L., Pettex, E., Plaisance, V., & Pompanon, F. (2002). Speciation in the globeflower fly Chiastocheta spp.(Diptera: Anthomyiidae) in relation to host plant species, biogeography, and morphology. *Molecular Phylogenetics and Evolution*, *22*(2), 258–268.

Dohm, J. C., Lottaz, C., Borodina, T., & Himmelbauer, H. (2008). Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Research*, *36*(16), e105–e105. https://doi.org/10.1093/nar/gkn425

Eaton, D. A. R., & Ree, R. H. (2013). Inferring Phylogeny and Introgression using RADseq Data : An Example from Flowering Plants (Pedicularis : Orobanchaceae), *62*(5), 689–706. https://doi.org/10.1093/sysbio/syt032

Eaton, D. (2014). PyRAD: assembly of de novo RADseq loci for phylogenetic analyses. *Bioinformatics*, *30*(13), 1844–1849. Retrieved from http://dx.doi.org/10.1093/bioinformatics/btu121

Eaton, D. (2015). ipyrad Documentation.

Eid, J., Fehr, A., Gray, J., Luong, K., Lyle, J., Otto, G., Peluso, P., Rank, D., Baybayan, P., Bettman, B. (2008). Real-time DNA sequencing from single polymerase molecules. *Science*.

Eisenstein, M. (2012). Oxford Nanopore announcement sets sequencing sector abuzz. *Nature Biotechnology*, *30*, 295. Retrieved from http://dx.doi.org/10.1038/nbt0412-295

Elshire, R. J., Glaubitz, J. C., Sun, Q., Poland, J. A., Kawamoto, K., Buckler, E. S., & Mitchell, S. E. (2011). A robust, simple genotyping-by-sequencing (GBS) approach for high diversity species. *PloS One*, *6*(5), e19379.

Emerson, K. J., Merz, C. R., Catchen, J. M., Hohenlohe, P. A., Cresko, W. A., Bradshaw, W. E., & Holzapfel, C. M. (2010). Resolving postglacial phylogeography using high-throughput sequencing. *Proceedings of the National Academy of Sciences*, 201006538.

English, A. C., Richards, S., Han, Y., Wang, M., Vee, V., Qu, J., Qin, X., Muzny, D. M., Reid, J. G., Worley, K. C., Gibbs, R. A. (2012). Mind the Gap: Upgrading Genomes with Pacific Biosciences RS Long-Read Sequencing Technology. *PLoS ONE*, *7*(11), e47768. https://doi.org/10.1371/journal.pone.0047768

Escudero, M., Eaton, D. A. R., Hahn, M., & Hipp, A. L. (2014). Genotyping-by-sequencing as a tool to infer phylogeny and ancestral hybridization : A case study in Carex (Cyperaceae). *Molecular Phylogenetics and Evolution*, *79*, 359–367. https://doi.org/10.1016/j.ympev.2014.06.026

Espíndola, A., Buerki, S., & Alvarez, N. (2012). Ecological and historical drivers of diversification in the fly genus Chiastocheta Pokorny. *Molecular Phylogenetics and Evolution*, *63*(2), 466–474.

Etter, P. D., Bassham, S., Hohenlohe, P. A., Johnson, E. A., & Cresko, W. A. (2011a). Molecular Methods for Evolutionary Genetics, *772*(2), 1–19. https://doi.org/10.1007/978-1-61779-228-1

Etter, P. D., Preston, J. L., Bassham, S., Cresko, W. A., & Johnson, E. A. (2011b). Local De Novo

Assembly of RAD Paired-End Contigs Using Short Sequencing Reads. *PLoS ONE*, *6*(4), e18561. https://doi.org/10.1371/journal.pone.0018561

Falush, D., Stephens, M., & Pritchard, J. K. (2003). Inference of Population Structure Using Multilocus Genotype Data: Linked Loci and Correlated Allele Frequencies. *Genetics*, *164*(4), 1567 LP-1587. Retrieved from http://www.genetics.org/content/164/4/1567.abstract

Falush, D., Stephens, M., & Pritchard, J. K. (2007). Inference of population structure using multilocus genotype data: dominant markers and null alleles. *Molecular Ecology Notes*, *7*(4), 574–578. https://doi.org/10.1111/j.1471-8286.2007.01758.x

Fedurco, M., Romieu, A., Williams, S., Lawrence, I., & Turcatti, G. (2006). BTA, a novel reagent for DNA attachment on glass and efficient generation of solid-phase amplified DNA colonies. *Nucleic Acids Research*, *34*(3), e22–e22. https://doi.org/10.1093/nar/gnj023

García-Barros, E., Gurrea, P., Luciáñez, M. J., Cano, J. M., Munguira, M. L., Moreno, J. C., Sainz, H., Sanz, M. J., Simón, J. C. (2002). Parsimony analysis of endemicity and its application to animal and plant geographical distributions in the Ibero-Balearic region (western Mediterranean). *Journal of Biogeography*, *29*(1), 109–124.

Gogala, M., & Gogala, A. (1999). A checklist and provisional atlas of the Cicadoidea fauna of Slovenia (Homoptera: Auchenorrhyncha). *Acta Entom Ologica Slovenica*, *7*, 2.

Gompert, Z., Lucas, L. K., Fordyce, J. A., Forister, M. L., & Nice, C. C. (2010). Secondary contact between Lycaeides idas and L. melissa in the Rocky Mountains: extensive admixture and a patchy hybrid zone. *Molecular Ecology*, *19*(15), 3171–3192.

Gut, I. G. (2013). New sequencing technologies. *Clinical and Translational Oncology*, *15*(11), 879–881. https://doi.org/10.1007/s12094-013-1073-6

Hall, N. (2007). Advanced sequencing technologies and their wider impact in microbiology. *Journal of Experimental Biology*, *210*(9), 1518 LP-1525. Retrieved from http://jeb.biologists.org/content/210/9/1518.abstract

Heather, J. M., & Chain, B. (2016). The sequence of sequencers: The history of sequencing DNA. *Genomics*, *107*(1), 1–8. https://doi.org/10.1016/j.ygeno.2015.11.003

Hertach, T. (2008). A new cicada species for Switzerland: Tettigetta argentata (Olivier, 1790)(Hemiptera: Cicadoidea). *Mitteilungen Der Schweizerischen Entomologischen Gesellschaft*, 209–214.

Hipp, A. L., Eaton, D. A. R., Cavender-Bares, J., Fitzek, E., Nipper, R., & Manos, P. S. (2014). A framework phylogeny of the American oak clade based on sequenced RAD data. *PLoS One*, *9*(4), e93975.

Hipp, A. (2015). *Should hybridization make us skeptical of the oak phylogeny? International Oaks: The Journal of the International Oak Society* (Vol. 26).

Hohenlohe, P. A., Bassham, S., Etter, P. D., Stiffler, N., Johnson, E. A., & Cresko, W. A. (2010). Population genomics of parallel adaptation in threespine stickleback using sequenced RAD tags. *PLoS Genetics*, *6*(2), e1000862.

Holley, R. W., Madison, J. T., & Zamir, A. (1964). A new method for sequence determination of large oligonucleotides. *Biochemical and Biophysical Research Communications*, *17*(4), 389–394. https://doi.org/https://doi.org/10.1016/0006-291X(64)90017-8

Holley, R. H., Apgar, J., A. Everett, G., T. Madison, J., Marquisee, M., H. Merrill, S., Penswick, J. R., Zamir, A. (1965). *Structure of a Ribonucleic Acid*. *Science (New York, N.Y.)* (Vol. 147). https://doi.org/10.1126/science.147.3664.1462

Hubisz, M. J., Falush, D., Stephens, M., & Pritchard, J. K. (2009). Inferring weak population structure with the assistance of sample group information. *Molecular Ecology Resources*, *9*(5), 1322–1332. https://doi.org/10.1111/j.1755-0998.2009.02591.x

Ilut, D. C., Nydam, M. L., & Hare, M. P. (2014). Defining loci in restriction-based reduced representation genomic data from nonmodel species: sources of bias and diagnostics for optimal clustering. *BioMed Research International*, *2014*.

Jain, M., Fiddes, I., Miga, K. H., Olsen, H. E., Paten, B., & Akeson, M. (2015). Improved data analysis for the MinION nanopore sequencer. *Nature Methods*, *12*(4), 351–356. https://doi.org/10.1038/nmeth.3290

Jong, H. D. E. (1998). In search of historical biogeographic patterns in. *Biological Journal of the*

Linnean Society, (July), 99–164.
Kremer, A., Abbott, A. G., Carlson, J. E., Manos, P. S., Plomion, C., Sisco, P., Staton, M. E., Ueno, S., Vendramin, G. G. (2012). Genomics of Fagaceae. *Tree Genetics & Genomes*, *8*(3), 583–610.
Kulski, J. K. (2016). Next-Generation Sequencing—An Overview of the History, Tools, and "Omic" Applications. In *Next Generation Sequencing-Advances, Applications and Challenges*. Intech.
Leong, U. I., Skinner, R. J., & Love, R. D. (2014). Application of Massively Parallel Sequencing in the Clinical Diagnostic Testing of Inherited Cardiac Conditions. *Medical Sciences* . https://doi.org/10.3390/medsci2020098
Levene, M. J., Korlach, J., Turner, S. W., Foquet, M., Craighead, H. G., & Webb, W. W. (2003). Zero-mode waveguides for single-molecule analysis at high concentrations. *Science*, *299*(5607), 682–686.
Lim, H. C., Zou, F., Taylor, S. S., Marks, B. D., Moyle, R. G., Voelker, G., & Sheldon, F. H. (2010). Phylogeny of magpie-robins and shamas (Aves: Turdidae: Copsychus and Trichixos): implications for island biogeography in Southeast Asia. *Journal of Biogeography*, *37*(10), 1894–1906. https://doi.org/10.1111/j.1365-2699.2010.02343.x
Lim, H. C., Rahman, M. A., Lim, S. L. H., Moyle, R. G., & Sheldon, F. H. (2011). REVISITING WALLACE'S HAUNT: COALESCENT SIMULATIONS AND COMPARATIVE NICHE MODELING REVEAL HISTORICAL MECHANISMS THAT PROMOTED AVIAN POPULATION DIVERGENCE IN THE MALAY ARCHIPELAGO. *Evolution*, *65*(2), 321–334. https://doi.org/10.1111/j.1558-5646.2010.01105.x
Liu, L., Li, Y., Li, S., Hu, N., He, Y., Pong, R., Lin, D., Lu, L., Law, M. (2012). Comparison of next-generation sequencing systems. *BioMed Research International*, *2012*.
Loman, N. J., Misra, R. V, Dallman, T. J., Constantinidou, C., Gharbia, S. E., Wain, J., & Pallen, M. J. (2012). Performance comparison of benchtop high-throughput sequencing platforms. *Nature Biotechnology*, *30*, 434. Retrieved from http://dx.doi.org/10.1038/nbt.2198
Loman, N. J., & Quinlan, A. R. (2014). Poretools: a toolkit for analyzing nanopore sequence data. *Bioinformatics*, *30*(23), 3399–3401. https://doi.org/10.1093/bioinformatics/btu555
Loman, N. J., Quick, J., & Simpson, J. T. (2015). A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nature Methods*, *12*, 733. Retrieved from http://dx.doi.org/10.1038/nmeth.3444
Loomis, E. W., Eid, J. S., Peluso, P., Yin, J., Hickey, L., Rank, D., McCalmon, S., Hagerman, R. J., Tassone, F., Hagerman, P. J. (2013). Sequencing the unsequenceable: Expanded CGG-repeat alleles of the fragile X gene. *Genome Research*, *23*(1), 121–128. https://doi.org/10.1101/gr.141705.112
Luikart, G., England, P. R., Tallmon, D., Jordan, S., & Taberlet, P. (2003). The power and promise of population genomics: from genotyping to genome typing. *Nature Reviews Genetics*, *4*(12), 981.
Manos, P. S., Cannon, C. H., & Oh, S.-H. (2008). Phylogenetic relationships and taxonomic status of the paleoendemic Fagaceae of western North America: recognition of a new genus, Notholithocarpus. *Madrono*, *55*(3), 181–190.
Mardis, E. R. (2008). Next-generation DNA sequencing methods. *Annu. Rev. Genomics Hum. Genet.*, *9*, 387–402.
Marguerat, S., Wilhelm, B. T., & Bähler, J. (2008). Next-generation sequencing: applications beyond genomes. Portland Press Limited.
Margulies, M., Egholm, M., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A., Berka, J., Braverman, M. S., Chen, Y., Chen, Z., Dewell, S. B., Du, L., Fierro, J. M., Gomes, X. V., Godwin, B. C., He, W., Helgesen, H., Ho, C. H., Irzyk, G. P., Jando, S. C., Alenquer, M. L., Jarvie, T. P., Jirage, K. B., Kim, J., Knight, J. R., Lanza, J. R., Leamon, J. H., Lefkowitz, S. M., Lei, M., Li, J., Lohman, K. L., Lu, H., Makhijani, V. B., McDade, K. E., McKenna, M. P., Myers, E. W., Nickerson, E., Nobile, J. R., Plant, R., Puc, B. P., Ronan, M. T., Roth, G. T., Sarkis, G. J., Simons, J. F., Simpson, J. W., Srinivasan, M., Tartaro, K. R., Tomasz, A., Vogt, K. A., Volkmer, G. A., Wang, S. H., Wang, Y., Weiner, M. P., Yu, P., Begley, R. F., Rothberg, J. M. (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, *437*, 376. Retrieved from http://dx.doi.org/10.1038/nature03959
Mastretta-Yanes, A., Arrigo, N., Alvarez, N., Jorgensen, T. H., Piñero, D., & Emerson, B. C. (2015).

Restriction site-associated DNA sequencing, genotyping error estimation and de novo assembly optimization for population genetic inference. *Molecular Ecology Resources*, *15*(1), 28–41.

McGinn, S., & Gut, I. G. (2013). DNA sequencing - spanning the generations. *New Biotechnology*, *30*(4), 366–372. https://doi.org/10.1016/j.nbt.2012.11.012

McKernan, K. J., Peckham, H. E., Costa, G. L., McLaughlin, S. F., Fu, Y., Tsung, E. F., Clouser, C. R., Dunca, C., Ichikawa, J. F., Lee, C. C., Zhang, Z., Ranade, S. S., Dimalanta, E. T., Hyland, F. C., Sokolsky, T. D., Zhang, L., Sheridan, A., Fu, H., Hendrickson, C. L., Li, B., Kotler, L., Stuart, J. R., Malek, J. A., Manning, J. M., Antipova, A. A., Perez, D. S., Moore, M. P., Hayashibara, K. C., Lyons, M. R., Beaudoin, R. E., Coleman, B. E., Laptewicz, M. W., Sannicandro, A. E., Rhodes, M. D., Gottimukkala, R. K., Yang, S., Bafna, V., Bashir, A., MacBride, A., Alkan, C., Kidd, J. M., Eichler, E. E., Reese, M. G., De La Vega, F. M., Blanchard, A. P. (2009). Sequence and structural variation in a human genome uncovered by short-read, massively parallel ligation sequencing using two-base encoding. *Genome Research*, *19*(9), 1527–1541. https://doi.org/10.1101/gr.091868.109

Medail, F., & Quezel, P. (1997). Hot-spots analysis for conservation of plant biodiversity in the Mediterranean Basin. *Annals of the Missouri Botanical Garden*, 112–127.

Mellmann, A., Harmsen, D., Cummings, C. A., Zentz, E. B., Leopold, S. R., Rico, A., Prior, K., Szczepanowski, R., Ji, Y., Zhang, W., McLaughlin, S. F., Henkhaus, J. K., Leopold, B., Bielaszewska, M., Prager, R., Brzoska, P. M., Moore, R. L., Guenther, S., Rothberg, J. M., Karch, H. (2011). Prospective Genomic Characterization of the German Enterohemorrhagic Escherichia coli O104:H4 Outbreak by Rapid Next Generation Sequencing Technology. *PLoS ONE*, *6*(7), e22751. https://doi.org/10.1371/journal.pone.0022751

Mendes, R., Nunes, V. L., Quartau, J. A., & Simões, P. C. (2014). Patterns of acoustic and morphometric variation in species of genus Tettigettalna (Hemiptera: Cicadidae): Sympatric populations show unexpected differences. *European Journal of Entomology*, *111*(3), 429–441. https://doi.org/10.14411/eje.2014.054

Metzker, M. L. (2009). Sequencing technologies — the next generation. *Nature Reviews Genetics*, *11*, 31. Retrieved from http://dx.doi.org/10.1038/nrg2626

Moreno Saiz, J. C., Castro Parga, I., & Sainz Ollero, H. (1998). Numerical analyses of distributions of Iberian and Balearic endemic monocotyledons. *Journal of Biogeography*, *25*(1), 179–194.

Morozova, O., & Marra, M. A. (2008). Applications of next-generation sequencing technologies in functional genomics. *Genomics*, *92*(5), 255–264. https://doi.org/10.1016/j.ygeno.2008.07.001

Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., & Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, *5*(7), 621.

Mota, J. F., Pérez-García, F. J., Jiménez, M. L., Amate, J. J., & Peñas, J. (2002). Phytogeographical relationships among high mountain areas in the Baetic Ranges (South Spain). *Global Ecology and Biogeography*, *11*(6), 497–504.

Nast, J. (1972). Palaearctic Auchenorrhyncha (Homoptera) an annotated check list. *Palaearctic Auchenorrhyncha (Homoptera) an Annotated Check List.*

Ng, E. Y. X., Garg, K. M., Low, G. W., Chattopadhyay, B., Oh, R. R. Y., Lee, J. G. H., & Rheindt, F. E. (2017). Conservation genomics identifies impact of trade in a threatened songbird. *Biological Conservation*, *214*(July), 101–108. https://doi.org/10.1016/j.biocon.2017.08.007

Nichols, R. A. (2017). Documentation for MavericK software :, 1–30.

Niedringhaus, T. P., Milanova, D., Kerby, M. B., Snyder, M. P., & Barron, A. E. (2011). Landscape of Next-Generation Sequencing Technologies. *Analytical Chemistry*, *83*(12), 4327–4341. https://doi.org/10.1021/ac2010857

Nunes, V. L., Mendes, R., Quartau, J., & Simões, P. (2014a). *Ecologi@ Current distribution raises concerns on the conservation of Tettigettalna mariae (Quartau & Boulard, 1995) (Hemiptera: Cicadoidea) in Portugal*.

Nunes, V. L., Mendes, R., Marabuto, E., Novais, B. M., Hertach, T., Quartau, J. A., Seabra, S. G., Paulo, O.S., Simões, P. C. (2014b). Conflicting patterns of DNA barcoding and taxonomy in the cicada genus Tettigettalna from southern Europe (Hemiptera: Cicadidae). *Molecular Ecology Resources*, *14*(1), 27–38. https://doi.org/10.1111/1755-0998.12158

Nyrén, P. (1987). Enzymatic method for continuous monitoring of DNA polymerase activity. *Analytical*

Biochemistry, *167*(2), 235–238. https://doi.org/https://doi.org/10.1016/0003-2697(87)90158-8
Peterson, B. K., Weber, J. N., Kay, E. H., Fisher, H. S., & Hoekstra, H. E. (2012). Double digest RADseq: an inexpensive method for de novo SNP discovery and genotyping in model and non-model species. *PloS One*, *7*(5), e37135.
Pham, K. K., Hipp, A. L., Manos, P. S., & Cronn, R. C. (2017). A time and a place for everything: phylogenetic history and geography as joint predictors of oak plastome phylogeny. *Genome*, *60*(9), 720–732.
Pritchard, J. K., Stephens, M., & Donnelly, P. (2000). Inference of Population Structure Using Multilocus Genotype Data. *Genetics*, *155*(2), 945 LP-959. Retrieved from http://www.genetics.org/content/155/2/945.abstract
Puissant, S., & Sueur, J. (2010). A hotspot for Mediterranean cicadas (Insecta: Hemiptera: Cicadidae): New genera, species and songs from southern Spain. *Systematics and Biodiversity*, *8*(4), 555–574. https://doi.org/10.1080/14772000.2010.532832
Puppo, P., Curto, M., & Meimberg, H. (2016). Genetic structure of Micromeria (Lamiaceae) in Tenerife, the imprint of geological history and hybridization on within-island diversification. *Ecology and Evolution*, *6*(11), 3443–3460.
Quail, M. A., Smith, M., Coupland, P., Otto, T. D., Harris, S. R., Connor, T. R., Bertoni, A., Swerdlow, H. P., Gu, Y. (2012). A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. *BMC Genomics*, *13*, 341. https://doi.org/10.1186/1471-2164-13-341
Quartau, J. A., & Boulard, M. (1995). Quartau&Boulard_1995_EPHE.pdf.
Quick, J., Quinlan, A. R., & Loman, N. J. (2014). A reference bacterial genome dataset generated on the MinIONTM portable single-molecule nanopore sequencer. *GigaScience*, *3*, 22. https://doi.org/10.1186/2047-217X-3-22
Radford, A. D., Chapman, D., Dixon, L., Chantrey, J., Darby, A. C., & Hall, N. (2012). Application of next-generation sequencing technologies in virology. *The Journal of General Virology*, *93*(Pt 9), 1853–1868. https://doi.org/10.1099/vir.0.043182-0
Razkin, O., Sonet, G., Breugelmans, K., Madeira, M. J., Gómez-Moliner, B. J., & Backeljau, T. (2016). Species limits, interspecific hybridization and phylogeny in the cryptic land snail complex Pyramidula: The power of RADseq data. *Molecular Phylogenetics and Evolution*, *101*, 267–278. https://doi.org/10.1016/j.ympev.2016.05.002
Reuter, J. A., Spacek, D. V., & Snyder, M. P. (2015). High-Throughput Sequencing Technologies. *Molecular Cell*, *58*(4), 586–597. https://doi.org/10.1016/j.molcel.2015.05.004
Ribera, I. (2000). Biogeography and conservation of Iberian water beetles. *Biological Conservation*, *92*(2), 131–150.
Rokas, A., & Abbot, P. (2009). Harnessing genomics for evolutionary insights. *Trends in Ecology & Evolution*, *24*(4), 192–200.
Ronaghi, M., Karamohamed, S., Pettersson, B., Uhlén, M., & Nyrén, P. (1996). Real-Time DNA Sequencing Using Detection of Pyrophosphate Release. *Analytical Biochemistry*, *242*(1), 84–89. https://doi.org/https://doi.org/10.1006/abio.1996.0432
Ronaghi, M., Uhlén, M., & Nyrén, P. (1998). A Sequencing Method Based on Real-Time Pyrophosphate. *Science*, *281*(5375), 363 LP-365. Retrieved from http://science.sciencemag.org/content/281/5375/363.abstract
Rothberg, J. M., Hinz, W., Rearick, T. M., Schultz, J., Mileski, W., Davey, M., Leamon, J. H., Johnson, K., Milgrew, M., Edwards, M., Hoon, J., Simons, J. F., Marran, D., Myers, J. W., Davidson, J. F., Branting, A., Nobile, J. R., Puc, B. P., Light, D., Clark, T. A., Huber, M., Branciforte, J. T., Stoner, I. B., Cawley, S. E., Lyons, M., Fu, Y., Homer, N., Sedova, M., Miao, X., Reed, B., Sabina, Y., Feierstein, E., Schorn, M., Alanjary, M., Dimalanta, E., Dressman, D., Kasinskas, R., Sokolsky, T., Fidanza, J. A., Namsaraev, E., McKernan, K. J., Williams, A., Roth, G. T., Bustillo, J. (2011). An integrated semiconductor device enabling non-optical genome sequencing. *Nature*, *475*, 348. Retrieved from http://dx.doi.org/10.1038/nature10242
Sanger, F., Air, G. M., Barrell, B. G., Brown, N. L., Coulson, A. R., Fiddes, J. C., Hutchison, C. A., Slocombe, P. M., Smith, M. (1977a). Nucleotide sequence of bacteriophage φX174 DNA. *Nature*, *265*, 687. Retrieved from http://dx.doi.org/10.1038/265687a0

Sanger, F., Nicklen, S., & Coulson, A. R. (1977b). DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America*, *74*(12), 5463–5467. Retrieved from http://www.ncbi.nlm.nih.gov/pmc/articles/PMC431765/

Schadt, E. E., Turner, S., & Kasarskis, A. (2010). A window into third-generation sequencing. *Human Molecular Genetics*, *19*(R2), 227–240. https://doi.org/10.1093/hmg/ddq416

Seabra, S. G., Silva, S. E., Nunes, V. L., Sousa, V. C., Martins, J., Marabuto, E., Rodrigues, A., Pina-Martins, F., Laurentino, T. G., Rebelo, M. T., Figueiredo, E., Paulo, O. S. (2019). Genomic signatures of introgression between commercial and native bumblebees, Bombus terrestris, in western Iberian Peninsula—Implications for conservation and trade regulation. *Evolutionary Applications*, *12*(4), 679–691. https://doi.org/10.1111/eva.12732

Shendure, J., Porreca, G. J., Reppas, N. B., Lin, X., McCutcheon, J. P., Rosenbaum, A. M., Wang, M. D., Zhang, K., Mitra, R. D., Church, G. M. (2005). Accurate Multiplex Polony Sequencing of an Evolved Bacterial Genome. *Science*, *309*(5741), 1728 LP-1732. Retrieved from http://science.sciencemag.org/content/309/5741/1728.abstract

Simeone, M. C., Piredda, R., Papini, A., Vessella, F., & Schirone, B. (2013). Application of plastid and nuclear markers to DNA barcoding of Euro-Mediterranean oaks (Quercus, Fagaceae): problems, prospects and phylogenetic implications. *Botanical Journal of the Linnean Society*, *172*(4), 478–499.

Simões, P. C., Nunes, V. L., Mendes, R., & Quartau, J. A. (2013). First record of Tettigettalna mariae Quartau & Boulard, 1995 (Insecta: Hemiptera: Cicadoidea) in Spain. *Biodiversity Data Journal*, *1*, e978. Retrieved from https://doi.org/10.3897/BDJ.1.e978

Simões, P., Nunes, V., Mendes, R., Seabra, S., Paulo, O., & Quartau, J. (2014). Tettigettalna josei (Boulard, 1982) (Hemiptera: Cicadoidea): first record in Spain, with notes on the distribution, genetic variation and behaviour of the species. *Biodiversity Data Journal*, *2*(May), e1045. https://doi.org/10.3897/BDJ.2.e1045

Smith, L. M., Sanders, J. Z., Kaiser, R. J., Hughes, P., Dodd, C., Connell, C. R., Heiner, C., Kent, S. B. H., Hood, L. E. (1986). Fluorescence detection in automated DNA sequence analysis. *Nature*, *321*, 674. Retrieved from http://dx.doi.org/10.1038/321674a0

Suchan, T., Espíndola, A., Rutschmann, S., Emerson, B. C., Gori, K., Dessimoz, C., Arrigo, N., Ronikier, N., Alvarez, N. (2017). Assessing the potential of RAD-sequencing to resolve phylogenetic relationships within species radiations: The fly genus Chiastocheta (Diptera: Anthomyiidae) as a case study. *Molecular Phylogenetics and Evolution*, *114*, 189–198. https://doi.org/10.1016/j.ympev.2017.06.012

Sueur, J., Puissant, S., Simões, P. C., Seabra, S., Boulard, M., & Quartau, J. A. (2004). Cicadas from Portugal: revised list of species with eco-ethological data (Hemiptera: Cicadidae). *Insect Systematics & Evolution*, *35*(2), 177–187.

Sueur, J. (2006). Insect Species and Their Songs. *Insect Sounds and Communication: Physiology, Behaviour, Ecology, and Evolution*, (November 2005), 207–217. https://doi.org/10.1201/9781420039337.ch15

Sun, M., Watson, G. S., Zheng, Y., Watson, J. A., & Liang, A. (2009). Wetting properties on nanostructured surfaces of cicada wings. *Journal of Experimental Biology*, *212*(19), 3148–3155. https://doi.org/10.1242/jeb.033373

Takahashi, T., Nagata, N., & Sota, T. (2014). Application of RAD-based phylogenetics to complex relationships among variously related taxa in a species flock. *Molecular Phylogenetics and Evolution*, *80*, 137–144. https://doi.org/10.1016/j.ympev.2014.07.016

Takahashi, T., & Moreno, E. (2015). A RAD-based phylogenetics for Orestias fishes from Lake Titicaca q. *MOLECULAR PHYLOGENETICS AND EVOLUTION*, *93*, 307–317. https://doi.org/10.1016/j.ympev.2015.08.012

Tawfik, D. S., & Griffiths, A. D. (1998). Man-made cell-like compartments for molecular evolution. *Nature Biotechnology*, *16*, 652. Retrieved from http://dx.doi.org/10.1038/nbt0798-652

Turcatti, G., Romieu, A., Fedurco, M., & Tairi, A.-P. (2008). A new class of cleavable fluorescent nucleotides: synthesis and optimization as reversible terminators for DNA sequencing by synthesis †. *Nucleic Acids Research*, *36*(4), e25–e25. Retrieved from http://dx.doi.org/10.1093/nar/gkn021

Van Tassell, C. P., Smith, T. P. L., Matukumalli, L. K., Taylor, J. F., Schnabel, R. D., Lawley, C. T.,

Haudenschild, C. D., Moore, S., Warren, W., Sonstegard, T. S. (2008). SNP discovery and allele frequency estimation by deep sequencing of reduced representation libraries. *Nature Methods*, *5*(3), 247.

Vargas, J. M., Real, R., & Guerrero, J. C. (1998). Biogeographical regions of the Iberian Peninsula based on freshwater fish and amphibian distributions. *Ecography*, *21*(4), 371–382.

Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., Smith, H. O., Yandell, M., Evans, C. A., Holt, R. A. (2001). The sequence of the human genome. *Science*, *291*(5507), 1304–1351.

Voelkerding, K. V, Dames, S. A., & Durtschi, J. D. (2009). Next-Generation Sequencing: From Basic Research to Diagnostics. *Clinical Chemistry*, *55*(4), 641 LP-658. Retrieved from http://clinchem.aaccjnls.org/content/55/4/641.abstract

WATSON, J. D., & CRICK, F. H. C. (1953). Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. *Nature*, *171*, 737. Retrieved from http://dx.doi.org/10.1038/171737a0

Wessel, A., Mühlethaler, R., Hartung, V., Kuštor, V., & Gogala, M. (2014). The tymbal: evolution of a complex vibration-producing organ in the Tymbalia (Hemiptera excl. Sternorrhyncha). In *Studying vibrational communication* (pp. 395–444). Springer.

Wilding, C. S., Grahame, J., & Mill, P. J. (2000). Mitochondrial DNA CoI haplotype variation in sibling species of rough periwinkles. *Heredity*, *85*(1), 62.

Williams, K. S., & Simon, C. (1995). The Ecology, Behavior, and Evolution of Periodical Cicadas. *Annual Review of Entomology*, *40*(1), 269–295. https://doi.org/10.1146/annurev.en.40.010195.001413

Xing, Y., Onstein, R. E., Carter, R. J., Stadler, T., & Peter Linder, H. (2014). Fossils and a large molecular phylogeny show that the evolution of species richness, generic diversity, and turnover rates are disconnected. *Evolution*, *68*(10), 2821–2832.

Yoshimura, J. (1997). The evolutionary origins of periodical cicadas during ice ages. *The American Naturalist*, *149*(1), 112–124.

Zhou, Y., Duvaux, L., Ren, G., Zhang, L., Savolainen, O., & Liu, J. (2017). Importance of incomplete lineage sorting and introgression in the origin of shared genetic variation between two closely related pines with overlapping distributions. *Heredity*, *118*(3), 211–220. https://doi.org/10.1038/hdy.2016.72

Chapter 6

Supporting Information

Table 6.1: Specimens of *Tettigettalna* cicadas under study with indication of identification barcode, species, sample location and data of the sample location.

#	Individual	Barcode	Species	Region	Country	Data
1	Tma560_allo	TAGCA	T. mariae_Allopatry	Cartaya	Spain	18/07/2013
2	Tma720_allo	TCAGA	T. mariae_Allopatry	Cartaya	Spain	24/07/2013
3	Tma556_allo	TCGAG	T. mariae_Allopatry	Cartaya	Spain	17/07/2013
4	Tma559_allo	TGACC	T. mariae_Allopatry	Cartaya	Spain	18/07/2013
5	Tma561_allo	TGGTT	T. mariae_Allopatry	Cartaya	Spain	18/07/2013
6	Tma718_allo	TTAAT	T. mariae_Allopatry	Cartaya	Spain	24/07/2013
7	Tma068_allo	TTGGC	T. mariae_Allopatry	Vale Judeu	Portugal	14/07/2011
8	Tma729_allo	AACCC	T. mariae_Allopatry	Vale Judeu	Portugal	25/07/2013
9	Tma071_allo	AATTT	T. mariae_Allopatry	Vale Judeu	Portugal	14/07/2011
10	Tma363_sym	ACCAT	T. mariae_Sympatry	Quinta do Lago	Portugal	09/08/2012
11	Tma352_sym	ACTGC	T. mariae_Sympatry	Quinta do Lago	Portugal	08/08/2012
12	Tma354_sym	AAAAA	T. mariae_Sympatry	Quinta do Lago	Portugal	08/08/2012
13	Tma327_sym	AAGGG	T. mariae_Sympatry	Quinta do Lago	Portugal	02/08/2012
14	Tma342_sym	ACACG	T. mariae_Sympatry	Quinta do Lago	Portugal	03/08/2012
15	Tma338_sym	ACGTA	T. mariae_Sympatry	Quinta do Lago	Portugal	02/08/2012
16	Tar310_sym	AGAGT	T. argentata South_sympatry	Quinta do Lago	Portugal	01/08/2012
17	Tar335_sym	AGGAC	T. argentata South_sympatry	Quinta do Lago	Portugal	02/08/2012
18	Tar321_sym	ATATC	T. argentata South_sympatry	Quinta do Lago	Portugal	01/08/2012
19	Tar357_sym	ATGCT	T. argentata South_sympatry	Quinta do Lago	Portugal	08/08/2012
20	Tar356_sym	CAACT	T. argentata South_sympatry	Quinta do Lago	Portugal	08/08/2012
21	Tar359_sym	CAGTC	T. argentata South_sympatry	Quinta do Lago	Portugal	08/08/2012
22	Tar553_allo	CCAAC	T. argentata South_allopatry	S. Bartolomeu de Messines	Portugal	17/07/2013
23	Tar552_allo	CCGGT	T. argentata South_allopatry	S. Bartolomeu de Messines	Portugal	17/07/2013
24	Tar260_allo	ATTAG	T. argentata South_allopatry	Espiel	Spain	10/07/2012
25	Tar769_allo	ATCGA	T. argentata South_allopatry	Portel	Portugal	08/07/2014
26	Tar770_allo	GTTGT	T. argentata South_allopatry	Portel	Portugal	08/07/2014
27	Tar771_allo	GTCAC	T. argentata South_allopatry	Portel	Portugal	08/07/2014
28	Tar526_c	GGTTC	T. argentata centre	Almaraz	Spain	26/06/2013

29	Tar528_c	GGCCT	T. argentata centre	Almaraz	Spain	26/06/2013
30	Tar547_c	GCTAA	T. argentata centre	Albarracin	Spain	28/06/2013
31	Tar529_N	GCCGG	T. argentata North	Almaraz	Spain	26/06/2013
32	Tar177_N	GACTA	T. argentata North	Sesimbra	Portugal	28/06/2012
33	Tar299_N	AGTCA	T. argentata North	Serra d' Aire e Candeeiros	Portugal	26/07/2012
34	Tan254	AGCTG	T. aneabi	Zagra	Spain	10/07/2012
35	Tan710	CTTCC	T. aneabi	Frailes	Spain	23/07/2013
36	Tan709	CTCTT	T. aneabi	Frailes	Spain	23/07/2013
37	Tan707	CGTAT	T. aneabi	Frailes	Spain	23/07/2013
38	Tan3755	CGCGC	T. aneabi	Granada	Spain	23/07/2013
39	Tan711	CCTTG	T. aneabi	Estepa	Spain	23/07/2013
40	Tjo3765	CCCCA	T. josei	Armação de Pêra	Portugal	08/06/2014

Table 6.2: Clustering threshold and minimum sample locus values for each of the assemblies performed on ipyrad program.

Assembly name	Clust_Threshold	Min_Sample_Locus
params	0,85	4
params1	0,9	4
params2	0,95	4
params3	0,8	4
params4	0,85	5
params5	0,9	5
params6	0,95	5
params7	0,85	7
params8	0,9	7
params9	0,95	7
params10	0,9	7
params11	0,9	5
params12	0,9	4
params13	0,9	5
params14	0,9	7

File 6.1 "barcodes_sem_aneabi.txt" file.

Each line corresponds to an individual followed by the barcode sequence that identifies the different individuals differing at least by 2 nucleotides. This file does not contain the *Tettigettalna aneabi* specimens. Total of individualds are 31 Tettigettalna's species.

Tma560_allo TAGCA
Tma720_allo TCAGA
Tma556_allo TCGAG
Tma559_allo TGACC
Tma718_allo TTAAT
Tma068_allo TTGGC
Tma729_allo AACCC
Tma071_allo AATTT
Tma363_sym ACCAT
Tma352_sym ACTGC
Tma354_sym AAAAA
Tma327_sym AAGGG
Tma342_sym ACACG
Tma338_sym ACGTA
Tar310_sym AGAGT
Tar335_sym AGGAC
Tar321_sym ATATC
Tar357_sym ATGCT
Tar356_sym CAACT
Tar359_sym CAGTC
Tar553_allo CCAAC
Tar552_allo CCGGT
Tar260_allo ATTAG
Tar769_allo ATCGA
Tar770_allo GTTGT
Tar526_c GGTTC
Tar528_c GGCCT
Tar547_c GCTAA
Tar529_N GCCGG
Tar177_N GACTA
Tar299_N AGTCA

File 6.2 "barcodes_aneabi.txt" file

Each line corresponds to an individual followed by the barcode sequence that identifies the different individuals differing at least by 2 nucleotides. This file contains the Tettigettalna aneabi specimens. Total of individuals are 37 *Tettigettalna's* specimens.

Tma560_allo TAGCA
Tma720_allo TCAGA
Tma556_allo TCGAG
Tma559_allo TGACC
Tma718_allo TTAAT
Tma068_allo TTGGC
Tma729_allo AACCC
Tma071_allo AATTT
Tma363_sym ACCAT
Tma352_sym ACTGC
Tma354_sym AAAAA
Tma327_sym AAGGG
Tma342_sym ACACG
Tma338_sym ACGTA
Tar310_sym AGAGT
Tar335_sym AGGAC
Tar321_sym ATATC
Tar357_sym ATGCT
Tar356_sym CAACT
Tar359_sym CAGTC
Tar553_allo CCAAC
Tar552_allo CCGGT
Tar260_allo ATTAG
Tar769_allo ATCGA
Tar770_allo GTTGT
Tar526_c GGTTC
Tar528_c GGCCT
Tar547_c GCTAA
Tar529_N GCCGG
Tar177_N GACTA
Tar299_N AGTCA
Tan254 AGCTG
Tan710 CTTCC
Tan709 CTCTT
Tan707 CGTAT
Tan3755 CGCGC
Tan711 CCTTG

File 6.3 “barcodes_com_josei.txt” file

Each line corresponds to an individual followed by the barcode sequence that identifies the different individuals differing at least by 2 nucleotides. Total of individuals are 21 *Tettigettalna*’s specimens.

Tma068_allo TTGGC
Tma729_allo AACCC
Tma071_allo AATTT
Tma363_sym ACCAT
Tma352_sym ACTGC
Tma354_sym AAAAA
Tma327_sym AAGGG
Tma342_sym ACACG
Tma338_sym ACGTA
Tar310_sym AGAGT
Tar335_sym AGGAC
Tar321_sym ATATC
Tar357_sym ATGCT
Tar356_sym CAACT
Tar359_sym CAGTC
Tar553_allo CCAAC
Tar552_allo CCGGT
Tar260_allo ATTAG
Tar769_allo ATCGA
Tar770_allo GTTGT
Tjo3765 CCCCA

File 6.4 “individuos_algarve.txt” file

Each line corresponds the population name of each individual. The presented order is the same order of the individuals in the *assembly_name*CenterSNP.vcf files with *Tettigettalna aneabi* individuals.

Tan
Tan
Tan
Tan
Tan
Tan
Tar_N
Tar_allo
Tar_N
Tar_sym
Tar_sym
Tar_sym
Tar_sym
Tar_sym
Tar_sym
Tar_c
Tar_c
Tar_N
Tar_c
Tar_allo
Tar_allo
Tar_allo
Tar_allo
Tma_sym
Tma_sym
Tma_sym
Tma_sym
Tma_sym
Tma_sym
Tma_sym
Tma_sym
Tma_allo
Tma_allo
Tma_allo
Tma_allo
Tma_allo
Tma_sym

File 6.5 “individuos_sem_aneabi_algarve.txt” file

Each line corresponds the population name of each individual. The presented order is the same order of the individuals in the *assembly_name*CenterSNP.vcf files without *Tettigettalna aneabi* individuals.

Tar_N
Tar_allo
Tar_N
Tar_sym
Tar_sym
Tar_sym
Tar_sym
Tar_sym
Tar_sym
Tar_c
Tar_c
Tar_N
Tar_c
Tar_allo
Tar_allo
Tar_allo
Tar_allo
Tma_sym
Tma_sym
Tma_sym
Tma_sym
Tma_sym
Tma_sym
Tma_sym
Tma_sym
Tma_allo
Tma_allo
Tma_allo
Tma_allo
Tma_allo
Tma_sym

File 6.6 “params-params.txt” file

File containing all the values for each of the parameters for the Ipyrad assembly params.

```
------- ipyrad params file (v.0.7.15)-------------------------------------------
params                    ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final  ## [1] [project_dir]: Project dir (made in curdir if not present)
                          ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_aneabi.txt   ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final/params_fastqs/*.gz ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                    ## [5] [assembly_method]: Assembly method (denovo, reference, denovo+reference, denovo-reference)
                          ## [6] [reference_sequence]: Location of reference sequence file
rad                       ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
TGCAG,                    ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                         ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                        ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                         ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                         ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                     ## [13] [maxdepth]: Max cluster depth within samples
0.85                      ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                         ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
0                         ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                        ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                         ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
5, 5                      ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus (R1, R2)
8, 8                      ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus (R1, R2)
4                         ## [21] [min_samples_locus]: Min # samples per locus for output
20, 20                    ## [22] [max_SNPs_locus]: Max # SNPs per locus (R1, R2)
8, 8                      ## [23] [max_Indels_locus]: Max # of indels per locus (R1, R2)
0.5                       ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus (R1, R2)
0, 0, 0, 0                ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, v                   ## [27] [output_formats]: Output formats (see docs)
                          ## [28] [pop_assign_file]: Path to population assignment file
```

File 6.7 “params-params1.txt” file

File containing all the values for each of the parameters for the Ipyrad assembly params1.

```
------- ipyrad params file (v.0.7.15)-------------------------------------------
params1                    ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final  ## [1] [project_dir]: Project dir (made in curdir if not present)
                           ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_aneabi.txt  ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final/params_fastqs/*.gz ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                     ## [5] [assembly_method]: Assembly method (denovo, reference, denovo+reference, denovo-reference)
                           ## [6] [reference_sequence]: Location of reference sequence file
rad                        ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
TGCAG,                     ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                          ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                         ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                          ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                          ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                      ## [13] [maxdepth]: Max cluster depth within samples
0.90                       ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                          ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
0                          ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                         ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                          ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
5, 5                       ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus (R1, R2)
8, 8                       ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus (R1, R2)
4                          ## [21] [min_samples_locus]: Min # samples per locus for output
20, 20                     ## [22] [max_SNPs_locus]: Max # SNPs per locus (R1, R2)
8, 8                       ## [23] [max_Indels_locus]: Max # of indels per locus (R1, R2)
0.5                        ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus (R1, R2)
0, 0, 0, 0                 ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                 ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, v                    ## [27] [output_formats]: Output formats (see docs)
                           ## [28] [pop_assign_file]: Path to population assignment file
```

File 6.8 “params-params2.txt” file

File containing all the values for each of the parameters for the Ipyrad assembly params2.

```
------- ipyrad params file (v.0.7.15)-------------------------------------------
params2                    ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final       ## [1] [project_dir]: Project dir (made in curdir if not present)
                           ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_aneabi.txt   ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final/params_fastqs/*.gz    ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                     ## [5] [assembly_method]: Assembly method (denovo, reference, denovo+reference, denovo-reference)
                           ## [6] [reference_sequence]: Location of reference sequence file
rad                        ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
TGCAG,                     ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                          ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                         ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                          ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                          ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                      ## [13] [maxdepth]: Max cluster depth within samples
0.95                       ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                          ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
0                          ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                         ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                          ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
5, 5                       ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus (R1, R2)
8, 8                       ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus (R1, R2)
4                          ## [21] [min_samples_locus]: Min # samples per locus for output
20, 20                     ## [22] [max_SNPs_locus]: Max # SNPs per locus (R1, R2)
8, 8                       ## [23] [max_Indels_locus]: Max # of indels per locus (R1, R2)
0.5                        ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus (R1, R2)
0, 0, 0, 0                 ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                 ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, v                    ## [27] [output_formats]: Output formats (see docs)
                           ## [28] [pop_assign_file]: Path to population assignment file
```

File 6.9 “params-params3.txt" file

File containing all the values for each of the parameters for the Ipyrad assembly params3.

```
------- ipyrad params file (v.0.7.15)-------------------------------------------
params3                    ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final       ## [1] [project_dir]: Project dir (made in curdir if not present)
                           ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_aneabi.txt   ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final/params_fastqs/*.gz  ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                     ## [5] [assembly_method]: Assembly method (denovo, reference, denovo+reference, denovo-reference)
                           ## [6] [reference_sequence]: Location of reference sequence file
rad                        ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
TGCAG,                     ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                          ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                         ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                          ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                          ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                      ## [13] [maxdepth]: Max cluster depth within samples
0.80                       ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                          ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
0                          ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                         ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                          ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
5, 5                       ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus (R1, R2)
8, 8                       ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus (R1, R2)
4                          ## [21] [min_samples_locus]: Min # samples per locus for output
20, 20                     ## [22] [max_SNPs_locus]: Max # SNPs per locus (R1, R2)
8, 8                       ## [23] [max_Indels_locus]: Max # of indels per locus (R1, R2)
0.5                        ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus (R1, R2)
0, 0, 0, 0                 ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                 ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, v                    ## [27] [output_formats]: Output formats (see docs)
                           ## [28] [pop_assign_file]: Path to population assignment file
```

File 6.10 “params-params4.txt” file

File containing all the values for each of the parameters for the Ipyrad assembly params4.

```
------- ipyrad params file (v.0.7.15)-------------------------------------------
params4                      ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final        ## [1] [project_dir]: Project dir (made in curdir if not present)
                             ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_aneabi.txt   ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final/params_fastqs/*.gz  ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                       ## [5] [assembly_method]: Assembly method (denovo, reference, denovo+reference, denovo-reference)
                             ## [6] [reference_sequence]: Location of reference sequence file
rad                          ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
TGCAG,                       ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                            ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                           ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                            ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                            ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                        ## [13] [maxdepth]: Max cluster depth within samples
0.85                         ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                            ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
0                            ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                           ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                            ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
5, 5                         ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus (R1, R2)
8, 8                         ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus (R1, R2)
5                            ## [21] [min_samples_locus]: Min # samples per locus for output
20, 20                       ## [22] [max_SNPs_locus]: Max # SNPs per locus (R1, R2)
8, 8                         ## [23] [max_Indels_locus]: Max # of indels per locus (R1, R2)
0.5                          ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus (R1, R2)
0, 0, 0, 0                   ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                   ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, v                      ## [27] [output_formats]: Output formats (see docs)
                             ## [28] [pop_assign_file]: Path to population assignment file
```

File 6.11 “params-params5.txt” file

File containing all the values for each of the parameters for the Ipyrad assembly params5.

```
------- ipyrad params file (v.0.7.15)-------------------------------------------
params5                    ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final       ## [1] [project_dir]: Project dir (made in curdir if not present)
                           ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_aneabi.txt   ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final/params_fastqs/*.gz  ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                     ## [5] [assembly_method]: Assembly method (denovo, reference, denovo+reference, denovo-reference)
                           ## [6] [reference_sequence]: Location of reference sequence file
rad                        ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
TGCAG,                     ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                          ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                         ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                          ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                          ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                      ## [13] [maxdepth]: Max cluster depth within samples
0.90                       ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                          ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
0                          ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                         ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                          ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
5, 5                       ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus (R1, R2)
8, 8                       ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus (R1, R2)
5                          ## [21] [min_samples_locus]: Min # samples per locus for output
20, 20                     ## [22] [max_SNPs_locus]: Max # SNPs per locus (R1, R2)
8, 8                       ## [23] [max_Indels_locus]: Max # of indels per locus (R1, R2)
0.5                        ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus (R1, R2)
0, 0, 0, 0                 ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                 ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, v                    ## [27] [output_formats]: Output formats (see docs)
                           ## [28] [pop_assign_file]: Path to population assignment file
```

File 6.12 “params-params6.txt” file

File containing all the values for each of the parameters for the Ipyrad assembly params6.

```
------- ipyrad params file (v.0.7.15)-------------------------------------------
params6                    ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final       ## [1] [project_dir]: Project dir (made in curdir if not present)
                           ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_aneabi.txt   ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final/params_fastqs/*.gz   ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                     ## [5] [assembly_method]: Assembly method (denovo, reference, denovo+reference, denovo-reference)
                           ## [6] [reference_sequence]: Location of reference sequence file
rad                        ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
TGCAG,                     ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                          ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                         ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                          ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                          ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                      ## [13] [maxdepth]: Max cluster depth within samples
0.95                       ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                          ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
0                          ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                         ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                          ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
5, 5                       ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus (R1, R2)
8, 8                       ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus (R1, R2)
5                          ## [21] [min_samples_locus]: Min # samples per locus for output
20, 20                     ## [22] [max_SNPs_locus]: Max # SNPs per locus (R1, R2)
8, 8                       ## [23] [max_Indels_locus]: Max # of indels per locus (R1, R2)
0.5                        ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus (R1, R2)
0, 0, 0, 0                 ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                 ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, v                    ## [27] [output_formats]: Output formats (see docs)
                           ## [28] [pop_assign_file]: Path to population assignment file
```

File 6.13 “params-params7.txt” file

File containing all the values for each of the parameters for the Ipyrad assembly params7.

```
------- ipyrad params file (v.0.7.15)-------------------------------------------
params7                    ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final       ## [1] [project_dir]: Project dir (made in curdir if not present)
                           ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_aneabi.txt   ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final/params_fastqs/*.gz     ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                     ## [5] [assembly_method]: Assembly method (denovo, reference, denovo+reference, denovo-reference)
                           ## [6] [reference_sequence]: Location of reference sequence file
rad                        ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
TGCAG,                     ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                          ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                         ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                          ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                          ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                      ## [13] [maxdepth]: Max cluster depth within samples
0.85                       ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                          ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
0                          ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                         ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                          ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
5, 5                       ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus (R1, R2)
8, 8                       ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus (R1, R2)
7                          ## [21] [min_samples_locus]: Min # samples per locus for output
20, 20                     ## [22] [max_SNPs_locus]: Max # SNPs per locus (R1, R2)
8, 8                       ## [23] [max_Indels_locus]: Max # of indels per locus (R1, R2)
0.5                        ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus (R1, R2)
0, 0, 0, 0                 ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                 ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, v                    ## [27] [output_formats]: Output formats (see docs)
                           ## [28] [pop_assign_file]: Path to population assignment file
```

File 6.14 “params-params8.txt” file

File containing all the values for each of the parameters for the Ipyrad assembly params8.

```
------- ipyrad params file (v.0.7.15)-------------------------------------------
params8                    ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final       ## [1] [project_dir]: Project dir (made in curdir if not present)
                           ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_aneabi.txt   ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final/params_fastqs/*.gz    ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                     ## [5] [assembly_method]: Assembly method (denovo, reference, denovo+reference, denovo-reference)
                           ## [6] [reference_sequence]: Location of reference sequence file
rad                        ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
TGCAG,                     ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                          ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                         ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                          ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                          ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                      ## [13] [maxdepth]: Max cluster depth within samples
0.90                       ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                          ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
0                          ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                         ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                          ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
5, 5                       ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus (R1, R2)
8, 8                       ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus (R1, R2)
7                          ## [21] [min_samples_locus]: Min # samples per locus for output
20, 20                     ## [22] [max_SNPs_locus]: Max # SNPs per locus (R1, R2)
8, 8                       ## [23] [max_Indels_locus]: Max # of indels per locus (R1, R2)
0.5                        ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus (R1, R2)
0, 0, 0, 0                 ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                 ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, v                    ## [27] [output_formats]: Output formats (see docs)
                           ## [28] [pop_assign_file]: Path to population assignment file
```

File 6.15 “params-params9.txt” file

File containing all the values for each of the parameters for the Ipyrad assembly params9.

```
------- ipyrad params file (v.0.7.15)-------------------------------------------
params9                      ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final       ## [1] [project_dir]: Project dir (made in curdir if not present)
                             ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_aneabi.txt   ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final/params_fastqs/*.gz   ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                       ## [5] [assembly_method]: Assembly method (denovo, reference, denovo+reference, denovo-reference)
                             ## [6] [reference_sequence]: Location of reference sequence file
rad                          ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
TGCAG,                       ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                            ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                           ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                            ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                            ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                        ## [13] [maxdepth]: Max cluster depth within samples
0.95                         ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                            ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
0                            ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                           ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                            ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
5, 5                         ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus (R1, R2)
8, 8                         ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus (R1, R2)
7                            ## [21] [min_samples_locus]: Min # samples per locus for output
20, 20                       ## [22] [max_SNPs_locus]: Max # SNPs per locus (R1, R2)
8, 8                         ## [23] [max_Indels_locus]: Max # of indels per locus (R1, R2)
0.5                          ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus (R1, R2)
0, 0, 0, 0                   ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                   ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, v                      ## [27] [output_formats]: Output formats (see docs)
                             ## [28] [pop_assign_file]: Path to population assignment file
```

File 6.16 “params-params10.txt” file

File containing all the values for each of the parameters for the Ipyrad assembly params10.

```
------- ipyrad params file (v.0.7.15)-------------------------------------------
params10                    ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final  ## [1] [project_dir]: Project dir (made in curdir if not present)
                            ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_sem_aneabi.txt  ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final/params10_fastqs/*.gz    ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                      ## [5] [assembly_method]: Assembly method (denovo, reference, denovo+reference, denovo-reference)
                            ## [6] [reference_sequence]: Location of reference sequence file
rad                         ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
TGCAG,                      ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                           ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                          ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                           ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                           ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                       ## [13] [maxdepth]: Max cluster depth within samples
0.90                        ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                           ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
0                           ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                          ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                           ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
5, 5                        ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus (R1, R2)
8, 8                        ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus (R1, R2)
7                           ## [21] [min_samples_locus]: Min # samples per locus for output
20, 20                      ## [22] [max_SNPs_locus]: Max # SNPs per locus (R1, R2)
8, 8                        ## [23] [max_Indels_locus]: Max # of indels per locus (R1, R2)
0.5                         ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus (R1, R2)
0, 0, 0, 0                  ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                  ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, v                     ## [27] [output_formats]: Output formats (see docs)
                            ## [28] [pop_assign_file]: Path to population assignment file
```

File 6.17 “params-params11.txt” file

File containing all the values for each of the parameters for the Ipyrad assembly params11.

```
------- ipyrad params file (v.0.7.15)-------------------------------------------
params11                       ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final       ## [1] [project_dir]: Project dir (made in curdir if not present)
                               ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_sem_aneabi.txt   ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final/params10_fastqs/*.gz    ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                         ## [5] [assembly_method]: Assembly method (denovo, reference, denovo+reference, denovo-reference)
                               ## [6] [reference_sequence]: Location of reference sequence file
rad                            ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
TGCAG,                         ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                              ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                             ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                              ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                              ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                          ## [13] [maxdepth]: Max cluster depth within samples
0.90                           ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                              ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
0                              ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                             ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                              ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
5, 5                           ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus (R1, R2)
8, 8                           ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus (R1, R2)
5                              ## [21] [min_samples_locus]: Min # samples per locus for output
20, 20                         ## [22] [max_SNPs_locus]: Max # SNPs per locus (R1, R2)
8, 8                           ## [23] [max_Indels_locus]: Max # of indels per locus (R1, R2)
0.5                            ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus (R1, R2)
0, 0, 0, 0                     ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                     ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, v                        ## [27] [output_formats]: Output formats (see docs)
                               ## [28] [pop_assign_file]: Path to population assignment file
```

File 6.18 “params-params12.txt” file

File containing all the values for each of the parameters for the Ipyrad assembly params12.

```
------- ipyrad params file (v.0.7.15)-------------------------------------------
params12                    ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final_josei  ## [1] [project_dir]: Project dir (made in curdir if not present)
                            ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_com_josei.txt    ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final_josei/params12_fastqs/*.gz  ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                      ## [5] [assembly_method]: Assembly method (denovo, reference, denovo+reference, denovo-reference)
                          ## [6] [reference_sequence]: Location of reference sequence file
rad                         ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
TGCAG,                        ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                         ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                        ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                         ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                         ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                       ## [13] [maxdepth]: Max cluster depth within samples
0.90                        ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                         ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
0                         ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                        ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                         ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
5, 5                      ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus (R1, R2)
8, 8                      ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus (R1, R2)
4                         ## [21] [min_samples_locus]: Min # samples per locus for output
20, 20                      ## [22] [max_SNPs_locus]: Max # SNPs per locus (R1, R2)
8, 8                      ## [23] [max_Indels_locus]: Max # of indels per locus (R1, R2)
0.5                       ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus (R1, R2)
0, 0, 0, 0                  ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                  ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, v                     ## [27] [output_formats]: Output formats (see docs)
                          ## [28] [pop_assign_file]: Path to population assignment file
```

File 6.19 “params-params13.txt” file

File containing all the values for each of the parameters for the Ipyrad assembly params13.

```
------- ipyrad params file (v.0.7.15)-------------------------------------------
params13                      ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final_josei    ## [1] [project_dir]: Project dir (made in curdir if not present)
                              ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_com_josei.txt    ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final_josei/params12_fastqs/*.gz ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                        ## [5] [assembly_method]: Assembly method (denovo, reference, denovo+reference, denovo-reference)
                              ## [6] [reference_sequence]: Location of reference sequence file
rad                           ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
TGCAG,                        ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                             ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                            ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                             ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                             ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                         ## [13] [maxdepth]: Max cluster depth within samples
0.90                          ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                             ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
0                             ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                            ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                             ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
5, 5                          ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus (R1, R2)
8, 8                          ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus (R1, R2)
5                             ## [21] [min_samples_locus]: Min # samples per locus for output
20, 20                        ## [22] [max_SNPs_locus]: Max # SNPs per locus (R1, R2)
8, 8                          ## [23] [max_Indels_locus]: Max # of indels per locus (R1, R2)
0.5                           ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus (R1, R2)
0, 0, 0, 0                    ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                    ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, v                       ## [27] [output_formats]: Output formats (see docs)
                              ## [28] [pop_assign_file]: Path to population assignment file
```

File 6.20 “params-params14.txt” file

File containing all the values for each of the parameters for the Ipyrad assembly params14.

```
------- ipyrad params file (v.0.7.15)-------------------------------------------
params14                       ## [0] [assembly_name]: Assembly name. Used to name output directories for assembly steps
/home/ioliveira/cicadas_final_josei    ## [1] [project_dir]: Project dir (made in curdir if not present)
                               ## [2] [raw_fastq_path]: Location of raw non-demultiplexed fastq files
/home/ioliveira/barcodes_com_josei.txt  ## [3] [barcodes_path]: Location of barcodes file
/home/ioliveira/cicadas_final_josei/params12_fastqs/*.gz ## [4] [sorted_fastq_path]: Location of demultiplexed/sorted fastq files
denovo                         ## [5] [assembly_method]: Assembly method (denovo, reference, denovo+reference, denovo-reference)
                               ## [6] [reference_sequence]: Location of reference sequence file
rad                            ## [7] [datatype]: Datatype (see docs): rad, gbs, ddrad, etc.
TGCAG,                         ## [8] [restriction_overhang]: Restriction overhang (cut1,) or (cut1, cut2)
5                              ## [9] [max_low_qual_bases]: Max low quality base calls (Q<20) in a read
33                             ## [10] [phred_Qscore_offset]: phred Q score offset (33 is default and very standard)
6                              ## [11] [mindepth_statistical]: Min depth for statistical base calling
6                              ## [12] [mindepth_majrule]: Min depth for majority-rule base calling
10000                          ## [13] [maxdepth]: Max cluster depth within samples
0.90                           ## [14] [clust_threshold]: Clustering threshold for de novo assembly
0                              ## [15] [max_barcode_mismatch]: Max number of allowable mismatches in barcodes
0                              ## [16] [filter_adapters]: Filter for adapters/primers (1 or 2=stricter)
35                             ## [17] [filter_min_trim_len]: Min length of reads after adapter trim
2                              ## [18] [max_alleles_consens]: Max alleles per site in consensus sequences
5, 5                           ## [19] [max_Ns_consens]: Max N's (uncalled bases) in consensus (R1, R2)
8, 8                           ## [20] [max_Hs_consens]: Max Hs (heterozygotes) in consensus (R1, R2)
7                              ## [21] [min_samples_locus]: Min # samples per locus for output
20, 20                         ## [22] [max_SNPs_locus]: Max # SNPs per locus (R1, R2)
8, 8                           ## [23] [max_Indels_locus]: Max # of indels per locus (R1, R2)
0.5                            ## [24] [max_shared_Hs_locus]: Max # heterozygous sites per locus (R1, R2)
0, 0, 0, 0                     ## [25] [trim_reads]: Trim raw read edges (R1>, <R1, R2>, <R2) (see docs)
0, 0, 0, 0                     ## [26] [trim_loci]: Trim locus edges (see docs) (R1>, <R1, R2>, <R2)
p, s, v                        ## [27] [output_formats]: Output formats (see docs)
                               ## [28] [pop_assign_file]: Path to population assignment file
```

File 6.21 Script written in python named "vcf_parser.py" that was used to filter the assembly_name.recode.vcf files

```
#!/usr/bin/python
# Copyright 2015 Diogo N. Silva <o.diogosilva@gmail.com>
# compare_pairs.py is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.

# Loci_counter is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.

# You should have received a copy of the GNU General Public License
# along with Loci_counter. If not, see <http://www.gnu.org/licenses/>.

# vcf_parser.py is that performs filtering and transformations steps on
# a VCF file that are not possible with vcftools

# Usage: python3 vcf_parser.py -h (will show all available options)

import argparse
import random
from collections import Counter

PARSER = argparse.ArgumentParser(description="Filtering of VCF files")

PARSER.add_argument("-vcf", dest="vcf_infile", help="Provide VCF "
                    " file(s).", required=True)
PARSER.add_argument("--remove-inv", dest="remove_inv", const=True,
                    action="store_const", help="Filters invariable SNP sites"
                    " from the VCF file (These sites may occurr when "
                    "individuals are removed from a VCF file).")
PARSER.add_argument("--remove-singletons", dest="remove_singletons", const=True,
                    action="store_const", help="Filters singletons SNP sites"
                    " from the VCF file.")
PARSER.add_argument("--one-snp", dest="one_snp", const=True,
                    action="store_const", help="Filters the VCF file so that"
                    " only one SNP per locus is retained - The first one")
PARSER.add_argument("--random-snp", dest="rnd_snp", const=True,
                    action="store_const", help="Filters the VCF file so that"
                    " only one random SNP per locus is retained.")
```

```
PARSER.add_argument("--center-snp", dest="center_snp", const=True,
                    action="store_const", help="Filters the VCF file so that"
                    " only the SNP closest to the locus center is retained.")

ARG = PARSER.parse_args()


def remove_sites(vcf_file, mode="inv", suffix="_NoInv.vcf"):
    """
    Removes invariable sites from a VCF file. This assumes that the genotype
    columns start at the 10th column until the last column
    :param vcf_file: string, path to vcf file
    :param mode: string, specifies the removal operation. May be one of the
    following:
        .:"inv": Removes invariable sites
        .:"singletons": Removes singleton sites
    :param suffix: string, the suffix of the output file.
    """

    vcf_output = vcf_file.split(".")[0] + suffix

    with open(vcf_file) as vcf_handle, open(vcf_output, "w") as vcf_out:

        for line in vcf_handle:

            if line.startswith("#"):
                vcf_out.write(line)

            elif line.strip() != "":

                # Get genotypes. Remove genotypes with no data.
                genotypes = [x.split(":")[0] for x in line.split()[9:] if x.split(":")[0] != "./."]

                if mode == "inv":
                    # If number of unique genotypes higher than 1, save SNP
                    if len(set(genotypes)) > 1:
                        vcf_out.write(line)

                elif mode == "singletons":
                    gens = Counter(genotypes)
                    # Remove most common genotype
                    del gens[gens.most_common()[0][0]]
                    # Check if remaining genotype count is higher than 1
                    if sum(gens.values()) > 1:
                        vcf_out.write(line)
```

```
def filter_one_snp(vcf_file):
    """
    Filters a VCF file so that only one SNP per locus (the first) is retained
    """

    vcf_output = vcf_file.split(".")[0] + "OneSNP.vcf"

    chrom_list = []

    with open(vcf_file) as vcf_handle, open(vcf_output, "w") as vcf_out:

        for line in vcf_handle:

            if line.startswith("#"):
                vcf_out.write(line)

            elif line.strip() != "":

                # Get chrom number
                chrom = line.split()[0]

                if chrom not in chrom_list:
                    vcf_out.write(line)
                    chrom_list.append(chrom)


def filter_random_snp(vcf_file):
    """
    Filters a VCF file so that only one random SNP per locus is retained.
    """

    vcf_output = vcf_file.split(".")[0] + "RandSNP.vcf"

    current_chrom = 0
    loci_snps = []

    with open(vcf_file) as vcf_handle, open(vcf_output, "w") as vcf_out:

        for line in vcf_handle:

            if line.startswith("#"):
                vcf_out.write(line)
            elif line.strip() != "":

                # Get chrom number
                chrom = line.split()[0]
```

```
            if chrom != current_chrom and loci_snps != []:
                choosen = random.choice(loci_snps)
                vcf_out.write(choosen)
                loci_snps = [line]
                current_chrom = chrom

            else:
                loci_snps += [line]
        vcf_out.write(random.choice(loci_snps))


def filter_center_snp(vcf_file):
    """
    Filters a VCF file so that only one SNP per locus (the first) is retained
    """

    vcf_output = vcf_file.split(".")[0] + "CenterSNP.vcf"

    current_chrom = ""
    line_list = []

    with open(vcf_file) as vcf_handle, open(vcf_output, "w") as vcf_out:

        for line in vcf_handle:

            if line.startswith("#"):
                vcf_out.write(line)

            elif line.strip() != "":

                # Get chrom number
                chrom = line.split()[0]

                # Get SNP position
                pos = int(line.split()[1])

                if chrom != current_chrom and current_chrom != "":
                    closest = min(pos_list, key=lambda x: abs(x - 45))
                    vcf_out.write(line_list[pos_list.index(closest)])
                    pos_list = [pos]
                    line_list = [line]
                    current_chrom = chrom
                elif chrom != current_chrom:
                    pos_list = [pos]
                    line_list = [line]
                    current_chrom = chrom
                else:
```

```
                pos_list += [pos]
                line_list += [line]

        closest = min(pos_list, key=lambda x: abs(x - 45))
        vcf_out.write(line_list[pos_list.index(closest)])

def main():
    """
    Main function that controls what to do.
    """
    # Args
    vcf_file = ARG.vcf_infile

    if ARG.remove_inv:
        remove_sites(vcf_file)

    if ARG.remove_singletons:
        remove_sites(vcf_file, mode="singletons", suffix="_NoSing.vcf")

    if ARG.one_snp:
        filter_one_snp(vcf_file)

    if ARG.rnd_snp:
        filter_random_snp(vcf_file)

    if ARG.center_snp:
        filter_center_snp(vcf_file)

main()
```

File 6.22 Script written in R named "vcf2PCA.R" used to generate Principal Component Analysis plots

```
source("http://bioconductor.org/biocLite.R")
biocLite("SNPRelate")
library ("SNPRelate")

#input
vcf.fn<-"/Users/inesoliveira/Desktop/pca_final/params11CenterSNP.vcf" #load the data
snpgdsVCF2GDS(vcf.fn,"test.gds",method="copy.num.of.ref", ignore.chr.prefix="chr")#import to
class. use method= biallelic.only for biallelic snps
#summary of the vcf imported
snpgdsSummary("test.gds")

#open the translated vcf imported
genofile<-snpgdsOpen("test.gds")

#import population names
pop_code <- scan("/Users/inesoliveira/Desktop/pca_final/individuos_sem_aneabi_algarve.txt",
what=character()) ## File 1

#PCA
pca<-snpgdsPCA(genofile, autosome.only = FALSE)
# variance proportion (%)
pc.percent<-pca$varprop*100
head(pc.percent)
#get sample ids
sample.id<-read.gdsn(index.gdsn(genofile,"sample.id"))

#data frame of eigenvectors
tab<-data.frame(sample.id= pca$sample.id,
        pop=factor(pop_code)[match(pca$sample.id, sample.id)],
        EV1= pca$eigenvect[,1],# first eignvector (can be changed)
        EV2= pca$eigenvect[,2],# the second eigenvector (can be changed)
        stringsAsFactors=FALSE)

par(mar=c(7.1, 4.1, 4.1, 9.5), xpd=TRUE)#creates space outside plot area
plot(tab$EV1, tab$EV2, col=as.integer(tab$pop), xlab="eigenvector 1",ylab="eigenvector 2",
pch=as.integer(tab$pop))
legend("topright", legend=levels(tab$pop),inset=c(-0.20,0), pch=1:nlevels(tab$pop),
col=1:nlevels(tab$pop))#inset forces legend outside area

text(tab$EV1, tab$EV2, labels=tab$sample.id)
write.csv(tab, "params11CenterSNP.csv")
```

```
#to print the eigenvalues
tab
snpgdsClose(genofile)
```

File 6.23 Script written in R named "MavericK1.0_functions.R" that came with the Maverick installation

```
# Author: Bob Verity
# Date: 30/03/2016

# Purpose:
# This R script contains a series of functions that produce simple plots from MavericK1.0 output. These functions were used to produce all the figures in the MavericK1.0 tutorial file.
# Eventually these functions will be wrapped up in a package, but for now they are free floating. Please feel free to modify them and tailor them to your needs!
# NB. Some of these functions make use of the data.table package for efficient reading in of data, and the RColorBrewer package for getting nice colour palettes.


#--- Diagnostic plots ---
# These functions can be used to produce plots that help diagnose the behaviour of an MCMC run.

# loadLikelihood() - read in an outputLikelihood.csv file as a data frame
# plotTrace() - plot simple MCMC trace of a particular column in likelihood data frame
# plotAcf() - produce autocorrelation plot of a particular column in likelihood data frame
# plotDensity() - produce kernel density plot of a particular column in likelihood data frame


#--- Qmatrix plots ---
# These functions produce posterior allocation plots from Qmatrix files. plotQmatrix_ind() and plotQmatrix_pop() accept both types of MavericK1.0 output; ordinary .csv files and Structure-style .txt files. Note that Structure-style files must have the header line stripped for these functions to work (this is done automatically in MavericK1.0 output).

# plotQmatrix_ind() - plot Qmatrix at individual level
# plotQmatrix_pop() - plot Qmatrix at population level
# plotQmatrix_gene() - plot Qmatrix at gene copy level
# plotQmatrixError_ind() - plot QmatrixError at individual level
# plotQmatrixError_pop() - plot QmatrixError at population level


#--- Evidence plots ---
# These functions read in outputEvidence.csv and outputEvidenceNormalised.csv files and produce simple plots over K.

# plotEvidence() - plot values in outputEvidence.csv file
# plotEvidenceNormalised() - plot values in outputEvidenceNormalised.csv file
# plotTI_path() - plot TIpoint_mean values from the outputEvidenceDetails.csv file


#--- Compare models ---
# model_normalise() - read in outputEvidence.csv files from multiple evolutionary models and calculates overall evidence, integrated over K.
# plotModel() - create barplot of values returned by model_normalise()
```

```
#--- Auxilliary functions ---
# These functions are called within the main functions listed above

# Qmatrix_ind_to_csv() - converts Structure-style individual Qmatrix into ordinary data frame
# Qmatrix_pop_to_csv() - converts Structure-style population Qmatrix into ordinary data frame
# defaultColours() - specifies some default colours from the RColorBrewer package
# errorBars() - add error bars to plot

# ------------------------------------------------------------------

## loadLikelihood
# Reads in likelihood file using the efficient fread() function from package "data.table". Gives an optional summary of file contents.

# fileName - name of likelihood file
# printSummary - whether to print basic file properties to console

#setwd("/home/inesoliveira/Desktop/pilot_run")

loadLikelihood <- function(fileName=file.choose(), printSummary=TRUE) {

	# load data.table package
	if (!"data.table"%in%rownames(installed.packages()))
		install.packages("data.table")
	require("data.table")

	# read in likelihood file
	df <- as.data.frame(fread(fileName))

	# summarise contents
	if (printSummary) {
		Kmin <- min(df$K)
		Kmax <- max(df$K)
		if (Kmin==Kmax) {
			cat(paste('K\t\t = ', Kmin, '\n', sep=''))
		} else {
			cat(paste('K\t\t = ', Kmin, ':', max(df$K), '\n', sep=''))
		}
		cat(paste('mainReps = ', max(df$mainRep), '\n', sep=''))
		burnin = 0
		if (any(df$MCMCsample<1))
			burnin = -min(df$MCMCsample)+1
		cat(paste('burn-in\t = ', burnin, '\n', sep=''))
		samples <- max(df$MCMCsample)
		cat(paste('samples\t = ', samples, '\n', sep=''))
	}

	return(df)
}

myLike <- loadLikelihood("./outputLikelihood.csv")

# ----------------------------------
```

```
## plotTrace
# Takes likelihood data frame (for example read in using loadLikelihood()) and produces trace plot of chosen column. Data frame is subset to a single value of K and a single mainRep prior to plotting.

# likelihood_data - data frame containing values formatted as in outputLikelihood.csv
# K - single value of K to plot
# mainRep - single value of mainRep to plot
# trimBurnin - whether to remove burn-in iterations prior to plotting
# column - which column in data do we want to plot
# xlab - x axis label
# ylab - y axis label
# main - main title

plotTrace <- function(likelihood_data, K, mainRep=1, trimBurnin=TRUE, column='loglike_marginal', xlab='iteration', ylab=column, main=column) {

	# ensure inputs are scalar valued
	K_ <- K[1]
	mainRep_ <- mainRep[1]

	# subset data frame and get plotting values
	df <- subset(likelihood_data, K==K_ & mainRep==mainRep_)
	if (trimBurnin) {
		df <- subset(df, MCMCsample>0)
	}
	if (0%in%dim(df)) {
		stop('once subset by K and mainRep likelihood data frame contains no fields')
	}
	if (column%in%names(df)) {
		y <- df[,column]
	} else {
		stop(paste('could not find column "',column,'" in likelihood file',sep=''))
	}

	# produce plot
	plot(df$MCMCsample, y, pch=4, cex=0.4, col=grey(0.7), xlab=xlab, ylab=ylab, main=main)
	if (!trimBurnin) {
		abline(v=0,lty=3)
	}
}

plotTrace(myLike, K=7, trimBurnin = FALSE)
plotTrace(myLike, K=7, column="alpha", trimBurnin=FALSE)

# ----------------------------------
## plotAcf
# Takes likelihood data frame (for example read in using loadLikelihood()) and produces an autocorrelation plot of chosen column. Data frame is subset to a single value of K and a single mainRep, and burn-in is discarded prior to plotting. The number of lags is chosen automatically.

# likelihood_data - data frame containing values formatted as in outputLikelihood.csv
# K - single value of K to plot
# mainRep - single value of mainRep to plot
# column - which column in data do we want to plot
```

```
# main - main title

plotAcf <- function(likelihood_data, K, mainRep=1, column='loglike_marginal', main='') {

	# ensure inputs are scalar valued
	K_ <- K[1]
	mainRep_ <- mainRep[1]

	# subset data frame and get plotting values
	df <- subset(likelihood_data, K==K_ & mainRep==mainRep_ & MCMCsample>0)
	if (0%in%dim(df)) {
		stop('once subset by K and mainRep likelihood data frame contains no fields')
	}
	if (column%in%names(df)) {
		y <- df[,column]
	} else {
		stop(paste('could not find column "',column,'" in likelihood file',sep=''))
	}

	# search various values of lag.max. Stop when autocorrelation drops below zero.
	lagMax_vec <- 10^(1:5)
	for (i in 1:length(lagMax_vec)) {
		lagMax <- lagMax_vec[i]
		c <- acf(y, lag.max=lagMax, plot=FALSE)
		c <- as.vector(c$acf)
		if (any(c<0)) {
			w <- which(c<0)[1]
			lagMax <- max(2*w, 30)
			break
		}
	}

	# final acf with chosen lags
	c <- acf(y, lag.max=lagMax, plot=FALSE)
	c <- as.vector(c$acf)

	# produce plot
	plot(0:(length(c)-1), c, type='h', ylim=c(min(c,na.rm=TRUE),1), xlab='Lag', ylab='ACF',
main=main)
	abline(h=0)
}

plotAcf(myLike, K=7)

# ---------------------------------
## plotDensity
# Takes likelihood data frame (for example read in using loadLikelihood()) and produces density plot
of chosen column. Data frame is subset to a single value of K and a single mainRep prior to plotting.

# likelihood_data - data frame containing values formatted as in outputLikelihood.csv
# K - single value of K to plot
# mainRep - single value of mainRep to plot
# trimBurnin - whether to remove burn-in iterations prior to plotting
# column - which column in data do we want to plot
```

```
# xlab - x axis label
# ylab - y axis label
# main - main title

plotDensity <- function(likelihood_data, K, mainRep=1, trimBurnin=TRUE,
column='loglike_marginal', xlab=column, ylab='probability density', main=column) {

	# ensure inputs are scalar valued
	K_ <- K[1]
	mainRep_ <- mainRep[1]
	# subset data frame and get plotting values
	df <- subset(likelihood_data, K==K_ & mainRep==mainRep_)
	if (trimBurnin) {
		df <- subset(df, MCMCsample>0)
	}
	if (0%in%dim(df)) {
		stop('once subset by K and mainRep likelihood data frame contains no fields')
	}
	if (column%in%names(df)) {
		x <- df[,column]
	} else {
		stop(paste('could not find column "',column,'" in likelihood file',sep=''))
	}

	# get plotting limits based on range of data
	xmin <- 1.5*min(x,na.rm=T)-0.5*max(x,na.rm=T)
	xmax <- 1.5*max(x,na.rm=T)-0.5*min(x,na.rm=T)
	if (column=='alpha') {
		xmin <- 0
		xmax <- min(xmax,10)
	}

	# plot kernel density
	fx = density(x,from=xmin,to=xmax)
	plot(fx,xlab=xlab,ylab=ylab,main=main)
	polygon(c(fx$x,rev(fx$x)), c(fx$y,rep(0,length(fx$y))), col=grey(0.7))

	# add quantiles to density plot
	quantiles = quantile(x,prob=c(0.025,0.975))
	fx_left_x = fx$x[fx$x<quantiles[1]]
	fx_left_y = fx$y[fx$x<quantiles[1]]
	polygon(c(fx_left_x,rev(fx_left_x)), c(fx_left_y,rep(0,length(fx_left_y))), col=colors()[373])
	fx_right_x = fx$x[fx$x>quantiles[2]]
	fx_right_y = fx$y[fx$x>quantiles[2]]
	polygon(c(fx_right_x,rev(fx_right_x)), c(fx_right_y,rep(0,length(fx_right_y))),
col=colors()[373])
	# add lines for median and mean
	best_median = which.min(abs(fx$x-median(x)))
	lines(rep(fx$x[best_median],2),c(0,fx$y[best_median]))
	best_mean = which.min(abs(fx$x-mean(x)))
	lines(rep(fx$x[best_mean],2),c(0,fx$y[best_mean]),lty=2)

	# add legend
```

```
	legend('topright', legend=paste(c('   mean=','median=','       sd=',' q0.025=',' q0.975='),
signif(c(mean(x), median(x), sd(x), quantiles[1],quantiles[2]), digits=3), sep=''),
lty=c(2,1,NA,NA,NA), seg.len=1, bg='#FFFFFF80', box.col='#00000050')

}

plotDensity(myLike, K=7, column="alpha")

######  Corri ate aqui para os restantes K's

# ----------------------------------
## Qmatrix_ind_to_csv
# Convert Structure format individual level Qmatrix to ordinary data frame.

# fileName - name of Qmatrix file

Qmatrix_ind_to_csv <- function(fileName) {

	# read in data
	Qmatrix <- read.delim(file=fileName,header=FALSE,sep='')

	# check whether using population info
	popCol_on <- TRUE
	if (Qmatrix[1,4]==":") popCol_on <- FALSE

	# produce data frame
	n <- nrow(Qmatrix)
	K <- ncol(Qmatrix)-4-popCol_on
	df <- data.frame(index=1:n, label=Qmatrix[,2])
	if (popCol_on) df$given_population <- as.factor(as.character(Qmatrix[,4]))
	df <- cbind(df,Qmatrix[,(ncol(Qmatrix)-K+1):ncol(Qmatrix)])
	names(df)[-(1:(2+popCol_on))] <- paste("deme",1:K,sep="")

	return(df)
}

# ----------------------------------
## Qmatrix_pop_to_csv
# Convert Structure format population level Qmatrix to ordinary data frame.

# fileName - name of Qmatrix file

Qmatrix_pop_to_csv <- function(fileName) {

	# read in data
	Qmatrix <- read.delim(file=fileName,header=FALSE,sep='')

	# produce data frame
	K <- ncol(Qmatrix)-2
	popNames <- unlist(strsplit(as.character(Qmatrix[,1]),":"))
	members <- as.numeric(as.character(Qmatrix[,K+2]))
	vals <- matrix(as.numeric(as.matrix(Qmatrix[,2:(K+1)])),length(popNames))
	df <- data.frame(given_population=popNames, members=members)
	df <- cbind(df, vals)
```

```
	names(df)[-(1:2)] <- paste("deme",1:K,sep="")

	return(df)
}

# ---------------------------------
## defaultColours
# Generate vector of K default colours using RColorBrewer package. Uses colorRampPalette function, meaning any value of K is supported.

# K - number of colours to produce
defaultColours <- function(K) {

	# load RColorBrewer package
	if (!"RColorBrewer"%in%rownames(installed.packages()))
			install.packages("RColorBrewer")
	require("RColorBrewer")

	# generate palette and colours
	myPalette <- colorRampPalette(brewer.pal(n=6,name="RdYlBu"))
	barCol <- myPalette(max(K,6))

	# if less than 6 colours then choose manually
	if (K==5) {
		barCol = barCol[c(1,2,3,5,6)]
	} else if (K==4) {
		barCol = barCol[c(1,2,3,5)]
	} else if (K==3) {
		barCol = barCol[c(1,3,5)]
	} else if (K==2) {
		barCol = barCol[c(1,5)]
	} else if (K==1) {
		barCol = barCol[1]
	}

	return(barCol)
}

# ---------------------------------
## plotQmatrix_ind
# Reads in a Qmatrix_ind file and produces a posterior allocation plot. Features such as border widths and colours can be set manually.

# fileName - path to input file
# StructureFormat - whether input file is in Structure format
# barCol - vector or colours for different demes. Leave blank to use default colours
# barBorderCol - colour of border around individual bars
# barBorderWidth - width of border around individual bars
# popBorderCol - colour of border around given populations
# popBorderWidth - width of border around given populations
# allBorderCol - colour of border around entire plot
# allBorderWidth - width of border around entire plot
# xlab - x axis label
# ylab - y axis label
```

```
# main - main title
# printK - whether to print value of K in top left corner

plotQmatrix_ind <- function(fileName=file.choose(), StructureFormat=FALSE, barCol=NA,
barBorderCol='white', barBorderWidth=0.5, popBorderCol='black', popBorderWidth=2,
allBorderCol='black', allBorderWidth=2, xlab='', ylab='', main='', printK=TRUE) {

	# read in Qmatrix file
	if (StructureFormat) df <- Qmatrix_ind_to_csv(fileName)
	else df <- read.csv(fileName)

	# extract basic quantities
	popCol_on <- "given_population"%in%names(df)
	if (popCol_on) {
		pop = df$given_population
		pops = length(unique(pop))
	}
	n = nrow(df)
	K = ncol(df)-2-popCol_on
	Q <- as.matrix(df[,(3+popCol_on):ncol(df)])

	# generate bar colours
	if (any(is.na(barCol)))
		barCol <- defaultColours(K)

	# produce barplot
	oldPar <- par(lwd=barBorderWidth, xpd=TRUE)
	barplot(t(Q), border=barBorderCol, col=barCol, space=0, ylim=c(-0.1,1.1), xlab=xlab,
ylab=ylab, main=main, axes=FALSE, names.arg=rep(NA,n), font.main=1)
	# if population data present
	if (popCol_on) {

		# count members in each population
		members <- NULL
		for (i in 1:pops)
			members = c(members,sum(pop==unique(pop)[i]))

		# add border around populations
		for (i in 1:pops) {
			xmax <- cumsum(members)[i]
			xmin <- xmax-members[i]
			polygon(c(xmin,xmax,xmax,xmin), c(0,0,1,1), lwd=popBorderWidth,
border=popBorderCol)
		}

		# add population labels
		midPoints = cumsum(members)-members/2
		text(midPoints, 0, pos=1, labels=unique(pop), cex=0.8)
	}

	# add outer frame
	polygon(c(0,n,n,0),c(0,0,1,1), lwd=allBorderWidth, border=allBorderCol)

	# print K
```

```
	if (printK)
		text(0,1,labels=paste('	K=',K,sep=''),pos=3)

	# restore old parameter values
	par(oldPar)
}

plotQmatrix_ind("./outputQmatrix_ind_K7.csv")

# -----------------------------------
## plotQmatrix_pop
# Reads in a Qmatrix_pop file and produces a posterior allocation plot. Features such as border widths and colours can be set manually.

# fileName - path to input file
# StructureFormat - whether input file is in Structure format
# barCol - vector or colours for different demes. Leave blank to use default colours
# barBorderCol - colour of border around individual bars
# barBorderWidth - width of border around individual bars
# popBorderCol - colour of border around given populations
# popBorderWidth - width of border around given populations
# allBorderCol - colour of border around entire plot
# allBorderWidth - width of border around entire plot
# xlab - x axis label
# ylab - y axis label
# main - main title
# printK - whether to print value of K in top left corner

plotQmatrix_pop = function(fileName=file.choose(), StructureFormat=FALSE, barCol=NA, barBorderCol='white', barBorderWidth=0.5, popBorderCol='black', popBorderWidth=2, allBorderCol='black', allBorderWidth=2, xlab='', ylab='', main='', printK=TRUE) {

	# read in Qmatrix file
	if (StructureFormat) df <- Qmatrix_pop_to_csv(fileName)
	else df <- read.csv(fileName)

	# extract basic quantities
	pops = nrow(df)
	n <- sum(df$members)
	K = ncol(df)-2
	Q <- as.matrix(df[,3:ncol(df)])

	# generate bar colours
	if (any(is.na(barCol)))
		barCol <- defaultColours(K)

	# produce barplot
	oldPar <- par(lwd=barBorderWidth, xpd=TRUE)
	barplot(t(Q), df$members, border=barBorderCol, col=barCol, space=0, ylim=c(-0.1,1.1), xlab=xlab, ylab=ylab, main=main, axes=FALSE, names.arg=rep(NA,pops), font.main=1)

	# add population labels
	midPoints = cumsum(df$members)-df$members/2
	text(midPoints, 0, pos=1, labels=df$given_population, cex=0.8)
```

```
	# add border around populations
	for (i in 1:pops) {
		xmax <- cumsum(df$members)[i]
		xmin <- xmax-df$members[i]
		polygon(c(xmin,xmax,xmax,xmin), c(0,0,1,1), lwd=popBorderWidth,
border=popBorderCol)
	}

	# add outer frame
	polygon(c(0,n,n,0),c(0,0,1,1), lwd=allBorderWidth, border=allBorderCol)

	# print K
	if (printK)
		text(0,1,labels=paste('	K=',K,sep=''),pos=3)

	# restore old parameter values
	par(oldPar)
}

# ----------------------------------
## plotQmatrix_gene
# Reads in a Qmatrix_gene file and produces a posterior allocation plot. Features such as border
widths and colours can be set manually. This function is fairly clunky, and may produce odd-looking
plots for very large data sets.

# fileName - path to input file
# xrange - left and right limits (as a proportion of total window width) occupied by bar plots
# yrange - bottom and top limits (as a proportion of total window height) occupied by bar plots
# barCol - vector or colours for different demes. Leave blank to use default colours
# barBorderCol - colour of border around individual bars
# barBorderWidth - width of border around individual bars
# geneBorderCol - colour of border around all loci corresponding to a particular gene copy
# geneBorderWidth - width of border around all loci corresponding to a particular gene copy
# indBorderCol - colour of border around each individual
# indBorderWidth - width of border around each individual
# indSpace - vertical space (as a proportion of row size) between individuals
# indLabels - whether to plot labels of each individual
# indFontSize - size of individual labels
# popSpace - vertical space (as a proportion of row size) between given populations
# xlab - x axis label
# xFontSize - size of x axis label
# main - main title

plotQmatrix_gene = function(fileName=file.choose(), xrange=c(0.2,0.8), yrange=c(0.1,0.9),
barCol=NA, barBorderCol='white', barBorderWidth=0.25, geneBorderCol='black',
geneBorderWidth=0.5, indBorderCol='black', indBorderWidth=1, indSpace=0.25, indLabels=TRUE,
indFontSize=0.5, popSpace=2, xlab='loci', xFontSize=1, main='') {

	# read in Qmatrix file
	df <- read.csv(fileName)

	# extract basic quantities
	popCol_on <- ("given_population"%in%names(df))
```

```
	pops <- NULL
	if (popCol_on) {
		pops <- unique(df$given_population)
		df <- df[order(df$given_population),]
	}
	n <- max(df$index)
	K <- ncol(df)-5
	loci <- max(df$locus)
	ploidy <- unlist(lapply(split(df$gene_copy, f=df$index),max))

	# generate bar colours
	if (any(is.na(barCol)))
		barCol <- defaultColours(K)

	# calculate size of each row
	delta_y <- diff(yrange)/(sum(ploidy)+(n-1)*indSpace+(length(pops)-1)*popSpace)

	# create empty plot
	oldPar <- par(fig=c(0,1,0,1), mar=c(0,0,0,0), lwd=barBorderWidth, xpd=TRUE)
	plot(1, type='n', xlim=c(0,1), ylim=c(0,1), xaxs='i', yaxs='i', axes=FALSE, xlab=NA,
ylab=NA)

	# loop through all individuals and all gene copies within an individual
	if (popCol_on)
		thisPop <- df$given_population[1]
	ymax <- yrange[2]
	for (ind in 1:n) {
		df_ind <- subset(df,index==ind)
		for (copy in 1:ploidy[ind]) {

			#??add space between populations
			df_copy <- subset(df_ind,gene_copy==copy)
			if (popCol_on) {
				if (df_copy$given_population[1]!=thisPop) {
					ymax = ymax - delta_y*popSpace
					thisPop <- df_copy$given_population[1]
				}
			}

			# define plotting frame for this row
			par(new=TRUE, fig=c(xrange[1],xrange[2],ymax-delta_y,ymax),
mar=c(0,0,0,0))

			# barplot for this row
			Q <- as.matrix(df_copy[,5:(4+K)+popCol_on])
			barplot(t(Q), names.arg=rep(NA,loci), border=barBorderCol, col=barCol,
space=0, ylim=c(0,1), xaxs='i', yaxs='i', axes=F)

			# reset plotting frame
			par(new=TRUE, fig=c(0,1,0,1), mar=c(0,0,0,0))
			plot(1, type='n', xlim=c(0,1), ylim=c(0,1), xaxs='i', yaxs='i', axes=FALSE,
xlab=NA, ylab=NA)

			# add border around all loci corresponding to this gene copy
```

```
                        polygon(c(xrange[1],xrange[2],xrange[2],xrange[1]), c(ymax-delta_y,ymax-
delta_y,ymax,ymax), lwd=geneBorderWidth, border=geneBorderCol)

                        # move down one row
                        ymax <- ymax - delta_y
                }

                # add border around this individual
                polygon(c(xrange[1],xrange[2],xrange[2],xrange[1]), c(ymax, ymax,
ymax+ploidy[ind]*delta_y, ymax+ploidy[ind]*delta_y), lwd=indBorderWidth, border=indBorderCol)

                # add individual label
                text(xrange[1],ymax+ploidy[ind]/2*delta_y, labels=as.character(df_ind$label[1]),
pos=2, cex=indFontSize)

                # create space between individuals
                ymax <- ymax - delta_y*indSpace
        }

        # add title and x axis label
        par(new=TRUE, fig=c(0,1,0,1), mar=c(0,0,0,0))
        plot(1, type='n', xlim=c(0,1), ylim=c(0,1), xaxs='i', yaxs='i', axes=FALSE, xlab=NA,
ylab=NA)
        text(mean(xrange),yrange[1],labels=xlab,cex=xFontSize,pos=1)
        text(mean(xrange),yrange[2],labels=main,pos=3)

        # restore old parameter values
        par(oldPar)
}

# ----------------------------------
## plotQmatrixError_ind
# Reads in a QmatrixError_ind file and plots the maximum standard error associated with each
individual assignment.

# fileName - path to input file
# barCol - colour of bars
# shadePops - whether to use alternating shading to represent different populations
# xlab - x axis label
# ylab - y axis label
# main - main title
# printK - whether to print value of K in top left corner

plotQmatrixError_ind <- function(fileName=file.choose(), barCol=NA, shadePops=TRUE, xlab='',
ylab='standard error', main='', printK=TRUE) {

        # read in Qmatrix file
        df <- read.csv(fileName)

        # extract basic quantities
        popCol_on <- "given_population"%in%names(df)
        if (popCol_on) {
                #df <- df[order(df$given_population),]
                pop = df$given_population
```

```
		popIndex <- match(pop,unique(pop))
		pops = length(unique(pop))
	}
	n = nrow(df)
	K = ncol(df)-2-popCol_on
	Q <- as.matrix(df[,(3+popCol_on):ncol(df)])

	# generate bar colours
	barCol <- barCol[1]
	if (is.na(barCol))
		barCol <- "#D73027"

	# produce plot
	maxError <- apply(Q,1,max)
	barplot(maxError, space=0, ylim=c(0,1.2*max(maxError)), col=barCol, xlab=xlab, ylab=ylab,
main=main)
	if (popCol_on & shadePops) {
		par(new=TRUE)
		barplot(maxError, space=0, ylim=c(0,1.2*max(maxError)), density=30*(1-
popIndex%%2), col='white', ann=FALSE, axes=FALSE)
	}

	# print K
	oldPar <- par(xpd=TRUE)
	if (printK)
		text(0,par('usr')[4],labels=paste('      K=',K,sep=''),pos=3)

	# restore old parameter values
	par(oldPar)
}

plotQmatrixError_ind("./outputQmatrixError_ind_K7.csv")

# ------------------------------------
## plotQmatrixError_pop
# Reads in a QmatrixError_pop file and plots the maximum standard error associated with each
population assignment.

# fileName - path to input file
# barCol - colour of bars
# shadePops - whether to use alternating shading to represent different populations
# xlab - x axis label
# ylab - y axis label
# main - main title
# printK - whether to print value of K in top left corner

plotQmatrixError_pop <- function(fileName=file.choose(), barCol=NA, shadePops=TRUE, xlab='',
ylab='standard error', main='', printK=TRUE) {

	# read in Qmatrix file
	df <- read.csv(fileName)

	# extract basic quantities
	pops = nrow(df)
```

```
	pop = df$given_population
	popIndex <- match(pop,unique(pop))
	K = ncol(df)-2
	Q <- as.matrix(df[,3:ncol(df)])

	# generate bar colours
	barCol <- barCol[1]
	if (is.na(barCol))
		barCol <- "#D73027"

	# produce plot
	maxError <- apply(Q,1,max)
	barplot(maxError, width=df$individuals, space=0, ylim=c(0,1.2*max(maxError)), col=barCol,
xlab=xlab, ylab=ylab, main=main)
	if (shadePops) {
		par(new=TRUE)
		barplot(maxError, width=df$individuals, space=0, ylim=c(0,1.2*max(maxError)),
density=30*(1-popIndex%%2), col='white', ann=FALSE, axes=FALSE)
	}

	# print K
	oldPar <- par(xpd=TRUE)
	if (printK)
		text(0,par('usr')[4],labels=paste('      K=',K,sep=''),pos=3)

	# restore old parameter values
	par(oldPar)
}

# ------------------------------------
## errorBars
# Produce error bars at given positions. Use se=TRUE to input mean and standard error, otherwise
input raw upper and lower limits.

# y1 - if (se==TRUE) this is a vector of means. Otherwise this is a vector of lower limits
# y2 - if (se==TRUE) this is a vector of standard errors. Otherwise this is a vector of upper limits
# x - x-positions of error bars
# se - whether to take means and standard errors as input, or lower and upper limits
# width - horizontal size of error bar whiskers
# col - colour of error bars
# lty - line type of error bars
# lwd - line width of error bars

errorBars <- function(y1, y2, x=1:length(y1), se=FALSE, width=1, col=1, lty=1, lwd=1) {

	# calculate limits based on method
	if (se) {
		LL <- y1-1.96*y2
		UL <- y1+1.96*y2
	} else {
		LL <- y1
		UL <- y2
	}
```

```
	# add error bars
	segments(x,LL,x,UL,lty=lty,lwd=lwd,col=col)
	segments(x-width/2,LL,x+width/2,LL,lty=lty,lwd=lwd,col=col)
	segments(x-width/2,UL,x+width/2,UL,lty=lty,lwd=lwd,col=col)
}

# ----------------------------------
## plotEvidence
# Reads in evidence file and plots the evidence for K in log space for the chosen estimation method.

# fileName - path to input file
# type - which estimator to plot. Options are 'exhaustive', 'harmonic', 'structure' and 'TI'
# xlab - x axis label
# ylab - y axis label
# main - main title

plotEvidence <- function(fileName, type='TI', xlab='K', ylab='log-evidence', main='') {

	# read in evidence file
	ev <- read.csv(fileName)

	# extract relevant columns
	df <- data.frame(K=ev$K)
	if (type=='exhaustive') {
		df$mean <- ev$logEvidence_exhaustive
		df$SE <- 0
	} else if (type=='harmonic') {
		df$mean <- ev$logEvidence_harmonic_grandMean
		df$SE <- ev$logEvidence_harmonic_grandSE
	} else if (type=='structure') {
		df$mean <- ev$logEvidence_structure_grandMean
		df$SE <- ev$logEvidence_structure_grandSE
	} else if (type=='TI') {
		df$mean <- ev$logEvidence_TI
		df$SE <- ev$logEvidence_TI_SE
	} else {
		stop('could not find evidence columns in chosen file')
	}

	# calculate upper and lower 95% confidence intervals
	df$UL <- df$mean+1.96*df$SE
	df$LL <- df$mean-1.96*df$SE

	# get plotting limits
	if (any(!is.na(df$mean))) {
		y_lim <- range(df$mean,na.rm=TRUE)
	} else {
		stop('chosen column contains all NA values')
	}
	if (any(!is.na(df$LL))) {
		y_lim[1] <- min(y_lim[1], min(df$LL,na.rm=TRUE))
	}
	if (any(!is.na(df$UL))) {
		y_lim[2] <- max(y_lim[2], max(df$UL,na.rm=TRUE))
```

```
	}

	# expand to add 10% rail around min and max values
	rail <- 0.1
	y_lim <- c(y_lim[1]*(1+rail/2)-y_lim[2]*rail/2, y_lim[2]*(1+rail/2)-y_lim[1]*rail/2)

	# produce plot
	plot(df$K, df$mean, pch=20, ylim=y_lim, xlab=xlab, ylab=ylab, main=main)
	errorBars(df$LL, df$UL, x=df$K, width=0.3)
}

plotEvidence("./outputEvidence.csv", type="TI")

# ----------------------------------
## plotEvidenceNormalised
# Reads in normalised evidence file and plots the evidence for K in for the chosen estimation method.

# fileName - path to input file
# type - which estimator to plot. Options are 'exhaustive', 'harmonic', 'structure' and 'TI'
# barCol - colour of bars
# xlab - x axis label
# ylab - y axis label
# main - main title

plotEvidenceNormalised <- function(fileName, type='TI', barCol='#64b4ff', xlab='K', ylab='posterior
probability', main='') {

	# read in normalised evidence file
	ev <- read.csv(fileName)
	# extract relevant columns
	df <- data.frame(K=ev$K)
	if (type=='exhaustive') {
		df$mean <- df$LL <- df$UL <- ev$posterior_exhaustive
	} else if (type=='harmonic') {
		df$mean <- ev$posterior_harmonic_mean
		df$UL <- ev$posterior_harmonic_UL
		df$LL <- ev$posterior_harmonic_LL
	} else if (type=='structure') {
		df$mean <- ev$posterior_structure_mean
		df$UL <- ev$posterior_structure_UL
		df$LL <- ev$posterior_structure_LL
	} else if (type=='TI') {
		df$mean <- ev$posterior_TI_mean
		df$UL <- ev$posterior_TI_UL
		df$LL <- ev$posterior_TI_LL
	} else {
		stop('could not find evidence columns in chosen file')
	}

	# get plotting limits
	if (any(!is.na(df$mean))) {
		y_max <- range(df$mean,na.rm=TRUE)
	} else {
		stop('chosen column contains all NA values')
```

```
	}
	if (any(!is.na(df$UL))) {
		y_max <- max(y_max, max(df$UL,na.rm=TRUE))
	}

	# expand to add 10% upper rail (but max-out at 1)
	rail <- 0.1
	y_max <- min(y_max*(1+rail), 1)

	# attract plotting limit to 1 if close by
	if (y_max>0.8) {
		y_max=1
	}

	# produce plot
	barplot(df$mean, names=df$K, space=0, ylim=c(0,y_max), col=barCol, xlab=xlab, ylab=ylab,
main=main)
	errorBars(df$LL, df$UL, x=1:nrow(df)-0.5, width=0.5)
}


plotEvidenceNormalised("./outputEvidenceNormalised.csv", type="TI" )

# ----------------------------------
## plotTI_path
# Reads in evidence details file and plots each of the TIpoint_mean values, along with 95% confidence
intervals given by TIpoint_SE.

# fileName - path to input file
# K - value of K to plot
# xlab - x axis label
# ylab - y axis label
# main - main title

plotTI_path <- function(fileName, K, xlab='beta', ylab='TI path', main=paste('K=',K,sep='')) {

	# read in evidence details file and subset to K
	ev <- read.csv(fileName)
	ev <- ev[ev[,1]==K,]

	# extract TIpoint_mean and TIpoint_SE values
	TIpoint_mean <- unlist(ev[,grep("TIpoint_mean",names(ev))])
	TIpoint_SE <- unlist(ev[,grep("TIpoint_SE",names(ev))])
	rungs <- length(TIpoint_mean)
	beta <- seq(0,1,l=rungs)

	# get upper and lower values
	TIpoint_UL <- TIpoint_mean + 1.96* TIpoint_SE
	TIpoint_LL <- TIpoint_mean - 1.96* TIpoint_SE

	# get plotting limits
	y_lim <- range(TIpoint_mean,na.rm=TRUE)
	if (any(!is.na(TIpoint_LL))) {
		y_lim[1] <- min(y_lim[1], min(TIpoint_LL,na.rm=TRUE))
```

```
	}
	if (any(!is.na(TIpoint_UL))) {
		y_lim[2] <- max(y_lim[2], max(TIpoint_UL,na.rm=TRUE))
	}

	# expand to add 10% rail
	rail <- 0.1
	y_lim <- c(y_lim[1]*(1+rail/2)-y_lim[2]*rail/2, y_lim[2]*(1+rail/2)-y_lim[1]*rail/2)

	# produce plot
	plot(beta, TIpoint_mean, type='o', pch=20, cex=0.7, ylim=y_lim, xlab=xlab, ylab=ylab,
main=main)
	errorBars(TIpoint_mean, TIpoint_SE, x=beta, se=TRUE, width=0.05)
}

plotTI_path("./outputEvidenceDetails.csv", K=7)
# ----------------------------------
## model_normalise
# Reads in evidence files from multiple models. Uses a simulation-based method to sum model
evidence over K. Outputs are given in log space, or in linear space after normalising to sum to 1 over
models. The latter is equivalent to the posterior distribution of the evolutionary model, integrated over
all K.

# fileNames - vector of file names, each of which is an outputEvidence.csv file
# logOutput - whether to produce output in log space
# reps - number of random draws used in simulation-based approach

model_normalise <- function(fileNames, logOutput=TRUE, reps=1e6) {

	# read in mean and SE evidence values from file
	for (i in 1:length(fileNames)) {

		# read file
		ev <- read.csv(fileNames[[i]])

		# check that correct columns are present
		if (!('K'%in%names(ev) & 'logEvidence_TI'%in%names(ev) &
'logEvidence_TI_SE'%in%names(ev))) {
			stop('cannot find TI values in chosen data frames')
		}

		# extract mean and SE into separate data frames
		ev_mean <- subset(ev, select=c(K,logEvidence_TI))
		ev_SE <- subset(ev, select=c(K,logEvidence_TI_SE))
		names(ev_mean)[2] <- paste('mean_model',i,sep='')
		names(ev_SE)[2] <- paste('SE_model',i,sep='')
		if (i==1) {
			df_mean <- ev_mean
			df_SE <- ev_SE
		} else {
			df_mean <- merge(df_mean,ev_mean)
			df_SE <- merge(df_SE,ev_SE)
		}
	}
```

```
	# subtract max value to value to avoid underflow
	df_mean_max <- max(df_mean[,-1])
	df_mean[,-1] <- df_mean[,-1] - df_mean_max

	# different methods for logged or linear output
	if (logOutput) {

		# initialise output object
		output <- data.frame(model=1:length(fileNames), logEvidence_mean=NA,
logEvidence_LL=NA, logEvidence_UL=NA)
		# loop through models
		for (i in 2:ncol(df_mean)) {

			# random draws
			X <- mapply(rnorm,n=reps,mean=df_mean[,i],sd=df_SE[,i])	# matrix of
random normal draws, with values taken from evidence file
			Y <- exp(X)	# exponentiate these draws
			Z <- log(rowMeans(Y)) + df_mean_max		# take log of mean over K and
add max value again

			# summarise output
			output$logEvidence_mean[i-1] <- mean(Z)
			output$logEvidence_LL[i-1] <- quantile(Z,probs=0.025)
			output$logEvidence_UL[i-1] <- quantile(Z,probs=0.975)
		}

	} else {

		# initialise output object
		output <- data.frame(model=1:length(fileNames), evidence_mean=NA,
evidence_LL=NA, evidence_UL=NA)

		# Z will contain the exponentiated values for all models
		Z <- NULL

		# loop through models
		for (i in 2:ncol(df_mean)) {

			# random draws
			X <- mapply(rnorm,n=reps,mean=df_mean[,i],sd=df_SE[,i])	# matrix of
random normal draws, with values taken from evidence file
			Y <- exp(X)	# exponentiate these draws
			Z <- rbind(Z, rowSums(Y)) # add these draws to Z
		}

		# normalise Z over models
		Z_sum <- colSums(Z)
		for (i in 1:length(fileNames)) {
			Z[i,] <- Z[i,]/Z_sum

			# summarise output
			output$evidence_mean[i] <- mean(Z[i,])
			output$evidence_LL[i] <- quantile(Z[i,],probs=0.025)
```

```
                output$evidence_UL[i] <- quantile(Z[i,],probs=0.975)
            }

        }
        return(output)
}

# ----------------------------------
## plotModel
# Takes object output from model_normalise() function and produces barplot of normalised evidence
(aka posterior distribution) with error bars.

# modelEvidence - object returned from model_normalise() function
# modelNames - vector of names to be printed under each bar
# col - colour of bars
# ylab - y-axis label

plotModel <- function(modelEvidence, modelNames, col='#64b4ff', ylab='posterior probability') {

        # check for presence of evidence columns
        if (!'evidence_mean'%in%names(modelEvidence) |
!'evidence_LL'%in%names(modelEvidence) | !'evidence_UL'%in%names(modelEvidence)) {
            stop('cannot find evidence columns in modelEvidence object')
        }

        # get basic properties
        models <- nrow(modelEvidence)
        xpos <- (1:models)*1.2-0.5
        barLabels <- sprintf("%.2f", modelEvidence$evidence_mean)
        for (i in which(modelEvidence$evidence_mean<0.005)) {
            barLabels[i] <- sprintf("%.2e", modelEvidence$evidence_mean[i])
        }

        # produce plot
        barplot(modelEvidence$evidence_mean, col=col, ylim=c(0,1.1), axes=FALSE, ylab=ylab)
        mtext(modelNames, side=1, at=xpos, line=2)
        axis(2, at=seq(0,1,0.2))
        axis(1, labels=FALSE, at=c(0,xpos,models*1.2+0.7))
        errorBars(modelEvidence$evidence_LL, modelEvidence$evidence_UL, x=xpos, width=0.4)
        text(xpos, modelEvidence$evidence_UL, barLabels, pos=3, cex=0.8)
}
```

File 6.24 MavericK "parameters.txt" file

File containing the MavericK parameters settings for the pilot runs performed for assemblies 1, 5, 8 and 10.

```
#### Data properties
headerRow_on    t
popCol_on       f

ploidyCol_on    f

ploidy  2
missingData     -9


#### Model parameters
Kmin    1
Kmax    10
admix_on        t
fixAlpha_on     f
alpha   1.0
alphaPropSD     0.10


#### Simulation parameters
exhaustive_on   f

mainRepeats     1
mainBurnin      500
mainSamples     5000

thermodynamic_on        f
thermodynamicRungs      20
thermodynamicBurnin     500
thermodynamicSamples    1000

EMalgorithm_on  f
EMrepeats       100
EMiterations    100


#### Basic output properties
outputLog_on    t
outputLikelihood_on     t
outputQmatrix_ind_on    t
outputQmatrix_pop_on    f
outputQmatrixError_ind_on       f
outputQmatrixError_pop_on       f
outputEvidence_on       f
outputEvidenceNormalised_on     f
outputEvidenceDetails_on        f
```

File 6.25 MavericK "parameters_final8_final.txt" file

File containing the parameter settings for the MavericK final run of assembly 8 with Thermodynamic Integration ON.

```
#### Data properties
headerRow_on    t
popCol_on       f

ploidyCol_on    f

ploidy  2
missingData     -9


#### Model parameters
Kmin    1
Kmax    7
admix_on        t
fixAlpha_on     f
alpha   1.0,0.838,0.539,0.455,0.25,0.193,0.066
alphaPropSD     1.0,0.171,0.091,0.074,0.038,0.
026,0.014


#### Simulation parameters
exhaustive_on   f

mainRepeats     5
mainBurnin      2000
mainSamples     10000

thermodynamic_on        t
thermodynamicRungs      20
thermodynamicBurnin     2000
thermodynamicSamples    10000

EMalgorithm_on  f
EMrepeats       100
EMiterations    100


#### Basic output properties
outputLog_on    t
outputLikelihood_on     t
outputQmatrix_ind_on    t
outputQmatrix_pop_on    f
outputQmatrixError_ind_on       t
outputQmatrixError_pop_on       f
outputEvidence_on       t
outputEvidenceNormalised_on     t
outputEvidenceDetails_on        t
```

File* 6.26 *MavericK "parameters_final10.txt" file

File containing the MavericK parameter settings for the final run of assembly 10 with Thermodynamic Integration ON.

```
#### Data properties
headerRow_on    t
popCol_on       f

ploidyCol_on    f

ploidy  2
missingData     -9

#### Model parameters
Kmin    1
Kmax    7
admix_on        t
fixAlpha_on     f
alpha   1.0,0.74,0.765,0.356,0.147,0.119,0.076
alphaPropSD     1.0,0.155,0.108,0.055,0.028,0.
022,0.021

#### Simulation parameters
exhaustive_on   f



mainRepeats     5
mainBurnin      2000
mainSamples     10000

thermodynamic_on        t
thermodynamicRungs      20
thermodynamicBurnin     2000
thermodynamicSamples    10000

EMalgorithm_on  f
EMrepeats       100
EMiterations    100


#### Basic output properties
outputLog_on    t
outputLikelihood_on     t
outputQmatrix_ind_on    t
outputQmatrix_pop_on    f
outputQmatrixError_ind_on       t
outputQmatrixError_pop_on       f
outputEvidence_on       t
outputEvidenceNormalised_on     t
outputEvidenceDetails_on        t
```

File 6.27 “IndsPops.txt” file for D-statistic

File used in the R script named “Dstat.R” for calculating the D-statistic. The first column corresponds to the Individual ID and the second column to the Population ID.

Tar260_allo	argentataAllopatric
Tar310_sym	argentataSympatric
Tar321_sym	argentataSympatric
Tar335_sym	argentataSympatric
Tar356_sym	argentataSympatric
Tar357_sym	argentataSympatric
Tar359_sym	argentataSympatric
Tar552_allo	argentataAllopatric
Tar553_allo	argentataAllopatric
Tar769_allo	argentataAllopatric
Tar770_allo	argentataAllopatric
Tjo3765	josei
Tma068_allo	mariaSympatric
Tma071_allo	mariaSympatric
Tma327_sym	mariaSympatric
Tma338_sym	mariaSympatric
Tma342_sym	mariaSympatric
Tma352_sym	mariaSympatric
Tma354_sym	mariaSympatric
Tma363_sym	mariaSympatric
Tma729_allo	mariaSympatric

File 6.28 Script written in R named “Dstat.R” used to calculate the D-statistics

```
# load the file with the function to compute the D-stat
source("Dstat_Jacknife.r")

# read a file where
# 1st column is the individual ID
# 2nd column is the pop ID
vcf.sortedindpopinfo <- read.table("IndsPops.txt", header=FALSE)

# Define the index of individuals that belong to each pop
index_p1 <- which(vcf.sortedindpopinfo[,2]=="argentataAllopatric") # vcf.sortedindpopinfo is a
matrix where the first column is the individual ID and the second is the pop ID
index_p2 <- which(vcf.sortedindpopinfo[,2]=="argentataSympatric")
index_p3 <- which(vcf.sortedindpopinfo[,2]=="mariaSympatric")
index_outg <- which(vcf.sortedindpopinfo[,2]=="josei")

# read the data with genotypes (GT matrix)
geno <- as.matrix(read.table("params14filtered.recode.GT", header=F, stringsAsFactors =
F,na.strings = "-1"))
# get the selected individuals from the matrix with all individuals
genotypes <- geno[,c(index_p1,index_p2, index_p3, index_outg)] # sorted_tgendata is the
genotype matrix with nsites (nrow) * nind (ncol)
# sample size for each pop
npop <- c(length(index_p1), length(index_p2), length(index_p3), length(index_outg))
# get index of individuals from each pop
index_d <- cumsum(c(1,npop))
# call function to compute d-stat
dstat(genotypes, index_d)
# get the estimates of significance based on the block-jacknife approach
dsj <- Djack(genotypes, index_inds = index_d, numblocks = 50)
dsj$D # D-statistic
# get the p-values assuming it is a one-sided test
pvals <- pnorm(-abs(dsj$z))
pvals
```

File 6.29 Output result obtained for params8.recode.vcf when using the R Script "vcf2PCA.R"

```
> vcf.fn<-"/Users/inesoliveira/Desktop/pca_final/params8.recode.vcf" #load the data
> snpgdsVCF2GDS(vcf.fn,"test.gds",method="copy.num.of.ref", ignore.chr.prefix="chr")#import to
class. use method= biallelic.only for biallelic snps
VCF Format ==> SNP GDS Format
Method: dosage (0,1,2) of reference allele for all variant sites
Number of samples: 37
Parsing "/Users/inesoliveira/Desktop/pca_final/params8.recode.vcf" ...
        import 8980 variants.
+ genotype   { Bit2 37x8980, 81.1K } *
Optimize the access efficiency ...
Clean up the fragments of GDS file:
   open the file 'test.gds' (122.1K)
   # of fragments: 40
   save to 'test.gds.tmp'
   rename 'test.gds.tmp' (121.9K, reduced: 240B)
   # of fragments: 20
> #summary of the vcf imported
> snpgdsSummary("test.gds")
Some of 'snp.allele' are not standard (e.g., A/T,C).
The file name: /Users/inesoliveira/Desktop/pca_final/test.gds
The total number of samples: 37
The total number of SNPs: 8980
SNP genotypes are stored in SNP-major mode (Sample X SNP).
The number of valid samples: 37
The number of biallelic unique SNPs: 8877
> #open the translated vcf imported
> genofile<-snpgdsOpen("test.gds")
> #import population names
> pop_code <- scan("/Users/inesoliveira/Desktop/pca_final/individuos_algarve.txt", what=character()) ##
File 1
Read 37 items
> pca<-snpgdsPCA(genofile, autosome.only = FALSE)
Principal Component Analysis (PCA) on genotypes:
Excluding 0 SNP (monomorphic: TRUE, MAF: NaN, missing rate: NaN)
Working space: 37 samples, 8,980 SNPs
   using 1 (CPU) core
PCA:    the sum of all selected genotypes (0,1,2) = 477246
CPU capabilities: Double-Precision SSE2
Mon Dec  3 20:00:24 2018    (internal increment: 9628)
[==================================================] 100%, completed in 0s
Mon Dec  3 20:00:24 2018    Begin (eigenvalues and eigenvectors)
Mon Dec  3 20:00:24 2018    Done.
> # variance proportion (%)
> pc.percent<-pca$varprop*100
> head(pc.percent)
[1] 7.982292 5.544426 4.719708 3.682021 3.288025 3.105270
```

File 6.30 Output result obtained for params10.recode.vcf when using the R Script "vcf2PCA.R"

```
> vcf.fn<-"/Users/inesoliveira/Desktop/pca_final/params10.recode.vcf" #load the data
> snpgdsVCF2GDS(vcf.fn,"test.gds",method="copy.num.of.ref", ignore.chr.prefix="chr")#import to
class. use method= biallelic.only for biallelic snps
VCF Format ==> SNP GDS Format
Method: dosage (0,1,2) of reference allele for all variant sites
Number of samples: 31
Parsing "/Users/inesoliveira/Desktop/pca_final/params10.recode.vcf" ...
        import 12059 variants.
+ genotype   { Bit2 31x12059, 91.3K } *
Optimize the access efficiency ...
Clean up the fragments of GDS file:
   open the file 'test.gds' (143.6K)
   # of fragments: 40
   save to 'test.gds.tmp'
   rename 'test.gds.tmp' (143.4K, reduced: 240B)
   # of fragments: 20
> #summary of the vcf imported
> snpgdsSummary("test.gds")
Some of 'snp.allele' are not standard (e.g., A/T,G).
The file name: /Users/inesoliveira/Desktop/pca_final/test.gds
The total number of samples: 31
The total number of SNPs: 12059
SNP genotypes are stored in SNP-major mode (Sample X SNP).
The number of valid samples: 31
The number of biallelic unique SNPs: 11915
> #open the translated vcf imported
> genofile<-snpgdsOpen("test.gds")
> #import population names
> pop_code <- scan("/Users/inesoliveira/Desktop/pca_final/individuos_sem_aneabi_algarve.txt",
what=character()) ## File 1
Read 31 items
> #PCA
> pca<-snpgdsPCA(genofile, autosome.only = FALSE)
Principal Component Analysis (PCA) on genotypes:
Excluding 0 SNP (monomorphic: TRUE, MAF: NaN, missing rate: NaN)
Working space: 31 samples, 12,059 SNPs
   using 1 (CPU) core
PCA:    the sum of all selected genotypes (0,1,2) = 538801
CPU capabilities: Double-Precision SSE2
Mon Dec  3 20:06:07 2018    (internal increment: 11492)
[==================================================] 100%, completed in 0s
Mon Dec  3 20:06:07 2018    Begin (eigenvalues and eigenvectors)
Mon Dec  3 20:06:07 2018    Done.
> # variance proportion (%)
> pc.percent<-pca$varprop*100
> head(pc.percent)
[1] 8.113234 5.923698 4.613793 3.864443 3.745702 3.704004
```

File 6.31 Output result obtained for params8CenterSNP.vcf when using the R Script "vcf2PCA.R"

```
> vcf.fn<-"/Users/inesoliveira/Desktop/pca_final/params8CenterSNP.vcf" #load the data
> snpgdsVCF2GDS(vcf.fn,"test.gds",method="copy.num.of.ref", ignore.chr.prefix="chr")#import to
class. use method= biallelic.only for biallelic snps
VCF Format ==> SNP GDS Format
Method: dosage (0,1,2) of reference allele for all variant sites
Number of samples: 37
Parsing "/Users/inesoliveira/Desktop/pca_final/params8CenterSNP.vcf" ...
        import 3091 variants.
+ genotype   { Bit2 37x3091, 27.9K } *
Optimize the access efficiency ...
Clean up the fragments of GDS file:
   open the file 'test.gds' (48.7K)
   # of fragments: 39
   save to 'test.gds.tmp'
   rename 'test.gds.tmp' (48.5K, reduced: 228B)
   # of fragments: 20
> #summary of the vcf imported
> snpgdsSummary("test.gds")
Some of 'snp.allele' are not standard (e.g., C/T,A).
The file name: /Users/inesoliveira/Desktop/pca_final/test.gds
The total number of samples: 37
The total number of SNPs: 3091
SNP genotypes are stored in SNP-major mode (Sample X SNP).
The number of valid samples: 37
The number of biallelic unique SNPs: 3065
> #open the translated vcf imported
> genofile<-snpgdsOpen("test.gds")
> #import population names
> pop_code <- scan("/Users/inesoliveira/Desktop/pca_final/individuos_algarve.txt", what=character()) ##
File 1
Read 37 items
> #PCA
> pca<-snpgdsPCA(genofile, autosome.only = FALSE)
Principal Component Analysis (PCA) on genotypes:
Excluding 0 SNP (monomorphic: TRUE, MAF: NaN, missing rate: NaN)
Working space: 37 samples, 3,091 SNPs
   using 1 (CPU) core
PCA:    the sum of all selected genotypes (0,1,2) = 166621
CPU capabilities: Double-Precision SSE2
Thu May  3 20:57:36 2018    (internal increment: 9628)
[==================================================] 100%, completed in 0s
Thu May  3 20:57:36 2018    Begin (eigenvalues and eigenvectors)
Thu May  3 20:57:36 2018    Done.
> # variance proportion (%)
> pc.percent<-pca$varprop*100
> head(pc.percent)
[1] 8.904754 6.162909 4.901049 3.528609 3.301333 3.147944
```

File 6.32 Output result obtained for params10CenterSNP.vcf when using the R Script "vcf2PCA.R"

```
> vcf.fn<-"/Users/inesoliveira/Desktop/pca_final/params10CenterSNP.vcf" #load the data
> snpgdsVCF2GDS(vcf.fn,"test.gds",method="copy.num.of.ref", ignore.chr.prefix="chr")#import to
class. use method= biallelic.only for biallelic snps
VCF Format ==> SNP GDS Format
Method: dosage (0,1,2) of reference allele for all variant sites
Number of samples: 31
Parsing "/Users/inesoliveira/Desktop/pca_final/params10CenterSNP.vcf" ...
        import 3763 variants.
+ genotype   { Bit2 31x3763, 28.5K } *
Optimize the access efficiency ...
Clean up the fragments of GDS file:
   open the file 'test.gds' (53.0K)
   # of fragments: 39
   save to 'test.gds.tmp'
   rename 'test.gds.tmp' (52.8K, reduced: 228B)
   # of fragments: 20
> #summary of the vcf imported
> snpgdsSummary("test.gds")
Some of 'snp.allele' are not standard (e.g., C/A,T).
The file name: /Users/inesoliveira/Desktop/pca_final/test.gds
The total number of samples: 31
The total number of SNPs: 3763
SNP genotypes are stored in SNP-major mode (Sample X SNP).
The number of valid samples: 31
The number of biallelic unique SNPs: 3729
> #open the translated vcf imported
> genofile<-snpgdsOpen("test.gds")
> #import population names
> pop_code <- scan("/Users/inesoliveira/Desktop/pca_final/individuos_sem_aneabi_algarve.txt",
what=character()) ## File 1
Read 31 items
> #PCA
> pca<-snpgdsPCA(genofile, autosome.only = FALSE)
Principal Component Analysis (PCA) on genotypes:
Excluding 0 SNP (monomorphic: TRUE, MAF: NaN, missing rate: NaN)
Working space: 31 samples, 3,763 SNPs
   using 1 (CPU) core
PCA:    the sum of all selected genotypes (0,1,2) = 170060
CPU capabilities: Double-Precision SSE2
Tue May  8 17:27:07 2018    (internal increment: 11492)
[==================================================] 100%, completed in 0s
Tue May  8 17:27:07 2018    Begin (eigenvalues and eigenvectors)
Tue May  8 17:27:07 2018    Done.
> # variance proportion (%)
> pc.percent<-pca$varprop*100
> head(pc.percent)
[1] 9.166461 6.353465 4.592294 4.009333 3.774590 3.715979
```