Ana Beatriz da Silva Bueno

Análise genômica de híbridos e não híbridos de Olhos-de-fogo (gênero *Pyriglena*, Aves: Thamnophilidae) da Floresta Atlântica

Genomic analysis of hybrids and non-hybrids of Fire-eyes (genus *Pyriglena*, Birds: Thamnophilidae) from the Atlantic Forest

São Paulo
2017

Ana Beatriz da Silva Bueno

Análise genômica de híbridos e não híbridos de Olhos-de-fogo (gênero *Pyriglena*, Aves: Thamnophilidae) da Floresta Atlântica

Genomic analysis of hybrids and non-hybrids of Fire-eyes (genus *Pyriglena*, Birds: Thamnophilidae) from the Atlantic Forest

Dissertação apresentada ao Instituto de Biociências da Universidade de São Paulo, para a obtenção de Título de Mestre em Ciências, na Área de Biologia/Genética.

Orientador(a): Profa. Dra. Cristina Yumi Miyaki

São Paulo
2017

Bueno, Ana Beatriz da Silva

Análise genômica de híbridos e não híbridos de Olhos-de-fogo (gênero *Pyriglena*, Aves: Thamnophilidae) da Floresta Atlântica

131 Páginas

Dissertação (Mestrado) - Instituto de Biociências da Universidade de São Paulo. Departamento de Genética e Biologia Evolutiva.

1. Palavra-Chave Principal: *Híbridos*
2. Palavra-Chave Secundária: Genômica
3. Palavra-Chave Secundária: Aves

I. Universidade de São Paulo. Instituto de Biociências. Departamento de Genética e Biologia Evolutiva.

Comissão Julgadora

_____ _____

Profª Drª **Prof Dr**

_____

Profª Drª Cristina Yumi Miyaki

Dedico à minha família, especialmente à minha avó Deise F.R. Soares e ao meu avô Orlando Soares, eterno vô "Pancho", com muito amor.

Foto: arquivo pessoal

“Sometimes the world seems against you
The journey may leave a scar
But scars can heal and reveal just
Where you are
The people you love will change you
The things you have learned will guide you
And nothing on earth can silence
The quiet voice still inside you
(…)
That come what may, I know the way
I am Moana!”

"I Am Moana (Song of Ancestors)" – Moana (2016 – Walt Disney Studios)
Música e letras por Lin-Manuel Miranda e Mark Mancina
Interpretada por Auli'i Cravalho (Moana) e Rachel House (Gramma Tala)

Agradecimentos

Agradeço...

À Dra. Cristina Yumi Miyaki pela orientação desde meu período de graduação, foram cinco anos de muito aprendizado nos quais amadureci cientifica e profissionalmente. Com sua orientação venci meus medos, ganhei confiança no meu conhecimento, saí da minha zona de conforto e assim adquiri a postura de uma verdadeira cientista. Minha eterna admiração e gratidão.

Às *Pyriglenas* por, literalmente, terem dado o sangue por esse projeto. Nunca serão esquecidas. #FireEyePower

Aos amigos do Laboratório de Genética e Evolução Molecular de Aves (LGEMA) pelas contribuições científicas, companheirismo e bons momentos regados a café. Minha doce família científica, nunca me esquecerei!

Aos professores do Departamento de Genética e Biologia Evolutiva do Instituto de Biociências (USP), pelas disciplinas oferecidas à pós-graduação. E aos funcionários do Departamento de Genética e Biologia Evolutiva e da pós-graduação do Instituto de Biociências (USP) pelo apoio.

Às pessoas e instituições que cederam o material biológico utilizado nesse estudo. À Fundação de Amparo a Pesquisa no Estado de São Paulo (Fapesp) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo financiamento da pesquisa e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela concessão da bolsa de mestrado, sem as quais não seria possível a realização do trabalho.

Ao casal Marcos Maldonado-Coelho e Linda Maria E. Svensson Coelho pela oportunidade de trabalhar com *Pyriglena* (lindezas dos olhos-de-fogo) e por todo o suporte e ajuda que me deram durante esses anos de mestrado.

À Dra. Sónia Cristina da Silva Andrade pelos conhecimentos passados, pela disponibilidade e disposição de me ajudar. Sua ajuda teve enorme impacto nas análises.

Às minhas amigas do interior: Daniela Coelho, Chiara Meneghel, Camille Constanza, Juliana Rodrigues e todas não citadas aqui, por todo o apoio, por me aguentarem por todo esse tempo, nos bons e maus momentos, pelas risadas compartilhadas, pela amizade, pela maquiagens, cosplays e lógico, pelas comilanças. Desde sempre para sempre. Amo vocês, suas danadinhas!

Aos meus amigos e às minhas amigas que fiz em São Paulo por encherem meus dias de alegria e risada nessa Selva de Pedra, que muitas vezes é bem solitária. Muito obrigada.

À Fernanda Midori, pelos tours malucos no centro, pelas gordices e por essa amizade praticamente irmandade que temos. Não existe distância que nos separe. Minha irmã nipônica.

À minha psicóloga por ouvir todas minhas angústias e inseguranças, por me fazer enfrentar meus medos e terminar esse mestrado mentalmente saudável.

À minha família: a minha admirável mãe Amantina Aparecida Soares Bueno, ao meu grande pai Rui da Silva Bueno Filho, as minhas avós Deise Francisca Reis Soares e Benedita da Silva Bueno, ao meu tio Orlando Soares Filho, por sempre acreditarem mais em mim do que eu mesma, pelo grande apoio, amor incondicional e pelo incentivo, mesmo não entendendo muito bem meu mestrado. Sempre estão e estarão comigo. Amo vocês!

Ao meu avô Orlando Soares, um agradecimento mais que especial, pois foi aquele que mais incentivou meu lado intelectual mesmo com pouca idade, sempre acreditou no meu potencial e incentivou meus sonhos. Onde estiver, espero que esteja orgulhoso. Sinto sua falta e para sempre te amarei.

Ao Thiago Lamente, que mesmo não entendendo 1% desse mestrado, foi meu maior apoio e minha família em São Paulo. Aquele que aguentou meus choros por não conseguir rodar alguma análise mesmo sabendo que 5 minutos depois o problema já estaria resolvido e eu estaria rindo. Seu amor incondicional me fez acreditar mais em mim e não desistir quando o medo era maior que minha confiança. *I love you to the moon and back. We're simply meant to be. Tico Tico <3*

Sumário

1. Introdução 01

1.1 Dinâmica de Espécies e Zona de Hibridação 01

1.2 A Mata Atlântica e o Gênero *Pyriglena* 04

2. Objetivos 10

3. Material e Métodos 11

3.1 Amostragem 11

3.2 Métodos Moleculares 12

3.2.1 Extração DNA 12

3.2.2 Preparo da Biblioteca e Sequenciamento 12

3.3 Análises Computacionais e Bioinformática 13

3.3.1 Tratamento dos Dados Brutos do Sequenciamento 13

3.3.2 Análises dos Dados 13

4. Resultados 17

5. Discussão 36

5.1 Estrutura populacional do complexo *P. atra/P. leucoptera*/Intermediários morfológicos 36

5.2 Hibridação e Seleção 37

6. Conclusões 40

Resumo 41

Abstract 42

Referências Bibliográficas 43

ANEXO 1 - Manual LGEMA: FastQC 51

ANEXO 2 - Manual LGEMA: pyRAD e PEAR para GBS paired-end 55

ANEXO 3 - Manual LGEMA: Bowtie2 82

ANEXO 4 - Manual LGEMA: sNMF 88

ANEXO 5 - Manual LGEMA: ADMIXTURE 99

ANEXO 6 - Manual LGEMA: fastSTRUCTURE 103

ANEXO 7 - Manual LGEMA: BayeScan e PGDSpider 112

ANEXO 8 - Manual LGEMA: TreeMix 124

1. Introdução

1.1 Dinâmica de Espécies e Zonas de Hibridação

Espécies são uma das unidades fundamentais da biologia, com sua importância comparada a genes, células e organismos, sendo utilizada em diversas áreas, de anatomia a comportamento, desenvolvimento, ecologia, genética, biologia molecular, paleontologia, fisiologia e sistemática (de Queiroz, 2005; Mayr, 1982).

Sua importância na biologia está intimamente ligada à sua importância na sistemática, que é responsável pela estrutura taxonômica usada em todos os ramos da biologia (de Queiroz, 2005). Sendo que por mais de 250 anos a sistemática tem sido fortemente influenciada pela hierarquia familiar das categorias taxonômicas originadas pelos trabalhos de Linnaeus publicados em 1753 e 1758, nos quais trata espécies como a menor e mais fundamental categoria taxonômica. O papel central da sistemática também é reforçado pela relação que une sistemática com biologia evolutiva (de Queiroz, 2005), como nos trabalhos: *On the Origin of Species* (Darwin, 1859), *Genetics and the Origin of Species* (Dobshansky, 1937), *Evolution, the Modern Synthesis* (Huxley, 1942) e *The Evolutionary Synthesis: Perspectives on the Unification of Biology* (Mayr, 1980).

Dentre esses trabalhos, pode-se destacar o livro de Mayr (1980) que levantou muita discussão sobre o conceito de espécie com a proposta de que "espécies são grupos de populações naturais que cruzam entre si, que são reprodutivamente isolados de tais outros grupos" (Mayr, 1980). Este conceito ficou conhecido como conceito biológico de espécie.

Apesar de o conceito de Mayr ter sido amplamente aceito, esta definição recebeu críticas e posteriores conceitos alternativos foram criados até que por volta da década de 90, Mayden (1997) identificou 24 conceitos de espécies, como o biológico, o fenético, o evolutivo, o ecológico e o filogenético (três versões). A diferença entre esses conceitos reside em diferentes propriedades biológicas consideradas por cada uma. Assim, por exemplo, o conceito biológico de espécies enfatiza a propriedade de isolamento reprodutivo (Dobzhansky, 1970; Mayr, 1942). O conceito ecológico enfatiza a ocupação de um nicho distinto ou zona adaptativa (Andersson, 1990; Van Valen, 1976). Uma das versões do conceito filogenético enfatiza a diagnose (Cracraft, 1983; Nixon, 1990) e outro, a monofilia (Donoghue, 1985; Rosen, 1979).

A essa diversidade advém dos diferentes interesses de cada subgrupo de biólogos: aqueles que estudam sistemática tendem a enfatizar a diagnose e monofilia, enquanto ecólogos tendem a enfatizar a diferença de nicho, paleontólogos enfatizam as diferenças

morfológicas, geneticistas as diferenças nos genes e os que estudam zona de hibridação tendem a enfatizar as barreiras reprodutivas (de Queiroz, 2005).

Entender a dinâmica das espécies e como delimitá-las tem intrigado evolucionistas e sistematas há muito tempo e muitos conceitos foram criados (de Queiroz, 1998; Harrison, 1998; Mayden, 1997; Stamos, 2003) pois casos que envolvem hibridação (cruzamento entre indivíduos de espécies diferentes com geração de descendentes), introgressão (fluxo de genes entre espécies que envolve hibridação e retrocruzamento) e retenção de polimorfismo ancestral com separação incompleta entre linhagens (Anderson, 1949; Anderson & Hubricht, 1938; Harrison, 1990; Harrison, 1993) desafiam a maioria dos conceitos de espécies.

Por ser um processo evolutivo que pode resultar em novidades genéticas mais rápido do que por meio de mutações (Anderson & Hubricht, 1938; Martinsen, 2001), a hibridação tem sido o foco de muitos estudos evolutivos. Por exemplo, discussão sobre as bases ecológicas e genéticas da hibridação (Anderson, 1949), os processos micro e macroevolutivos envolvidos na hibridação natural (Arnold, 1992), contribuição da hibridação na origem de plantas (Rieseberg & Carney, 1998) e a invasão do genoma animal por meio de introgressão com consequentes mudanças no padrão de especiação dos mesmos (Dowling, 1997; Jiggins, 2008; Mallet, 2005; Schwenk *et al.*, 2008).

Espera-se que um conjunto de híbridos deve conter maior variedade de alelos do que cada parental isoladamente, portanto alelos seletivamente neutros nos parentais que estejam em nova combinação nos híbridos, poderiam sofrer pressão seletiva. Com isso, uma variante genética adaptativa importante poderia resultar em alterações no híbrido (Choler *et al*., 2004; Castric *et al*., 2008; Kim *et al*., 2008; Martin *et al*., 2006).

Os resultados evolutivos da hibridação podem ser dos mais variados tipos: inclui a manutenção ou aumento da diversidade resultando em zonas de hibridação estáveis; a aquisição de nova variedade alélica pode acarretar na origem ou transferência de adaptações, tendo efeito positivo no resgate de populações pequenas puras; o reforço do isolamento reprodutivo e a formação de novas linhagens híbridas (Abbott, 1992; Abbott *et al*., 2013; Anderson, 1949; Arnorld, 1996; Ellstrand & Schierenbeck, 2000; Frankham, 2015; Mallet, 2007; Stebins, 1959). A hibridação também pode resultar em especiação se o híbrido se tornar ecológica e/ou espacialmente muito divergente dos parentais (Abbott *et al*., 2010; Gross & Rieseberg, 2005).

No entanto, a hibridação pode reduzir a diversidade, pois a quebra das barreiras reprodutivas e a mescla de populações anteriormente distintas pode acarretar na extinção de uma das populações ou espécies envolvidas na hibridação (Allendorf *et al*., 2001; Buerkle *et*

al., 2003; Ellstrand, 1992; Levin *et al*., 1996; Rieseberg *et al*., 1989; Rhymer & Simberloff 1996; Vuillaume *et al.*, 2015).

Em geral, a extinção resultante de hibridação pode ocorrer via dois processos:

(1) se o *fitness* do híbrido for menor do que o *fitness* dos parentais e a hibridação for muito frequente, uma ou ambas as linhagens parentais podem declinar em termos demográficos resultando na sua extinção. Esse processo é conhecido como submersão demográfica (*demographic swamping*; Ellstrand *et al*., 1999; Ellstrand & Elam, 1993; Levin *et al*., 1996; Wolf *et al.*, 2001);

(2) se a depressão por exocruzamento for menos severa e a taxa de crescimento populacional dos híbridos superar a taxa de crescimento das populações parentais, uma (ou ambas) linhagem(ns) parental(is) pode(m) ser substituída(s) por híbridos. Esse processo é a submersão genética (*genetic swamping*; Heusmann, 1974; Johnsgard, 1961; Rhymer & Simberloff, 1996; Todesco *et al.*, 2016).

A extinção relacionada à hibridação pode ser importante em conservação caso uma das espécies parentais seja ameaçada de extinção. Assim, entender as consequências da hibridação é um pontos chave para a proteção e conservação dessa espécie.

Diversos métodos são utilizados para detectar hibridação, incluindo exames morfológicos, citológicos, químicos e marcadores moleculares (Riseberg & Ellstrand, 1993). Sendo que o aumento de estudos com marcadores moleculares ocorre devido a possibilidade de analisar de diversas formas a teoria evolutiva com abordagens estatísticas robustas (Rieseberg *et al.,* 2000). Análises genômicas comparativas de híbridos e seus parentais podem permitir a identificação de genes que possivelmente contribuem no isolamento reprodutivo e da frequência que ocorre hibridação/introgressão entre as espécies envolvidas e podem auxiliar a caracterizar o padrão de seleção dos alelos (Arnold, 1992; Buerkle, 2007; Barton & Gale, 1993; Barton & Hewitt, 1985; Twyford & Ennos, 2012). Para realizar esse tipo de análises, o método de Sanger de sequenciamento de DNA (Sanger & Coulson, 1975; Sanger *et al.*, 1977) foi utilizado por mais de duas décadas para obter genomas completos de numerosas espécies. No entanto, novas tecnologias de sequenciamento foram desenvolvidas produzindo muito mais leituras, com custo mais baixo e menos trabalho laboratorial. Esse Sequenciamento de Nova Geração (do inglês, *Next Generation Sequencing* – NGS) tem sido bastante utilizado em estudos genômicos que abordam hibridação e introgressão (Margullies *et al.*, 2005; McCormack, 2012; Metzker, 2010; Schendure *et al.*, 2005; Twyford & Ennos, 2011).

A crescente acessibilidade às tecnologias de NGS e o desenvolvimento simultâneo de

ferramentas bioinformáticas inovadoras para analisar grandes conjuntos de dados tornaram essa metodologia uma ferramenta amplamente difundida em pesquisas biológicas (Kumar & Kocour, 2017), o que pode ser evidenciado por um grande número de publicações científicas usando tal metodologia.

1.2. Mata Atlântica e o Gênero *Pyriglena*

A região neotropical é conhecida por sua heterogeneidade de condições ambientais e grande diversidade de fitofisionomias (Oliveira-Filho & Fontes, 2000) que vão desde florestas úmidas a desertos, sendo que destas, as florestas tropicais são reconhecidas como detentoras de boa parte da biodiversidade (Price *et al.*, 2008). Entre as florestas tropicais, a Mata Atlântica se destaca por ser considerada a segunda maior floresta pluvial tropical do continente americano e por ser uma das 25 áreas mundialmente mais importantes (*hotspots*) de biodiversidade, abrigando por volta de 1 a 8% de toda a biodiversidade do planeta (Silva & Casteleti, 2003), com mais de 8.000 espécies endêmicas (Tabarelli *et al.,* 2005), sendo essa diversidade composta por mais de 207 aves endêmicas (Brooks *et al.,* 1999).

No passado, a Mata Atlântica era constituída de mais de 1,5 milhões de km² distribuídos desde o nordeste do Brasil até o sul do país, também chegando ao leste do Paraguai e nordeste da Argentina (Câmara, 2003). Deste total, 92% se encontram em território brasileiro (Fundação SOS Mata Atlântica & INPE, 2001; Galindo-Leal & Câmara, 2003). Atualmente é considerada como um dos biomas mais degradados (Myers *et al.,* 2000) pois restam cerca de 11,4% a 16% dessa floresta (Ribeiro *et al.,* 2009). Tornam-se urgentes ações e estudos que documentem a biota da Mata Atlântica, pois além da perda de território, nela também se encontram a maioria das espécies oficialmente ameaçadas de extinção no Brasil (Tabarelli *et al,*. 2003).

Dentre as aves em perigo de extinção que habitam a Mata Atlântica, está *Pyriglena atra* (IUCN, 2012), popularmente conhecida como papa-taoca-da-Bahia, da ordem Passeriformes. O gênero *Pyriglena* (Cabanis 1847) é considerado monofilético baseado em dados filogenéticos, morfológicos e comportamentais (Isler *et al.*, 2013). Comparado às demais espécies da sua família Thamnophilidae, o gênero *Pyriglena* divergiu relativamente recentemente do seu grupo irmão (Isler *et al.*, 2013). Habitam florestas de terra firme e florestas em geral na região neotropical (Ridgely & Tudor, 1994; Willis & Oniki, 1982; Zimmer & Isler, 2003). Todos os indivíduos do gênero parecem evitar o sub-bosque aberto da floresta madura e preferir emaranhados densos nas falhas de luz (especialmente em torno de cachoeiras), perto da borda da floresta e em área de crescimento secundário. Eles forrageiam

principalmente em poleiros quase horizontais 0-3m acima do solo, mas às vezes ascendem a 10m em vinhas ou outra vegetação espessa (Isler & Maldonado-Coelho, 2017). Os indivíduos do gênero *Pyriglena* seguem formigas durante sua alimentação, mas não dependem delas, pois capturam presas longe de formigas (Willis, 1981), especialmente em áreas com baixa densidade de formigas-correição (Zimmer & Isler, 2003). Eles são, portanto, considerados seguidores regulares mais do que seguidores obrigatórios de correição de formigas (Brumfield *et al.*, 2007).

Considerado endêmico da América do Sul, o gênero é atualmente reconhecido como composto de três espécies (Remsen *et al.*, 2016): *Pyriglena leuconota* (Spix 1824), *Pyriglena leucoptera* (Vieillot 1818) e *Pyriglena atra* (Swaison 1825). Os machos dessas três espécies apresentam plumagem preta e as fêmeas, marrom. Machos de *P. atra* possuem uma mancha branca interescapular com franjas em sua parte inferior. Já os machos de *P. leucoptera* possuem manchas brancas lineares e paralelas nas asas enquanto que em *P. leuconota* tanto os machos quantos as fêmeas possuem uma mancha branca interescapular (Zimmer & Isler, 2003; Figura 1).

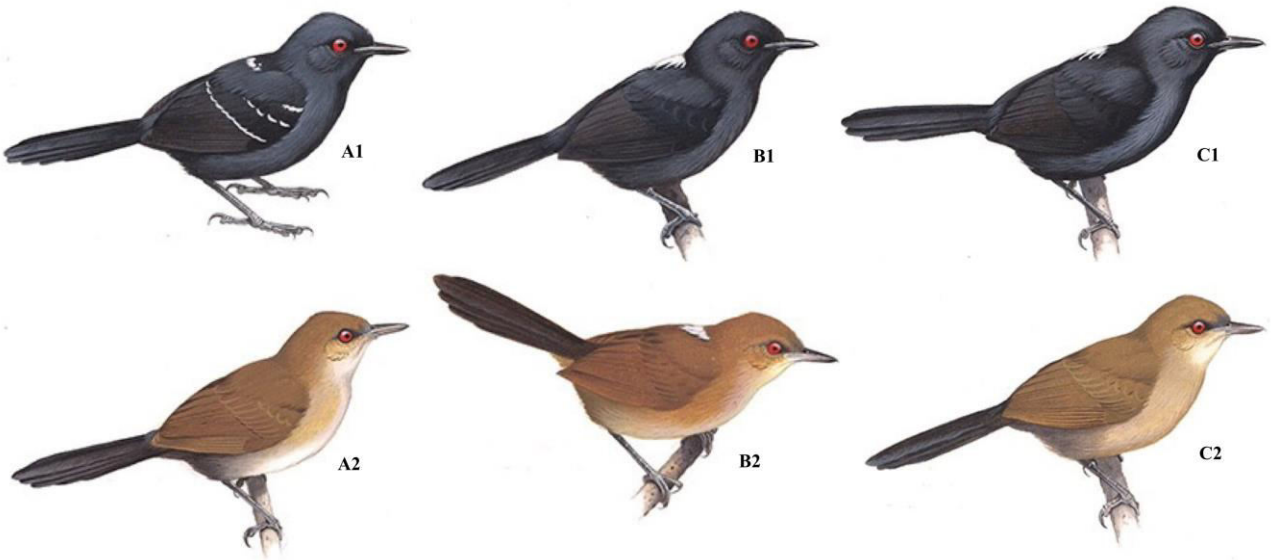


Figura 1 - *Pyriglena leucoptera* macho (A1), *Pyriglena leucoptera* fêmea (A2), *Pyriglena leuconota* macho (B1), *Pyriglena leuconota* fêmea (B2), *Pyriglena atra* macho (C1) e *Pyriglena atra* fêmea (C2). Prancha de Eduardo Brettas (2009).

P. atra e *P. leucoptera* são endêmicas da Mata Atlântica. *P. atra* ocorre em uma pequena região entre os estados brasileiros de Sergipe e Bahia. *P. leucoptera* distribui-se

desde o norte da Bahia e vai se estendendo em direção ao Rio Grande do Sul, chegando no Paraguai e Argentina. *P. leuconota* distribui-se principalmente na Amazônia, possuindo, também, indivíduos na Mata Atlântica localizados entre os estados brasileiros de Pernambuco e Alagoas (Zimmer & Isler, 2003; Figura 2).

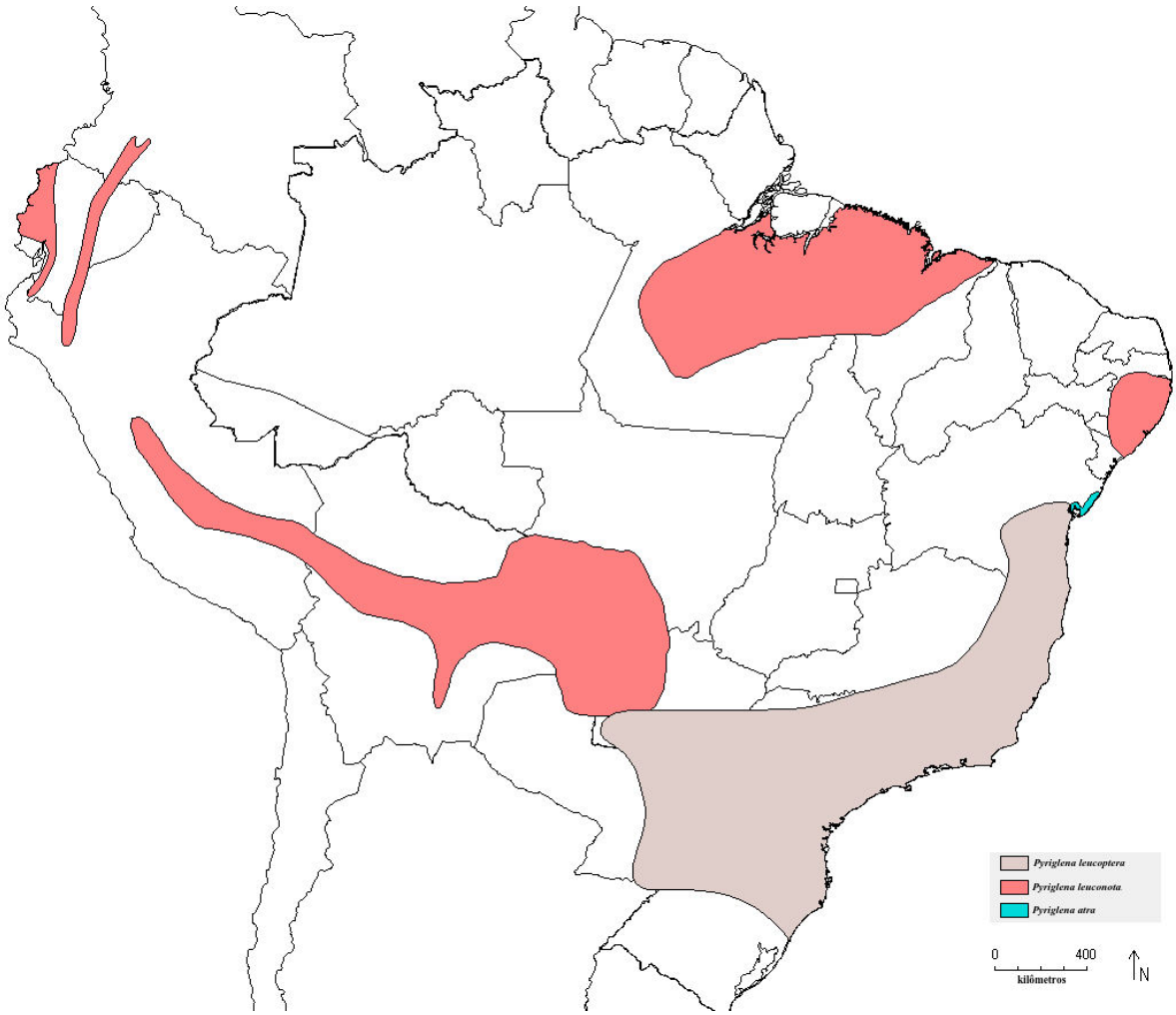


Figura 2 – Distribuição de espécies do gênero *Pyriglena* na América do Sul (BirdLife International, 2014 (http://www.birdlife.org) plotados em um único mapa no DIVA-GIS (Hijmans *et al*., 2001). Salmão: *P. leuconota*; cinza: *P. leucoptera*; azul turquesa: *P. atra.*

Em uma análise filogenética baseada em dados mitocondriais *P. leuconota* foi distribuída em dois clados, um com indivíduos da Mata Atlântica e outro com indivíduos amazônicos; já *P. atra* e *P. leucoptera* foram recuperadas em um único clado, mas não aparecem como reciprocamente monofiléticas (Maldonado-Coelho, 2010; Figura 3). Esse resultado foi corroborado na rede de haplótipos (Figura 4) onde há compartilhamento de um haplótipo de alta frequência entre *P. atra* e a linhagem mais ao norte de *P. leucoptera*.

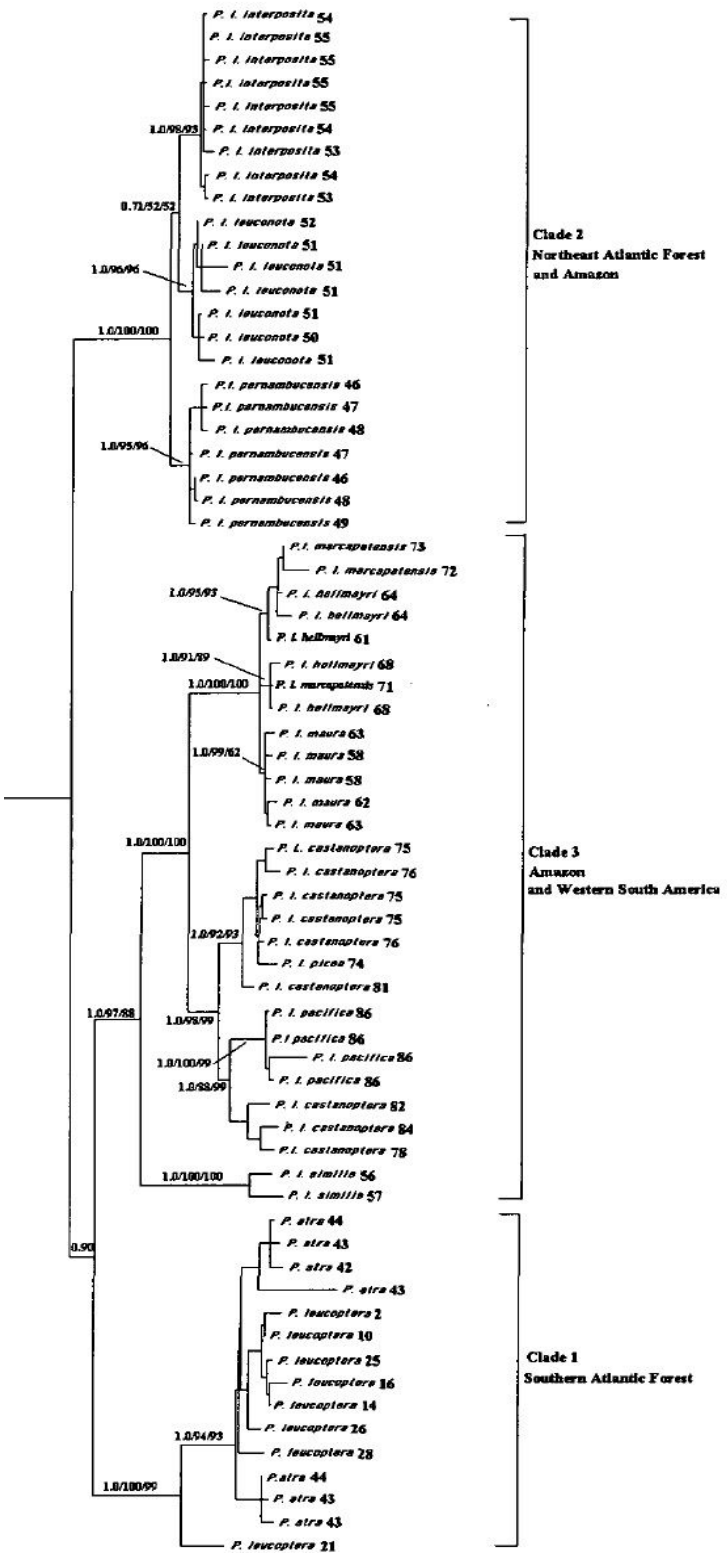


Figura 3 – Árvore de máxima verossimilhança das espécies do gênero *Pyriglena* baseada em 3213pb de quatro genes mitocondriais (modificado de Maldonado-Coelho, 2010). Clado 1: *P. atra* e *P. leucoptera*; clados 2 e 3: *P. leuconota*. Grupos externos retirados dessa imagem.

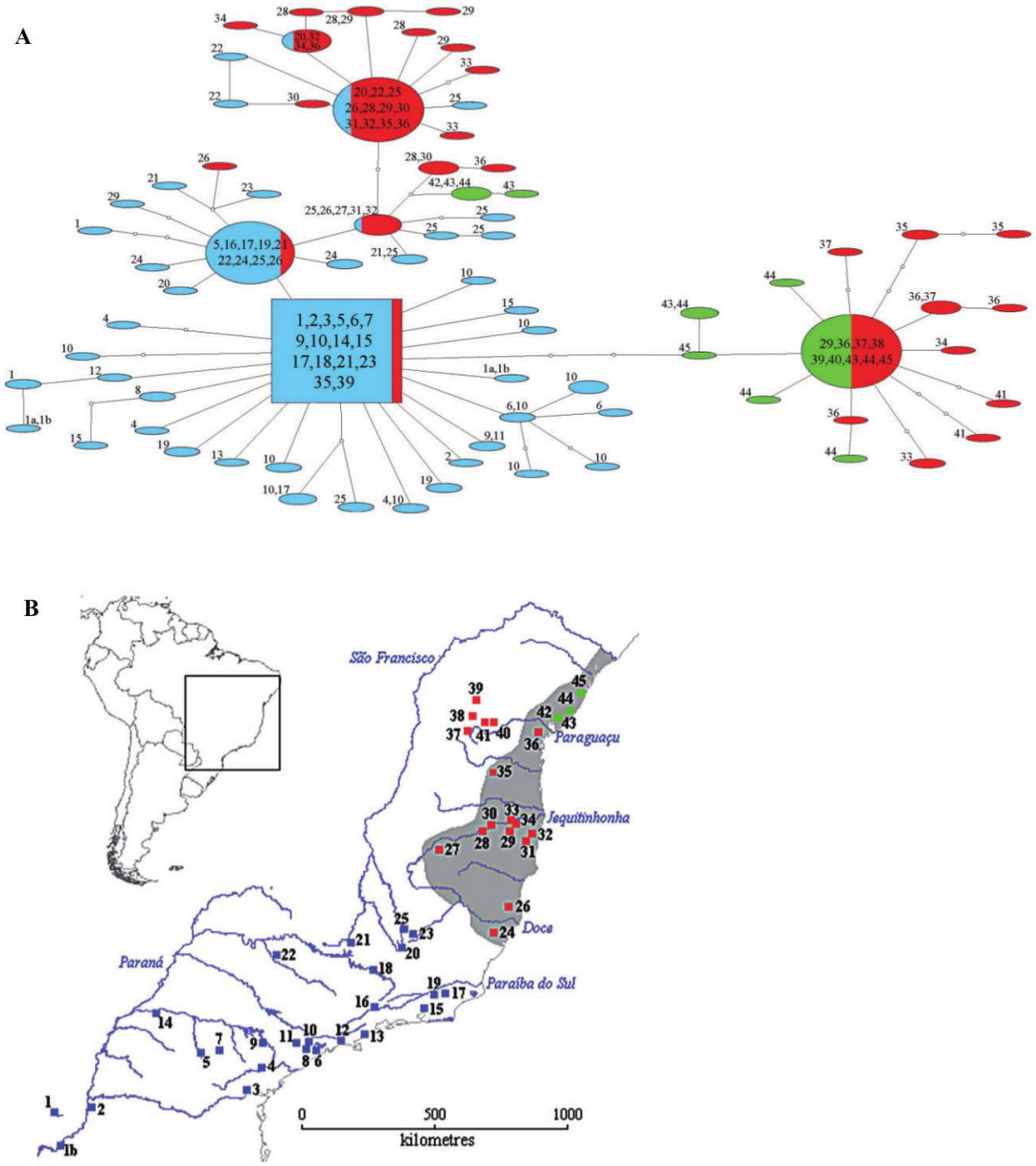


Figura 4 – (A) Rede de haplótipos de *Pyriglena leucoptera* e *P. atra* baseada em 1041pb da subunidade II de NADH desidrogenase 2. Cada círculo representa um haplótipo diferente com tamanho proporcional à sua frequência. Em azul estão representadas sequências de indivíduos de *P. leucoptera* do sul da Mata Atlântica, em vermelho, as de indivíduos de *P. leucoptera* da região central da Mata Atlântica e em verde, as de indivíduos de *P. atra*. Os números correspondem às localidades de amostragem identificados no mapa (B). Fonte: Maldonado-Coelho (2012).

Outra característica muito interessante desse grupo de aves é que há intermediários morfológicos entre *P. atra* e *P. leucoptera* (Willis & Oniki 1982; Figura 5) na região do vale do Rio Paraguaçu, próximo a Salvador/BA, mais precisamente nos arredores da cidade de Ipirá/BA (Maldonado-Coelho 2012 ; Willis & Oniki, 1982). A presença de intermediários morfológicos indica a existência de hibridação e ainda, possível introgressão.

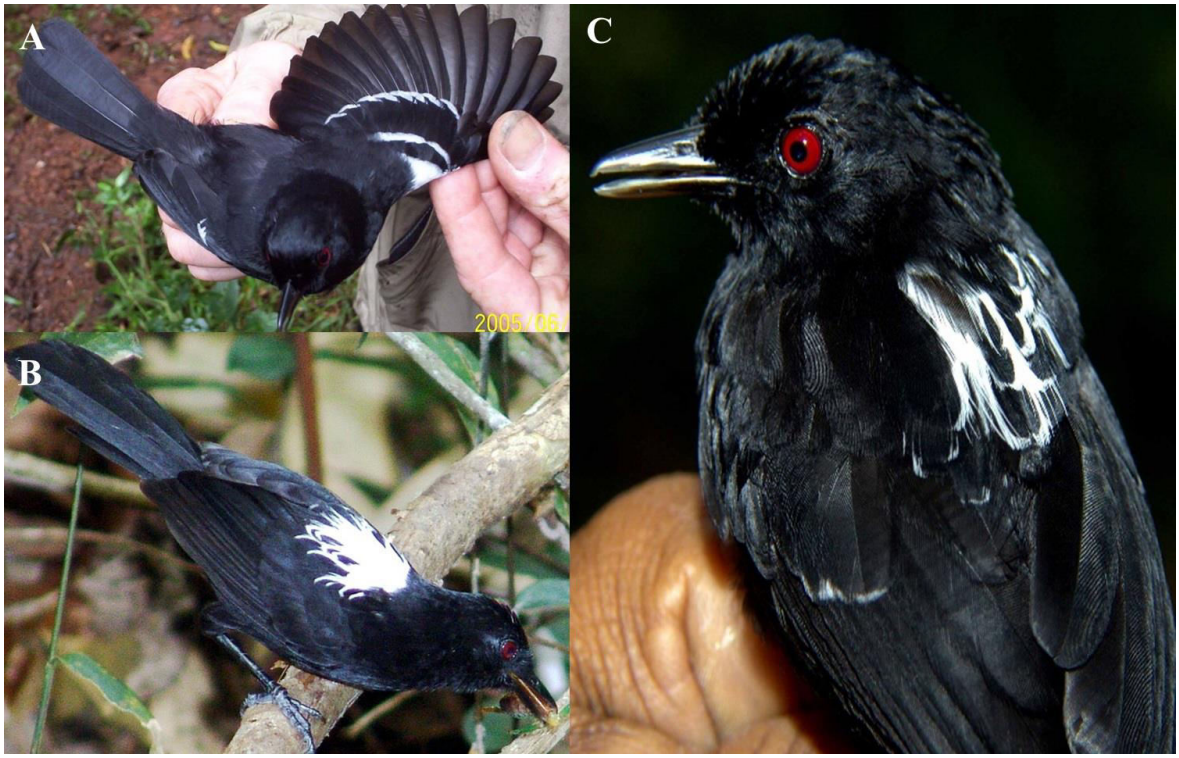


Figura 5 – (A) Detalhe das barras brancas na asa de um macho de *P. leucoptera*. (B) *P. atra* com a mancha interescapular branca típica. (C) Indivíduo intermediário morfológico com barras brancas na asa e mancha interescapular branca. Fotos: Marcos Maldonado-Coelho.

As consequências dessa hibridação são especialmente importantes no caso de *P. atra* que ocorre em uma área restrita e bastante fragmentada nos estados de Sergipe e Bahia (Figura 2) e é considerada ameaçada de extinção (IUCN, 2016). Enquanto *P. leucoptera* é mais abundante. No caso de *P. atra*, devido ao pequeno tamanho populacional, alelos bem adaptados poderiam ser substituídos mesmo com baixas taxas de hibridação. Com isso, pode haver redução do *fitness* devido à depressão por endogamia (Edmands, 2007; Templeton, 1986) e devido à deriva genética (Shrader-Frechette & McCoy, 1993). Portanto, caracterizar se há introgressão em direção à *P. atra* é especialmente importante para planejar ações para sua conservação. Como o estudo da dinâmica dessa região de contato e das forças evolutivas envolvidas ainda foi pouco explorada, o presente estudo pretende compreender o processo de hibridação nessa região baseado em marcadores em escala sub-genômica.

2. Objetivos

No presente estudo analisamos milhares de marcadores de indivíduos das espécies parentais *P. leucoptera* e *P. atra* e de indivíduos morfologicamente intermediários de modo a compreender e caracterizar o processo de hibridação natural e as consequências do fluxo gênico na manutenção de espécies.

Buscamos responder as seguintes perguntas:

- Qual a estrutura genômica populacional das espécies e como os intermediários morfológicos se inserem?

- Existe introgressão genômica? Em qual direção?

- Existem regiões genômicas sob seleção natural?

3. Material e Métodos

3.1. Amostragem

Para o desenvolvimento deste estudo foram selecionados 48 indivíduos identificados morfologicamente, sendo 13 *P. atra*, 20 *P. leucoptera* e 15 indivíduos da zona de hibridação (Tabela 1). Todas as amostras estão depositadas na coleção de tecidos do Laboratório de Genética e Evolução Molecular de Aves da Universidade de São Paulo (LGEMA-IBUSP). Os indivíduos selecionados são de ambos os sexos, mas o número total de machos é maior (Tabela 1), pois machos das duas espécies parentais podem ser facilmente identificados pela plumagem, ao contrário das fêmeas (Willis & Oniki 1982). Assim, com os machos podemos testar se há correlação entre genótipo e fenótipo. A amostragem das espécies parentais foi planejada de modo a incluir localidades equidistantes em relação à zona de hibridação, ou seja, localidade 1 de *P. atra* e localidade 4 de *P. leucoptera* que distam cerca de 285 km da localidade 3 (Figura 6). Já a localidade 2 (*P. atra*) dista de 42 km da localidade 3. Também foram amostradas localidades de *P. leucoptera* bem distantes da zona de hibridação (localidades 5 e 6; Figura 6) de modo a termos indivíduos mais provavelmente "puros" (com pouca ou nenhuma introgressão).

Tabela 1 – Amostragem de *Pyriglena atra* e *P. leucoptera*. O número (No.) corresponde aos indicados no mapa da Figura 6, localidade (cidade/estado), espécie e número de indivíduos (N), número da coleção LGEMA e sexo (M ou F). * - híbridos morfológicos.

No.	Localidade	Espécie (N)	LGEMA (sexo)
1	Itaporanga D'ajuda/ SE	*Pyriglena atra* (10)	15362 (M), 16545(M), 16548(M), 16550(M), 16551(M), 16552 (F), 16553 (F), 16554 (F), 16555(M), 16556(M)
2	Serra Preta/ BA	*Pyriglena atra* (3)	17112 (M), 17113 (M), 17114 (F)
3	Ipirá/ BA	Zona de Hibridação (15)	17921* (M), 17922* (M), 17923 (M), 17924 (F), 17925 (M), 17926* (M), 17927 (M), 17928 (M), 17929 (F), 17930 (M), 17931* (M), 17934* (M), 17935 (M), 17936 (F), 17937* (M)
4	Itacaré/ BA	*Pyriglena leucoptera* (10)	16685 (M), 16702 (M), 16704 (M), 16705 (F), 16706 (F), 16707 (M), 16708 (M), 16709 (M), 16710 (M), 16711 (F)
5	Telemaco Borba/ PR	*Pyriglena leucoptera* (3)	MMC232 (M), MMC233 (M), MMC234 (M)
6	Foz do Iguaçu/PR	*Pyriglena leucoptera* (7)	MMC221 (F), MMC222 (F), MMC223 (M), MMC224 (M), MMC225 (M), MMC226(M), MMC230 (M)

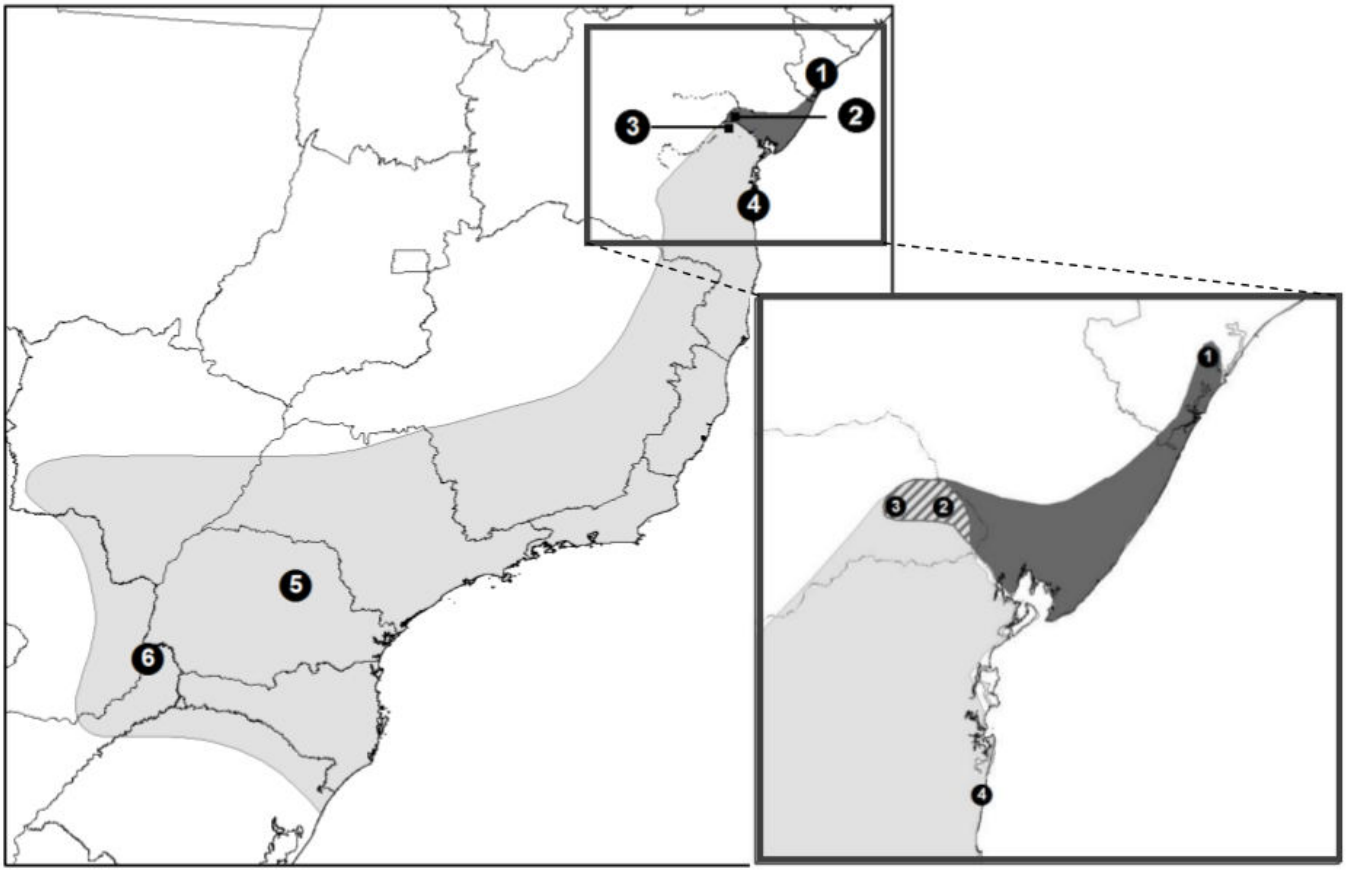


Figura 6 – Distribuição geográfica de *P. atra* (cinza escuro) e de *P. leucoptera* (cinza claro). Localidades numeradas conforme Tabela 1. No detalhe, a região que abrange a zona de hibridação (hachurado).

3.2. Métodos moleculares

3.2.1. Extração do DNA

Para realizar a extração de DNA utilizamos o Kit DNeasy Blood & Tissue (Qiagen) e o protocolo padrão fornecido pela empresa. A qualidade do DNA extraído foi verificada em gel de agarose 1%. A quantificação foi realizada com Qubit 2.0 Fluorometer. A amostra final tinha volume de 20 µl com aproximadamente 400 ng (concentração de 20 ng/µl).

3.2.2. Preparo da Biblioteca e Sequenciamento

A biblioteca foi preparada na empresa ECOMOL Consultoria e Projetos LTDA ME conforme o protocolo de *Genotyping-by-sequencing* (GBS) padronizado por Elshire *et al.* (2011). A enzima de restrição utilizada foi a *Pst* I com sítio de reconhecimento de 6 pb (5'-CTGCAG-3')

que também foi utilizada por Weir *et al.* (2015). A plataforma Illumina HiSeq 2000 da Fundação de Estudos Agrários Luiz de Queiroz (Piracicaba/SP) foi utilizada para o sequenciamento *paired-end* gerando dois arquivos de sequências, cada um correspondente a uma das leituras (R1 e R2).

3.3. Análises Computacionais e Bioinformática

As análises foram realizadas com dois conjuntos de dados:

1. Todas as amostras das 6 localidades,
2. Amostras das localidades 1 a 4 de modo a concentrar na zona de hibridação e seu entorno.

Para cada programa computacional usado no presente trabalho foi produzido um anexo com a tradução do manual original do programa adicionado de informações e procedimentos que realizei. Esse material ficará disponível na página na internet do nosso laboratório para *download* e consulta.

3.3.1. Tratamento dos Dados Brutos do Sequenciamento

Os dados brutos do sequenciamento foram analisados quanto à qualidade no FastQC (Andrews, 2012 – Anexo 1). Como esperado, parte dos dados brutos apresentou baixa qualidade em relação aos valores padrão, assim, foi necessário filtrar os dados e para isso utilizamos o pyRAD (Eaton, 2014 – Anexo 2) que é um conjunto de *scripts* destinado a processar dados obtidos de bibliotecas que utilizam enzima de restrição, como foi o caso do presente estudo. Com o pyRAD foi possível separar as leituras (*reads*) referentes à amostra de cada indivíduo, como também retirar as sequências dos adaptadores e sequências com baixa cobertura. Além do pyRAD foi utilizado o programa PEAR (Zhang *et al.*, 2014 – Anexo 2), que consiste em um conjunto de *scripts* desenvolvido para a comparar e alinhar arquivos *paired-end* (nomeados R1 e R2). Em seguida a qualidade das sequências foi novamente avaliada no programa FastQC para verificar se a limpeza realizada ocorreu a contento. Somente foram mantidas sequências de boa qualidade e com boa cobertura, separadas conforme o *barcode* da amostra e em formatos e extensões para os passos seguintes de análise.

3.3.2. Análise dos Dados

Após a filtragem, utilizei o Bowtie 2 (Langmead & Salzberg, 2012 – Anexo 3) que é uma

ferramenta para alinhar leituras (*reads*) com alguma sequência de referência. Isso foi feito visando avaliar se no conjunto de dados existiam *reads* pertencentes ao parasita do gênero *Plasmodium*, ao genoma mitocondrial e ao cromossomo Z. Como alguns dos indivíduos sequenciados estavam infectados com esse parasita (M. Svensson-Coelho, com. pess.), para identificar se haviam *reads* dele, foi realizado o alinhamento dos *reads* obtidos com a sequência FASTA de *Plasmodium gallinaceum* obtido do banco de dados *online* do gênero *Plasmodium* (http://plasmodb.org/common/downloads/). Como o genoma mitocondrial, o cromossomo Z e os autossomos possuem padrões de herança diferentes, avaliamos se haviam *reads* pertencentes ao genoma mitocondrial e ao cromossomo Z para que eles pudessem ser analisados separadamente em relação aos dados autossômicos. Para averiguar se haviam *reads* mitocondriais foi realizado o alinhamento com o genoma mitocondrial de *Pyriglena leucoptera* obtido por Amaral (com. pess.). Para conferir se haviam *reads* do cromossomo Z foi utilizada a sequência do cromossomo Z de *Gallus gallus gallus* (Bellott *et al.*, 2017; acesso no GenBank: GENL00000000).

Para inferir a estrutura populacional, o número de populações distintas, atribuir indivíduos às populações e identificar possíveis migrantes e indivíduos miscigenados foi utilizado o *sNMF* (Frichot *et al.*, 2014 – Anexo 4). O ADMIXTURE (Alexander *et al.*, 2009 – Anexo 5) e o fastStructure (Raj *et al.*, 2014 – Anexo 6) também foram testados com os meus dados e produzi manuais traduzidos sobre eles. Porém, os resultados do ADMIXTURE e do fastStructure indicaram ausência de mistura de ancestralidade (dados não apresentados) e, assim, não parecem ser biologicamente reais, enquanto os resultados do sNMF mostraram certa mistura e foram considerados biologicamente mais consistentes. Além disso, o sNMF permite realizar uma análise adicional de mascaramento parcial de dados, ou seja, conjuntos com parte dos SNPs (no presente caso, 100 conjuntos com 95% dos SNPs) foram amostrados de modo aleatório e cada um deles foi analisado. O resultado final considerou os resultados de todas essas análises.

O sNMF utiliza algoritmos baseados em fatorização de matriz não-negativa esparsa (NMF) e otimização de mínimo-quadrado. Os algoritmos NMF são flexíveis e robustos e se baseiam em suposições de modelos genéticos populacionais tradicionais. A análise foi realizada variando o valor de K (número de populações) de 1 a 10, com 100 corridas independentes para cada um, e cada valor de K foi testado com quatro valores de alpha: 1, 10, 100 e 1000. A avaliação do melhor K foi realizada utilizando o Critério de Entropia Cruzada e os valores de cada corrida foram plotados em um gráfico para avaliar o valor de K com menor valor de entropia usando um

script no R.

Para avaliar se há SNPs candidatos a estarem sob seleção foi realizada análise no BayeScan (Fischer *et al.*, 2011; Foll & Gaggiotti, 2008; Foll *et al.*, 2010 – Anexo 7) utilizando diferenças nas frequências de alelos entre as populações observadas. O BayeScan é baseado no modelo multinomial-Dirichlet e utiliza diferenças nas frequências de alelos como explicado a seguir. Um dos cenários mais simples que pode ser assumido é constituído por um modelo de ilha no qual as frequências de alelos das populações são comparadas às frequências de um *pool* de alelos migrantes compartilhados em diferentes níveis entre as populações, ou seja, esse *pool* corresponde a alelos em comum entre pelo menos duas populações. A diferença nas frequências de alelos entre este *pool* alélico e cada população é medida por F_{ST}. Esse modelo permite considerar cenários ecológicos relativamente realistas nos quais o tamanho efetivo e a taxa de imigração podem diferir entre as populações. Assim, para testar se há sinal de seleção é realizada a decomposição dos coeficientes de F_{ST} em um componente específico populacional compartilhado por todos os locos (beta) e em um componente específico de locos compartilhados por todas as populações (alfa) usando uma regressão logística. A ausência de neutralidade em um dado loco é assumida quando o componente loco-específico é necessário para explicar o padrão de diversidade observado, ou seja, alfa significativamente diferente de 0. Um valor positivo de alfa sugere presença de seleção diversificadora, enquanto valores negativos sugerem seleção balanceadora ou purificadora. Assim, para cada loco o alfa pode ou não ser considerado no modelo de seleção. Para cada loco, um MCMC de salto reversível explora modelos com e sem seleção (componente alfa presente ou ausente, respectivamente) e estima sua probabilidade posterior. Para utilizar o programa, o arquivo de extensão ".vcf" foi convertido em extensão reconhecível pelo BayeScan com o uso do PGDSpider (Lisher & Excoffier, 2009 – Anexo 7). Os parâmetros foram definidos para uso de dados de SNPs e as populações foram definidas conforme os resultados do sNMF (2, 3 e 4 populações, ver Resultados). Os testes de seleção foram realizados utilizando os parâmetros padrões do BayeScan nas populações indicadas pelos resultados do sNMF.

Para verificar se há evidência de eventos de migração foi utilizado o TreeMix (Pickrell & Pritchard, 2012 – Anexo 8). Esse programa também infere as relações históricas entre as populações. Utilizei arquivo no formato .geno que foi convertido no arquivo de entrada .treemix usando o *script* input_treemix_generator.py (Alvarado-Serrano, 2014). A análise foi realizada

com sete arquivos de entrada diferentes: (1 e 2) considerando duas populações que correspondem às duas espécies, *P. atra* e *P. leucoptera*, sendo que os indivíduos da zona de hibridação foram considerados como *P. atra*; nesse caso foram analisados indivíduos somente das localidades de 1 a 4, ou todas as seis localidades (Tabela 1, Figura 6); (3 e 4) considerando as três populações indicadas pelo sNMF; também analisando indivíduos somente das localidades de 1 a 4, ou todas as seis localidades; (5 e 6) considerando quatro populações conforme indicadas pelo sNMF; novamente analisando indivíduos somente das localidades de 1 a 4, ou todas as seis localidades; e (7) considerando seis populações com todas as seis localidades amostradas (Tabela 1, Figura 6), sendo considerado cada localidade como uma população distinta. Em todas as corridas foram permitidos que os eventos de migração variassem de zero (m=0; ausência de migração) até quatro (m=4). Com os arquivos de saída foi gerada uma representação gráfica em formato de árvore cujos terminais correspondiam às populações e possíveis eventos de migração. Para confirmar os resultados gerados pelo Treemix, utilizamos as estatísticas f3 (Teste de 3 populações; Reich *et al.*, 2009) e f4 (Teste de 4 populações; Reich *et al.*, 2009) para os dados que envolviam três e quatro populações, respectivamente. O Teste de 3 populações é baseado em uma "estatística f3", uma generalização de F_{ST} entre três populações. Esta estatística é igual ao produto interno das diferenças de frequência entre um grupo X e dois outros grupos A e B, é proporcional à deriva genética correlacionada entre os grupos A e X e os grupos A e B. Se X não apresenta sinal de ancestralidade misturada, é esperado que esse valor seja positivo, porque a deriva genética ao longo da linhagem que leva do antepassado ao X deve ser positiva. Em contraste, se o grupo X surgiu de uma mistura de grupos relacionados a A e B, o valor da estatística f3 pode ser significativamente negativo, o que é interpretado como um sinal de mistura. O teste de 4 populações segue o mesmo modelo do Teste de 3 populações mas é mais sensível. Nele é possível avaliar se uma árvore filogenética não enraizada é consistente com os dados de frequências alélicas. É calculado uma "estatística f4", que deverá ser proporcional à correlação nas diferenças de frequência alélica entre pares de grupos. Se a topologia ((A, B),(C, D)) estiver correta, as diferenças de frequência entre A e B devem refletir a deriva genética que não está correlacionada com a de C e D. Assim, o valor esperado do produto das diferenças de frequência é zero. Os desvios significativos de 0 das estatísticas f4 são interpretados para as três possíveis topologias como evidência de que os quatro grupos não podem ser relacionados por uma filogenia sem mistura.

4. Resultados

Após o sequenciamento foram obtidos 476.168.518 *reads* totalizando 59.997.233.268 pb. Após a triagem inicial dos dados (*reads*) utilizando o FastQC e o pyRAD e adotando máximo de 6,25% de *missing data* foi obtido o total de 23.179 SNPs de 48 indivíduos amostrados em seis localidades. Para as análises com indivíduos somente das quatro localidades do nordeste (zona de hibridação e seu entorno) foi obtida uma matriz de 28.743 SNPs de 38 indivíduos (também com máximo de 6,25% de *missing data*)

Os alinhamentos no Bowtie 2 dessas matrizes de 23.179 e 28.743 SNPs com o genoma de *Plasmodium gallinaceum*, o mitogenoma de *Pyriglena leucoptera* e o cromossomo Z de *Gallus gallus gallus* não mostraram evidências de presença de *reads* desses genomas (Tabela 2), assim, tratamos todos esses SNPs como autossômicos.

Tabela 2 – Porcentagem de *reads* de origem do genoma mitocondrial, do cromossomo Z e do genoma de *Plasmodium gallinaceum* após alinhamento no Bowtie 2 das matrizes com indivíduos das seis e quatro localidades. Amostras com asterisco indicam indivíduos infectados por *Plasmodium* (M. Svensson-Coelho, comunicação pessoal).

	6 localidades (23179 SNPs)			**4 localidades (28743 SNPs)**		
Amostra	**% mtDNA**	**% Z**	***% Plasmodium***	**% mtDNA**	**% Z**	***% Plasmodium***
16705*	0	0	0,03	0	0,01	0,02
16706	0	0	0,03	0	0,01	0,02
16709*	0	0	0,03	0	0	0,02
17112	0	0	0,03	0	0,01	0,02
MMC223*	0	0	0,02	-	-	-
MMC224	0	0	0,03	-	-	-
MMC226	0	0	0,03	-	-	-
MMC233*	0	0	0,03	-	-	-

A análise de estrutura populacional utilizando o sNMF indicou, tanto para a amostragem total (todas as seis localidades) quanto para a subamostragem (quatro localidades), que os melhores números de populações (K) seriam 2, 3 e 4, como ilustrado nos gráficos de valores de entropia cruzada (Figuras 7 e 8). O critério adotado para selecionar o melhor K foi o de possuir o menor valor de entropia. Nesses gráficos é possível observar que em geral os valores de alfa testados resultaram em distribuições sobrepostas de valores de entropia cruzada. Porém para a amostragem das seis localidades o alfa 10 apresentou menores valores para K= 3 e 4, enquanto o alfa 100, para o K= 2 (Figura 7). Já para a subamostragem de quatro localidades, o alfa 10

apresentou menores valores para K= 2 e o alpha 1, para K= 3 e 4 (Figura 8).

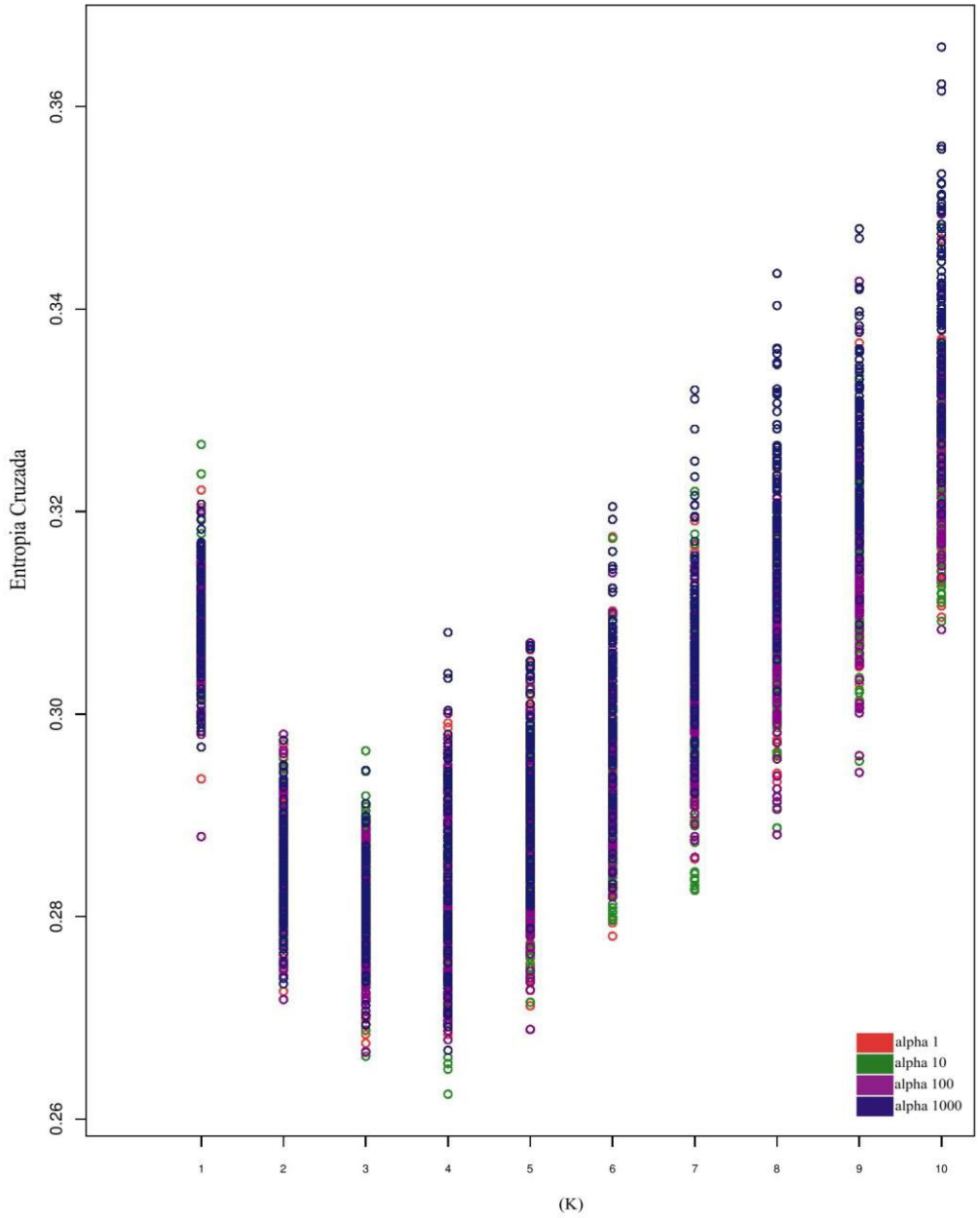


Figura 7 – Gráfico de Entropia Cruzada gerado com os dados do sNMF para diferentes valores de alfa (1,10, 100 e 1000) e de número de populações (K; 1 a 10) considerando amostragem total (seis localidades). Foram realizadas 100 repetições para cada alfa e cada K.

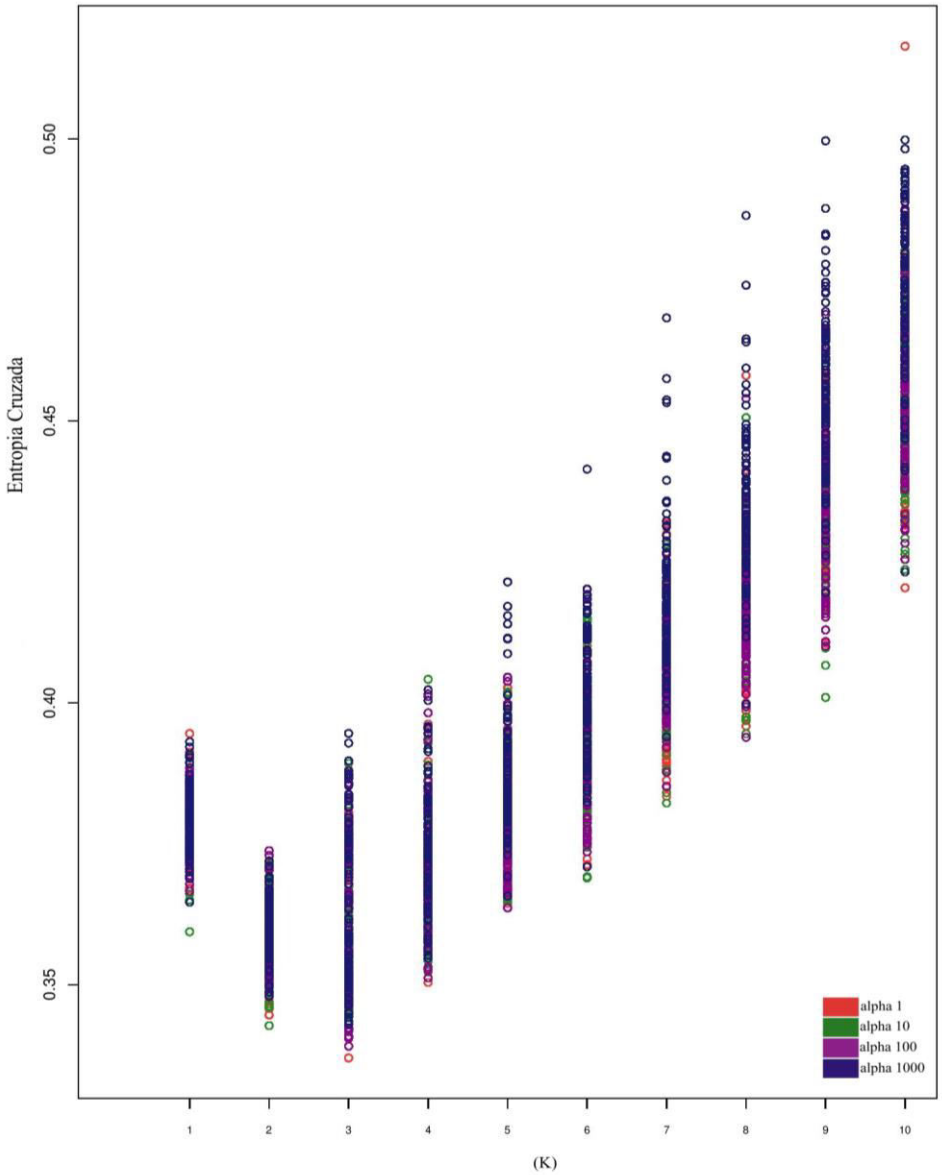


Figura 8 – Gráfico de Entropia Cruzada gerado com os dados do sNMF para diferentes valores de alfa (1,10, 100 e 1000) e de número de populações (K; 1 a 10) considerando subamostragem (quatro localidades). Foram realizadas 100 repetições para cada alfa e cada K.

A composição genética dos 48 indivíduos adotando K= 2 (Figura 9A), indica separação de acordo com as espécies, sendo os indivíduos da zona de hibridação incluídos no grupo de *P. atra.* Adotando K= 3 (Figura 9B), indivíduos de uma das duas localidades onde *P. atra* foi amostrada compuseram um grupo separado dos demais, indivíduos da outra localidade com *P. atra* se agruparam com indivíduos da zona de hibridação e indivíduos de *P. leucoptera* continuaram

agrupados em uma única população. Já para K= 4 (Figura 9C), em relação aos grupos descritos para K= 3 surge uma subdivisão dentro do grupo de *P. leucoptera* separando indivíduos da localidade central (4) e as localidades ao sul (5 e 6). No caso da subamostragem que considerou somente as quatro localidades no ou próximos do rio Paraguaçu, para K= 2 (Figura 10A), as populações correspondem às duas espécies com a inclusão dos indivíduos da zona de hibridação no agrupamento de *P. atra.* Para K= 3 (Figura 10B), indivíduos de uma das duas localidades onde *P. atra* foi amostrada compuseram um grupo separado dos demais, indivíduos da outra localidade com *P. atra* se agruparam com indivíduos da zona de hibridação e indivíduos de *P. leucoptera* continuaram agrupados em uma única população. Para K= 4 (Figura 10C), em relação aos grupos descritos para K= 3 surge a separação de indivíduos de *P. atra* (localidade 2) e os da zona de hibridação.

Considerando K= 2, tanto com as seis localidades quanto considerando apenas quatro localidades, é possível notar que existe mistura genética principalmente em indivíduos das localidades 2, 3 e 4 (Figuras 9A e 10A). Sendo a contribuição do primeiro agrupamento (representado pela cor rosa) de 10% até 20% em todos os indivíduos da localidade 4 e de 8% em um indivíduo da localidade 5; enquanto a contribuição do segundo agrupamento foi de no máximo a 10% nas localidades 2 e 3.

O padrão se repete com K= 3 (Figuras 9B e 10B) com as localidades 2, 3 e 4 apresentando maior porcentagem de mistura, com destaque para a localidade 2 que possui indivíduos com 21% a 33% de contribuição do agrupamento rosa e por volta de 10% de contribuição do agrupamento amarelo. Os três agrupamentos apresentam indivíduos com mistura genética, porém o agrupamento amarelo apresenta mais indivíduos com mistura. A localidade 3, que é a zona de hibridação, apresentou 33% dos indivíduos com alguma mistura e esta representa uma porcentagem baixa. É ainda interessante destacar que a maioria dos híbridos morfológicos (indicados com asterisco) não apresentou nenhuma mistura, com exceção do indivíduo 17931 com 6,5% de sua composição indicando mistura genética.

Para K= 4 (Figuras 9C e 10C) é observado um padrão diferente. Os agrupamentos rosa e azul possuem contribuição dos agrupamentos amarelo e verde na faixa de 1% a 6%, enquanto esses dois últimos agrupamentos recebem uma contribuição que varia de 6% a 15% dos dois primeiros agrupamentos, ou seja, recebem o dobro de contribuição.

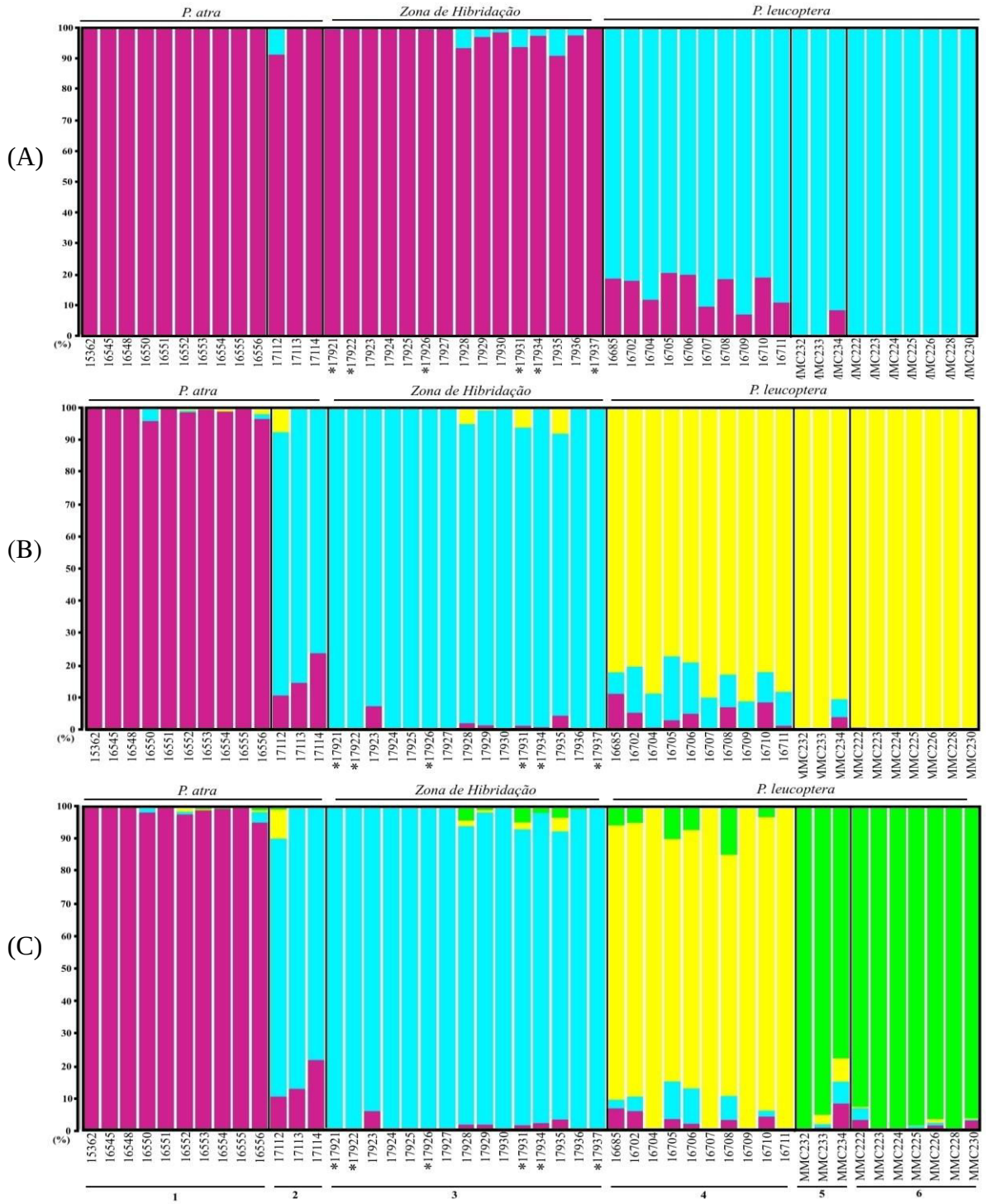


Figura 9 - Resultados das análises no sNMF de *P. atra* (N=13)/ *P. leucoptera* (N=20)/ zona de hibridação (N= 15) baseadas em 23.179 SNPs. A identificação taxonômica dos indivíduos foi realizada segundo sua morfologia externa. Diferentes cores representam os agrupamentos genéticos distintos observados. Os números correspondem aos indivíduos amostrados (Tabela 1). Barras verticais pretas separam as localidades amostradas (1 a 6). Asterisco ao lado do número indica indivíduo intermediário morfológico (híbrido morfológico).

(A) K= 2. (B) K= 3. (C) K= 4.

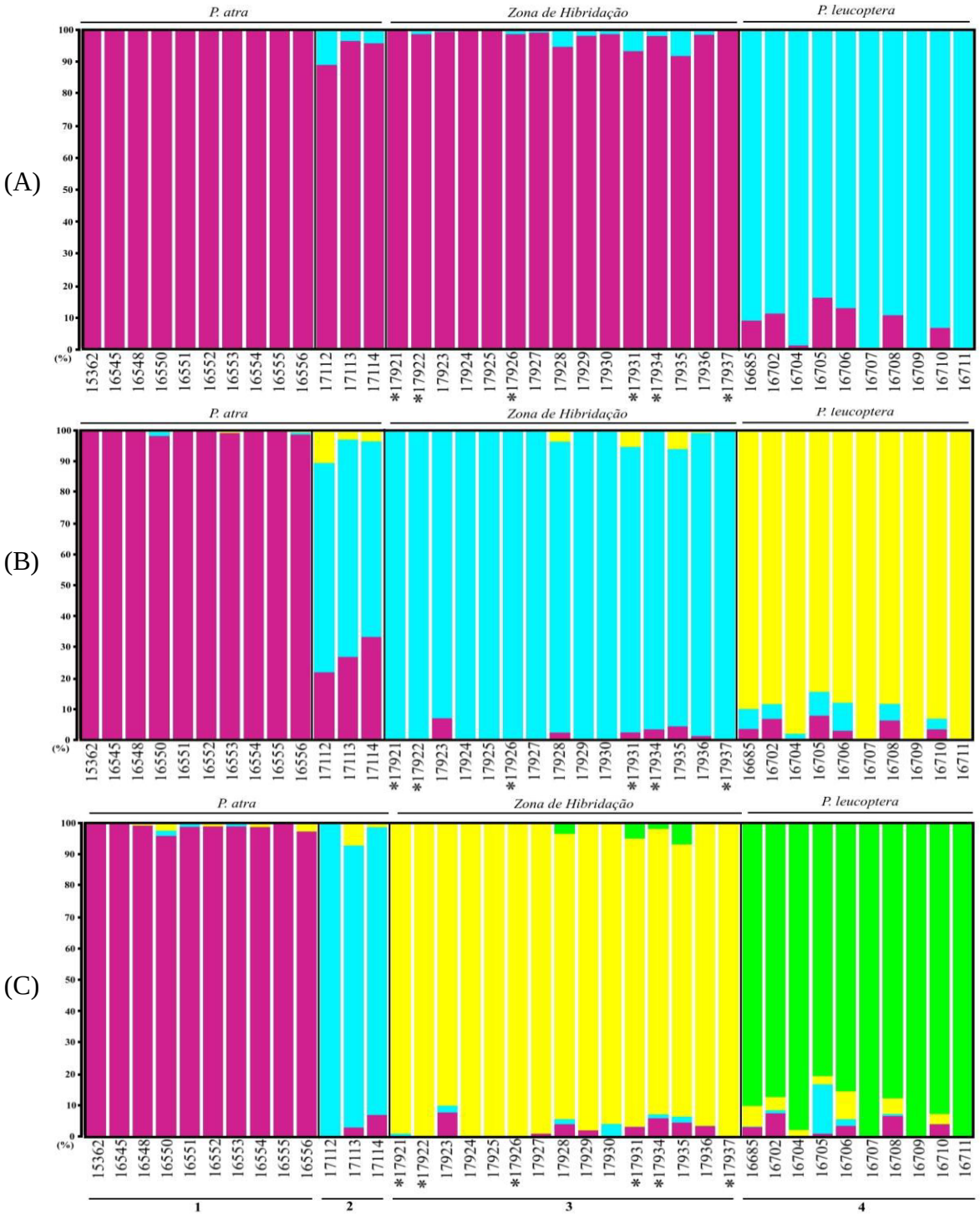


Figura 10 - Resultados das análises no sNMF de *P. atra* (N=13)/ *P. leucoptera* (N=10)/ zona de hibridação (N= 15) baseadas em 28.743 SNPs. A identificação taxonômica dos indivíduos foi realizada segundo sua morfologia externa. Diferentes cores representam os agrupamentos genéticos distintos observados. Os números correspondem aos indivíduos amostrados (Tabela 1). Barras verticais pretas separam as localidades amostradas (1 a 4). Asterisco ao lado do número indica indivíduo intermediário morfológico (híbrido morfológico). (A) K= 2. (B) K= 3. (C) K= 4.

A análise do BayeScan foi realizada levando em conta as divisões populacionais dos melhores valores de K (2, 3 e 4), sendo considerados os resultados tanto para amostragem total (seis localidades) quanto para a submostragem (quatro localidades). Quando foram adotados os grupos resultantes do K= 2 de ambos conjuntos de dados, teríamos de aceitar um valor de falsos negativos de SNPs *outliers* superior a 30% (FDR=0.3), o que seria pouco estringente e, portanto, os resultados considerando esses grupos de K= 2 foram descartados. Foi estabelecido um máximo de 5% de falsos negativos (FDR=0.05) de SNPs *outliers*, o que é bastante estringente. Para a matriz com a amostragem total e considerando os grupos resultantes de K= 3 não foram detectados SNPs *outliers* (Tabela 3). Ainda com essa matriz da amostragem total e considerando os grupos resultantes de K= 4 foram observados 56 SNPs *outliers*, cerca de 0,24% do total de 23179 SNPs (Tabela 3). Dentre os 56 *outliers* (Tabela 3), apenas três apresentaram valores negativos de alfa (indicativo de seleção balanceadora ou purificadora) e os demais poderiam estar sob seleção positiva. Já para a matriz de dados subamostrados, ao considerar os grupos de K= 3 e K= 4 foram detectados 76 (0,3% de um total de 28743 SNPs) e 192 (0,7%) SNPs *outliers*, respectivamente (Tabela 4). Dentre os 56 *outliers* (Tabela 3), apenas três apresentaram valores negativos de alfa (seleção balanceadora ou purificadora) e os demais podem ser considerados como sob seleção positiva. Somente um dos 76 *outliers* da análise considerando K= 3 indicou sinal de estar sob seleção negativa e os demais indicaram seleção positiva. Todos os 192 *outliers* da análise considerando K= 4 indicaram seleção positiva.

Tabela 3 – SNPs identificados como *outliers* na análise no BayeScan da amostragem total (48 indivíduos de seis localidades, 23.179 SNPs) considerando os grupos (populações) do resultado do sNMF de K= 4. SNP: identificação do SNP; q-valor: valores de q-value, alpha: valor de alfa e F_{ST}: valor de F_{ST}.

SNP	qval	alpha	F_{ST}	SNP	qval	alpha	F_{ST}
50	0.020265	1.5934	0.81001	**8699**	0.022333	1.569	0.80632
200	0.021764	1.5614	0.80512	**9055**	0.021589	1.5768	0.80671
201	0.018984	1.5709	0.80653	**9227**	0.022562	1.5494	0.80368
202	0.020512	1.5652	0.8062	**9365**	0.021589	1.5561	0.80481
203	0.019524	1.5835	0.80788	**9367**	0.021985	1.5698	0.80606
341	0.018729	1.5775	0.80831	**9460**	0.021676	1.5651	0.80613
342	0.021313	1.5755	0.80722	**11314**	0.042569	-1.3391	0.26414
1977	0.030364	1.1679	0.73405	**11729**	0.020763	1.5799	0.80779
2207	0.017137	1.575	0.80782	**12063**	0.018729	1.6016	0.81121
2281	0.021125	1.5648	0.80571	**12132**	0.020844	1.5765	0.80667
2282	0.019909	1.5718	0.80696	**12133**	0.022102	1.5849	0.80784
2284	0.019187	1.5941	0.80958	**12134**	0.022448	1.5949	0.80879
2533	0.019629	1.5804	0.80783	**14410**	0.018084	1.586	0.80935
2583	0.020265	1.5788	0.80728	**15365**	0.02094	1.5876	0.80867
2848	0.019524	1.5796	0.80779	**15756**	0.022784	1.5557	0.80385
3551	0.019187	1.5844	0.80836	**16004**	0.019994	1.582	0.808
4246	0.016703	1.5812	0.8083	**17038**	0.020597	1.5688	0.80587
4660	0.022784	1.5649	0.80556	**17200**	0.019909	1.5895	0.80911
4684	0.034407	1.1631	0.73284	**18466**	0.020512	1.5734	0.80677
5084	0.021589	1.5791	0.80688	**18710**	0.046627	0.96501	0.70473
5103	0.017704	1.5563	0.80528	**19026**	0.020512	1.5827	0.80809
5235	0.021873	1.5678	0.80581	**19027**	0.018729	1.577	0.80741
6052	0.022913	1.5763	0.80647	**20604**	0.038382	-1.3409	0.26127
6059	0.021216	1.5772	0.80706	**20687**	0.021125	1.5857	0.80799
6625	0.018984	1.5935	0.80982	**21510**	0.019909	1.5719	0.80693
6987	0.026205	-1.4705	0.23931	**21758**	0.02008	1.5453	0.8031
6994	0.019296	1.5762	0.80765	**22001**	0.022218	1.5608	0.80469
8653	0.016603	1.5771	0.8082	**22337**	0.020763	1.5798	0.80777

Tabela 4 – SNPs identificados como *outliers* na análise no BayeScan da subamostragem (38 indivíduos de quatro localidades, 28.743 SNPs) considerando os grupos (populações) do resultado do sNMF de K= 3 e K= 4. SNP: identificação do SNP; q-valor: valores de q-value, alpha: valor de alfa e F_{ST}: valor de F_{ST}.

	K=3			K=4		
SNP	**qval**	**alpha**	**F_{ST}**	**qval**	**alpha**	**F_{ST}**
2	-	-	-	0.016714	1.3674	0.71738
76	0.019644	1.5952	0.77139	0.0018819	1.8006	0.78468
173	0.021087	1.5597	0.76624	0.0016281	1.7971	0.78461
306	-	-	-	0.025036	1.3282	0.70989
308	-	-	-	0.015065	1.372	0.71821
748	-	-	-	0.047122	1.103	0.66593
917	-	-	-	0.013905	1.3706	0.71808
1059	-	-	-	0.014782	1.3752	0.71863
1147	-	-	-	0.013905	1.3748	0.71844
1327	-	-	-	0.032957	1.1892	0.68388
1355	0.02003	1.5756	0.76851	0.00093352	1.8141	0.7863
1356	0.02003	1.573	0.7681	0.0015217	1.8116	0.78674
1502	-	-	-	0.037009	1.1822	0.68189
1503	0.019774	1.5832	0.77021	0.0020198	1.7947	0.78404
1504	-	-	-	0.041248	1.1627	0.67844
1536	-	-	-	0.017736	1.3708	0.71809
1537	-	-	-	0.017736	1.3763	0.71887
1657	-	-	-	0.02305	1.3584	0.71582
1658	-	-	-	0.021388	1.3595	0.71636
2478	0.036302	0.8644	0.63875	0.0029979	1.4218	0.72739
2540	0.020543	1.5758	0.76842	0.0018819	1.7944	0.78358
2541	0.021487	1.5911	0.77079	0.0016281	1.8024	0.78538
2542	0.020204	1.5972	0.77204	0.0015217	1.8089	0.78644
2768	-	-	-	0.019178	1.3619	0.71623
2892	0.02184	1.5611	0.76648	0.0015217	1.8109	0.78641
2952	0.018004	1.5795	0.77	0.0013669	1.7994	0.78481
2997	-	-	-	0.0095592	1.3752	0.7188
3293	0.018658	1.5784	0.76934	0.0012184	1.8105	0.78622
3338	0.020467	1.5871	0.77027	0.0015217	1.8115	0.78644
3339	0.020846	1.5696	0.76744	0.00093352	1.8006	0.78459
3672	0.018964	1.5765	0.76918	0.0020974	1.7784	0.78128
3877	-	-	-	0.011682	1.3625	0.71733
3878	-	-	-	0.0084203	1.3765	0.71906
3879	-	-	-	0.014782	1.3686	0.7171
3914	0.02003	1.5675	0.76752	0.00080016	1.8105	0.78672
3915	0.020204	1.5797	0.76911	0.0015217	1.7932	0.78349
3916	0.020393	1.5694	0.76783	0.0017213	1.8076	0.78615
3917	0.017537	1.5874	0.77061	0.0012184	1.8042	0.78559

continuação da Tabela 4

	K=3			K=4		
SNP	**qval**	**alpha**	**F_{ST}**	**qval**	**alpha**	**F_{ST}**
4067	-	-	-	0.024491	1.3261	0.71032
4109	0.019098	1.5632	0.76686	0.0017213	1.7856	0.78299
4161	-	-	-	0.0080275	1.384	0.72006
4200	0.019774	1.5697	0.76747	0.0018819	1.8072	0.7857
4237	0.02198	1.5577	0.76558	0.0016281	1.7986	0.78477
4241	-	-	-	0.0376	1.192	0.68371
4524	-	-	-	0.026392	1.2927	0.70358
4979	-	-	-	0.043196	1.1565	0.67625
5024	-	-	-	0.031438	1.2612	0.69685
5029	-	-	-	0.020523	1.376	0.71915
5201	0.022357	1.5664	0.7666	0.0018819	1.7819	0.78253
5256	-	-	-	0.0088064	1.3906	0.72119
5314	0.022724	1.5399	0.76253	0.0021946	1.7661	0.77973
5857	0.045309	0.8061	0.62717	0.0033601	1.4143	0.72595
5858	-	-	-	0.0063741	1.3777	0.7198
6848	-	-	-	0.012666	1.3971	0.7223
7020	0.01984	1.5894	0.77041	0.0013669	1.7976	0.78472
7739	-	-	-	0.02008	1.3576	0.71575
8059	0.018964	1.597	0.77168	0.0016281	1.8077	0.78619
8097	-	-	-	0.0455	1.1289	0.66957
8112	0.019644	1.576	0.76853	0.001792	1.7912	0.78391
8265	0.021772	1.5678	0.76719	0.0019147	1.8066	0.78581
8512	-	-	-	0.022843	1.3681	0.71743
8649	-	-	-	0.015908	1.376	0.71906
8651	-	-	-	0.012982	1.3671	0.7172
8785	-	-	-	0.031911	1.2387	0.69331
8786	-	-	-	0.028822	1.2759	0.69999
8854	0.018493	1.5855	0.77055	0.0017213	1.8207	0.78801
8994	-	-	-	0.019178	1.368	0.71776
8996	-	-	-	0.005515	1.386	0.72071
9001	-	-	-	0.016714	1.3622	0.71662
9002	-	-	-	0.0046373	1.3914	0.72186
9020	0.022204	1.5525	0.76467	0.0020198	1.782	0.78205
9043	-	-	-	0.027166	1.2978	0.70349
9182	-	-	-	0.011682	1.3721	0.71803
9311	-	-	-	0.023728	1.344	0.71299
9397	-	-	-	0.04796	1.1003	0.66507
9706	-	-	-	0.042516	1.1797	0.68083
9952	0.020619	1.5636	0.76638	0.0018819	1.817	0.7874
10043	0.019029	1.5871	0.77042	0.0013669	1.8047	0.7859
10214	-	-	-	0.046309	1.1139	0.66769
10215	-	-	-	0.048792	1.1014	0.66516
10258	0.020326	1.5829	0.76983	0.0020198	1.8018	0.78533

continuação da Tabela 4

	K=3			K=4		
SNP	**qval**	**alpha**	**F_{ST}**	**qval**	**alpha**	**F_{ST}**
10397	-	-	-	0.024753	1.3417	0.71238
10398	-	-	-	0.023974	1.3422	0.71311
10401	-	-	-	0.02423	1.3533	0.71503
10431	-	-	-	0.016714	1.3723	0.71812
10432	-	-	-	0.0067989	1.3749	0.71885
10518	0.023045	1.5337	0.76141	0.0021534	1.7358	0.77562
10580	-	-	-	0.026773	1.291	0.70291
10991	0.022878	1.5524	0.76432	0.0020198	1.7553	0.77767
11240	0.020326	1.5619	0.76678	0.0013669	1.7986	0.7848
11307	0.017103	1.6043	0.77308	0.0017213	1.8057	0.7855
11371	0.019314	1.5909	0.77147	0.0010669	1.7867	0.78288
11731	-	-	-	0.017736	1.3514	0.71427
11961	-	-	-	0.030999	1.2476	0.69483
11978	-	-	-	0.023263	1.3452	0.7134
12580	-	-	-	0.044692	1.1284	0.67075
12619	-	-	-	0.0091859	1.377	0.71944
12821	-	-	-	0.033511	1.2281	0.69037
12974	0.016203	1.5819	0.76952	0.0015217	1.804	0.78526
13215	-	-	-	0.020956	1.3671	0.71724
13216	-	-	-	0.0059462	1.3745	0.7184
13633	-	-	-	0.035248	1.2084	0.68623
13758	-	-	-	0.02564	1.3383	0.71135
13779	-	-	-	0.032399	1.1491	0.67876
14023	0.019385	1.5869	0.77087	0.0017213	1.8021	0.78517
14144	0.022439	1.556	0.76532	0.0021198	1.7729	0.78054
14160	-	-	-	0.014782	1.3642	0.71727
14162	-	-	-	0.02008	1.3744	0.71847
14515	-	-	-	0.040019	1.1889	0.68258
15208	-	-	-	0.0080275	1.3591	0.7163
15322	-	-	-	0.015908	1.3686	0.71745
15344	0.032069	0.85796	0.63773	0.0037287	1.4152	0.72624
15462	-	-	-	0.022431	1.3714	0.7177
15490	-	-	-	0.004185	1.3864	0.72077
15491	-	-	-	0.021812	1.35	0.71443
15492	-	-	-	0.010292	1.37	0.71792
15524	0.022521	1.5776	0.76826	0.001792	1.7997	0.78482
15732	0.019171	1.5761	0.7689	0.0015217	1.7912	0.78386
15733	-	-	-	0.04188	1.1613	0.6777
15784	-	-	-	0.021812	1.3587	0.7161
16178	0.021772	1.5696	0.76732	0.0016281	1.8038	0.78533
17070	0.021487	1.571	0.76761	0.0017213	1.8047	0.78512
17262	-	-	-	0.027975	1.2808	0.70069
17397	-	-	-	0.0050805	1.3826	0.72042

continuação da Tabela 4

	K=3			K=4		
SNP	**qval**	**alpha**	**F_{ST}**	**qval**	**alpha**	**F_{ST}**
17484	-	-	-	0.012666	1.3639	0.71692
17485	-	-	-	0.012666	1.3619	0.7168
17530	0.022052	1.57	0.76755	0.0020567	1.7729	0.78045
17934	0.021772	1.5641	0.76622	0.0010669	1.8028	0.78511
18199	0.018289	1.5873	0.77067	0.0013669	1.7859	0.78295
18258	-	-	-	0.041248	1.1856	0.6819
18292	-	-	-	0.034656	1.2217	0.68861
18403	0.02117	1.5886	0.77022	0.0012184	1.8066	0.78624
18405	0.020771	1.5757	0.76831	0.0015217	1.7933	0.78366
18431	-	-	-	0.018226	1.3766	0.71853
18744	0.020771	1.5975	0.7716	0.0020198	1.7844	0.78268
18929	0.021772	1.5671	0.7673	0.001792	1.7999	0.78471
19187	0.021487	1.5667	0.76741	0.0015217	1.7921	0.78405
19459	-	-	-	0.038191	1.178	0.68106
19753	-	-	-	0.015349	1.3611	0.71596
19756	-	-	-	0.022227	1.3589	0.71546
19845	-	-	-	0.0080275	1.3715	0.71881
20084	0.023342	1.523	0.75972	0.002316	1.7335	0.77441
20085	0.023653	1.5174	0.75854	0.0022462	1.7452	0.77646
20162	0.018964	1.5817	0.76978	0.0020974	1.7957	0.78455
20219	-	-	-	0.035839	1.167	0.67974
20557	-	-	-	0.018226	1.3749	0.71865
20583	-	-	-	0.036425	1.1772	0.68135
20945	-	-	-	0.022638	1.3619	0.71662
21010	0.040829	0.81769	0.62955	0.0026339	1.415	0.72597
21090	-	-	-	0.023481	1.3643	0.71647
21093	-	-	-	0.011	1.3777	0.71921
21470	-	-	-	0.018707	1.3641	0.71668
21471	-	-	-	0.020523	1.3493	0.71434
21472	-	-	-	0.02008	1.3725	0.71823
21473	-	-	-	0.020741	1.3583	0.7157
21909	0.018289	1.5823	0.77013	0.0010669	1.8197	0.78779
22249	-	-	-	0.018468	1.3603	0.71617
22298	-	-	-	0.028392	1.2918	0.70241
22341	-	-	-	0.013295	1.3717	0.71832
22566	-	-	-	0.034068	1.2242	0.68931
22704	0.021004	1.5679	0.76715	0.0018819	1.7981	0.78449
23206	0.018493	1.5772	0.7696	0.0019147	1.7941	0.78434
23310	-	-	-	0.030119	1.2728	0.69916
23311	-	-	-	0.029685	1.2824	0.70082
23312	-	-	-	0.029685	1.2732	0.6992
23313	-	-	-	0.030555	1.2712	0.69857
23373	0.021004	1.5765	0.76837	0.0013669	1.795	0.78424

continuação da Tabela 4

	K=3			K=4		
SNP	**qval**	**alpha**	**F_{ST}**	**qval**	**alpha**	**F_{ST}**
23397	0.027803	-0.94971	0.28276	-	-	-
23733	-	-	-	0.04394	1.1617	0.67656
23859	-	-	-	0.026013	1.2913	0.70297
23985	-	-	-	0.011	1.3769	0.7191
24390	0.017804	1.5869	0.76998	0.0016281	1.8082	0.78591
24496	-	-	-	0.049637	1.0936	0.66356
24649	0.022357	1.5714	0.7676	0.0016281	1.7887	0.7835
24834	-	-	-	0.0099287	1.3702	0.71803
25297	0.022614	1.5633	0.76652	0.001792	1.8034	0.78563
25532	-	-	-	0.02008	1.3634	0.71654
25533	-	-	-	0.016975	1.3736	0.71829
26085	0.018658	1.5719	0.76827	0.0020198	1.7955	0.78393
26312	0.019314	1.5851	0.77002	0.0012184	1.808	0.78599
26313	0.022128	1.5639	0.76675	0.001792	1.8049	0.78533
26582	-	-	-	0.038778	1.1734	0.68079
26708	0.019644	1.5653	0.76716	0.0017213	1.8019	0.78511
27051	-	-	-	0.025319	1.3287	0.71001
27275	0.019644	1.5936	0.77142	0.0016281	1.8037	0.78507
27277	0.02191	1.5762	0.76834	0.0013669	1.7884	0.78321
27278	0.018964	1.5883	0.77111	0.0012184	1.815	0.78624
27446	-	-	-	0.027572	1.282	0.70087
27482	-	-	-	0.021388	1.3671	0.71729
27622	0.021487	1.5592	0.76585	0.0020567	1.771	0.78082
28250	-	-	-	0.039398	1.1627	0.6784
28507	-	-	-	0.022227	1.3594	0.71576
28542	0.020204	1.5931	0.77132	0.0015217	1.7877	0.78326

O Treemix permite analisar se há evidências de eventos de migração entre pelo menos três populações. Considerando como terminais os grupos resultantes de K= 3 para a amostragem total (Figura 11) e para a subamostragem (Figura 12) não foi detectado sinal de migração e/ou fluxo gênico entre as populações, mesmo permitindo considerar até quatro eventos de migração. Já adotando os agrupamentos resultantes de K= 4, seja com a amostragem total (Figura 13) ou com a subamostragem (Figura 14), foi detectado sinal para apenas um evento de migração, mesmo permitindo considerar mais eventos nas análises. O resultado com a amostragem total (Figura 13) indicou que um evento de migração ocorreu de *P. atra* para a localidade de *P. leucoptera* mais próxima (localidade 4) e isso representou uma contribuição de 39% de *P. atra* para *P. leucoptera*. Já o resultado com a subamostragem (Figura 14) indicou que o evento migração foi da população de *P. atra* (localidade 1) para a zona de hibridação

(localidade 3) com contribuição de 10%. Ao considerar cada uma das localidades como uma população o número de eventos de migração detectados corresponderam ao número testado (Figura 15). O mais significativo foi assumindo quatro eventos de migração (m=4). Nesse cenário foi detectado um evento de migração de *P. leucoptera* em direção a *P. atra* (localidade 4 para a 1) com contribuição de 1%, outro de *P. atra* para *P. leucoptera* (localidade 1 para a 6) com contribuição de 18% e dois eventos, sendo de via dupla, entre a localidade 3 (zona de hibridação) e a localidade 5 de *P. leucoptera*, sendo a contribuição da zona de hibridação para a localidade 5 de 14% e a outra via, de 8% de contribuição. O Teste de 3 populações (Tabela 5) não resultou em valores negativos, o que indicou ausência de evidência de migração e foi congruente com as topologias do Treemix. Enquanto o Teste de 4 populações (Tabela 5) indicou que na amostragem total há valores negativos indicando presença de migração em dois agrupamentos ((le,at;zh,lc) e (le,zh;at,lc)) e na subamostragem, há indicativo de migração para o agrupamento (1,4;3,2).

Tabela 5 – Resultados do Teste 3 populações (f3) e do Teste 4 populações (f4) baseados na amostragem total e subamostragem. Agrupamento conforme siglas utilizadas nas árvores do TreeMix, estatística f para cada agrupamento e seu respectivo desvio padrão e valor de Z-score.

Amostragem total K=3			
Agrupamento	estatística f4	Desvio padrão	Z-score
at;le,zh	0.00706871	0.000271905	25.997
le;zh,at	0.013967	0.000902449	15.4768
zh;le,at	0.00251314	0.000203712	12.3367
Amostragem total K=4			
Agrupamento	estatística f4	Desvio padrão	Z-score
le,at;zh,lc	-0.0101343	0.000891634	-11.366
le,lc;zh,at	0.000935798	0.000106808	8.76152
le,zh;at,lc	-0.0110701	0.000902604	-12.2646
Subamostragem K=3			
Agrupamento	estatística f3	Desvio padrão	Z-score
1;3,2	0.0101091	0.000276799	36.5214
2;1,3	0.00423849	0.000222726	19.0301
3;1,2	0.0179998	0.000770377	23.365
Subamostragem K=4			
Agrupamento	estatística f4	Desvio padrão	Z-score
1,2;3,4	0.000448459	0.000213028	2.10516
1,3;2,4	0.0023032	0.000242335	9.50423
1,4;3,2	-0.00185475	0.000242421	-7.65094

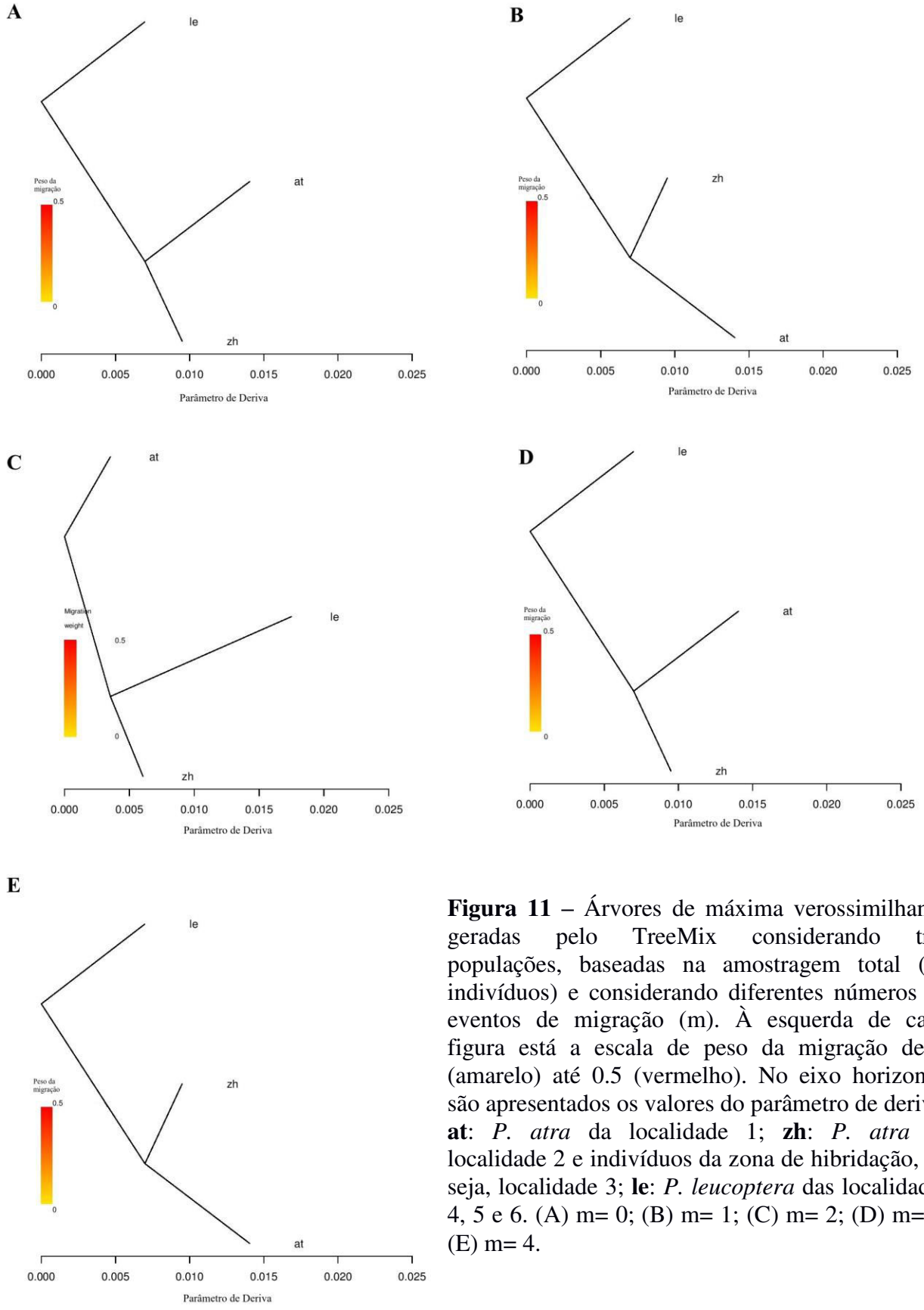


Figura 11 – Árvores de máxima verossimilhança geradas pelo TreeMix considerando três populações, baseadas na amostragem total (48 indivíduos) e considerando diferentes números de eventos de migração (m). À esquerda de cada figura está a escala de peso da migração de 0 (amarelo) até 0.5 (vermelho). No eixo horizontal são apresentados os valores do parâmetro de deriva. **at**: *P. atra* da localidade 1; **zh**: *P. atra* da localidade 2 e indivíduos da zona de hibridação, ou seja, localidade 3; **le**: *P. leucoptera* das localidades 4, 5 e 6. (A) m= 0; (B) m= 1; (C) m= 2; (D) m= 3; (E) m= 4.

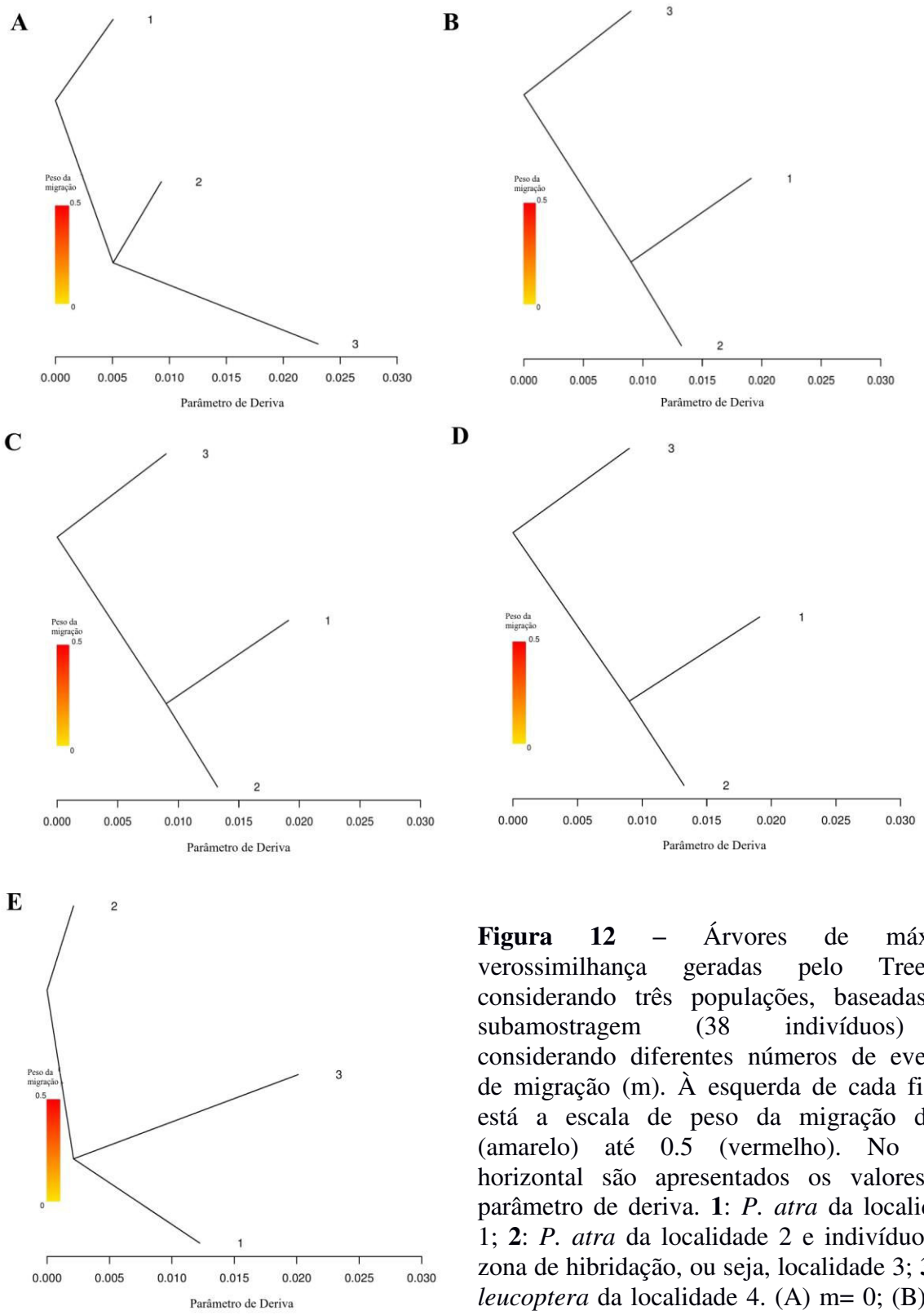


Figura 12 – Árvores de máxima verossimilhança geradas pelo TreeMix considerando três populações, baseadas na subamostragem (38 indivíduos) e considerando diferentes números de eventos de migração (m). À esquerda de cada figura está a escala de peso da migração de 0 (amarelo) até 0.5 (vermelho). No eixo horizontal são apresentados os valores do parâmetro de deriva. **1**: *P. atra* da localidade 1; **2**: *P. atra* da localidade 2 e indivíduos da zona de hibridação, ou seja, localidade 3; **3**: *P. leucoptera* da localidade 4. (A) m= 0; (B) m= 1; (C) m= 2; (D) m= 3; (E) m= 4.

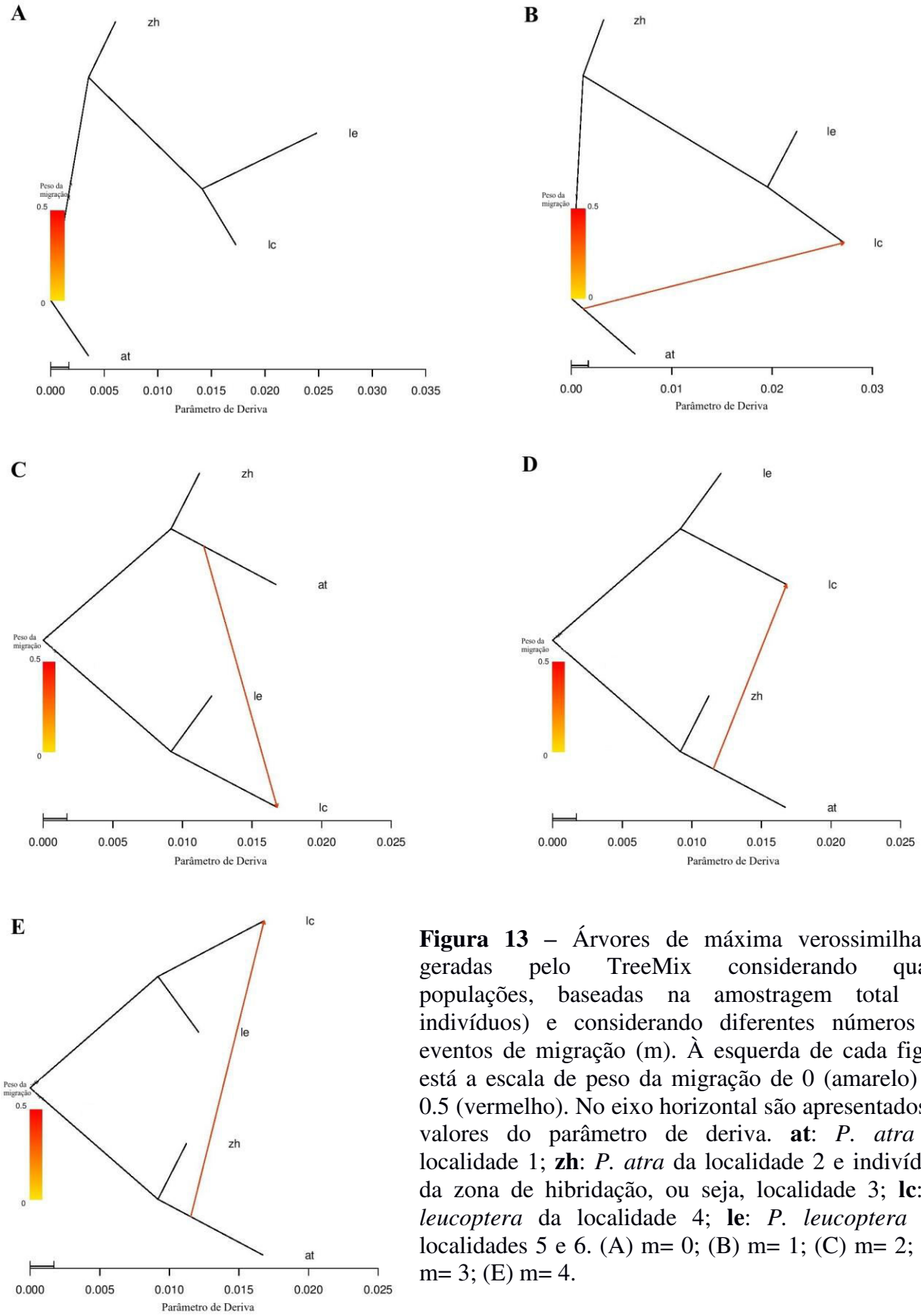
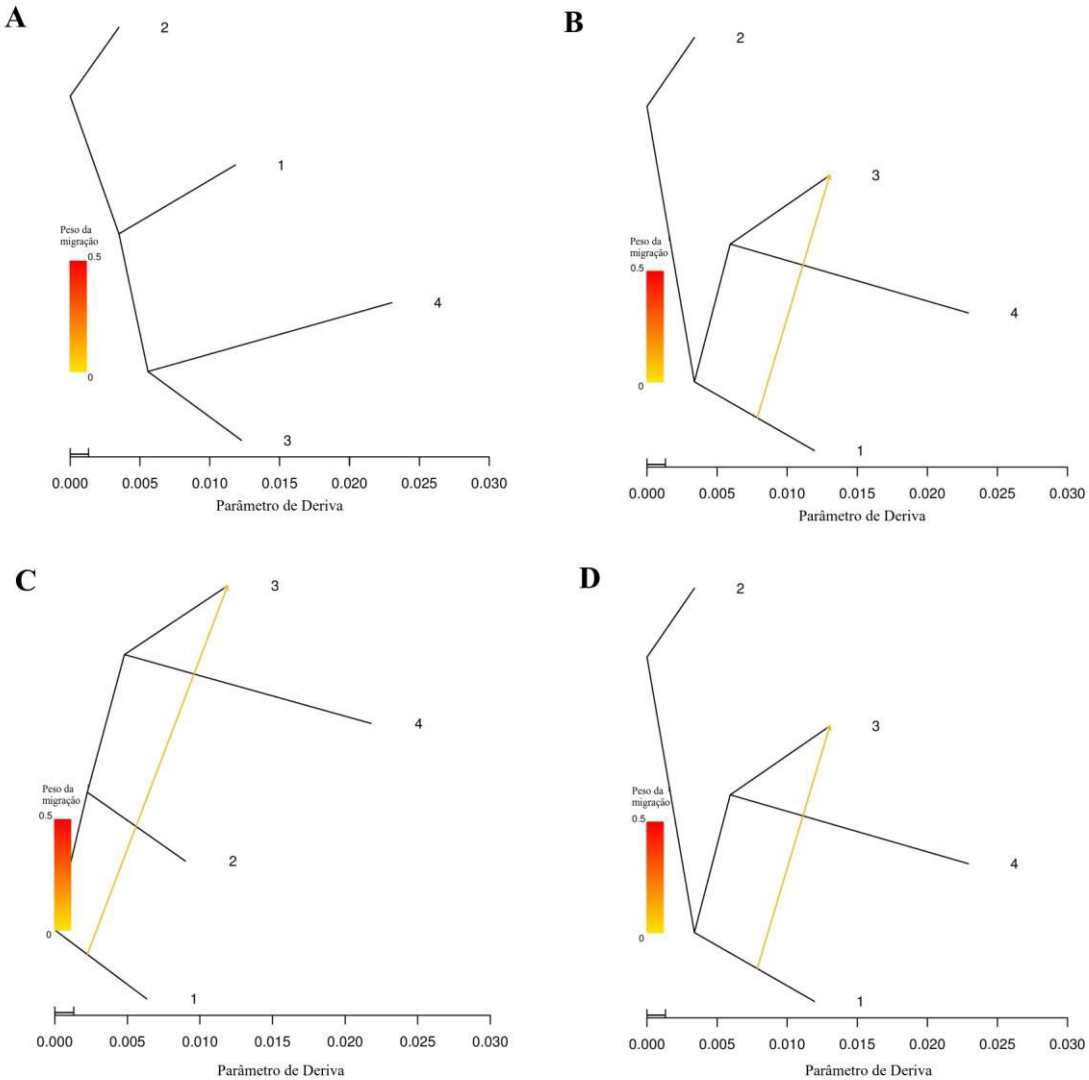


Figura 13 – Árvores de máxima verossimilhança geradas pelo TreeMix considerando quatro populações, baseadas na amostragem total (48 indivíduos) e considerando diferentes números de eventos de migração (m). À esquerda de cada figura está a escala de peso da migração de 0 (amarelo) até 0.5 (vermelho). No eixo horizontal são apresentados os valores do parâmetro de deriva. **at**: *P. atra* da localidade 1; **zh**: *P. atra* da localidade 2 e indivíduos da zona de hibridação, ou seja, localidade 3; **lc**: *P. leucoptera* da localidade 4; **le**: *P. leucoptera* das localidades 5 e 6. (A) m= 0; (B) m= 1; (C) m= 2; (D) m= 3; (E) m= 4.



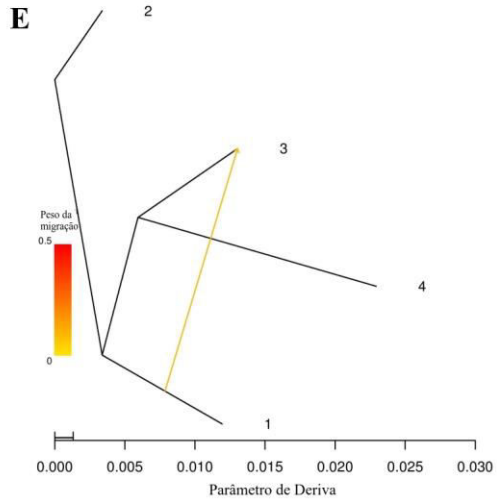


Figura 14 – Árvores de máxima verossimilhança geradas pelo TreeMix considerando quatro populações, baseadas na subamostragem (38 indivíduos) e considerando diferentes números de eventos de migração (m). À esquerda de cada figura está a escala de peso da migração de 0 (amarelo) até 0.5 (vermelho). No eixo horizontal são apresentados os valores do parâmetro de deriva. **1**: *P. atra* da localidade 1; **2***: P. atra* da localidade 2 e indivíduos da zona de hibridação, ou seja, localidade 3; **3**: *P. leucoptera* da localidade 4. (A) m= 0; (B) m= 1; (C) m= 2; (D) m= 3; (E) m= 4.

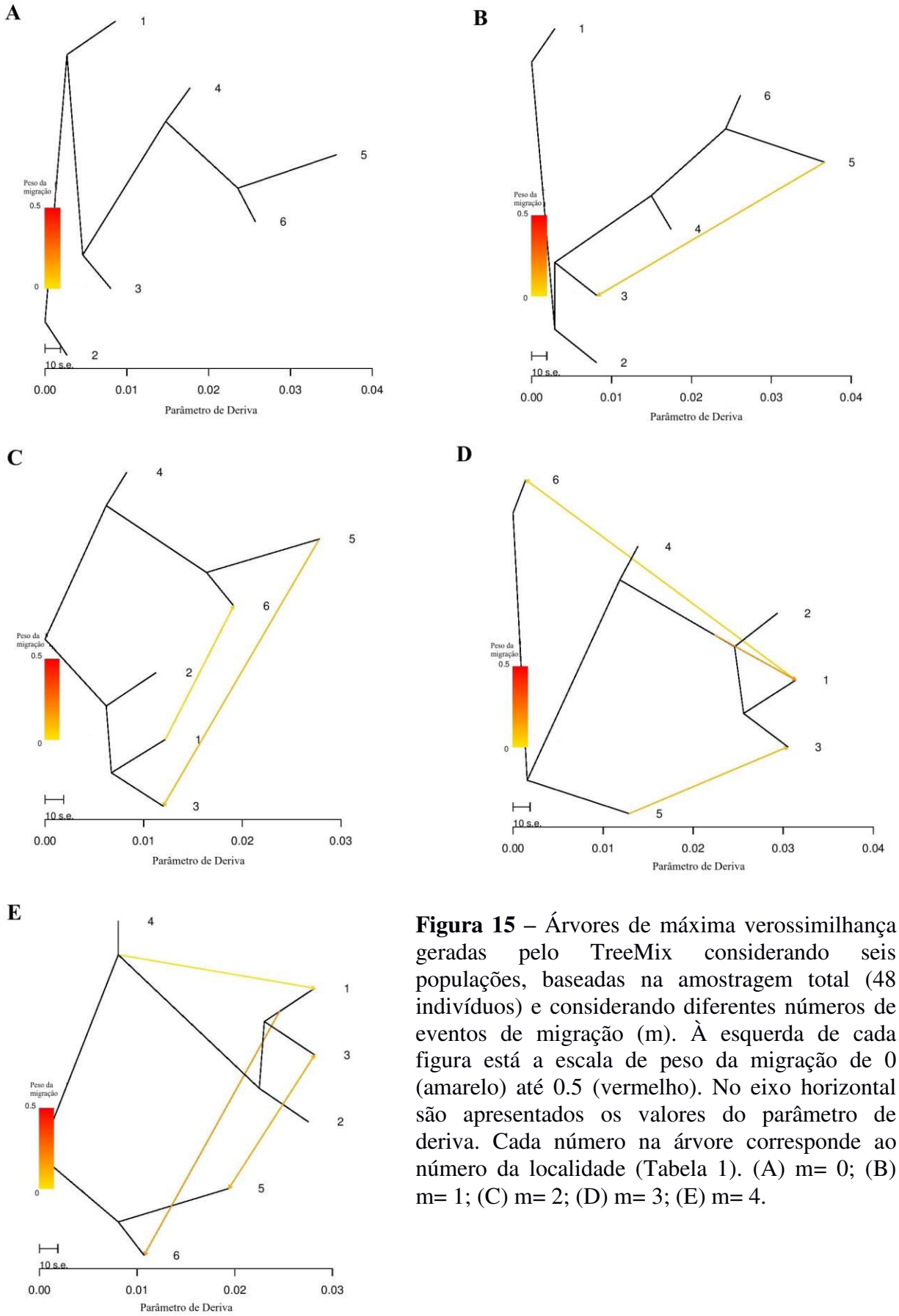


Figura 15 – Árvores de máxima verossimilhança geradas pelo TreeMix considerando seis populações, baseadas na amostragem total (48 indivíduos) e considerando diferentes números de eventos de migração (m). À esquerda de cada figura está a escala de peso da migração de 0 (amarelo) até 0.5 (vermelho). No eixo horizontal são apresentados os valores do parâmetro de deriva. Cada número na árvore corresponde ao número da localidade (Tabela 1). (A) m= 0; (B) m= 1; (C) m= 2; (D) m= 3; (E) m= 4.

5. Discussão

5.1 Estrutura populacional do complexo *P. atra*/ *P. leucoptera*/ intermediários morfológicos

Os resultados de SAMOVA de um estudo utilizando sequência de apenas um gene mitocondrial de 442 indivíduos de *P. leucoptera* e de *P. atra* (Maldonado-Coelho, 2010) mostraram aumento no valor de F_{CT} acompanhando o aumento de número de populações (K) e, portanto, não foi possível identificar grupos genéticos distintos. Esse autor sugeriu que essa ausência de estrutura populacional poderia ser resultado do pequeno volume de dados (um gene) ou a presença de isolamento por distância. Os resultados da presente análise de estrutura genética populacional obtidos com mais de 20 mil SNPS no sNMF e avaliando de dois a quatro grupos (K; Figuras 9 e 10) indicaram que os grupos aqui obtidos possuem forte congruência com as espécies envolvidas e/ou localidades de amostragem. Isso pode também ser observado nos resultados dos valores de entropia cruzada que apresentaram grande sobreposição para os diversos valores de K testados (Figuras 7 e 8). Assim, mesmo utilizando mais de vinte mil SNPs o padrão populacional obtido parece poder ser resultado de isolamento por distância.

Como a diversificação do gênero *Pyriglena* parece ter se iniciado no começo do Pleistoceno (por volta de 2,5 milhões de anos) e parece ter sido rápida (Maldonado-Coelho, 2010), a diversificação entre *P. leucoptera* e *P. atra* seria ainda mais recente. Assim, foi sugerido que o tempo relativamente curto não teria sido suficiente para a molécula de DNA mitocondrial ter acumulado sinal filogenético capaz de resolver o relacionamento entre os clados de *Pyriglena* (Maldonado-Coelho, 2010). Apesar de que seria ainda menos esperado que o DNA nuclear (que evolui com taxa menor do que o genoma mitocondrial) possuísse sinal suficiente para definir as espécies parentais, os mais de 20 mil SNPs utilizados no presente estudo distinguiram esses dois táxons. No entanto, a associação de genótipo e fenótipo não foi clara nos seis indivíduos intermediários morfológicos (todos machos, pois as fêmeas das duas espécies não são distinguíveis morfologicamente). Esses indivíduos podem não ter acumulado mudanças genéticas suficientes para serem distinguíveis dos parentais e ainda, quatro deles não possuem evidência de possuírem mistura genética, com exceção dos indivíduos 17931 e 17934 (Figuras 9 e 10; intermediários morfológicos indicados com asterisco). Esses resultados indicam que houve um desacoplamento entre a diferenciação morfológica e a diferenciação genética, ou seja, intermediários morfológicos perfeitos (com

caracteres morfológicos diagnósticos das duas espécies) não apresentaram mistura genética em níveis elevados, ou seja, no presente caso a evolução morfológica e a molecular não foram congruentes. Essa diferença pode ser resultante do fato de a grande maioria dos SNPs analisados aparentemente não apresentarem sinal de seleção (ver Tabela 3), enquanto a morfologia pode estar sob alguma forma de seleção, talvez sexual.

Indivíduos das duas localidades adjacentes (2 e 4) à localidade 3 (zona de hibridação) apresentaram valores maiores de mistura genética (Figuras 9 e 10). Ainda, nota-se que a porcentagem de mistura em indivíduos da localidade 4 em relação aos da localidade 2 foi mais que o dobro. Isso sugere que alelos de *P. atra* e dos intermediários morfológicos foram transferidos para *P. leucoptera* em maior proporção do que o contrário. Ainda, a diferença na proporção de mistura de cada agrupamento se torna mais evidente quando são comparados indivíduos das localidades 1 e 4 (Figuras 9 e 10). Ambas estão à mesma distância (aproximadamente 285 km) da zona de hibridação (localidade 3) e apresentam muita diferença nas porcentagens de contribuição genética de outros agrupamentos. Enquanto na localidade 4 (*P. leucoptera*) por volta de 7 a 15% dos alelos vieram de outra população, os indivíduos da localidade 1 (*P. atra*) receberam ao redor de 3 a 5%. Essa diferença de contribuição genética de outra origem foi de praticamente o triplo.

O local de contato das distribuições geográficas das espécies parentais coincide com o Rio Paraguaçu, com *P. atra* na margem ao norte e *P. leucoptera*, ao sul. A hipótese que parece mais aceitável de como esse contato ocorreu foi a de uma expansão de *P. leucoptera* acompanhando o curso do rio e encontrando *P. atra* na margem norte (Maldonado-Coelho & Sampaio, comunicação pessoal). Essa expansão geográfica de *P. leucoptera* junto com indícios de maior mistura em *P. leucoptera* e sinal de introgressão de *P. atra* para *P. leucoptera* poderiam ser explicadas pelo mecanismo de *genetic surfing* (Edmonds *et al.*, 2004; Klopfstein *et al.*, 2006). Segundo esse mecanismo, como a frente de expansão é constituída de um tamanho populacional pequeno, quando ela encontra uma espécie proximamente relacionada estabelecida ocorre introgressão assimétrica da espécie local para a espécie em expansão (Currat & Excoffier, 2004).

5.2 Hibridação e Seleção

As aves suboscines (subordem Tyranni, Passeriformes), caso das espécies analisadas no presente estudo, parecem não possuírem aprendizado vocal e em geral, apresentam raras ocorrências de hibridação (Graves, 1992; Cadena *et al.,* 2007). Acredita-se que a divergência

acústica com isolamento geográfico e uma consequente diferenciação genética podem explicar o isolamento pré-zigótico neste grupo (Seddon & Tobias, 2007). Entretanto, em uma região de contato entre *P. leucoptera* e *P. atra* próximo ao Rio Paraguaçu (BA) e, mais precisamente, na cidade de Ipirá (BA) foram encontrados e amostrados indivíduos intermediários morfológicos (Maldonado-Coelho & Sampaio, comunicação pessoal), Ou seja, esse é um relativamente raro exemplo de zona de hibridação de suboscines. No entanto, evidências de hibridação e introgressão morfológica entre essas duas espécies tem sido descritas desde a década de 1980, quando indivíduos machos depositados em coleções e identificados como *P. leucoptera* foram descritos como possuindo morfologia intermediária entre *P. leucoptera* e *P. atra* (Willis & Oniki, 1982). Além disso, dois tipos de vocalização, o canto e o chamado, que são significativamente diferentes entre as espécies *P. leucoptera* e *P. atra,* apresentaram uma versão intermediária em dois indivíduos coletados no estado da Bahia, indicando que a hibridação está ocorrendo naquela região (Isler & Maldonado-Coelho, 2017).

Apesar de possuírem diferenças significativas e diagnósticas na plumagem de machos e em vocalizações, não há monofilia recíproca entre as espécies parentais baseado em dados de locos nucleares e mitocondriais (Maldonado-Coelho, comunicação pessoal). Isso também foi observado na comparação entre as árvores de máxima verossimilhança baseadas em mais de 20 mil SNPs e geradas pelo TreeMix (Figuras 11 a 15). Em alguns casos a variação na quantidade de eventos de migração resultou em topologias diferentes (principalmente Figura 15), o que parece indicar que cada nova simulação resulta em uma nova topologia. Assim, como não parece haver um sinal forte evidenciando apenas um dos resultados, vamos discutir apenas alguns padrões recorrentes e realizar algumas inferências sobre fluxo gênico nesse complexo. Quando foi considerada a amostragem com as seis localidades divididas em quatro grupos (Figura 13) foi sugerido que existe uma contribuição de 39% de *P. atra* na composição genética de *P. leucoptera* da localidade próxima à zona de hibridação. Ou seja, pode-se inferir que a direção da introgressão é de *P. atra* para *P. leucoptera,* assim como foi indicado pelos resultados do sNMF (Figuras 9 e 10). A árvore considerando cada localidade como um grupo distinto (Figura 15) sugere que, assim como na rede de haplótipos mitocondriais de Maldonado-Coelho (2012) existe contribuição e compartilhamento de alelos entre indivíduos do nordeste e do sul do Brasil, assim como do sul com a região central. Ou seja, em geral, os resultados obtidos no presente trabalho são congruentes com resultados previamente obtidos.

Um processo que pode ser bastante importante em casos de hibridação é a seleção natural, pois a hibridação pode gerar rápidas mudanças evolutivas e as consequências

ecológicas, fenotípicas e genéticas desses eventos de hibridação têm sido o foco de alguns estudos (Grant & Grant, 1992; Seehausen, 2004; Baack & Rieseberg, 2007). Assim, zonas de hibridação são conhecidas principalmente pela possibilidade de apresentar uma maior variabilidade de fenótipos e genótipos, uma vez que muitas vezes incluem indivíduos híbridos e retrocruzados com novas combinações ou extremos em relação a populações parentais puras (Parsons *et al.,* 1993; Grant & Grant, 1994; Rieseberg *et al.*, 1999; Chiba, 2005). Em híbridos, os traços quantitativos podem aumentar consideravelmente seu efeito por conta do potencial aumento de heterozigosidade e da introgressão de alelos, ou seja, as variabilidades genética e fenotípica podem aumentar (Barton & Gale, 1993).

Assim, no presente estudo avaliamos se os marcadores utilizados possuíam sinal de estar sob seleção. Os resultados indicaram um baixo número de SNPs com sinal de estar sob seleção (menos de 1% do total de SNPs obtidos; Tabelas 2 e 3). Como comparação, em um estudo intraespecífico de 69 indivíduos de duas espécies de aves, *Ammodramus caudacutus* e *A. nelsoni*, foram encontrados 5% de um total de 1929 SNPs como possivelmente sob seleção (Walsh *et al.,* 2017). É possível que a pequena porcentagem de SNPs *outliers* observada no presente estudo esteja relacionada com a proximidade genética dos parentais envolvidos. Como alguns desses parentais compartilham variantes genéticas em relativamente alta quantidade (ex. localidade 4 de *P. leucoptera*, Figuras 9A e 10A; localidade 2 de *P. atra*, Figuras 9B e 10B), os híbridos teriam composição genética pouco diferenciada daquela de alguns indivíduos das espécies parentais. Além disso, a zona de hibridação parece ser bastante estreita e em uma região de intensa fragmentação de habitat devido ao desmatamento (Maldonado-Coelho, 2010; Isler & Maldonado-Coelho, 2017; Maldonado-Coelho, comunicação pessoal). Assim, é possível que as populações remanescentes nessa região estejam sob forte efeito de deriva pelo isolamento. Adicionalmente, devido à grande perda de área da Mata Atlântica, especialmente na região próxima à essa zona de hibridação, é possível que populações que seriam importantes para reconstruir essa história já tenham sido perdidas.

6. Conclusões

1) As sequências obtidas utilizando a metodologia de *Genotyping by Sequencing* não continham dados de DNA mitocondrial e de cromossomos sexuais. Esses marcadores estão sendo analisados em um outro estudo do laboratório e a futura comparação dos resultados deverá permitir realizar uma análise mais abrangente.

2) A falta de sinal claro sobre o número de populações e sua estrutura baseado na análise de mais de 20 mil SNPs pode ser reflexo tanto de isolamento por distância, quanto da hibridação e introgressão detectadas, ou inclusive a combinação de ambos. Como a genômica de organismos não modelo é relativamente recente, esperamos que novas ferramentas de análise estejam disponíveis em um futuro próximo, permitindo testar nossos dados quanto ao isolamento por distância. Além disso, seria importante descrever melhor a história demográfica desses organismos.

3) A existência de um genoma montado e anotado de uma das espécies (*Pyriglena leucoptera* ou *P. atra*) ou de uma espécie próxima seria útil na montagem e comparação das *reads* e permitiria obter dados mais detalhados sobre os SNPs *outliers* (com sinal de estar sob seleção).

4) *P. leucoptera* e *P. atra* apresentam pouco compartilhamento de SNPs, inclusive alguns dos intermediários morfológicos não apresentaram mistura genética. É possível que isso seja reflexo de isolamento por distância. Como a área ao redor da zona de hibridação está muito degrada, é possível que parte da história desses táxons tenha sido perdida. No entanto, é possível que a adição de mais localidades amostradas possa ajudar a reconstruir essa história.

5) O baixo número de SNPs potencialmente com sinal de estarem sob seleção, pode ser resultado de termos analisado apenas dados nucleares autossômicos, possivelmente marcadores neutros. Assim, seria bastante importante ampliar o estudo com a adição de dados mitocondriais e de cromossomos sexuais que podem ser mais adequados para detectar sinal de seleção.

Resumo

A hibridação pode ter várias consequências, incluindo a manutenção ou aumento da diversidade resultando em zonas de hibridação estáveis; a introdução de nova variedade alélica que pode acarretar na origem ou transferência de adaptações, que pode ter efeito positivo no resgate de populações pequenas puras; o reforço do isolamento reprodutivo e a formação de novas linhagens híbridas. Além disso, a hibridação pode reduzir a diversidade, pois a quebra das barreiras reprodutivas e a mescla de populações anteriormente distintas podem acarretar na extinção de uma das populações ou espécies envolvidas na hibridação. Nesse cenário, nosso estudo teve como objetivo estudar uma zona de hibridação no estado da Bahia entre duas espécies de aves, *Pyriglena atra* (Swaison 1825) e *P. leucoptera* (Vieillot 1818). Foram analisados mais de 20.000 SNPs de 20 indivíduos de *P. leucoptera*, 22 de *P. atra* e seis indivíduos intermediários morfológicos. Encontramos pouca a nenhuma mistura genética nos intermediários morfológicos, o que indica um descompasso entre genótipo e fenótipo, o que poderia estar associado ao fato de os marcadores genéticos analisados terem se mostrado neutros, enquanto a morfologia externa pode estar sob pressão de seleção sexual. Foi detectado que as localidades adjacentes à zona de hibridação apresentaram mistura genética na maioria dos indivíduos, um indicativo de que além de hibridação, esteja ocorrendo introgressão entre as espécies, sendo maior em *P. leucoptera*, o que pode ser consequência de "genetic surfing".

Abstract

Hybridization can have many consequences, including maintenance or increase of diversity resulting in stable hybridization zones; introduction of a new allelic variety that can lead to the origin or transfer of adaptations, that may result on a positive effect on the rescue of small pure populations; enhancement of reproductive isolation and the origin of new species. In addition, hybridization may reduce diversity, as the break of reproductive barriers and the mixture of previously distinct populations may lead to the extinction of one of the populations or species. In this scenario, our study aimed to study a hybridization zone in the state of Bahia between two bird species, *Pyriglena atra* (Swaison 1825) and *P. leucoptera* (Vieillot 1818). We analyzed more than 20,000 SNPs from 20 individuals of *P. leucoptera*, 22 *P. atra* and six morphological intermediate individuals. We found little or no genetic mixture in the morphological intermediates, which indicates a mismatch between genotype and phenotype, possibly because these molecular markers were tested as neutral and these morphological characters could be under sexual selection. The majority of the individuals from localities adjacent to the hybrid zone have mixed ancestry, an indication that, besides hybridization, introgression among the species is occurring but higher into *P. leucoptera* as a possible consequence of genetic surfing.

Referências

Abbott, R. J. (1992) Plant invasions, interspecific hybridization and the evolution of new plant taxa. **Trends in Ecology and Evolution 7**: 401–405.

Abbott, R., Albach, D., Ansell, S., Arntzen, J. W., Baird, S. J. E., Bierne, N., Boughman, J., Brelsford, A., Buerkle, C. A., Buggs, R., Butlin, R. K., Dieckmann, U., Eroukhmanoff, F., Grill, A., Cahan, S. H., Hermansen, J. S., Hewitt, G., Hudson, A. G., Jiggins, C., Jones, J., Keller, B., Marczewski, T., Mallet, J., Martinez-Rodriguez, P., Möst, M., Mullen, S., Nichols, R., Nolte, A. W., Parisod, C., Pfennig, K., Rice, A. M., Ritchie, M. G., Seifert, B., Smadja, C. M., Stelkens, R., Szymura, J. M., Väinölä, R., Wolf, J. B. W., Zinner, D. (2013) Hybridization and speciation. **Journal of Evolutionary Biology 26**: 229–246.

Abbott, R.J.; Hegarty, M.J.; Hiscock, S.J.; Brennan, A.C. (2010) Homoploid hybrid speciation in action. **Taxon 59**: 1375–1386.

Allendorf, F. W.; Leary, R.F.; Spruell, P.; Wenburg, J. K. (2001) The problems with hybrids: setting conservation guidelines. **Trends in Ecology and Evolution 16**: 613–622.

Alexander, D.H.; Novembre, J.; Lange, K. (2009) Fast model-based estimation of ancestry in unrelated individuals. **Genome Research 19**:1655–1664.

Anderson, E. (1949) **Introgressive Hybridization.** New York: Wiley.

Anderson, E.; Hubricht, L. (1938) Hybridization in *Tradescantia*. III The evidence for introgressive hybridization. **American Journal of Botany 25**: 396–402.

Andersson, L. (1990) The driving force: Species concepts and ecology. **Taxon 3**: 375-382.

Andrews, S. (2012) FastQC A quality control tool for high throughput sequence data. [http://www.bioinformatics. bbsrc.ac.uk/projects/fastqc/].

Arnold, M.L. (1992) Natural hybridization as an evolutionary process. **Annual Review of Ecology, Evolution, and Systematics 23**: 237–261.

Arnold, M. L. (1996) **Natural Hybridization and Evolution.** New York: Oxford University Press.

Baack, E. J.; Rieseberg, L. H. (2007) A genomic view of introgression and hybrid speciation. **Current Opinion in Genetics and Development 17**: 513-518.

Barton, N. H.; Hewitt, G. M. (1985) Analysis of hybrid zones. **Annual Review of Ecology, Evolution, and Systematics 16**: 113-148.

Barton, N. H.; Hewitt, G. M. (1989) Adaptation, speciation and hybrid zones. **Nature 341**: 497-503.

Barton, N. H.; Gale, K. S. (1993). Genetic analysis of hybrid zones. *In*: R.G. Harrison (ed.). **Hybrid zones and the evolutionary process.** New York: Oxford University Press. Pp.: 13-45.

Bellott, D. W.; Skaletsky, H.; Cho, T. J.; Brown, L.; Locke, D.; Chen, N.; Galkina, S.; Pyntikova, T.; Koutseva,N.; Graves, T.; Kremitzki, C.; Warren, W.C.;Clark, A.G.; Gaginskaya, E.; Wilson. R.K.; Page, D.C. (2017). Avian W and mammalian Y chromosomes convergently retained dosage-sensitive regulators. **Nature Genetics 49.**3: 387-394.

BirdLife International (2014) Ficha de informações sobre espécies: *Pyriglena atra, Pyriglena leucoptera, Pyriglena leuconota*. Baixado de http://www.birdlife.org em 16/11/2014.

Brooks, T.; Tobias, J.; Balford, A. (1999) Deforestation and bird extinction in the Atlantic Forest. **Animal Conservation 2**: 211-222

Buerkle, C.A.; Wolf, D.E.; Rieseberg, L.H. (2003) The origin and extinction of species through hybridization. *In*: Brigham, C. A., Schwartz, M. W. (eds.) **Population Viability in Plants: Conservation, Management, and Modelling of Rare Plants.** Heidelberg: Springer Verlag. Pp. 117–141.

Buggs, R.J.A. (2007). Empirical study of hybrid zone movement. **Heredity 99**: 301–312.

Brumfield, R. T.; Tello, J. G.; Cheviron, Z. A.; Carling, M. D.; Crochet, N.; Rosenberg, K. V. (2007) Phylogenetic conservatism and antiquity of a tropical specialization: army-ant-following in the typical antbirds (Thamnophilidae). **Molecular Phylogenetics and Evolution 45**: 1-13.

Cadena, C. D.; Lopez-Lanus, B.; Bates, J. M.; Krabbe, N. ; Rice, N. H.; Stiles, F. G.; Palacio, J. D.; Salaman, P. (2007) A rare case of interspecific hybridization in the tracheophone suboscines: Chestnut-naped Antpitta *Grallaria nuchalis* X Chestnut-crowned Antpitta *G. ruficapilla* in a fragmented landscape. **Ibis 149**: 814-825.

Câmara, I.G. (2003) Brief history of conservation in the Atlantic forest. *In:* C. Galindo-Leal & I.G. Câmara (eds.). **The Atlantic Forest of South America: biodiversity status, threats, and outlook.** Washington D.C.: Center for Applied Biodiversity Science & Island Press. Pp.: 31-42.

Castric, V.; Bechsgaard, J.; Schierup, M.H.; Vekemans, X. (2008) Repeated adaptive introgression at a gene under multiallelic balancing selection. **PLoS Genet 4**: e1000168.

Chiba, S. (2005) Appearance of morphological novelty in a hybrid zone between two species of land snail. **Evolution 59**: 1712-1720.

Choler, P.; Erschbamer, B.; Tribsch, A.; Gielly, L.; Taberlet, P. (2004) Genetic introgression as a potential to widen a species' niche: insights from alpine *Carex curvula*. **Proceedings of the National Academy of Sciences of the United States of America 101**: 171–176.

Cracraft, J. (1983) Species concepts and speciation analysis. *In*: R.F. Johnston (ed.). **Current Ornithology.** New York: Plenum Press. Pp: 159-187.

Currat, M.; Excoffier, L. (2004). Modern humans did not admix with Neanderthals during their range expansion into Europe. **PLoS Biology 2**.12: e421.

Darwin, C. (1859) **On the Origin of Species by Natural Selection**. London: Murray.

De Queiroz, K. (1998) The general lineage concept of species, species criteria, and the process of speciation: A conceptual unification and terminological recommendations. *In:* Howard, D. J.; Berlocher, S. H. (eds.) **Endless forms: Species and Speciation.** New York: Oxford University Press. Pp: 57–75.

De Queiroz, K. (2005) Ernst Mayr and the modern concept of species. **Proceedings of the National Academy of Sciences of the United States of America 102**: 6600-6607.

Dobzhansky, T. (1970) **Genetics of the Evolutionary Process** (Vol. 139). New York: Columbia University Press.

Dobzhansky, T.; Dobzhansky, T. G. (1937) **Genetics and the Origin of Species** (No. 11). New York: Columbia University Press.

Donoghue, M. J. (1985) A critique of the biological species concept and recommendations for a phylogenetic alternative. **Bryologist 3**: 172-181.

Dowling, T. E.; Secor, C. L. (1997) The role of hybridization and introgression in the diversification of animals. **Annual Review of Ecology, Evolution, and Systematics 28**: 593-619.

Eaton, D. A. (2014) PyRAD: assembly of de novo RADseq loci for phylogenetic analyses. **Bioinformatics** btu121.

Edmands, S. (2007) Between a rock and a hard place: evaluating the relative risks of inbreeding and outbreeding for conservation and management. **Molecular Ecology 16**: 463–475.

Edmonds, C. A.; Lillie, A. S.; Cavalli-Sforza, L. L. (2004) Mutations arising in the wave front of an expanding population. **Proceedings of the National Academy of Sciences of the United States of America 101**: 975-979.

Ellstrand, N. C. (1992) Gene flow by pollen: implications for plant conservation genetics. **Oikos 63**: 77–86.

Ellstrand, N. C.; Elam, D. R. (1993) Population genetic consequences of small population size: implications for plant conservation. **Annual Review of Ecology and Systematics 24**: 217-242.

Ellstrand, N. C.; Prentice, H. C.; Hancock, J. F. (1999) Gene flow and introgression from domesticated plants into their wild relatives. **Annual Review of Ecology and Systematics 30**: 539-563.

Ellstrand, N. C.; Schierenbeck, K. A. (2000) Hybridization as a stimulus for the evolution of invasiveness in plants? **Proceedings of the National Academy of Sciences of the United States of America 97**: 7043–7050.

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**: e19379.

Frankham, R. (2015) Genetic rescue of small inbred populations: meta-analysis reveals large and consistent benefits of gene flow. **Molecular Ecology 24**: 2610–2618.

Foll, M.; Gaggiotti, O. E. (2008) A genome scan method to identify selected loci appropriate for both dominant and codominant markers: A Bayesian perspective. **Genetics 180**: 977-993

Foll, M.; Fischer, M. C.; Heckel, G.; Excoffier, L. (2010) Estimating population structure from AFLP amplification intensity. **Molecular Ecology 19**: 4638-4647

Fischer, M. C.; Foll, M.; Excoffier, L.; Heckel, G. (2011) Enhanced AFLP genome scans detect local adaptation in high-altitude populations of a small rodent (*Microtus arvalis*). **Molecular Ecology 20**: 1450-1462.

Frichot, E.; Mathieu, F.; Trouillon, T.; Bouchard, G.; François, O. (2014) Fast and efficient estimation of individual ancestry coefficients. **Genetics 196**: 973-983.

Fundação SOS Mata Atlântica & INPE (Instituto Nacional de Pesquisas Espaciais). 2001. **Atlas dos Remanescentes Forestais da Mata Atlântica e Ecossistemas Associados no Período de 1995–2000**. São Paulo: Fundação SOS Mata Atlântica e INPE.

Galindo-Leal, C.; Câmara, I.G. (2003) Atlantic forest hotspots status: an overview. *In:* Galindo-Leal, C.*;* Câmara, I.G. (eds.). **The Atlantic Forest of South America: Biodiversity Status, Threats, and Outlook.** Washington, D.C.: Center for Applied Biodiversity Science & Island Press. Pp.: 3-11.

Grant, P. R.; Grant, B. R. (1992) Hybridization of bird species. **Science 256:** 193-197.

Grant, P. R.; Grant, B. R. (1994) Phenotypic and genetic effects of hybridization in Darwin's finches. **Evolution 48**: 297-316.

Graves, G. R. (1992) Diagnosis of a hybrid antbird (*Phlegopsis nigromaculata* X *Phlegopsis erythroptera*) and the rarity of hybridization among suboscines. **Proceedings of the Biological Society of Washington 105**: 834-840.

Gross, B.L.; Rieseberg, L.H. (2005) The ecological genetics of homoploid hybrid speciation. **Journal of Heredity 96**: 241–252.

Levin, D. A.; Francisco-Ortega, J.; Jansen, R. K. (1996) Hybridization and the extinction of rare plant species. **Conservation Biology 10**: 10–16.

Harrison, R. G. (1990) Hybrid zones: windows on evolutionary process. *In:* Futuyma, D., Antonovics, J. (eds.) **Oxford Surveys in Evolutionary Biology.** New York: Oxford University Press. Pp.: 69–128.

Harrison, R. G. (Ed.) (1993) **Hybrid Zones and the Evolutionary Process.** New York: Oxford University Press.

Harrison, R. G. (1998) Linking evolutionary pattern and process. *In:* Howard, D. J. and Berlocher, S. H. (eds.) **Endless Forms: Species and Speciation.** New York: Oxford University Press. Pp: 19-31.

Heusmann, H. W. (1974) Mallard-black duck relationships in the northeast. **Wildlife Society Bulletin 2**: 171-177.

Huxley, J. (1942) **Evolution. The Modern Synthesis.** London: George Alien & Unwin Ltd.

Isler, M. L.; Bravo, G. A.; Brumfield, R. T. (2013) Taxonomic revision of *Myrmeciza* (Aves: Passeriformes: Thamnophilidae) into 12 genera based on phylogenetic, morphological, behavioral, and ecological data. **Zootaxa 3717**: 469–497.

Isler, M. L.; Maldonado-Coelho, M. (2017) Calls distinguish species of Antbirds (Aves: Passeriformes: Thamnophilidae) in the genus *Pyriglena*. **Zootaxa: No prelo.**

IUCN (2016) **Red List of Threatened Species.** Versão 2015.4. Disponível em <http://iucnredlist.org>. Acesso em [01.2016].

Jiggins, C.D.; Salazar, C.; Linares, M.; Mavarez, J. (2008) Hybrid trait speciation and *Heliconius* butterflies. **Philosophical Transactions of the Royal Society of London. Series B 363**: 3047–3054.

Johnsgard, P. A. (1961) Evolutionary relationships among the North American mallards. **The Auk** 78: 3-43.

Kim, M.; Cui, M.-L.; Cubas, P.; Gillies, A.; Lee, K.; Chapman, M.A; Abbott, R.J.; Coen, E. (2008) Regulatory genes control a key morphological and ecological trait transferred between species. **Science 322**: 1116–1119.

Klopfstein, S.; Currat, M.; Excoffier, L. (2005). The fate of mutations surfing on the wave of a range expansion. **Molecular biology and evolution 23.3**: 482-490.

Kumar, G., & Kocour, M. (2017). Applications of next-generation sequencing in fisheries research: A review. **Fisheries Research 186**: 11-22.

Langmead, B.; Salzberg, S. (2012) Fast gapped-read alignment with Bowtie 2. **Nature Methods 9**: 357-359.

Lischer, H. E. L.; Excoffier, L. (2012) PGDSpider: An automated data conversion tool for connecting population genetics and genomics programs. **Bioinformatics 28**: 298-299.

Maldonado-Coelho, M. (2010) **Evolution and Biogeography of South American Fire-Eyes (Genus *Pyriglena*): Insights from Molecules and Songs**. PhD Dissertation. St Louis, University of Missouri.

Maldonado-Coelho, M. (2012) Climatic oscillations shape the phylogeographical structure of Atlantic Forest fire-eye antbirds (Aves: Thamnophilidae). **Biological Journal of the Linnean Society 105**: 900-924.

Mallet, J. (2005) Hybridization as an invasion of the genome. **Trends in Ecology and Evolution 20**: 229–237.

Mallet, J. (2007) Hybrid speciation. **Nature 446**: 279–283.

Margulies, M.; Egholm, M.; Altman, W. E.; Attiya, S.; Bader, J. S.; Bemben, L. A.; Volkmer, G. A. (2005) Genome sequencing in microfabricated high-density picolitre reactors. **Nature 437**: 376-380.

Martin, N.H.; Bouck, A.C.; Arnold, M.L. (2006) Detecting adaptive trait introgression between *Iris fulva* and *I. brevicaulis* in highly selective field conditions. **Genetics 172**: 2481–2489.

Martinsen, G.D.; Whitham, T.G.; Turek, R.J.; Keim, P. (2001) Hybrid populations selectively filter gene introgression between species. **Evolution 55**: 1325–1335.

Mayden, R. L. (1997) A hierarchy of species concepts: the denouement in the saga of the species problem. *In*: M. F. Claridge, H. A. Dawah, M. R. Wilson (eds.) **Species: The units of diversity.** London: Chapman and Hall. Pp: 381–423.

Mayr, E. (1942) **Systematics and the Origin of Species, from the Viewpoint of a Zoologist.** Cambridge: Harvard University Press.

Mayr, E.; Provine, W. B., (eds). (1980) **The Evolutionary Synthesis: Perspectives on the Unification of Biology.** Cambridge: Harvard University Press.

Mayr, E. (1982) **The Growth of Biological Thought: Diversity, Evolution, and Inheritance.** Cambridge: Belknap Press of Harvard University Press.

McCormack, J. E.; Maley, J. M.; Hird, S. M.; Derryberry, E. P.; Graves, G. R.; Brumfield, R. T. (2012) Next-generation sequencing reveals phylogeographic structure and a species tree for recent bird divergences. **Molecular Phylogenetics and Evolution 62**: 397-406.

Metzker, M.L. (2010) Sequencing technologies - the next generation. **Nature Reviews Genetics 11**: 31–46.

Myers, N.; Mittermeier, R.A.; Mittermeier, C.G.; Fonseca, G.A.B.; Kent, J. (2000) Biodiversity hotspots for conservation priorities. **Nature 403**: 853-845.

Nixon, K. C.; Wheeler, Q. D. (1990) An amplification of the phylogenetic species concept. **Cladistics 6**: 211-223.

Oliveira-Filho, A.T.; Fontes, M.A.L. (2000) Patterns of floristic differentiation among Atlantic Forests in southeastern Brazil and the influence of climate. **Biotropica 32**: 793-810.

Parsons, T. J.; Olson, S. L.; Braun, M. J. (1993) Unidirectional spread of secondary sexual plumage traits across an avian hybrid zone. **Science 11**: 1643-1646.

Pickrell, J. K.; Pritchard, J. K. (2012) Inference of population splits and mixtures from

genome-wide allele frequency data. **PLoS Genet 8**: e1002967.

Pritchard, J. K.; Stephens, M.; Donnelly, P. (2000) Inference of population structure using multilocus genotype data. **Genetics 155**: 945–959.

Price, T. (2008) **Speciation in Birds.** Greenwood Village: Roberts and Co..

Raj, A.; Stephens, M.; Pritchard, J. K. (2014) fastSTRUCTURE: variational inference of population structure in large SNP data sets. **Genetics 197**: 573-589.

Reich, D.; Thangaraj, K.; Patterson, N.; Price, A.L.; Singh L. (2009) Reconstructing Indian population history. **Nature 461**: 489–494.

Remsen Jr, J. V.; Cadena, C. D.; Jaramillo, A., Nores, M., Pacheco, J. F., Pérez-Emán, J.; Robbins, M.B.; Stiles, F.G.; Stotz, D.F.; Zimmer, K. J. (2016). A Classification of the Bird Species of South America. Version 20 September 2016. **American Ornithologists' Union. Available from: http://www.museum.lsu.edu/~Remsen/SACCBaseline.html.**

Rhymer, J. M.; Simberloff, D. (1996) Extinction by hybridization and introgression. **Annual Review of Ecology and Systematics 27**: 83–109.

Ribeiro, M. C.; Metzger, J. P.; Martensen, A. C.; Ponzoni, F. J.; Hirota, M. M. (2009) The Brazilian Atlantic Forest: How much is left, and how is the remaining forest distributed? Implications for conservation. **Biological Conservation 142**: 1141-1153.

Ridgely, R. S.; Tudor, G. (1984) **The Birds of South America. The Suboscine Passerines. Volume II.** Austin: University of Texas.

Rieseberg, L. H.; Archer, M. A.; Wayne, R. K. (1999) Transgressive segregation, adaptation and speciation. **Heredity 83**: 363-372.

Rieseberg, L.H.; Baird, S.J.E.; Gardner, K.A. (2000). Hybridization, introgression, and linkage evolution. **Plant Molecular Biology 42**: 205–224.

Rieseberg, L.H.; Carney, S.E. (1998) Tansley review no. 102 Plant hybridization. **New Phytologist 140**: 599–624.

Rieseberg, L.H.; Ellstrand, N.C. (1993). What can molecular and morphological markers tells us about Plant hybridization? **Critical Reviews in Plant Sciences 12**: 213–241.

Rieseberg, L.H.; Zona, S.; Aberbom, L.; Martin, T.D. (1989) Hybridization in the island endemic, Catalina mahogany. **Conservation Biology 3**: 52-58.

Rosen, D. E. (1979) Fishes from the uplands and intermontane basins of Guatemala: revisionary studies and comparative geography. **Bulletin of the British Museum of Natural History 162**: 267-376.

Sanger, F.; Coulson, A. R. (1975). A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. **Journal of Molecular Biology 94**: 441IN19447-446IN20448.

Sanger, F.; Nicklen, S.; Coulson, A. R. (1977) DNA sequencing with chain-terminating inhibitors. **Proceedings of the National Academy of Sciences of the United States of America 74**: 5463-5467.

Schwenk, K.; Brede, N.; Streit, B. (2008) Introduction. Extent processes and evolutionary impact of interspecific hybridization in animals. **Philosophical Transactions of the Royal Society of London. Series B 363**: 2805–2811.

Seddon, N.; Tobias, J. A. (2007) Song divergence at the edge of Amazonia: an empirical test

of the peripatric speciation model. **Biological Journal of the Linnean Society 90**: 173-188.

Seehausen, O. (2004) Hybridization and adaptive radiation. **Trends in Ecology and Evolution 19**: 198-207.

Shendure, J.; Porreca, G. J.; Reppas, N. B.; Lin, X.; McCutcheon, J. P.; Rosenbaum, A. M.; Church, G. M. (2005) Accurate multiplex polony sequencing of an evolved bacterial genome. **Science 309**: 1728-1732.

Shrader-Frechette, K. S.; McCoy, E. D. (1993) **Methods in Ecology: Strategies for Conservation.** Cambridge: Cambridge University Press.

Silva, J.M.C; Casteleti, C.H.M. (2003) Status of the biodiversity of the Atlantic Forest of Brazil. *In:* C. Galindo-Leal & I.G. Câmara (eds.). **The Atlantic Forest of South America: Biodiversity Status, Trends, and Outlook.** Washington, D.C.: Center for Applied Biodiversity Science & Island Press. Pp.: 43-59.

Stamos, D.N. (2003) **The Species Problem: Biological Species, Ontology, and the Metaphysics of Biology.** Lanham: Lexington Books.

Stebbins, G. L. (1959) The role of hybridization in evolution. **Proceedings of the American Philosophical Society 103**: 231–251.

Tabarelli, M.; Pinto, L.P.; Silva, J.M.C.; Costa, C.M.R. (2003) The Atlantic Forest of Brazil: endangered species and conservation planning. *In:* C. Galindo-Leal & I.G. Câmara (eds.). **The Atlantic Forest of South America: Biodiversity Status, Trends, and Outlook.** Washington, D.C.: Center for Applied Biodiversity Science & Island Press. Pp.: 86-94.

Tabarelli, M.; Pinto, L.P.; Silva, J.M.C.; Hirota, M.M.; Bedê, L.C. (2005) Desafios e oportunidades para a conservação da biodiversidade na Mata Atlântica brasileira. **Megadiversidade 1**: 132-138.

Templeton, A. R. (1986) Coadaptation and outbreeding depression. *In*: Soulé, N. (ed.) **Conservation Biology: the Science of Scarcity and Diversity.** Sunderland: Sinauer Associates. Pp.: 105–116.

Todesco, M.; Pascual, M. A.; Owens, G. L.; Ostevik, K. L.; Moyers, B. T.; Hübner, S.; Heredia, S. M.; Hajn, M. A.; Caseys, C.; Bock, D. G.; Rieseberg, L. H. (2016) Hybridization and extinction. **Evolutionary Applications 9**: 892-908.

Twyford, A. D.; Ennos, R. A. (2012) Next-generation hybridization and introgression. **Heredity 108**: 179-189.

Van Valen, L. (1976) Ecological species, multispecies, and oaks. **Taxon 25**: 233-239.

Vuillaume, B.; Valette, V.; Lepais, O.; Grandjean, F.; Breuil, M (2015) Genetic evidence of hybridization between the endangered native species *Iguana delicatissima* and the invasive *Iguana iguana* (Reptilia, Iguanidae) in the Lesser Antilles: management implications. **PloS ONE 10**: e0127575.

Walsh, J.; Lovette, I. J.; Winder, V.; Elphick, C. S.; Olsen, B. J.; Shriver, G.; Kovach, A. I. (2017). Subspecies delineation amid phenotypic, geographic and genetic discordance in a songbird. **Molecular Ecology 26**: 1242-1255.

Weir, J. T.; Faccio, M. S.; Pulido-Santacruz, P.; Barrera-Guzmán, A. O.; Aleixo, A. (2015) Hybridization in headwater regions, and the role of rivers as drivers of speciation in Amazonian birds. **Evolution 69**: 1823-1834.

Willis, E. O. (1981) Diversity in adversity: the behaviors of two subordinate antbirds.

Arquivos de Zoologia 30: 159-234.

Willis, E. O.; Oniki, Y. (1982) Behavior of Fringe-backed Fire-eyes (*Pyriglena atra*, Formicariidae): a test case for taxonomy versus conservation. **Revista Brasileira de Biologia 42**: 213-223.

Wolf, D. E.; Takebayashi, N.; Rieseberg, L. H. (2001) Predicting the risk of extinction through hybridization. **Conservation Biology 15**: 1039–1053.

Zhang, J.; Kobert, K.; Flouri, T.; Stamatakis, A. (2014) PEAR: a fast and accurate Illumina Paired-End reAd mergeR. **Bioinformatics 30:** 614-620.

Zimmer, K. J.; Isler, M. L. (2003) Family Thamnophilidae (typical antbirds). *In*: del Hoyo, J.; Elliott, A.; Christie, D.A. (eds.) **Handbook of Birds of the World**. Barcelona: Lynx Edicións. Pp.: 448-592.

Zhou, H.; Alexander, D. H.; Lange, K. (2011) A quasi-Newton method for accelerating the convergence of iterative optimization algorithms. **Statistics and Computing 21**.2: 261-273.

ANEXO 1 - Manual LGEMA: FastQC

Tradução por Ana Beatriz da Silva Bueno (ana_bia_bueno@hotmail.com) das páginas citadas a seguir (consulta: 30/03/2016) acrescentada de imagens.

Página da *internet* do FastQC:
http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
Página da *internet* com manual:
http://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/
Página da *internet* para *download* do programa FastQC:
http://www.bioinformatics.babraham.ac.uk/projects/download.html#fastqc

1. Introdução

O que é FastQC (Andrews, 2012)?

Os sequenciadores modernos de alto rendimento podem gerar centenas de milhões de sequências em uma única execução. Antes de analisar esta sequência para tirar conclusões biológicas, você sempre deve realizar o controle da qualidade dos dados brutos diminuindo a posssibilidade de haver problemas ou vieses em seus dados.

A maioria dos sequenciamentos gera um relatório de QC como parte de seu *pipeline* de análise, mas geralmente ele é focado na identificação de problemas durante o sequenciamento. O FastQC fornece um relatório de *quality control* (QC) que detecta problemas que ocorreram no sequenciamento ou no material da biblioteca de amostras.

FastQC pode ser executado de dois modos alternativos. Ele pode ser executado como um aplicativo interativo para a análise imediata de poucos arquivos FastQ, ou pode ser executado de modo não interativo para o qual seria adequado integrar um *pipeline* de análise maior para o processamento sistemático de grande número de arquivos.

2. Instalação dos pacotes necessários

2.1. Java

Para conferir se possui o Java instalado e qual a versão, digitar no terminal:

$ java -version

Caso não tenha, o Java deve ser instalado com a seguinte linha de comando no terminal:

$ sudo apt-get install default-jve

Após a intalação, repetir a verificação do Java:

$ java -version

2.2. FastQC

Pode ser obtido por *download* direto do site:

http://www.bioinformatics.babraham.ac.uk/projects/download.html#fastqc

Ou pode ser obtido por linha de comando no terminal:

$ sudo apt-get fastqc

O arquivo baixado deve ser descompactado digitando no terminal:

$ unzip fastqc_v.0.11.5.zip

3. Rodando o FastQC

Você pode executar FastQC de dois modos alternativos, como um aplicativo gráfico interativo no qual você pode carregar dinamicamente arquivos FastQ e exibir seus resultados (maneira como executei o programa e será explicado a seguir).

Alternativamente, você pode executar o FastQC em um modo não interativo pelo qual você especifica os arquivos que deseja processar na linha de comando e o FastQC irá gerar um relatório HTML para cada arquivo sem lançar uma interface do usuário. Isso permitiria o FastQC ser executado como parte de um *pipeline* de análise (mais detalhes desse tipo de execução no link: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/INSTALL.txt).

3.1. Rodando interativamente

No Linux: o *wrapper script* chamado 'fastqc' é a maneira mais fácil de iniciar o programa. O *wrapper* está no nível superior da instalação FastQC. Talvez seja necessário tornar este arquivo executável:

$ chmod 755 fastqc

Assim que realizar o comando acima, pode rodar diretamente com:

$ ./fastqc

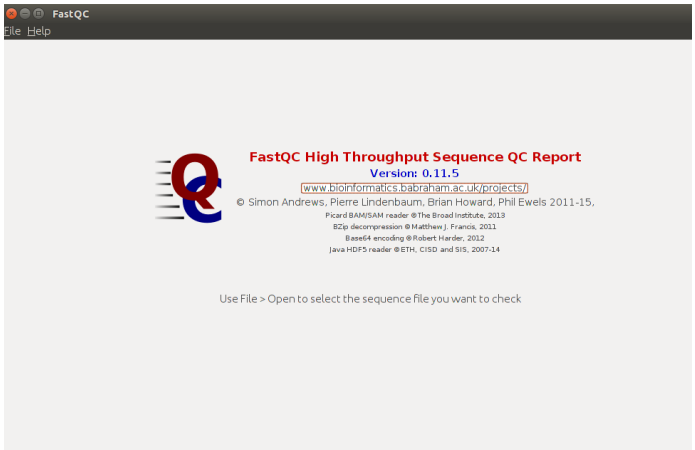
Ou colocar o link em /usr/local/bin e permitir que o programa seja executado em qualquer localização:

$ sudo ln -s /path/to/FastQC/fastqc /usr/local/bin/fastqc

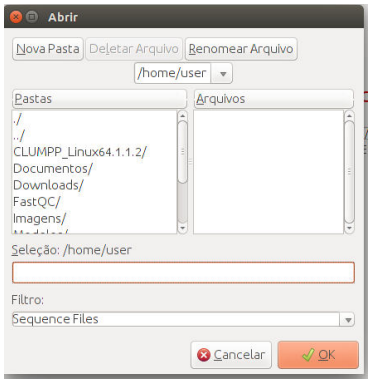
4. Executando o programa

Ao digitar no terminal: $ ./fastq

A seguinte página deverá abrir:



Clicar em File > Open… , assim a seguinte página irá abrir:

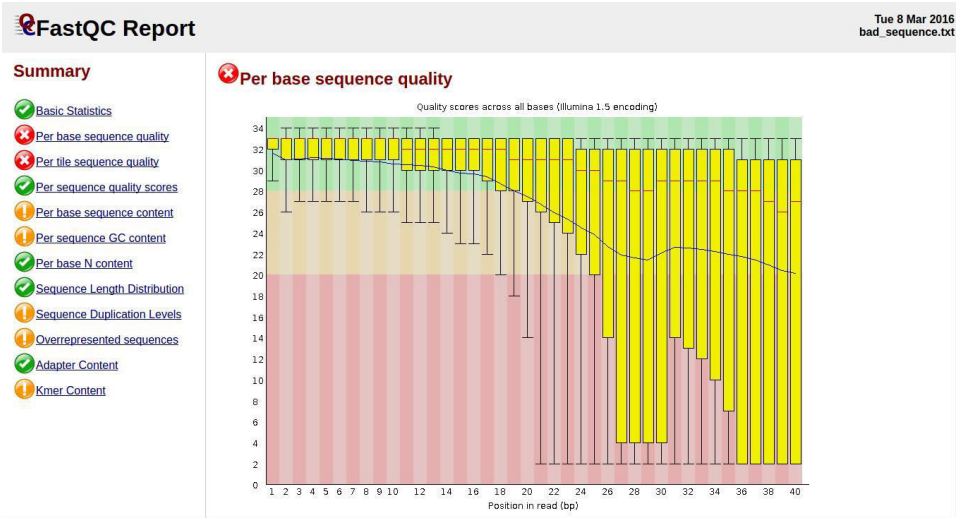


Você deverá indicar a localização do arquivo que você quer analisar. Assim que o encontrar, selecione o Filtro (tipo de arquivo que será analisado) e clique em OK. A análise começará logo em seguida.

5. Exemplos de resultados para dados plataforma Illumina

5.1. Dados com boa qualidade

5.2. Dados com má qualidade



6. Dúvidas

Em casa de dúvidas, consulte os tutoriais oficiais do autor do programa (ver na 1ª página desse manual), ou entre em contato com o autor do programa, Simon Andrews: simon.andrews@babraham.ac.uk

7. Referência

Andrews, S. (2012) FastQC. A quality control tool for high throughput sequence data. **[http://www.bioinformatics. bbsrc.ac.uk/projects/fastqc/]**.

ANEXO 2 - Manual LGEMA: pyRAD e PEAR para GBS *paired-end*

Tradução por Ana Beatriz da Silva Bueno (ana_bia_bueno@hotmail.com) das páginas citadas a seguir (consulta: 20/07/2016) acrescentada de imagens e dados.

Página da *internet* do pyRAD:
http://dereneaton.com/software/pyrad/
Tutorial do pyRAD v.3.0:
http://nbviewer.jupyter.org/gist/dereneaton/af9548ea0e94bff99aa0/pyRAD_v.3.0.ipynb
Tutorial de exemplo de análise de dados de sequenciamento GBS *paired-end*:
http://nbviewer.jupyter.org/gist/dereneaton/1f661bfb205b644086cc/PE-GBS_empirical.ipynb
Página da *internet* do PEAR:
http://sco.h-its.org/exelixis/web/software/pear/doc.html

1. Introdução

Segue uma breve descrição do que os *scripts* do pyRAD e PEAR realizam com os dados.

pyRAD (Eaton, 2014) foi escrito com o objetivo de ser fácil de utilizar, rápido e flexível. Sua instalação é simples, não requer nenhuma biblioteca ou extensões difíceis, somente requer alguns argumentos na linha de comando. É rápido, usado tanto no multi-processamento e multi-tratamento, e também oferece uma série heurística para acelerar o agrupamento de conjuntos de dados extremamente grandes. É flexível pois foi desenvolvido para diferentes tipos de biblioteca (RAD, ddRAD, GBS, PE-ddRAD, PE-GBS). Realiza também a filtragem de qualidade, incluindo a sobreposição de *reads* pareadas e as reversas, e a presença de sequências de adaptador. Assim, os dados não precisam ser filtrados antes da análise pelo pyRAD. A análise pode ser realizada em 7 passos sucessivos de modo a permitir a verificação do resultado de cada passo antes de proceder para o seguinte. Os passos serão detalhados mais para frente nesse manual.

PEAR (Zhang *et al*. 2014) foi escrito para ser rápido, eficiente e preciso na fusão de arquivos pareados (R1 e R2). É totalmente paralelizado e pode ser executado com kilobytes de memória em um computador de mesa padrão. Ele avalia as sobreposições nas *reads* pareadas, sem exigir o tamanho do fragmento alvo como entrada. Além disso, ele executa um teste estatístico para minimizar resultados falsos positivos.

2. Instalação dos pacotes necessários

2.1. Numpy e Scipy

No terminal: $ sudo dnf install numpy scipy

2.2. muscle

muscle é um programa para realizar alinhamento múltiplo com precisão e velocidade (centenas de sequências em segundos).

Baixar em: http://www.drive5.com/muscle/downloads.htm

No terminal: $ tar -zxvf muscle3.8.31_i86linux64.tar.gz

$ chmod +x muscle3.8.31_i86linux64

2.3. vsearch

VSEARCH faz uma busca heurística rápida mas sensível baseada em segmentos compartilhados entre a *read* alvo e as demais *reads* para identificar *reads* similares. Em seguida, executa alinhamento global de sequências fazendo a busca contra várias sequências alvo potenciais em paralelo.

No terminal:

(pré-requisitos):

$ sudo dnf install autoconf automake libtool

$ sudo dnf install make automake gcc gcc-c++ kernel-devel

(vsearch):

$ wget https://github.com/torognes/vsearch/archive/v1.11.1.tar.gz

$ tar xzf v1.11.1.tar.gz

$ cd vsearch-1.11.1

$ ./autogen.sh

$ ./configure

$ make

$ sudo make install

2.4. pyRAD

No terminal:

$ git clone https://github.com/dereneaton/pyrad.git

```
$ cd pyrad
$ sudo python setup.py install
$ pyrad -h    #-h = help - mostra as opções
```

Obs.: sem instalar você pode chamar o pyRAD.py de sua localização usando o python

```
$ python pyrad/pyrad/pyRAD.py -h
```

2.5. PEAR

No terminal:

(pré-requisitos):

```
$ sudo dnf install autoconf automake libtool
$ sudo dnf install make automake gcc gcc-c++ kernel-devel
```

(PEAR):

```
$ git clone https://github.com/xflouris/PEAR.git
$ cd PEAR
$ ./autogen.sh
$ ./configure
$ make
$ sudo make install
```

(testando PEAR):

```
$ cd test
$ ./test.py
```

3. Entrada de dados

3.1. Arquivo de *barcodes*

Esse arquivo contém uma tabela que informa o *barcode* de cada amostra/indivíduo. Os *barcodes* podem ser de diferentes comprimentos. Cada linha deve ter um nome e um *barcode*, separados por uma tabulação ou espaço. O nome do arquivo deve ser “barcode.txt” e deve ficar na pasta “pyrad”. A seguir, segue a imagem do meu arquivo “barcode.txt”:

barcodes.txt

```
MMC234  AACCA
MMC232  CCACG
MMC233  TATAA
15362   GAGCG
16545   ACATA
16548   CTCAG
16550   TCTGG
16551   CGGCA
16552   GAAGA
16553   ACGAG
16555   TTCTA
16556   AGTCG
16554   GCCAA
17112   TAATG
17113   GTTCA
17114   GGCGG
MMC224  CAGAT
MMC228  ACTGCA
MMC230  CTGCCA
MMC223  TGATCA
MMC226  AAGTCA
MMC222  CGTACA
MMC225  TTCGCA
16704   GTAACA
16705   GATCCA
16706   ACATCG
16707   TCGCAA
16708   CGCTGA
16709   AACGTG
16710   CCTAAG
16711   TTACGA
16685   ATGGCG
16702   GGACTA
17921   CACTAG
17922   TCTAGCA
17923   GTGGCCA
17924   AGAATCA
17925   CATCGCA
17926   TCCGACA
17927   AGGTCCA
17928   GAATACA
17929   CCTGTCA
17930   TTCCGCA
17931   TAGACCA
17934   GGCCTCA
17935   ACATGCA
17936   CTTGCAA
17937   ATGAACA
```

Texto sem formatação Largura da tabulação: 8 Lin 28, Col 15 INS

Se os dados já estiverem ordenados em arquivos separados para cada amostra, esse arquivo de *barcodes* não é necessário.

3.2. Arquivo de parâmetros

O arquivo de parâmetros, chamado params.txt, pode ser criado com a opção -n no pyRAD, como abaixo.

No terminal: $ cd pyrad
$ pyrad –n

Este arquivo lista todas as opções ou configurações necessárias para uma análise. Cada linha se refere a um parâmetro (por exemplo, um número ou cadeia de caracteres) seguido por qualquer número de espaços ou tabulações e, em seguida, duas marcas de *hash* ("##"), após a qual o parâmetro é descrito e/ou comentários pessoais podem ser adicionados. Entre parênteses é indicado o passo da análise no qual o parâmetro será usado. Se uma linha for deixada em branco, o valor padrão (*default*) será usado.

A seguir o arquivo assim que é gerado pelo *software*:

```
params.txt ×
==** parameter inputs for pyRAD version 3.0.66  **======================= affected step ==
./                      ## 1. Working directory                          (all)
./*.fastq.gz            ## 2. Loc. of non-demultiplexed files (if not line 18)  (s1)
./*.barcodes            ## 3. Loc. of barcode file (if not line 18)      (s1)
vsearch                 ## 4. command (or path) to call vsearch (or usearch)    (s3,s6)
muscle                  ## 5. command (or path) to call muscle           (s3,s7)
TGCAG                   ## 6. Restriction overhang (e.g., C|TGCAG -> TGCAG)     (s1,s2)
2                       ## 7. N processors (parallel)                    (all)
6                       ## 8. Mindepth: min coverage for a cluster       (s4,s5)
4                       ## 9. NQual: max # sites with qual < 20 (or see line 20)(s2)
.88                     ## 10. Wclust: clustering threshold as a decimal        (s3,s6)
rad                     ## 11. Datatype: rad,gbs,pairgbs,pairddrad,(others:see docs)(all)
4                       ## 12. MinCov: min samples in a final locus       (s7)
3                       ## 13. MaxSH: max inds with shared hetero site    (s7)
c88d6m4p3               ## 14. Prefix name for final output (no spaces)   (s7)
==== optional params below this line ==================================  affected step ==
                      ## 15.opt.: select subset (prefix* only selector)            (s2-s7)
                      ## 16.opt.: add-on (outgroup) taxa (list or prefix*)         (s6,s7)
                      ## 17.opt.: exclude taxa (list or prefix*)                   (s7)
                      ## 18.opt.: loc. of de-multiplexed data                      (s2)
                      ## 19.opt.: maxM: N mismatches in barcodes (def= 1)          (s1)
                      ## 20.opt.: phred Qscore offset (def= 33)                    (s2)
                      ## 21.opt.: filter: def=0=NQual 1=NQual+adapters. 2=strict   (s2)
                      ## 22.opt.: a priori E,H (def= 0.001,0.01, if not estimated) (s5)
                      ## 23.opt.: maxN: max Ns in a cons seq (def=5)               (s5)
                      ## 24.opt.: maxH: max heterozyg. sites in cons seq (def=5)   (s5)
                      ## 25.opt.: ploidy: max alleles in cons seq (def=2;see docs) (s4,s5)
                      ## 26.opt.: maxSNPs: (def=100). Paired (def=100,100)         (s7)
                      ## 27.opt.: maxIndels: within-clust,across-clust (def. 3,99) (s3,s7)
                      ## 28.opt.: random number seed (def. 112233)              (s3,s6,s7)
                      ## 29.opt.: trim overhang left,right on final loci, def(0,0) (s7)
                      ## 30.opt.: output formats: p,n,a,s,v,u,t,m,k,g,* (see docs) (s7)
                      ## 31.opt.: maj. base call at depth>x<mindepth (def.x=mindepth) (s5)
                      ## 32.opt.: keep trimmed reads (def=0). Enter min length.    (s2)
                      ## 33.opt.: max stack size (int), def= max(500,mean+2*SD)    (s3)
                      ## 34.opt.: minDerep: exclude dereps with <= N copies, def=1 (s3)
                      ## 35.opt.: use hierarchical clustering (def.=0, 1=yes)      (s6)
                      ## 36.opt.: repeat masking (def.=1='dust' method, 0=no)      (s3,s6)
                      ## 37.opt.: vsearch max threads per job (def.=6; see docs)   (s3,s6)
==== optional: list group/clade assignments below this line (see docs) ==================
Texto sem formatação   Largura da tabulação: 8   Lin 10, Col 87   INS
```

3.2.1. Definição de cada linha de comando

A seguir vamos descrever o que cada linha faz no arquivo de parâmetros. As primeiras 14 linhas tem preenchimento obrigatório. As demais linhas são opções avançadas e não necessitam de preenchimento (neste caso, serão usados os valores padrões).

Linhas 1 a 14 - Obrigatórias

Linha 1: Diretório de trabalho para a sua análise.

Este é o local para onde todos os arquivos de saída serão direcionados.

Exemplos: ## 1. usa o diretório atual como diretório de trabalho

./ ## 1. usa o diretório atual como diretório de trabalho

/home/user/RAD/ ## 1. usa o diretório “RAD” como diretório de trabalho

Linha 2: Localização dos dados brutos de sequências em fastqc.

Use o caractere * para selecionar vários arquivos no mesmo diretório. Dados pareados devem possuir o mesmo nome, apenas diferindo pela presença de “R1” ou “R2” no final dos nomes. Os arquivos podem estar compactados.

Exemplos: ./raw/*.fastq ## 2. caminho para buscar os dados brutos

raw/*.fastq.gz ## 2. caminho para buscar os dados brutos zipados

Linha 3: Localização do arquivo *barcode*.

Se os seus dados já estão separados, então as linhas 2 e 3 podem ser deixadas em branco, mas a linha 18 deve ser preenchida. Caso precise separar as *reads* segundo os *barcodes*, incluir:

Exemplos: ./barcodes.txt ## 3. caminho para buscar o arquivo *barcode*

mydata.barcodes ## 3. caminho para buscar o arquivo *barcode*

Linha 4: Comando para chamar vsearch (ou usearch).

Se vsearch está na sua $ PATH, então simplesmente liste o nome do arquivo executável aqui. Caso contrário, digite o caminho completo para vsearch para que você possa chamá-lo pela linha de comando.

Exemplos: vsearch ## 4. no PATH do sistema

/user/bin/usearch7.0.1090_i86linux32 ## 4. caminho completo

usearch7.0.1090_i86linux32 ## 4. no PATH do sistema

Linha 5: Comando para chamar muscle.

Semelhante ao descrito para a linha 4.

Exemplos: muscle ##5. caminho direto

muscle3.8.31_x86 linux32 ## 5. caminho completo

Linha 6: Sequência do sítio de restrição.

Se seu dado for ddRAD, liste o sítio da enzima mais comum (que consta nas *reads* R1) seguido daquele da enzima mais rara (que consta nas *reads* R2), separado por vírgulas. Se não tiver certeza dos sítios, buscar nos seus dados brutos as sequências mais comuns encontradas na extremidade 5' de suas *reads.* As enzimas podem conter bases ambíguas (Exemplo: CWGC).

Exemplos: TGCAG ## 6. fita simples do corte de *Pst*I C|TGCAG

CWGC, AATT ## 6. no caso de ddRAD com *ape*KI e, *Eco*RI

Linha 7: Número de análises em paralelo que serão executadas.

Para a maioria das etapas que não envolvem agrupamento (*clustering*) este é o número de processadores a serem utilizados. O processamento paralelo em pyRAD geralmente ocorre por amostra individual separada em processadores diferentes, então raramente se ganha velocidade ao indicar mais processadores do que o número de amostras. No entanto, para os passos de agrupamento 3 e 6, o número de tarefas (*threads*) em paralelo pode ser configurado usando mais processadores (ver linha 36).

Exemplo: 8 ## 7. 8 processadores

Linha 8: Profundidade mínima de cobertura para fazer uma chamada de base em cada sítio de um agrupamento.

Dependendo da taxa de erro de sequenciamento inferida no passo 4, o valor de cobertura mínima pode ser modificado, mas em geral é inicialmente adotada cobertura mínima de 5 para chamar uma base. Para análises populacionais é melhor usar cobertura mínima mais elevada, ex. 10. Se seus dados tem cobertura muito baixa, você pode chamar por consenso de maioria. Para isso, defina um número baixo aqui (por exemplo, 2) e use a regra de maioria para a chamada (linha 29; ver abaixo).

Exemplo: 5 ## 8. cobertura mínima de 5 para a chamada de bases

Obs: O consenso de maioria é estabelecido após o alinhamento das *reads* e a base que aparece em maior frequência é a que será escolhida. Por exemplo, se em um sítio tiver C/T/T/T, a T será escolhida pelo programa. Porém, se ocorrer C/C/T/T, a C será escolhida pois o programa considera a ordem alfabética.

O parâmetro da linha 8 é a cobertura mínima para fazer a chamada da base, se escolher 10, será necessário ter pelo menos 10 *reads* alinhadas para fazer a chamada da base. Esse parâmetro está ligado ao da linha 31. Se a linha 31 for ignorada, o valor de 10 da linha 8 será a cobertura mínima para chamar a base. Se considerar a linha 31 (chamada de consenso simples) o consenso de maioria será considerado. Dessa forma, se linha #8=10 e linha #31=5, todos os agrupamentos com profundidade > 10 terão base chamada com probabilidade significativa e os agrupamentos com cobertura entre 5 e 10 terão base chamada baseado no consenso de maioria. Dessa forma se perde menos informação, mas aumenta o risco de inferir a base errada. Essa estratégia pode ser útil se a cobertura nos agrupamentos for baixa.

Linha 9: Número máximo de sítios de baixa qualidade ou indeterminados ("N") nas sequências após a filtragem do Passo 2 (abaixo).

Este número deve ser selecionado de acordo com o limiar a ser adotado para o agrupamento (ver linha a seguir). Um limite baixo de agrupamento (.85) permite ter mais sítios indeterminados; enquanto que um limite alto de agrupamento (.95) exclui *reads* com muitos sítios indeterminados ou afetará o agrupamento.

Exemplo: 4 ## 9. maxN: máximo de 4 bases N em cada *read*

Linha 10: Limiar de similaridade usado pelo vsearch para o agrupamento de sequências.

Este valor é usado tanto para agrupamento intra- quanto entre-amostras. Durante o agrupamento, as bases de restrição da enzima permanecem nas *reads*, então todos os locos terão um conjunto de 4-6 bases idênticas em uma extremidade. Para uma *read* de 100 pb contendo um sítio de 5 pb e com um *barcode* de 5 pb retirado, teremos 95 pb de dados, sendo 5 pb idênticos. Então se você quer permitir 5 bases de diferença entre as *reads* (90/95 = .947), você deve usar um valor de .94. Regiões repetidas são mascaradas usando como padrão o método 'dust' (veja mais na linha 35)

Exemplo: .90 ## 10. limiar de agrupamento de 0.90

Linha 11: Tipo de biblioteca utilizada.

Há seis opções possíveis, atualmente. O tipo deve ser colocado em letras minúsculas: rad, ddrad, gbs, pairddrad, pairgbs, merge.

Exemplos: rad ## 11. Dados de RAD-Seq

pairgbs ## 11. Dados de GBS *paired-end*

Linha 12: Cobertura mínima de táxons.

Indica o número mínimo de amostras para um determinado loco para esse loco permanecer no conjunto final de dados. Se você entrar um número igual ao número total de amostras do seu conjunto então só locos com todas as amostras serão retidos. Se colocar um valor baixo, a matriz terá bastante *missing data*. Adotei mínimo de 90% de amostras por loco, ou seja, 43 amostras.

Exemplo: 43 ## 12. cobertura mínima de 43 amostras

Linha 13: Número (ou proporção) máximo de sítios polimórficos compartilhados em um loco.

Entre um número, ou um decimal com o prefixo “p” (por exemplo p.10 para 10%). Esta opção é usada para detectar parálogos potenciais, pois um sítio heterozigoto compartilhado entre muitas amostras pode representar o agrupamento de parálogos com uma diferença fixada em vez de um verdadeiro sítio heterozigoto. Usei máximo de 10% do número de amostras, ou seja, 4,8 e assim, adotei máximo de 5 sítios polimórficos compartilhados por loco.

Exemplo: 5 ## 13. máximo de 5 sítios heterozigotos compartilhados

Linha 14: Nome para os arquivos *output* finais.

Geralmente colocar além do nome, alguns dos parâmetros citados acima.

Exemplo: arabidopsis_c90d5m4p4 ## 14. nome do *output* é arabidopsis_c90d5m4p4

Linhas 15 a 37 – opcionais

Linha 15: Seleção de subconjunto para os passos 2-7.

Se, por exemplo, você acha que sua amostragem inclui dois clados diferentes e você nomeou todas as amostras de um dos clados começando com a letra "A", você pode selecionar apenas essas amostras para análise, entrando com o prefixo "A*" nesta linha.

Exemplo: A* ## 15. seleciona o subconjunto de arquivos com nome iniciando com "A*

Linha 16: Indicação de grupo externo.

O parâmetro na linha 12 indica ao pyRAD para manter apenas locos com dados de pelo menos N amostras. Para maximizar a cobertura dentro de um certo clado você pode indicar os outros táxons como sendo do "grupo externo" e assim, apenas locos com o mínimo de cobertura do grupo interno serão mantidos e qualquer combinação com o "grupo externo" será incluído como dado, mas não levado em conta no número mínimo de amostras. Liste os nomes das amostras do grupo externo aqui separado por vírgulas.

Exemplo: outg1,outg2 ## 16. selecionar grupo externo para o passo 7

Linha 17: Exclusão de táxons.

Ao montar o conjunto de dados no passo 7, você pode excluir táxons visando maximizar a cobertura dentre as amostras incluídas. Por exemplo, você pode querer excluir amostras que tem pouco dado, ou excluir o grupo externo para análises populacionais. Liste o nome das amostras separados por vírgulas.

Exemplo: sample3,sample4 ## 17. excluir amostras 3 e 4 no passo 7

Linha 18: Localização dos arquivos fastq separadas por *barcode*.

Se seus dados já estão separados em arquivos diferentes por *barcode* você pode pular o passo 1 e ir direto para o passo 2. Mas você precisa indicar a localização dos seus arquivos fastq separados nesta linha. As *reads* devem estar sem o *barcode*. Se o sítio de corte tiver sido retirado então coloque o simbolo "@" no começo da linha. Arquivos podem estar

compactados ou não. No caso de *reads* R1 e R2 que foram consolidadas com os scripts de PEAR, a localização dos novos arquivos gerados para serem usados a partir do passo 2 deve ser indicada aqui.

Exemplos: ~/RAD/fastqc/*.fastqc ## 18. arquivos separados por amostra

@~/RAD/fastqc/*.fastqc ## 18. arquivos separados por amostra e sem sítio de restrição

~/RAD/fastqc/*.assembled.fastqc ## 18. arquivos separados por amostra e após consolidação de R1 e R2 pelo PEAR.

Linha 19: Número máximo de *mismatches* em um *barcode* durante a separação por amostra.

O padrão é 1, não é necessário ajustar.

Exemplo: 1 ## 19. máximo de 1 *mismatch* no *barcode*

Linha 20: Índice Phred de qualidade (usualmente 33, às vezes 64).

Base com valor phred abaixo de 20 é convertida a N. Se você quer que esse valor seja mais estrito, pode aumentar o índice. Por exemplo, para aumentar o phred mínimo para 30 indique 43. Se deixar em branco, será considerado phred de 33.

Exemplo: 33 ## 20. phred mínimo de 33 considerado no passo 2

Linha 21: Nível do rigor de filtragem no passo 2.

(0) é padrão e significa nenhuma filtragem para *barcode*, adaptador e sítio de corte, apenas corrige para qualidade de chamada de base.

(1) busca e retira sítios de corte e adaptadores

. (2) busca sítios de corte e adaptadores, mas permite alguns erros, assim forçando uma filtragem mais rígida. Para a maioria do conjunto de dados que não possuem muitos fragmentos com sobreposição e curtos, as opções 0 e 1 são recomendadas. Se você preparou sua biblioteca ao invés de usar o serviço comercial, é recomendado utilizar pelo menos a opção 1 para checar por adaptadores.

Exemplo: 0 ## 21. sem filtragem de *barcode*, adaptador e sítio de corte

Linha 22: Taxa de erro de sequenciamento *a priori* e heterozigosidade.

O passo 4 da análise estima conjuntamente a taxa de erro e a heterozigosidade, e o passo 5 usa esses valores para fazer a chamada de base. Se por algum motivo você deseja ignorar a

estimativa da taxa de erro (por exemplo, suas amostras são poliplóides), você pode incluir uma estimativa *a priori* da taxa de erro aqui. Essa taxa pode ser estimada baseada em de dados de um parente diplóide, por exemplo. Mas cuidado: geralmente é melhor o programa estimá-la no passo 4, pois seus dados passaram por alguns filtros antes de estimar a sua taxa de erro e as taxas de erro baseadas em outros dados podem ser incorretas. Dependendo do comprimento e da química da sequência tipicamente a taxa de erro é de 0.0001 – 0.002.

Exemplo: 0.0005,0.001 ## 22. Taxa de erro e heterozigosidade, respectivamente

Linha 23: Número máximo de Ns na sequência consenso.

Agrupar entre amostras pode ser afetado pelo número de Ns, e este número deve ser escolhido de acordo com a linha 6 e o comprimento das *reads*. O padrão é 4.

Exemplo: 4 ## 23. máximo de 4 N na sequência consenso

Linha 24: Número máximo de Hs (sítios heterozigotos) na sequência consenso.

Entre diversos métodos para excluir potenciais parálogos ou regiões altamente repetitivas do genoma, você pode adotar um valor máximo de sítios heterozigotos na sequência consenso. O padrão é 4.

Exemplo: 4 ## 24. máximo de 4 Hs na sequência consenso

Linha 25: Ploidia (opções = 1,2, ou qualquer número acima de 2).

Padrão = 2 permite ter apenas 2 haplótipos em uma sequência consenso após correção devido a erros de sequenciamento e exclui sequências consensos com mais de 2 alelos.

Exemplo: 2 ## 25. diplóide, filtro máximo de haplótipos 2

Linha 26: Número máximo de SNPs em um loco final.

Isto pode remover os efeitos potenciais de alinhamento ruim em regiões repetitivas do conjunto de dados final por excluir locos com mais de *n* SNPs. Para dados pareados você pode colocar o máximo para cada *read* R1 e R2 separadamente, ou usar um valor para o par.

Exemplos: 10 ## 26. máximo de 10 SNPs no loco final

8,12 ## 26. Máximo de 8 e 12 SNPs na R1 e R2,leitura, respectivamente

Linha 27: Número máximo de inserções/deleções.

Se indicar apenas um valor, ele se aplica ao agrupamento intra-amostra. Se indicar dois valores, o primeiro será intra-amostra e o segundo entre amostras. Para dados pareados, inclua

quatro valores: os dois primeiros serão os intraR1 e intraR2 e os dois últimos serão os entreR1 e entreR2. O padrão é 3,6,99,99.

Exemplos: 3 ## 27. máximo de 3 *indels* em agrupamentos intra-amostra

3,10 ## 27. máximo de 3 *indels* intra-amostra e de 10 *indels* máximo entre-amostras

3,6,10,10 ## 27. Máximo de 3 *indels* intraR1, máximo de 6 *indels* intraR2, máximo de 10 *indels* entreR1, máximo de 10 *indels* entreR2

Linha 28: Número aleatório de entrada de *reads*.

A ordem de *input* das *reads* é randomizada antes do agrupamento nos passos 3 e 6. Isto pode afetar (geralmente minimamente) os resultados. Para *outputs* de SNPS não-ligados (.usnps, .treemix, .str, .geno) um SNP é aleatoriamente amostrado de cada loco variável no passo 7. A entrada padrão é 112233, assim, as análises são repetíveis por padrão. Você pode mudar para ter diferentes amostragens.

Exemplo: 998877 ## 28. número aleatório de alimentação

Linha 29: Permite extremidades simples fita das *reads* no conjunto final de dados.

Se as *reads* tem comprimentos ou sobreposição em diferentes graus, 1,1 vai retirar a menor *read* em cada extremidade do loco; 0,1 vai retirar a da extremidade 3'; 0,0 permite que ambas extremidades tenham fitas simples. Para dados pareados 1,1,1,1 vai retirar fitas simples nas extremidades 5'R1, 3'R1, 5'R2, 3'R2.

Exemplo: 1,1 ## 29. aparar extremidades do loco

Linha 30: Formatos de *outputs*.

* seleciona todas possíveis.

Opções = (p,n,a,s,v,u,t,m,k,g).

Indicar as letras separadas por vírgulas. Ver seção sobre os formatos de *outputs* abaixo.

Exemplo: a,n ## 30. formatos de *outputs*

Linha 31: Chamada de consenso simples (base em maioria) em agrupamentos com profundidade menor que a profundidade mínima.

Permite que coloque na linha 6 valor tão baixo quanto 2. Mas ainda faz chamada de base em sítios de alta cobertura.

Exemplo: 1 ## 31. chama base por maioria em sítios de baixa profundidade

Linha 32: Retenção de sequências aparadas maiores que X.

Padrão = 0 = não mantém; essa opção deve ser usada apenas para dados ruins (GBS, ddRAD) gerados devido a problema com o método de seleção de tamanho dos fragmentos da biblioteca que resulta em muitos fragmentos curtos e assim, adaptadores são frequentemente sequenciados. Ao invés de descartar essas *reads* com adaptadores, esta opção retira os adaptadores e mantém a sequência se ela for maior que X.

Exemplo: 0 ## 32. não mantém sequências aparadas

50 ## 32. mantém sequências aparadas maiores que 50

Linha 33: Cobertura máxima em um agrupamento intra-amostra.

O padrão (vazio) corresponde ao maior valor entre o máximo de cobertura média mais duas vezes o desvio padrão da cobertura do agrupamento ou 500. Se você quiser indicar um valor absoluto deve colocar aqui. Ou entrar um valor absurdamente grande para não ter filtragem de máximo de cobertura.

Exemplo: ## 33. opção padrão de máximo de cobertura

50000 ## 33. máxima cobertura de 50000

999999999 ## 33. sem máxima cobertura

Linha 34: Número mínimo de cópias de *reads* não replicadas no agrupamento.

Padrão = 1. As *reads* são colapsadas (*dereplicated*) em uma única *read* e o número de réplicas é registrado. O padrão é não excluir nenhum dado. Mas para análises exploratórias, o mínimo de 2 pode aumentar drasticamente a velocidade do agrupamento intra-amostra. Em geral não é recomendado descartar dados na análise final, mas para conjuntos de dados com grande cobertura, tem pouco efeito nos resultados e melhora bastante o tempo de agrupamento.

Exemplo: 2 ## 34. exclui únicos

5 ## 34. exclui *reads* que ocorrem menos que 5 vezes

Linha 35: Agrupamento hierárquico.

Padrão = No = 0. Ver a seção de agrupamento hierárquico. Isto pode ser útil para acelerar agrupamento entre amostras, e também melhorar agrupamentos de parentes próximos quando envolve um conjunto de dados contendo muitos não aparentados.

Exemplo: 1 ## 35. usa agrupamento hierárquico

Linha 36: Identificação de sequências repetitivas.

Padrão = 1 = usa o método padrão de identificação de sequ6encias repetitivas (*repeat masking*) em vsearch/usearch. Se colocar 0 então nenhuma busca será realizada. Vsearch usa o método de *'dust' masking* enquanto o usearch tem um método customizado.

Exemplo: 0 ## 36. desliga a busca por sequências repetitivas

Linha 37: Número máximo de trabalhos (*threads*) em paralelo por agrupamento usado pelo vsearch.

Padrão = 6, mas ainda não foi otimizado. É mais rápido que usearch mesmo quando o número de processadores (*cores*) X *threads* é maior do que o número disponível de processadores, apesar de usar mais memória.

Exemplo: 6 ## 37. 6 trabalhos em paralelo de vsearch

Linhas restantes – atribuição a grupo/clado

Usado para arquivos de *outputs* de populações (ex. treemix e migrate-n) e para agrupamento hierárquico. Cada grupo é listado em uma linha separada, pode deixar linhas em branco. Cada linha deve conter três elementos separados por espaço ou tab:

(1) nome do grupo (por exemplo, A ou 1), esses nomes são arbitrários, apenas use um nome diferente para cada grupo;

(2) o tamanho mínimo de um clado dentro desse agrupamento, qualquer clado menor do que esse tamanho não será agrupado ou não constará no output;

(3) o nome dos indivíduos/amostras neste grupo. Nomes de amostras podem ser individualmente listadas e separadas por vírgulas, ou usando um *.

Exemplos: A 4 1A0, 1B0, 1C0, 1D0

B 4 2*

C 4 3*

Arquivo params.txt final

Depois de conferir a função de cada linha e como os parâmetros podem ser colocados, utilizei o arquivo abaixo para meus dados. Ele pode ser visualizado e editado em editor de texto ou no terminal.

```
params.txt ×
==** parameter inputs for pyRAD version 3.0.66  **======================== affected step ==
./                                                  ## 1. Working directory (all)
/home/biabueno/Documents/Sequences/*.fastq          ## 2. Loc. of non-demultiplexed files (if not line 18)  (s1)
/home/biabueno/pyrad/barcodes.txt                   ## 3. Loc. of barcode file (if not line 18)             (s1)
vsearch                                             ## 4. command (or path) to call vsearch (or usearch)    (s3,s6)
/home/biabueno/muscle3.8.31_i86linux64              ## 5. command (or path) to call muscle                  (s3,s7)
TGCAG                                               ## 6. Restriction overhang (e.g., C|TGCAG -> TGCAG)     (s1,s2)
25                                                  ## 7. N processors (parallel)                           (all)
6                                                   ## 8. Mindepth: min coverage for a cluster              (s4,s5)
6                                                   ## 9. NQual: max # sites with qual < 20 (or see line 20)(s2)
.85                                                 ## 10. Wclust: clustering threshold as a decimal        (s3,s6)
merged                                              ## 11. Datatype: rad,gbs,pairgbs,pairddrad,(others:see docs)(all)
45                                                  ## 12. MinCov: min samples in a final locus             (s7)
5                                                   ## 13. MaxSH: max inds with shared hetero site          (s7)
pyriglena_c85d6m4p3                                 ## 14. Prefix name for final output (no spaces)         (s7)
==== optional params below this line ==================================  affected step ==
                                                ## 15.opt.: select subset (prefix* only selector)            (s2-s7)
                                                ## 16.opt.: add-on (outgroup) taxa (list or prefix*)         (s6,s7)
                                                ## 17.opt.: exclude taxa (list or prefix*)                   (s7)
/home/biabueno/pyrad/fastq/*.assembled.fastq    ## 18.opt.: loc. of de-multiplexed data                      (s2)
0                                               ## 19.opt.: maxM: N mismatches in barcodes (def= 1)          (s1)
                                                ## 20.opt.: phred Qscore offset (def= 33)                    (s2)
                                                ## 21.opt.: filter: def=0=NQual 1=NQual+adapters. 2=strict   (s2)
                                                ## 22.opt.: a priori E,H (def= 0.001,0.01, if not estimated) (s5)
                                                ## 23.opt.: maxN: max Ns in a cons seq (def=5)               (s5)
                                                ## 24.opt.: maxH: max heterozyg. sites in cons seq (def=5)   (s5)
                                                ## 25.opt.: ploidy: max alleles in cons seq (def=2;see docs) (s4,s5)
                                                ## 26.opt.: maxSNPs: (def=100). Paired (def=100,100)         (s7)
                                                ## 27.opt.: maxIndels: within-clust,across-clust (def. 3,99) (s3,s7)
                                                ## 28.opt.: random number seed (def. 112233)              (s3,s6,s7)
                                                ## 29.opt.: trim overhang left,right on final loci, def(0,0) (s7)
*                                               ## 30.opt.: output formats: p,n,a,s,v,u,t,m,k,g,* (see docs) (s7)
                                                ## 31.opt.: maj. base call at depth>x<mindepth (def.x=mindepth) (s5)
50                                              ## 32.opt.: keep trimmed reads (def=0). Enter min length.    (s2)
9999999999                                      ## 33.opt.: max stack size (int), def= max(500,mean+2*SD)    (s3)
                                                ## 34.opt.: minDerep: exclude dereps with <= N copies, def=1 (s3)
                                                ## 35.opt.: use hierarchical clustering (def.=0, 1=yes)      (s6)
                                                ## 36.opt.: repeat masking (def.=1='dust' method, 0=no)      (s3,s6)
24                                              ## 37.opt.: vsearch max threads per job (def.=6; see docs)   (s3,s6)
==== optional: list group/clade assignments below this line (see docs) ==================
1       10      15362,16545,16548,16550,16551,16552,16553,16554,16555,16556
2       3       17112,17113,17114
3       15      17921,17922,17923,17924,17925,17926,17927,17928,17929,17930,17931,17934,17935,17936,17937
4       10      16685,16702,16704,16705,16706,16707,16708,16709,16710,16711
5       3       MMC232,MMC233,MMC234
6       7       MMC222,MMC223,MMC224,MMC225,MMC226,MMC228,MMC230
Texto sem formatação ⌄   Largura da tabulação: 8 ⌄   Lin 24, Col 118   INS
```

4. Rodando pyRAD

pyRAD pode ser chamado a partir do se próprio diretório (isto é, com os seus outros arquivos .py), ou você pode criar um *link* simbólico para chamar o programa de qualquer lugar. Por exemplo, o comando abaixo criaria um *link* para chamar o pyRAD ao digitar 'pyRAD':

$ ln -s ~/pyrad_v.3.0/pyRAD pyRAD

Se o *link* simbólico parece confuso então pode chamar o pyRAD de seu diretório. Por exemplo, para chamar o pyRAD que está no home/directory digite o comando a seguir:

$ ~/pyrad_v.3.0/pyRAD -h

E se você criou um *link* simbólico digite:

$ pyRAD -h

Existem seis opções para a linha de comando do pyRAD, que devem ser indicadas para que o *script* receba o comando da atividade que irá realizar:

-h (tela de ajuda)

-n (gera um novo arquivo de parâmetros)

-p (nomeia arquivo de parâmetros)

-s (seleciona os passos)

-d (arquivo de entrada do D-test)

-D (gera novo arquivo de entrada D-test)

O arquivo de entrada de parâmetros (params.txt) deve ser criado usando a opção -n e depois

preenchido de acordo com sua análise. Se o arquivo de parâmetros está devidamente preenchido, toda a análise - conversão de dados RADseq brutos em uma montagem *de novo* - pode ser feita digitando:

$ pyrad -p params.txt

Isto inicializará todos os passos (1-7) da análise. Mas cuidado, é melhor executar apenas um ou alguns passos de cada vez, utilizando a opção -s (passos). Por exemplo, para selecionar apenas o primeiro passo:

$ pyrad -p params.txt -s 1

Para rodar os passos 2-7, digite:

$ pyrad -p params.txt -s 234567

Se quiser rodar novamente o passo 7 com novos parâmetros, digite:

$ pyrad-p params.txt -s 7

A opção -s é muito útil, pois interrompe a análise de acordo com os passos que são realizados sequencialmente. Desta forma, é possível avaliar o resultado do passo 1 antes de continuar para o passo 2. Se então o passo 2 falhar, você guardou os resultados do passo 1 e pode começar de novo a partir do passo 2, onde surgiu o erro. É recomendado utilizar os conjuntos de dados de exemplo (tutoriais) para familiarizar-se com os passos da análise, os arquivos de saída e as estatísticas antes de iniciar a sua própria análise.

5. Passo 1 – separando por *barcode*

Esta etapa usa as informações do arquivo de *barcode* para separar as *reads* de seus arquivos fastq brutos em arquivos separados para cada amostra. Estes são colocados em um novo diretório, dentro do seu diretório de trabalho, chamado "fastq/". Os nomes dos arquivos não são importantes para os dados finais individuais. No entanto, para *reads* pareadas é necessário que os nomes de arquivos de dados brutos sigam um formato específico: o primeiro arquivo deve conter "_R1_" no nome, e o segundo arquivo deve ter nome idêntico ao do primeiro arquivo, mas com " _R2_" no lugar de "_R1_ ". Relembrando que se tiver as *reads* de cada amostra devidamente separadas pode pular essa etapa. Aqui está um exemplo de como devem ficar os nomes dos arquivos:

yourfilename_R1_001.fastq

yourfilename_R2_001.fastq

No terminal, dar o comando para o passo 1 ($ pyrad -p params.txt -s 1) e aparecerá no terminal:

$ pyrad -p params.txt -s 1

```
-------------------------------------------------------------
 pyRAD : RADseq for phylogenetics & introgression analyses
-------------------------------------------------------------
   step 1: sorting reads by barcode
    ..............................................................
```

Quando terminar, você pode digitar o comando para ver o resultado ou entrar no diretório 'stats" e ver o arquivo s1.sorting.txt.

$ head -n 100 analysis_pyrad/stats/s1.sorting.txt

A seguir mostro as primeiras linhas do meu arquivo s1.sorting.txt. Nas primeiras duas linhas estão os dados sobre o conteúdo do meu arquivo. Em seguida estão as 48 amostras com seus respectivos *barcodes* e os números de *reads*. Em seguida estão dados de *reads* descartadas por não possuírem *barcode*, nomeadas como 'nomatch' com as suas sequências iniciais e número de ocorrências. Esse arquivo continua por mais 3724 linhas.

```
s1.sorting.txt ×
file                Nreads  cut_found       bar_matched
GBS_Aves_TAAGGCGA_L008_.fastq   238084259       232381518       230659373


sample  true_bar        obs_bars        N_obs
MMC234          AACCA           AACCA   1334751
16709           AACGTG          AACGTG  4975009
MMC226          AAGTCA          AAGTCA  3714134
16545           ACATA           ACATA   5200681
16706           ACATCG          ACATCG  4193911
17935           ACATGCA         ACATGCA 1387088
16553           ACGAG           ACGAG   10314885
MMC228          ACTGCA          ACTGCA  4362395
17924           AGAATCA         AGAATCA 5678080
17927           AGGTCCA         AGGTCCA 3054900
16556           AGTCG           AGTCG   7007621
17937           ATGAACA         ATGAACA 3105267
16685           ATGGCG          ATGGCG  6166641
17921           CACTAG          CACTAG  9848892
MMC224          CAGAT           CAGAT   7787303
17925           CATCGCA         CATCGCA 4381860
MMC232          CCACG           CCACG   3426809
16710           CCTAAG          CCTAAG  3842830
17929           CCTGTCA         CCTGTCA 2933336
16708           CGCTGA          CGCTGA  4090827
16551           CGGCA           CGGCA   4815696
MMC222          CGTACA          CGTACA  2074297
16548           CTCAG           CTCAG   3008330
MMC230          CTGCCA          CTGCCA  2461929
17936           CTTGCAA         CTTGCAA 4122018
16552           GAAGA           GAAGA   7942362
17928           GAATACA         GAATACA 1811989
15362           GAGCG           GAGCG   8501043
16705           GATCCA          GATCCA  2613562
16554           GCCAA           GCCAA   6958470
16702           GGACTA          GGACTA  6002785
17934           GGCCTCA         GGCCTCA 1702205
17114           GGCGG           GGCGG   10974271
16704           GTAACA          GTAACA  2666893
17923           GTGGCCA         GTGGCCA 6389374
17113           GTTCA           GTTCA   5076777
17112           TAATG           TAATG   9701247
17931           TAGACCA         TAGACCA 1251375
MMC233          TATAA           TATAA   3111203
17926           TCCGACA         TCCGACA 5436825
16707           TCGCAA          TCGCAA  3929917
17922           TCTAGCA         TCTAGCA 6179945
16550           TCTGG           TCTGG   3830576
MMC223          TGATCA          TGATCA  6761282
16711           TTACGA          TTACGA  3401841
17930           TTCCGCA         TTCCGCA 2993805
MMC225          TTCGCA          TTCGCA  3657030
16555           TTCTA           TTCTA   6475106

nomatch                         5702741
```

6. PEAR – montagem de *reads* R1 e R2

A montagem das *reads* R1 e R2 é essencial para prosseguir a análise e para isso pode ser utilizado o PEAR. Em geral, a maioria das *reads* pareadas apresenta sobreposição e o arquivo gerado será utilizado para os próximos passos da análise do pyRAD.

A primeira etapa é descompactar os arquivos. Se você separou seus dados com pyRAD, então eles estarão no diretório 'fastq'

No terminal: $ gunzip analysis_pyrad/fastq/*.gz ## seleção dos arquivos com extensão .gz que serão descompactados

Em seguida, chame o PEAR, indique os arquivos R1 e R2 e informe como o *output* será nomeado. Os argumentos necessários são:

-f <str> (especifica o nome do arquivo que contém as *reads* R1)

-r <str> (especifica o nome do arquivo que contém as *reads* R2)

-o <str> (especifica os nomes dos arquivos de *output.* São 4 arquivos: um com as sequências montadas com extensão "assembled.fastq", dois com as *reads* não montadas sendo a primeira de R1 e a segunda de R2 com extensão "unassembled.forward.fastq" e "unassembled.reverse.fastq" e um arquivo com as *reads* descartadas com extensão "discarded.fastq")

Os argumentos opcionais são:

-p <str> (especifica um valor p para o teste estatístico. Se o valor de p calculado de uma montagem excede ao valor de p especificado, a sequência consolidada não será montada. As opções válidas são: 0,0001, 0,001, 0,01, 0,05 e 1,0. Configuração 1.0 desabilita o teste. Padrão: 0,01.)

-v <int> (especifica o tamanho mínimo de sobreposição. A sobreposição mínima pode ser definida como 1 quando o teste estatístico é utilizado. No entanto, ao restringir ainda mais o tamanho mínimo de sobreposição pode reduzir montagem de falsos-positivos. Padrão: 10)

-m <int> (especifica o comprimento máximo das sequências montadas. Definir esse valor como 0 desativa a restrição e sequências montadas podem ser arbitrariamente longas. Padrão: 0)

-n <int> (especifica o comprimento mínimo possível das sequências montadas. Definir esse valor como 0 desativa a restrição e sequências montadas podem ser arbitrariamente curtas. Padrão: 50)

-t <int> (especifica o comprimento mínimo de leituras após o corte da parte de baixa qualidade -ver '-q'. Padrão: 1)

-q <int> (especifica o limite de qualidade para aparar segmento com baixa qualidade de uma *read*. Se a qualidade de duas bases consecutivas for inferior ao limiar especificado, o restante da *read* será aparada. Padrão: 0)

-u <float> (especifica a proporção máxima de bases não chamados em uma leitura. Definir esse valor como 0 descartará todas as leituras contendo bases não chamadas. Adotando 1, todas as *reads* serão processadas independente do número de bases não chamadas. Padrão:1)

-g <int> (especifica o tipo de teste estatístico a ser usado para testar se as *reads* R1 e R2

podem ou não ser combinadas. O padrão é usar o teste 1). Duas opções de teste estão disponíveis:

1. Considerando a mínima sobreposição necessária, será utilizada uma taxa de alinhamento observado pelo esperado (*observed expected alignment score*, OES) mais elevada para ser testada. Note que, devido à sua natureza discreta, este teste geralmente produz um valor de p inferior para a sequência montada do que o especificado. Por exemplo, se for definido p=0,05 a sequência montada pode ter p=0,02.

2. Estimativa do máximo *a posteriori* (*maximum a posteriori estimation*, m.a.p). Este método assume que a sobreposição mínima é a sobreposição observada com o maior valor de OES, ao invés daquela especificada pelo -v. Portanto, este não é um teste estatístico válido e o valor de p é na verdade a probabilidade máxima para aceitar a montagem. No entanto, observou-se que em casos de tamanhos reais de sobreposição relativamente pequenos esse teste permite montar corretamente mais *reads* com um pouco mais de falsos-positivos.

-e (desativa frequências de base empíricas. Padrão: utilização de frequências de base empírica.)

-e <int> (especifica o método de pontuação. Padrão: 2);

1. OES com +1 para o compatibilidade (*match*) e -1 para incompatibilidade (*mismatch*).
2. Pontuação de montagem (Assembly score - AS). Use +1 para a compatibilidade e -1 para incompatibilidade multiplicado pelos índices de qualidade de base.
3. Ignorar índices de qualidade e usar +1 para uma compatibilidade e -1 para uma incompatibilidade.

-b <int> (índice Phred de qualidade da base. Padrão: 33.)

-y <str> (especifica a quantidade de memória a ser usada. O número pode ser seguido por uma das letras K, M, ou L denotando kilobytes, megabytes e gigabytes, respectivamente. Bytes são assumidos caso nenhuma letra seja especificada.)

-j <int> (Número de trabalhos [*threads*] a serem usados)

Obs..: A sigla entre os símbolos “< >” informa o tipo de valor que deve ser colocado após a letra:

<str> = string, conjunto de letras. Ex: pyriglena_R1.fastq

<int> = inteiro, valores inteiros. Ex: 1

<float> = variável, aceita valores decimais. Ex: 0.1

Para meus dados utilizei apenas os argumentos necessários (indiquei os dois arquivos que

deveriam ser compilados e o nome do arquivo resultante), deixando os opcionais com seus valores padrões (*default*). No terminal, digitei:

$ ./PEAR/src/pear – f pyrad/fastq/filename_R1.fq -r pyrad/fastq/filename_R2.fq -o pyrad/fastq/filename

```
[biabueno@lgema ~]$ ./PEAR/src/pear -f pyrad/fastq/15362_R1.fq -r pyrad/fastq/15362_R2.fq -o pyrad/fastq/15362
 ____  _____      _    ____
|  _ \| ____|  / \  |  _ \
| |_) |  _|   / _ \ | |_) |
|  __/| |___ / ___ \|  _ <
|_|   |_____/_/   \_\_| \_\

PEAR v0.9.10 [May 30, 2016]

Citation - PEAR: a fast and accurate Illumina Paired-End reAd mergeR
Zhang et al (2014) Bioinformatics 30(5): 614-620 | doi:10.1093/bioinformatics/btt593

Forward reads file.................: pyrad/fastq/15362_R1.fq
Reverse reads file.................: pyrad/fastq/15362_R2.fq
PHRED..............................: 33
Using empirical frequencies........: YES
Statistical method.................: OES
Maximum assembly length............: 999999
Minimum assembly length............: 50
p-value............................: 0.010000
Quality score threshold (trimming).: 0
Minimum read size after trimming...: 1
Maximal ratio of uncalled bases....: 1.000000
Minimum overlap....................: 10
Scoring method.....................: Scaled score
Threads............................: 1

Allocating memory..................: 200,000,000 bytes
Computing empirical frequencies....: DONE
  A: 0.240789
  C: 0.251272
  G: 0.284911
  T: 0.223029
  1079943 uncalled bases
Assemblying reads: 100%

Assembled reads ...................: 7,658,795 / 8,501,043 (90.092%)
Discarded reads ...................: 913 / 8,501,043 (0.011%)
Not assembled reads ...............: 841,335 / 8,501,043 (9.897%)
Assembled reads file...............: pyrad/fastq/15362.assembled.fastq
Discarded reads file...............: pyrad/fastq/15362.discarded.fastq
Unassembled forward reads file.....: pyrad/fastq/15362.unassembled.forward.fastq
Unassembled reverse reads file.....: pyrad/fastq/15362.unassembled.reverse.fastq
[biabueno@lgema ~]$
```

Após a montagem das *reads* R1 e R2 no PEAR o arquivo de parâmetros deve ser modificado. Na linha 11 deve-se trocar 'pairgbs' por 'merged' e na linha 18 deve-se indicar o caminho para o arquivo de sequências compiladas com a extensão '.assembled.fastq'.

7. Passo 2 – filtragem

Esta etapa usa o índice de qualidade Phred registrado nos arquivos fastq para filtrar as chamadas de base de baixa qualidade. Sítios com valores de Phred abaixo do limite são transformados em Ns e *reads* com mais Ns do que o mínimo permitido são descartadas. Os arquivos são gravados no diretório 'edits/' com o sufixo ".edit". Filtros opcionais também podem ser usados.

No terminal, dar o comando para o passo 2:

$ pyrad -p params.txt -s 2

```
    ------------------------------------------------------------
     pyRAD : RADseq for phylogenetics & introgression analyses
    ------------------------------------------------------------
```

```
    sorted .fastq from analysis_pyrad/fastq/*.assembled.fastq being used
    step 2: editing raw reads
    .......................
```

Quando a análise terminar, um arquivo chamado 's2.rawedit.txt' será gerado no diretório "stats/" e pode ser visto em editor de texto comum ou no terminal:

```
$ cat pyrad/stats/s2.rawedit.txt
```

8. Passo 3 – agrupamentos

Este passo retira as *reads* em réplica após a filtragem do passo 2 e registra o número de vezes que cada *read* única é observada. Em seguida as *reads* são agrupadas usando VSEARCH e guarda os valores de similaridade entre elas. Sequências que foram agrupadas são então alinhadas e enviadas para um novo arquivo no diretório "clust.xx/" com a terminação ".clustS.gz".

No terminal, dar o comando para o passo 3:

```
$ pyrad -p params.txt -s 3
  -------------------------------------------------------------
   pyRAD : RADseq for phylogenetics & introgression analyses
  -------------------------------------------------------------
    de-replicating files for clustering...
    step 3: within-sample clustering of 48 samples at
          '.85' similarity. Running 25 parallel jobs
          with up to 24 threads per job. If needed,
          adjust to avoid CPU and MEM limits
    .......................
```

Quando a análise terminar, um arquivo chamado 's3.clusters.txt' será gerado no diretório "stats/" e pode ser visto em um editor de texto comum ou no terminal:

```
$ cat pyrad/stats/s3.clusters.txt
```

9. Passo 4 – taxa de erro e estimativas de heterozigosidade

O passo 4 utiliza a equação de máxima verossimilhança de Lynch (2008) para estimar conjuntamente a taxa de erro de sequenciamento e a heterozigosidade média a partir das frequências de bases em cada sítio em todos conjuntos de *reads* de um indivíduo (considerando o mínimo de cobertura) e utiliza estes valores para calcular a probabilidade

binomial de um sítio ser homozigótico (aa ou bb) ou heterozigótico (ab). Os resultados são gravados no arquivo 'Pi_estimate.txt' no diretório 'stats/'.

No terminal, dar o comando para o passo 4:

```
$ pyrad -p params.txt -s 4
    -------------------------------------------------------------
     pyRAD : RADseq for phylogenetics & introgression analyses
    -------------------------------------------------------------
       step 4: estimating error rate and heterozygosity
       .......................
```

Quando a análise terminar, o arquivo gerado pode ser visto em um editor de texto comum ou no terminal:

```
$ cat pyrad/stats/Pi_E_estimate.txt
```

10. Passo 5 – criar sequências consensos por amostra

Baseado na taxa de erro média e na heterozigosidade estimada no passo anterior, o passo 5 gera sequências consensos para cada agrupamento (amostra). Sequências com menos do que o mínimo de cobertura, mais do que o número máximo de sítios indeterminados (Ns), mais do que o número máximo de sítios heterozigotos (Hs), ou mais do que o número permitido de alelos, são descartadas. Em organismos diplóides se dois alelos estão presentes, eles são mantidos nas sequências consenso.

Você pode querer ajustar:

- cobertura máxima do agrupamento (linha 33)
- número máximo de Ns (linhas 9 e 23)
- número máximo de Hs (linhas 13 e 24)

No terminal, dar o comando para o passo 5:

```
$ pyrad -p params.txt -s 5
    -------------------------------------------------------------
     pyRAD : RADseq for phylogenetics & introgression analyses
    -------------------------------------------------------------
       step 5: creating consensus seqs for 48 samples, using H=0.00694 E=0.00066
       .......................
```

Quando a análise terminar, o arquivo gerado estará no diretório 'stats/' e pode ser visto em um editor de texto comum ou no terminal:

$ cat pyrad/stats/s5.consens.txt

11. Passo 6 – agrupamentos entre amostras

As sequências consenso são agrupadas entre amostras utilizando os mesmos parâmetros utilizados no passo 3. No caso de um indivíduo heterozigoto um de seus alelos é amostrado e usado neste passo aleatoriamente, embora ambos os alelos são retidos no conjunto de dados final.

No terminal, dar o comando para o passo 6:

$ pyrad -p params.txt -s 6

```
-------------------------------------------------------------
 pyRAD : RADseq for phylogenetics & introgression analyses
-------------------------------------------------------------
   step 6: clustering across 48 samples at '.85' similarity
   ........................
```

Nenhum arquivo é criado no diretório 'stats/', mas essa mensagem aparece no terminal:

```
vsearch v1.11.1_linux_x86_64, 62.8GB RAM, 32 cores
https://github.com/torognes/vsearch

Reading file /home/biabueno/pyrad/clust.85/cat.haplos_ 100%
456910277 nt in 3754019 seqs, min 36, max 246, avg 122
Counting unique k-mers 100%
Clustering 100%
Sorting clusters 100%
Writing clusters 100%
Clusters: 148272 Size min 1, max 133, avg 25.3
Singletons: 12339, 0.3% of seqs, 8.3% of clusters

        finished clustering
```

12. Passo 7 – montagem do conjunto final de dados

Aqui é feito o alinhamento, filtragem dos parálogos e montagem dos arquivos *outputs* em formato fasta (.loci). Um grande número de formatos alternativos estão disponíveis (ver abaixo). As estatísticas de montagem final são disponibilizadas no diretório 'stas/' no arquivo com extensão .stats. Este passo é relativamente rápido, e pode ser repetidos com valores diferentes para as opções das linhas 12,13,14,16,17, etc. para criar diferentes conjuntos de dados que podem ser otimizados quanto à cobertura para diferentes amostras.

12.1. Tipos de formatos de saída

12.1.1. Arquivos de sequência

Por padrão o pyRAD fornecerá um conjunto final de dados e os locos excluídos (que não passaram na filtragem do passo 7) em arquivos com extensão, respectivamente, ".loci" e ".excluded_loci"

Estes dados podem ser transformados em formatos adicionais, cada um listado por uma única letra no arquivo de parâmetros:

a: (.alleles): arquivo .loci com dois alelos separados (apenas análises de diplóides).

p: (.phy): arquivo PHYLIP não intercalado que pode ser analisado no raxml.

n: (.nex): arquivo nexus intercalado que pode ser analisado no BEAST.

12.1.2. Arquivo de SNP

s: (.snps): todos os SNPs do conjunto de dados montado.

v: (.vcf): matriz que indica base "ancestral" (0), base "derivada" (1) ou *missing data* (., [ponto final]) de cada SNP para cada amostra/indivíduo.

Obs.: Se não tiver genoma de referência disponível, o pyRAD considera a base mais frequente como referência (ancestral) e as menos como alternativas (derivadas).

12.1.3. Arquivo de SNPs selecionados

u: (.usnps): SNPs não ligados, um SNP aleatoriamente sorteado por loco.

k: (.str): arquivo de SNPs não ligados com formato para o STRUCTURE.

g: (.geno): arquivo de SNPs não ligados com formato para o ADMIXTURE.

12.1.4. Arquivos de sequências para análises populacionais

Dois formatos de arquivos de saída com indivíduos organizados em populações/clados usando as atribuições do arquivo de parâmetros.

t: (.treemix): arquivo para TreeMix de SNPs não ligados com amostras reunidas e filtradas pela cobertura da população.

m: (.migrate): locos para migrate-n com amostras reunidas e filtradas por cobertura da população.

12.2. Rodando o passo 7

É importante conferir se o arquivo de parâmetros foi preenchido depois da linha 37. No meu caso eu poderia dividir a amostragem conforme as localidades (ver tabela a seguir) se já houvesse indicação prévia de que essa é a estrutura populacional.

No.	Localidade	Espécie (N)	LGEMA (sexo)
1	Itaporanga D'ajuda/SE	*Pyriglena atra* (10)	15362 (M), 16545(M), 16548(M), 16550(M), 16551(M), 16552 (F), 16553 (F), 16554 (F), 16555(M), 16556(M)
2	Serra Preta/BA	*Pyriglena atra* (3)	17112 (M), 17113 (M), 17114 (F)
3	Ipirá/BA	*Zona de Hibridização* (15)	17921 (M), 17922 (M), 17923 (M), 17924 (F), 17925 (M), 17926 (M), 17927 (M), 17928 (M), 17929 (F), 17930 (M), 17931 (M), 17934 (M), 17935 (M), 17936 (F), 17937 (M)
4	Itacaré/BA	*Pyriglena leucoptera* (10)	16685 (M), 16702 (M), 16704 (M), 16705 (F), 16706 (F), 16707 (M), 16708 (M), 16709 (M), 16710 (M), 16711 (F)
5	Telemaco Borba/PR	*Pyriglena leucoptera* (3)	MMC232 (M), MMC233 (M), MMC234 (M)
6	Foz do Iguaçu/PR	*Pyriglena leucoptera* (7)	MMC222 (F), MMC223 (M), MMC224 (M), MMC225 (M), MMC226(M), MMC228 (F), MMC230 (M)

Assim, se cada localidade da tabela fosse considerada como uma população, o final do meu arquivo de parâmetros deveria ser mudado para o seguinte:

```
*params.txt ×
==== optional: list group/clade assignments below this line (see docs) ==================
1       10      15362,16545,16548,16550,16551,16552,16553,16554,16555,16556
2       3       17112,17113,17114
3       15      17921,17922,17923,17924,17925,17926,17927,17928,17929,17930,17931,17934,17935,17936,17937
4       10      16685,16702,16704,16705,16706,16707,16708,16709,16710,16711
5       3       MMC232,MMC233,MMC234
6       7       MMC222,MMC223,MMC224,MMC225,MMC226,MMC228,MMC230
```

Obs.: O correto é assumir as populações que forem indicadas pelos próprios dados como, por exemplo, após a análise no STRUCTURE, separar os indivíduos no arquivo 'params.txt' e rodar novamente o passo 7.

No terminal, dar o comando para o passo 7:

$ pyrad -p params.txt -s 7

Como no meu arquivo 'params.txt' eu solicitei todas as extensões possíveis, os seguintes arquivos serão produzidos:

pyriglena_c85d6m4p3.alleles

pyriglena_c85d6m4p3.excluded_loci

pyriglena_c85d6m4p3.gphocs

pyriglena_c85d6m4p3.loci

pyriglena_c85d6m4p3.migrate

pyriglena_c85d6m4p3.nex

pyriglena_c85d6m4p3.phy

pyriglena_c85d6m4p3.phy.partitions

pyriglena_c85d6m4p3.snps

pyriglena_c85d6m4p3.snps.geno

pyriglena_c85d6m4p3.str

pyriglena_c85d6m4p3.treemix.gz

pyriglena_c85d6m4p3.unlinked_snps

pyriglena_c85d6m4p3.usnps.geno

pyriglena_c85d6m4p3.vcf

Além de gerar os formatos de saída, o passo 7 gera um arquivo com extensão '.stats' no diretório 'stats'. Com meus dados criei o 'pyriglena_c85d6m4p3.stats'. Para visualizar o arquivo pode usar um editor de texto comum ou no terminal: $ cat pyrad/stats/pyriglena_c85d6m4p3.stats

Obs.: Para conferir o resultado do pyRAD verifique se seu arquivo '.vcf' possui muitos './.' (ponto final, *missing data*). A presença de muito *missing data* (ver exemplo abaixo) significa que poucos indivíduos tem o mesmo sítio representado e esses dados podem não ser informativos. Nesse caso, altere a linha 12 aumentando o número mínimo de amostras representadas.

Exemplo de arquivo .vcf que indica que os dados estão ruins:

```
pyriglena_c85d6m4p3.vcf ×
##fileformat=VCFv4.1
##fileDate=20160729
##source=pyRAD.v.3.0.66
##reference=common_allele_at_each_locus
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO            FORMAT  15362.assembled 16545.assembled 16548.assembled 16550.assembled
16551.assembled 16552.assembled 16553.assembled 16554.assembled 16555.assembled 16556.assembled 16685.assembled 16702.assembled 16704.assembled
16705.assembled 16706.assembled 16707.assembled 16708.assembled 16709.assembled 16710.assembled 16711.assembled 17112.assembled 17113.assembled
17114.assembled 17921.assembled 17922.assembled 17923.assembled 17924.assembled 17925.assembled 17926.assembled 17927.assembled 17928.assembled
17929.assembled 17930.assembled 17931.assembled 17934.assembled 17935.assembled 17936.assembled 17937.assembled MMC222.assembled
MMC223.assembled        MMC224.assembled        MMC225.assembled        MMC226.assembled        MMC228.assembled        MMC230.assembled
MMC232.assembled        MMC233.assembled        MMC234.assembled
1       121     .       A       G       20      PASS    NS=12;DP=6
GT      ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     0|0     1|0     0|0     1|
0       ./.     ./.     ./.     ./.     ./.     0|0     0|0     0|0     0|0     0|0     0|0     ./.     ./.     ./.     ./.     0|
0       ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     0|
0       ./.     ./.     ./.     ./.     ./.     ./.     ./.
1       125     .       G       T       20      PASS    NS=12;DP=6
GT      ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     0|0     0|0     0|0     0|
0       ./.     ./.     ./.     ./.     ./.     0|0     0|0     0|0     1|1     0|0     0|0     ./.     ./.     ./.     ./.     0|
0       ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     0|
1       ./.     ./.     ./.     ./.     ./.     ./.     ./.
1       131     .       A       G       20      PASS    NS=12;DP=6
GT      ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     0|0     1|0     0|0     1|
0       ./.     ./.     ./.     ./.     ./.     0|0     0|0     0|0     0|0     0|0     0|0     ./.     ./.     ./.     ./.     0|
0       ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     0|
0       ./.     ./.     ./.     ./.     ./.     ./.     ./.
1       137     .       C       T       20      PASS    NS=12;DP=6
GT      ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     ./.     0|0     0|0     0|0     0|
0       ./.     ./.     ./.     ./.     ./.     0|0     0|0     0|0     0|0     0|0     1|0     ./.     ./.     ./.     ./.     0|
                                                                                Texto sem formatação  Largura da tabulação: 8  Lin 1, Col 1  INS
```

Exemplo de arquivo .vcf que indica que os dados podem ser bons:

```
pyriglena_c85d6m4p3.vcf ×
##fileformat=VCFv4.1
##fileDate=20160914
##source=pyRAD.v.3.0.66
##reference=common_allele_at_each_locus
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO            FORMAT  15362.assembled 16545.assembled 16548.assembled 16550.assembled
16551.assembled 16552.assembled 16553.assembled 16554.assembled 16555.assembled 16556.assembled 16685.assembled 16702.assembled 16704.assembled
16705.assembled 16706.assembled 16707.assembled 16708.assembled 16709.assembled 16710.assembled 16711.assembled 17112.assembled 17113.assembled
17114.assembled 17921.assembled 17922.assembled 17923.assembled 17924.assembled 17925.assembled 17926.assembled 17927.assembled 17928.assembled
17929.assembled 17930.assembled 17931.assembled 17934.assembled 17935.assembled 17936.assembled 17937.assembled MMC222.assembled
MMC223.assembled        MMC224.assembled        MMC225.assembled        MMC226.assembled        MMC228.assembled        MMC230.assembled
MMC232.assembled        MMC233.assembled        MMC234.assembled
1       27      .       G       A       20      PASS    NS=48;DP=6      GT      0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0
0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     1|1     0|0     1|1     0|1     0|0     0|1
0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     1|1     0|1     0|0     0|0     0|0     0|0     0|0
0|0     0|0     0|0     0|0
1       30      .       C       A       20      PASS    NS=48;DP=6      GT      0|1     0|1     0|1     0|0     1|1     0|0     0|0     1|1
0|0     0|1     1|1     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0
0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0
0|0     0|0     0|0     0|0
3       1       .       T       A       20      PASS    NS=48;DP=6      GT      0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0
0|0     0|0     0|1     0|0     0|0     0|1     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0
0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|1     0|0     0|1     0|1     0|0     0|0
0|0     1|1     0|0     ./.
3       60      .       C       G       20      PASS    NS=48;DP=6      GT      0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0
0|0     0|0     0|0     1|1     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     1|0     0|0     1|0     0|0     0|0
1|0     ./.     1|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0     1|0
0|0     0|0     0|0     0|0
5       11      .       A       G       20      PASS    NS=48;DP=6      GT      0|0     0|0     0|0     0|0     0|0     0|0     0|0     0|0
0|0     0|0     0|0     0|0     1|0     0|0     0|0     0|0     0|0     0|0     0|0     1|0     0|0     0|0     0|0     0|0     0|0     0|0
Texto sem formatação  Largura da tabulação: 8  Lin 5, Col 1  INS
```

13. Dúvidas

Em casa de dúvidas, consulte os tutoriais oficiais feitos pelo autor do programa (ver na 1ª página desse manual) e grupos no Google para esclarecer dúvidas:

Grupo do pyRAD - https://groups.google.com/forum/#!forum/pyrad-users

Grupo do PEAR - https://groups.google.com/forum/#!forum/pear-users

14. Referências

Eaton, D. A. (2014) PyRAD: assembly of de novo RADseq loci for phylogenetic analyses. **Bioinformatics 30**:1844–1849.

Lynch, M. (2008) Estimation of nucleotide diversity, disequilibrium coefficients, and mutation rates from high-coverage genome-sequencing projects. **Molecular Biology and Evolution 25**: 2409–2419.

Zhang, J., Kobert, K., Flouri, T. & Stamatakis, A. (2014) PEAR: a fast and accurate Illumina Paired-End reAd mergeR. **Bioinformatics 30**: 614-620.

ANEXO 3 - Manual LGEMA: Bowtie2

Tradução por Ana Beatriz da Silva Bueno (ana_bia_bueno@hotmail.com) das páginas citadas a seguir (consulta: 23/03/2017) acrescentada de dados.

Página da *internet* do Bowtie2:
http://bowtie-bio.sourceforge.net/bowtie2/index.shtml
Página da *internet* com manual:
http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml
Página da internet para *download* do programa Bowtie2:
https://sourceforge.net/projects/bowtie-bio/files/bowtie2/

1. Introdução

O que é Bowtie 2 (Langmead & Salzberg, 2012)?
Bowtie 2 é uma ferramenta rápida e eficiente em memória para alinhar sequências de *reads* com sequências de referência longas. É particularmente bom para alinhar *reads* de cerca de 50 até 100 ou 1.000 caracteres com genomas relativamente longos. Bowtie 2 atribui ao genoma de referência um Índice FM (baseado na transformação Burrows-Wheeler ou BWT) para manter sua "pegada de memória" pequena. Por exemplo, para o genoma humano a "pegada de memória" é tipicamente cerca de 3,2 gigabytes de RAM. Bowtie 2 suporta alinhamento com *gap*, alinhamento local e entre pares. Vários processadores podem ser usados simultaneamente para alcançar maior velocidade de alinhamento. Bowtie 2 alinha em formato SAM, permitindo a interoperação com um grande número de outras ferramentas (por exemplo SAMtools, GATK) que usam SAM. Bowtie 2 é distribuído sob a licença GPLv3 e é executado por linhas de comando em Windows, Mac OS X e Linux.

2. Obtendo Bowtie 2

Download do pacote em:
https://sourceforge.net/projects/bowtie-bio/files/bowtie2/
Descompactar o arquivo baixado no terminal:
$ unzip bowtie2-2.3.1-linux-x86_64.zip

Execução do programa através do terminal:
$ cd bowtie2-2.3.1/

$ ./bowtie2

OBS: Eu fiz instalação local portanto para rodar o programa tenho que entrar na pasta do *software*, inclusive os dados que utilizei foram colocados na mesma pasta. Para fazer a instalação global conferir o manual dos autores (*link* no início deste Manual).

3. Criando o *index* do genoma (*bowtie2-build indexer*)

Antes de começar o alinhamento propriamente dito, temos que criar arquivos que comporão o *index* do genoma de referência que será utilizado no alinhamento. Depois que o *index* é obtido, o(s) arquivo(s) original(is) FASTA do genoma não será(ão) mais usado(s) pelo Bowtie 2.

Bowtie2-build constrói um *index* Bowtie a partir de um ou mais arquivos FASTA do genoma. Inicialmente o Bowtie2-build produz um conjunto de seis arquivos com sufixos .1.bt2, .2.bt2, .3.bt2, .4.bt2, .rev.1.bt2 e .rev.2.bt2. No caso de um *index* grande (com mais de milhões de pares de base) estes sufixos receberão a terminação bt2l. O conjunto desses arquivos constitui o *index*: somente eles são necessários para alinhar as *reads* com o genoma de referência. O formato de *index* .bt2 do Bowtie 2 é diferente do formato .ebwt do Bowtie 1. E eles não são compatíveis entre si.

Os comandos serão dados no terminal e devem ter a seguinte configuração:

$ ./bowtie2-build [options*] <reference_in> <bt2_base>

Descrição:

./bowtie2-build = comando para iniciar o programa de criação de *index* do Bowtie 2

[options*] = opções e parâmetros que podem ser acrescentados no seu alinhamento

<reference_in> o arquivo FASTA com o genoma de referência a ser usado. Caso tenha mais de um arquivo FASTA, separá-los com vírgulas (por exemplo: chr1.fa,chr2.fa,chrX.fa,chrY.fa). Caso queira buscar determinadas sequências, utilizar a opção -c e colocar as sequências desejadas (por exemplo: -c GGTCATCCT,ACGGGTCGT,CCGTTCTATGCGGCTTA).

<bt2_base> o nome de base dos arquivos *index* para gravar. Por padrão, bowtie2-build gera arquivos nomeados NAME.1.bt2, NAME.2.bt2, NAME.3.bt2, NAME.4.bt2, NAME.rev.1.bt2 e NAME.rev.2.bt2, onde NAME é <bt2_base>.

(Para mais informações e mais detalhes: http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml#the-bowtie2-build-indexer)

Como eu utilizei:

- Como alguns indivíduos de *Pyriglena* que sequenciei estavam infectados com *Plasmodium*, averiguei se haviam *reads* de *Plasmodium* no meu conjunto de *reads*. Assim, baixei um genoma montado de *Plasmodium gallinaceum* em formato FASTA do *site* http://plasmodb.org/plasmo/ (*Plasmodium gallinaceum* é a espécie mais aparentada com genoma completo disponível àquela que infecta o gênero *Pyriglena*). Nomeei esse arquivo como ' Pgallinaceum.fasta' e para que o Bowtie 2 gerasse arquivos nomeados de 'Pgallinaceum' usei a seguinte linha de comando:

$ ./bowtie2-build -f Pgallinaceum.fasta Pgallinaceum

(a opção -f indica que estou apresentando um arquivo de extensão FASTA)

Com esse comando gerei os seguintes arquivos:

Pgallinaceum.1.bt2

Pgallinaceum.2.bt2

Pgallinaceum.3.bt2

Pgallinaceum.4.bt2

Pgallinaceum.rev.1.bt2

Pgallinaceum.rev.2.bt2

- Para separar as *reads* dos indivíduos de *Pyriglena* que sequenciei e que fossem do genoma mitocondrial, utilizei o genoma mitocondrial montado de *Pyriglena leucoptera* por F. Amaral (comunicação pessoal). Nomeei esse arquivo como 'Pyriglenamitocondrial.fasta' e para que o Bowtie 2 gerasse arquivos nomeados como 'Pyriglenamitocondrial' usei a seguinte linha de comando:

$ ./bowtie2-build -f Pyriglenamitocondrial.fasta Pyriglenamitocondrial

(a opção -f indica que estou apresentando um arquivo de extensão FASTA)

Com esse comando gerei os seguintes arquivos:

Pyriglenamitocondrial.1.bt2

Pyriglenamitocondrial.2.bt2

Pyriglenamitocondrial.3.bt2

Pyriglenamitocondrial.4.bt2

Pyriglenamitocondrial.rev.1.bt2

Pyriglenamitocondrial.rev.2.bt2

4. Rodando o programa

A linha de comando base para alinhar as *reads* com o genoma de referência é a seguinte:

bowtie2 [options]* -x <bt2-idx> {-1 <m1> -2 <m2> | -U <r>} -S [<hit>]

Descrição:

bowtie2 = comando para executar o programa

[options]* = opções e parâmetros para seu alinhamentos

-x <bt2-idx> = nome do arquivo de índices; mesmo que tenha vários, só é necessário citar o inicial e não precisa adicionar os sufixos (.1.bt2, .2.bt2, .3.bt2, .4.bt2, .rev.1.bt2 e .rev.2.bt2)

{-1 <m1> -2 <m2> | -U <r>} = arquivos a serem alinhados com os índices

-S [<hit>] = arquivo de saída do formato SAM.

Mais opções e uma descrição mais completa podem ser encontrados em:

http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml#the-bowtie2-aligner

Minha linha básica de comando foi a seguinte:

$./bowtie2 –verysensitive-local -p8 -x tituloíndice -U readsparaalinhar.fasta -S saída.sam –un nãopareadas.fasta –al pareadas.fasta

Minha linha de comando para o alinhamento com genoma de *Plasmodium* foi a seguinte:

$ ./bowtie2 –verysensitive-local -p8 -x Pgallinaceum -U 16709.fasta -S 16709.sam –un 16709naoalinhadas.fasta –al 16709alinhadas.fasta

Descrição:

./bowtie2 = comando para executar o programa Bowtie 2

–verysensitive-local = um dos tipos de alinhamento com parâmetros já definidos

-p8 = número de processadores utilizados no alinhamento (eu utilizei 8)

-x Pgallinaceum = título dos meus arquivos de índices (apenas coloquei Pgallinaceum, não é necessário colocar os sufixos .1.bt2, .2.bt2, .3.bt2, .4.bt2, .rev.1.bt2 e .rev.2.bt2).

-U 16709.fasta = arquivo a ser alinhado com o índice

-S 16709.sam = arquivo de saída SAM

–un 16709naoalinhadas.fasta = as *reads* que não foram alinhadas com o genoma serão listadas nesse arquivo de saída

–al 16709alinhadas.fasta = as *reads* que foram alinhadas com o genoma serão listadas nesse arquivo de saída

O seguinte texto apareceu no meu terminal:

```
[biabueno@lgema bowtie2-2.3.1]$ ./bowtie2 –very-sensitive-local -p8 -x Pgallinaceum -U 16709.sam –un 16709naoalinhadas.fastqc –al 16709alinhadas.fastqc
4641414 reads; of these:
        4641414 (100.00%)  were unpaired; of these:
                4640010 (99.97%) aligned 0 times
                534 (0.01%) aligned exactly 1 time
                870 (0.02%) aligned >1 times
0.03% overall alignment rate
```

Nesse caso o resultado foi um total de 0,03% de alinhamento com o genoma de *P. gallinaceum*, o que o programa considera não significativo e indica como 0% alinhado.

Minha linha de comando para o alinhamento com o genoma mitocondrial foi:

```
$ ./bowtie2 –verysensitive-local -p8 -x Pyriglenamitocondrial -U 17112.fasta -S 17112.sam –un 17112nuclear.fasta –al 17112mitocondrial.fasta
```

Descrição:

./bowtie2 = comando para executar o programa Bowtie 2

–verysensitive-local = um dos tipos de alinhamento com parâmetros já definidos

-p8 = número de processadores utilizados no alinhamento (eu utilizei 8)

-x Pyriglenamitocondrial = título dos meus arquivos de índices (apenas coloquei Pyriglenamitocondrial, não é necessário colocar os sufixos .1.bt2, .2.bt2, .3.bt2, .4.bt2, .rev.1.bt2 e .rev.2.bt2).

-U 17112.fasta = arquivo a ser alinhado com o índice

-S 17112.sam = arquivo de saída SAM

–un 17112nuclear.fasta = as *reads* que não foram alinhadas serão listadas nesse arquivo de saída, ou seja, representam as *reads* do genoma nuclear.

–al 17112mitocondrial.fasta = as *reads* que foram alinhadas serão listadas nesse arquivo de saída, ou seja, representam as *reads* do genoma mitocondrial.

O seguinte texto apareceu no terminal:

```
[biabueno@lgema bowtie2-2.3.1]$ ./bowtie2 –verysensitive-local -p8 -x
Pyriglenamitocondrial -U 17112.fasta -S 17112.sam –un 17112nuclear.fasta –al
17112mitocondrial.fasta
8580657 reads; of these:
        8580657 (100.00%)  were unpaired; of these:
                8580625 (100.00%) aligned 0 times
                32 (0.00%) aligned exactly 1 time
                0 (0.00%) aligned >1 times
0.00% overall alignment rate
```

Nesse caso houve 0,00% de alinhamento com o genoma mitocondrial, portanto não havia mtDNA em meu conjunto de dados.

5. Dúvidas

Caso ocorram dúvidas, sugiro olhar as páginas citadas no início deste Manual para buscar mais informações bem como a leitura do artigo do programa (citado abaixo).

6. Referência

Langmead B, Salzberg S. (2012) Fast gapped-read alignment with Bowtie 2. **Nature Methods 9**: 357-359.

ANEXO 4 - Manual LGEMA: sNMF

Tradução por Ana Beatriz da Silva Bueno (ana_bia_bueno@hotmail.com) das páginas citadas a seguir (consulta: 15/01/2017) acrescentada de imagens, tabelas e dados.

Página da *internet* do sNMF:
http://membres-timc.imag.fr/Olivier.Francois/snmf/index.htm
Página da *internet* com manual:
http://membres-timc.imag.fr/Olivier.Francois/snmf/tutorial.htm
Página da *internet* para *download* do programa sNMF:
http://membres-timc.imag.fr/Olivier.Francois/snmf/software.htm

1. Introdução

A inferência dos coeficientes de ancestralidade individuais, que são importante para estudos genéticos e associativos de populações, é comumente realizada usando algoritmos de probabilidade de uso intensivo do computador. Com a disponibilidade de grandes conjuntos de dados genômicos populacionais, versões rápidas de algoritmos de probabilidade têm atraído atenção considerável. O sNMF (Frichot *et al.* 2014) é um método rápido e eficiente para estimar coeficientes de ancestralidades individuais com base em algoritmos de fatorização de matrizes não negativas esparsas. Sem perda de precisão, o sNMF calcula estimativas de coeficientes de ancestralidade em tempos 10 a 30 vezes mais curtos que os de ADMIXTURE (ver Anexo 5).

2. Instalação

O pacote pode baixado do seguinte *link*:
http://membres-timc.imag.fr/Olivier.Francois/snmf/software.htm
O arquivo baixado tem o seguinte nome: 'sNMF_CL_v1.2.zip'
Em seguida, no terminal, descompacte o arquivo:

```
$ unzip  sNMF_CL_v1.2.zip
$ cd  sNMF_CL_v1.2
$ ./install.command
```

3. Formato dos dados

Dados de entrada: um único arquivo de genótipos no formato '.geno'

OBS: Os formatos '.ped', '.vcf', '.ancestrymap', '.lfmm' podem ser convertidos em arquivos '.geno' pois o sNMF possui em seu pacote *scripts* que realizam essa conversão (mais detalhes em http://membres-timc.imag.fr/Olivier.Francois/snmf/tutorial.htm).

Dados de saída: o sNMF gera dois tipos de arquivos:
- arquivo '.Q': contém os coeficientes individuais de mistura. Ele contém uma matriz com n linhas (número de indivíduos) e K colunas (número de populações ancestrais).
- arquivo '.G': frequência genotípica ancestral. Ele contém uma matriz com naxL linhas (número de alelos vezes o número de SNPs) e K colunas (número de populações ancestrais). Para um SNP diplóide, a primeira linha contém a frequência de alelos igual a 0, a segunda linha contém a frequência ancestral para o número de alelos igual a 1, a terceira linha contém a frequência ancestral para o número de alelos igual a 2.

4. Executando o programa

O sNMF pode ser executado pela linha de comando padrão:

$ ./bin/sNMF -x genotype_file.geno -K n_anc_pop

Todas as opções seguintes são obrigatórias. Não existe uma ordem para a linha de comando.
-x genotype_file.geno = caminho para o arquivo de genótipo ('.geno')
-k 1 = número de populações ancestrais é igual a 1 (tem que ser número inteiro)

Opções adicionais que estão disponíveis:
-a alpha (padrão: 10) = valor do parâmetro de regularização. Os resultados dependem do valor desse parâmetro, especialmente pequenos conjuntos de dados.
-q output_Q ('.K.Q') = caminho para o arquivo de saída contendo os coeficientes de ancestralidade. O padrão é sair com o mesmo nome que o arquivo de entrada.
-g output_G ('.K.G') = caminho para o arquivo de saída contendo as frequências genotípicas das populações ancestrais. Por padrão, o nome da saída será o mesmo de entrada mas com extensão '.K.G'.
- c prec = porcentagem de genótipos mascarados. Indica a porcentagem do total dos SNPs, escolhida aleatoriamente, que será desconsiderada na análise em questão. É uma maneira de avaliar a consistência dos dados (subamostragens dos dados que são analisadas independentemente e no final todos os resultados são considerados). Se essa opção for

definida, o critério de entropia cruzada (ver a seguir) será calculado. A porcentagem padrão é 5%.

-e tolerance = tolerância de erro no algoritmo de otimização do sNMF (padrão: 0,0001).

-i interation_number = número máximo de interações do algoritmo (padrão: 200).

-I nb_SNPs = começa o algoritmo com um subconjunto aleatório de SNPs. Pode agilizar no caso de muitos dados.

-Q input_Q = caminho para um arquivo inicial Q contendo coeficientes de mistura individual. Se ambos -I e -Q são colocados, só -Q é escolhido.

-s seed = semente para iniciar a geração de números aleatórios.

-m ploidy = 1 para haplóide e 2 para diplóide (padrão: 2).

-p = número de CPUs que serão utilizados (padrão: 1).

-h = ajuda; mostra todos os parâmetros.

5. Critério de Entropia Cruzada

Objetivo: o valor de critério de entropia cruzada indica se a predição de genótipos mascarados é boa, ou seja, ajuda a avaliar a qualidade de estimação da população ancestral. Assim, a entropia cruzada vai ajudar a identificar o número de populações ancestrais (K) ou a melhor corrida em um conjunto de corridas.

Programa: o critério é útil para escolher o número de populações ancestrais (K) e os valores do parâmetro de regularização (alpha). Um menor valor de entropia cruzada usando dados mascarados significa uma melhor predição dos dados mascarados.

A linha de comando base é:

```
$ ./bin/cross-Entropy -x genotype_file.geno -K n_anc_pop
```

As opções obrigatórias são:

-x genotype_file.geno = caminho para o arquivo de extensão '.geno'

-K n_anc_pop = número de populações ancestrais

Os arquivos de saída do sNMF, os arquivos com dados mascarados, arquivos originais e arquivos de resultados são armazenados em uma única pasta.

Outras opções:

-m (ploidia, 1 haplóide e 2 diplóide; padrão:2)

6. Analisando os dados

1º Criar dados mascarados:

$ ./bin/createDataSet -x genotype_file.geno

Como eu fiz:

$ ./sNMF/bin/createDataSet -x /pyrad/outfiles/pyriglena.geno

Esse comando criou um arquivo chamado 'pyriglena_I.geno'

2º Analisar os dados:

$ ./bin/sNMF -x examples/panel_I.geno

Como eu fiz: utilizei um *script* com comando *bash* (explicação no próximo item)

3º Computando o critério de entropia cruzada:

$ ./bin/crossEntropy -x examples/panel11.geno -K 5

Como eu fiz: utilizei um *script* com comando *bash* (explicação no próximo item)

7. Utilizando o sNMF de forma automatizada ('snmf_script_loop.sh')

Conteúdo do *script* 'snmf_script_loop.sh':

```
for i in {1..10}
do
for j in {1..100}
do
./sNMF/bin/sNMF -x pyriglena_c85d6m4p3.snps.geno -K ${i} -a 1 -q
pyriglena_a1_${i}_${j}.Q -g pyriglena_a1_${i}_${j}.G -c > pyriglena_a1_${i}_${j}.log
done
done

for i in {1..10}
do
for j in {1..100}
do
./sNMF/bin/sNMF -x pyriglena_c85d6m4p3.snps.geno -K ${i} -a 10 -q
pyriglena_a10_${i}_${j}.Q -g pyriglena_a10_${i}_${j}.G -c > pyriglena_a10_${i}_${j}.log
```

```
done
done

for i in {1..10}
do
for j in {1..100}
do
./sNMF/bin/sNMF -x pyriglena_c85d6m4p3.snps.geno -K ${i} -a 100 -q
pyriglena_a100_${i}_${j}.Q -g pyriglena_a100_${i}_${j}.G -c >
pyriglena_a100_${i}_${j}.log
done
done

for i in {1..10}
do
for j in {1..100}
do
./sNMF/bin/sNMF -x pyriglena_c85d6m4p3.snps.geno -K ${i} -a 1000 -q
pyriglena_a1000_${i}_${j}.Q -g pyriglena_a1000_${i}_${j}.G -c >
pyriglena_a1000_${i}_${j}.log
done
done
```

Descrição:

- a letra *i* corresponde ao K (número de populações ancestrais). Testei K de 1 até 10.

for i in {1..10}

do

- a letra *j* corresponde ao número de repetições para cada valor de K (no meu caso foram 100 repetições)

for j in {1..100}

do

./sNMF/bin/sNMF = localização do arquivo que executa o programa sNMF

-x pyriglena_c85d6m4p3.snps.geno = nome do arquivo '.geno'
-K ${i} = valor do número de populações ancestrais (*i* vai variar de 1 a 10, como é um *script* que se repete várias vezes, a cada rodada o valor de K vai mudar automaticamente).
-a 1 = valor de alfa (repeti o procedimento quatro vezes, pois em cada rodada foi usado um valor diferente de alfa: 1, 10, 100, 1000)
-q pyriglena_a1_${i}_${j}.Q = cria o nome do arquivo "K.Q" relacionando o número da repetição e o número de K utilizado
-g pyriglena_a1_${i}_${j}.G = cria o nome do arquivo "K.G" relacionando o número da repetição e o número de K utilizado
-c > pyriglena_a1_${i}_${j}.log = direciona o resultado do critério de entropia em um arquivo cujo nome está relacionando o número da repetição e o número de K utilizado

Para rodar o *script* descrito acima, abrir o terminal e colocar no diretório onde esse arquivo se encontra e digitar:
$ bash snmf_script_loop.sh

8. 'plot_snmf.R' (parte 1)
Importante: para essa etapa é necessário que o pacote do R esteja devidamente instalado no computador.

```
##############################################################################
###
#                    Script para gerar figuras sNMF                    #
#                    Escrito por: Roberta Damasceno                     #
#                    Adaptação por: Ana Beatriz Bueno                    #
#                    Última versão em 13/02/2017                       #
##############################################################################
###

## Obter valores de entropia cruzada com comandos unix (pode ser feito no terminal)
grep "Cross-Entropy (masked data):" *.log >> pyriglena_cross_entropy.txt

## Limpando o terminal:
rm(list=ls())
```

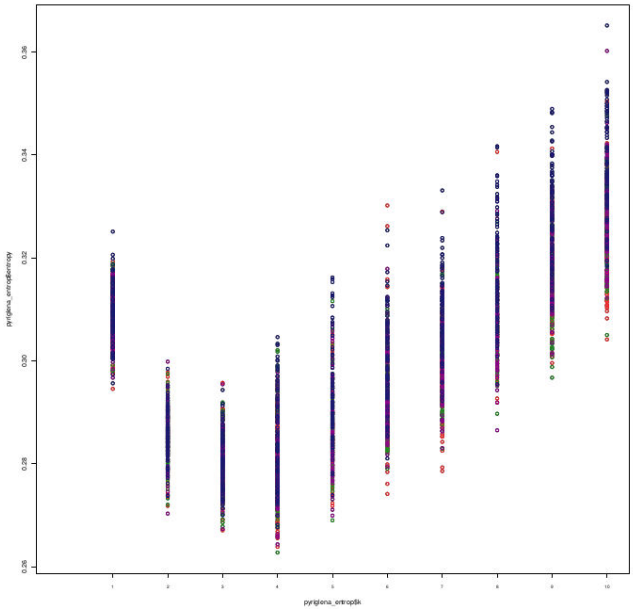
```
## Escolhendo diretório para rodar o script:
setwd ("/home/user/Documentos")

## Lendo os dados:
pyriglena_entrop <- read.table (file = 'pyriglena_cross_entropy.txt', sep = "\t", header =
TRUE)
head (pyriglena_entrop)
dim(pyriglena_entrop)

## Encontrando o menor valor de entropia e a corrida correspondente:
pyriglena_entrop[which(pyriglena_entrop$entropy == min(pyriglena_entrop$entropy)), ]
#       dataset   a k  r  entropy
# 2470 pyriglena a10 4 72 0.262461

# Conferir se nenhum número absurdo foi corrigido (problemas com virgula/ponto)
pyriglena_entrop[which(pyriglena_entrop$entropy == max(pyriglena_entrop$entropy)), ]
#     dataset     a  k  r  entropy
# 90 pyriglena a1000 10 90 0.365883

## Código para diferentes alfas
a1_pyriglena<-pyriglena_entrop[pyriglena_entrop$a=="a1",]
a10_pyriglena<-pyriglena_entrop[pyriglena_entrop$a=="a10",]
a100_pyriglena<-pyriglena_entrop[pyriglena_entrop$a=="a100",]
a1000_pyriglena<-pyriglena_entrop[pyriglena_entrop$a=="a1000",]
head(a1_pyriglena)
head(a10_pyriglena)
head(a100_pyriglena)
head(a1000_pyriglena)
dim(a1_pyriglena)
dim(a10_pyriglena)
dim(a100_pyriglena)
dim(a1000_pyriglena)
```

```
## Plotando todos os valores de entropias para diferentes Ks
pdf("pyriglena_number_ks.pdf", width = 16, height = 16)
plot(pyriglena_entrop$k,pyriglena_entrop$entropy,ylim=c(min(pyriglena_entrop$entropy),m
ax(pyriglena_entrop$entropy)), xlim=c(0, 10), type = "p", xaxt = "n")
# ann=FALSE impede a impressão dos nomes dos eixos; xaxt = "n" impede que marcas sejam
impressas no eixo.
axis (side = 1, at = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10), labels = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10), cex.axis
= 0.7)
# cex.axis = muda o tamanho da fonte no texto do eixo (não é o rótulo do eixo)
par (new=T)
points(x=a1_pyriglena$k,y=a1_pyriglena$entropy, col = "firebrick1")
par (new=T)
points(x=a10_pyriglena$k,y=a10_pyriglena$entropy, col = "forestgreen")
par (new=T)
points(x=a100_pyriglena$k,y=a100_pyriglena$entropy, col = "magenta4")
par (new=T)
points(x=a1000_pyriglena$k,y=a1000_pyriglena$entropy, col = "midnightblue")
dev.off()
```

Para utilizar este *script*, deve ter o R instalado e entrar nele pelo terminal:

```
$ R
R version 3.3.2 Patched (2017-01-18 r72012) -- "Sincere Pumpkin Patch"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
R é um software livre e vem sem GARANTIA ALGUMA.
Você pode redistribuí-lo sob certas circunstâncias.
Digite 'license()' ou 'licence()' para detalhes de distribuição.
R é um projeto colaborativo com muitos contribuidores.
Digite 'contributors()' para obter mais informações e 'citation()' para saber como citar o R ou
pacotes do R em publicações.
Digite 'demo()' para demonstrações, 'help()' para o sistema on-line de ajuda ou 'help.start()'
para abrir o sistema de ajuda em HTML no seu navegador.
Digite 'q()' para sair do R.
>
```

Assim que mudar o símbolo do terminal de '$' para '>', você copia todo o conteúdo mostrado acima e assim a figura a seguir deve ser gerada:



9. 'plot_snmf.R' (parte 2)

Importante: para essa etapa é necessário que o pacote do R esteja devidamente instalado no computador.

Antes de começar essa etapa, deve ser criado uma planilha de extensão '.csv' unindo as matrizes Q que possuem o menor valor de entropia cruzada. No meu caso selecionei os valores correspondentes aos K= 2, K= 3, K= 4 e K= 5. A planilha ficou com a seguinte configuração:

geog_order	id	k1_2clusters	k2_clusters	k1_3clusters	k2_3clusters	k3_3clusters	k1_4clusters	k2_4clusters	k3_4clusters
1	15362	0.9999	0.0001	0.9998	0.00009999	0.00009999	0.9997	0.00009998	0.00009998
1	16545	0.9999	0.0001	0.9998	0.00009999	0.00009999	0.9997	0.00009998	0.00009998
1	16548	0.9999	0.0001	0.998972	0.000099991	0.000928382	0.998843	0.000099982	0.000099982
1	16550	0.9999	0.0001	0.955139	0.0447615	0.00009999	0.980765	0.0180781	0.000784187
1	16551	0.9999	0.0001	0.9998	0.00009999	0.00009999	0.9997	0.00009998	0.00009998
1	16552	0.9999	0.0001	0.982638	0.00497051	0.0123915	0.974959	0.00859702	0.0163443
1	16553	0.9999	0.0001	0.991638	0.000099991	0.00826195	0.987802	0.00009999	0.00342329
1	16554	0.9999	0.0001	0.984928	0.000099991	0.0149722	0.992661	0.000588031	0.000099991
1	16555	0.9999	0.0001	0.9998	0.00009999	0.00009999	0.996965	0.000099982	0.000099982
1	16556	0.9999	0.0001	0.962128	0.0146762	0.0231961	0.949144	0.0332244	0.00576171
2	17112	0.910738	0.089262	0.103987	0.81651	0.0795024	0.0993374	0.800308	0.0903674
2	17113	0.994341	0.00565936	0.142837	0.853825	0.00333868	0.123507	0.871665	0.00472776
2	17114	0.9999	0.0001	0.235154	0.764746	0.000099991	0.213043	0.786757	0.000099982
3	17921	0.992162	0.00783773	0.00009999	0.9998	0.00009999	0.00009998	0.9997	0.00009998
3	17922	0.992952	0.00704804	0.00009999	0.9998	0.00009999	0.00009998	0.9997	0.00009998
3	17923	0.9999	0.0001	0.0707423	0.929158	0.000099991	0.0545262	0.945274	0.000099982
3	17924	0.996673	0.00332672	0.00009999	0.9998	0.00009999	0.00009998	0.9997	0.00009998
3	17925	0.9999	0.0001	0.00009999	0.9998	0.00009999	0.00009998	0.9997	0.00009998
3	17926	0.992519	0.00748134	0.00009999	0.9998	0.00009999	0.00009998	0.9997	0.00009998
3	17927	0.994061	0.00593875	0.00009999	0.9998	0.00009999	0.00009998	0.9997	0.00009998
3	17928	0.931505	0.0684952	0.0179024	0.928048	0.0540499	0.0121877	0.925937	0.0176854
3	17929	0.967461	0.0325386	0.0112232	0.977412	0.011365	0.0128188	0.967975	0.00775087
3	17930	0.982391	0.0176095	0.00009999	0.9998	0.00009999	0.000099991	0.99569	0.000724799

geog_order = número correspondente a localidade geográfica

id = número da amostra

as demais = as matrizes Q de K= 2, seguido de K= 3 e K= 4.

Com essa segunda parte serão gerados os gráficos de barra que mostram a estrutura populacional.

```
## Codificando para diferentes Ks (valores selecionados na figura anterior)
k2_pyriglena<-pyriglena_entrop[pyriglena_entrop$k=="2",]
k3_pyriglena<-pyriglena_entrop[pyriglena_entrop$k=="3",]
k4_pyriglena<-pyriglena_entrop[pyriglena_entrop$k=="4",]
k5_pyriglena<-pyriglena_entrop[pyriglena_entrop$k=="5",]
dim(k2_pyriglena)
dim(k3_pyriglena)
dim(k4_pyriglena)
dim(k5_pyriglena)

### Encontrando a corrida com menor entropia para diferentes Ks
# vou repetir para cada K que na figura anterior apresentou baixa entropia
k2_pyriglena[which(k2_pyriglena$entropy == min(k2_pyriglena$entropy)), ]
k3_pyriglena[which(k3_pyriglena$entropy == min(k3_pyriglena$entropy)), ]
k4_pyriglena[which(k4_pyriglena$entropy == min(k4_pyriglena$entropy)), ]
k5_pyriglena[which(k5_pyriglena$entropy == min(k5_pyriglena$entropy)), ]

## Plotando coeficientes Q por K (corridas com menor entropia)

# Lendo matriz Q
Q_pyriglena <- read.csv(file = 'pyriglena_Q_matrix.csv', header = TRUE)
head(Q_pyriglena)

#### k2 - [, 3:4]
#### k3 - [, 5:7]
#### k4 - [, 8:11]
#### k5 - [, 12:16]
```

```
# Plotando e criando pdf em ordem de K (menor para maior) - localidades geográficas de norte para o sul
pdf('pyriglena_k2.pdf', )
barplot(t(as.matrix(Q_pyriglena[,3:4])), col= c('violetred' , 'turquoise1'), border=NA, axes = FALSE)
dev.off()

pdf('pyriglena_k3.pdf', )
barplot(t(as.matrix(Q_pyriglena[,5:7])), col= c('violetred' , 'turquoise1', 'yellow'), border=NA, axes = FALSE)
dev.off()

pdf('pyriglena_k4.pdf', )
barplot(t(as.matrix(Q_pyriglena[,8:11])), col= c('violetred' , 'turquoise1', 'yellow', 'green'), border=NA, axes = FALSE)
dev.off()

pdf('pyriglena_k5.pdf', )
barplot(t(as.matrix(Q_pyriglena[,12:16])), col= c('violetred' , 'turquoise1', 'yellow', 'green', 'orange'), border=NA, axes = FALSE)
dev.off()
```

10. Dúvidas

Em caso de dúvidas com relação a execução do programa sNMF, aconselho a ler os manuais e artigos dos autores pois possuem mais detalhes.

Caso a dúvida seja relacionado com os scripts para gerar figuras e automatizar sNMF pode me mandar email (ana_bia_bueno@hotmail.com)

11. Referência

Frichot, E.; Mathieu, E.; Trouillon, T.; Bouchard, G.; François,O. (2014) Fast and efficient estimation of individual ancestry coefficients. **Genetics 196** (4): 973-983.

ANEXO 5 - Manual LGEMA: ADMIXTURE

Tradução por Ana Beatriz da Silva Bueno (ana_bia_bueno@hotmail.com) das páginas citadas a seguir (consulta: 01/04/2017).

Página da *internet* do ADMIXTURE:
https://www.genetics.ucla.edu/software/admixture/download.html
Página da *internet* com manual:
https://www.genetics.ucla.edu/software/admixture/admixture-manual.pdf
Página da *internet* para *download* do programa ADMIXTURE:
https://www.genetics.ucla.edu/software/admixture/download.html

1. Introdução

ADMIXTURE é um *software* para estimativa de máxima verossimilhança de ancestralidade individual de conjuntos de dados de SNP multilocus. Ele usa o mesmo modelo estatístico que STRUCTURE, mas calcula estimativas muito mais rapidamente usando um algoritmo de otimização numérica rápida. Os arquivos de entrada do ADMIXTURE podem ser PLINK binário (.bed), PLINK normal (.ped) ou EIGENSTRAT (.geno) e seus arquivos de saída são delimitados por espaço contendo as estimativas de frequências alélicas. Para usar o ADMIXTURE, além do arquivo de entrada, é necessário fornecer uma hipótese *a priori* do número de populações ancestrais (K). Você também deve ter os arquivos de suporte associados no mesmo diretório do seu arquivo de entrada principal (conferir no manual original os demais arquivos, neste manual utilizarei apenas o formato '.geno')

2. Instalação

- Baixar o arquivo 'admixture_linux-1.3.0.tar.gz' do *site*:
https://www.genetics.ucla.edu/software/admixture/download.html
- No terminal, descompactar o arquivo:
$ tar -vzxf admixture_linux-1.3.0.tar.gz
- Para executar o programa, entre na pasta do programa e depois dê o comando de execução:
$ cd admixture_linux-1.3.0
$ ./admixture

3. Rodando os dados

- O comando básico para rodar seu dados é:

$ ./admixture [opções] inputfile K

- Exemplo utilizando K com valor 3:

$ ./admixture exemplo.geno 3

Outras opções:

-B erro padrão (se deixar “B” o valor será 200, se quiser outro valor, digitar logo após a letra o valor desejado, sem espaço).

-j núcleos de computador utilizados na análise (número inteiro e sem espaço entre letra e número)

--cv validação cruzada (método para escolha do valor de K com a melhor precisão preditiva para seu conjunto de dados; um bom valor de K exibirá um baixo erro de validação cruzada em comparação com outros valores de K).

-s 12345 para mudar a *seed* (maneira como a análise começa; evita viés de análise)

-m mudar o algoritmo utilizado (o padrão é o algoritmo de relaxamento de bloco, mas tem o algoritmo EM que é o mesmo utilizado pelo *software* FRAPPE). Ex: -m EM

-C critério de convergência máximo (o critério padrão é parar a análise quando a probabilidade de log aumenta menos de $e=10^{-4}$, com '-C' você escolhe o valor máximo que pode atingir ou o número máximo de iterações -N).

-c critério de convergência mínimo (o critério padrão é parar a análise quando a probabilidade de log aumenta menos de $e=10^{-4}$, com '-c' você escolhe o valor mínimo que pode atingir ou o número mínimo de iterações -N).

-a mudar o valor do método de aceleração (por padrão é utilizado o método de aceleração de convergência quasi-Newton com q = 3. Para usar a aceleração quasi-Newton com um valor diferente, utilizar '-a', como, por exemplo: -a qn2)

--help aparece na tela as opções que podem ser usadas

- Após rodar os dados são gerados dois arquivos de saída:

exemplo.3.Q (nome do arquivo original, valor de K - nesse caso 3 - e extensão .Q)

exemplo.3.P (nome do arquivo original, valor de K - nesse caso 3 - e extensão .P)

- Mas se utilizar a opção -B, um terceiro arquivo de saída será criado contendo os valores de erro padrão calculados com o procedimento de *bootstrap:*

exemplo.3.Q_se (nome do arquivo original, valor de K – nesse caso 3 – e extensão .Q_se)

- Para verificar qual o melhor valor de K para seus dados, a opção de validação cruzada (--cv) é fornecida e pode ser usada no *bash* da seguinte maneira:

```
for K in 1 2 3 4 5;  \
do admixture --cv hapmap3.bed $K | tee log${K}.out; done
```

- Com o comando acima você roda o ADMIXTURE com o K variando de 1 a 5 e realizando o teste de validação cruzada. Os valores devem ser reorganizados em um novo arquivo para gerar os gráficos. Como exemplo, segue a imagem das cinco primeiras linhas do meu arquivo. A 1ª coluna se refere à localidade de amostragem (geog_order); a 2ª coluna, à identificação da amostra/ indivíduo (id); as 3ª e 4ª colunas se referem às porcentagens de ancestralidade desse determinado indivíduo em relação à população 1 (k1-2clusters) e à população 2 (k2_2clusters) quando K= 2, respectivamente; as colunas 5 a 7 se referem às porcentagens de ancestralidade desse determinado indivíduo em relação à população 1 (k1-3clusters), à população 2 (k2_3clusters) e à população 3 (k3_3clusters) quando K= 3, respectivamente; e assim por diante para os vários K testados.

geog_order	id	k1_2clusters	k2_clusters	k1_3clusters	k2_3clusters	k3_3clusters	k1_4clusters	k2_4clusters	k3_4clusters
1	15362	0.9999	0.0001	0.9998	0.00009999	0.00009999	0.9997	0.00009998	0.00009998
1	16545	0.9999	0.0001	0.9998	0.00009999	0.00009999	0.9997	0.00009998	0.00009998
1	16548	0.9999	0.0001	0.998972	0.000099991	0.000928382	0.998843	0.000099982	0.000099982
1	16550	0.9999	0.0001	0.955139	0.0447615	0.00009999	0.980765	0.0180781	0.000784187

Minha linha de comando foi:

$ ./admixture -j10 pyriglena.geno 1 –cv

Descrição:

./admixture - rodar o *software* ADMIXTURE

-j10 – utilizei 10 núcleos do servidor para rodar minha análise

pyriglena.geno – arquivo de entrada na extensão '.geno' chamado 'pyriglena'

1 – valor de K utilizado na análise, nesse caso foi K=1.

--cv – pedi para que o teste de validação cruzada fosse feito.

4. Gerando gráficos

Foi utilizado o mesmo *script* R usado para o sNMF (“Manual LGEMA: sNMF”, itens 8 e 9), a única alteração que fiz foi o nome do arquivo a ser utilizado como entrada.

5. Dúvidas

Em casa de dúvidas, além desse tutorial traduzido e com minhas informações, consulte os tutoriais oficiais do autor do programa (ver Referências) ou entrar em contato com o autor David Alexander (dalexander@ucla.edu).

6. Referências

Alexander, D.H.; Novembre, J.; Lange, K. (2009) Fast model-based estimation of ancestry in unrelated individuals. **Genome Research 19**:1655–1664.

Zhou, H.; Alexander, D. H.; Lange, K. (2009) A quasi-Newton method for accelerating the convergence of iterative optimization algorithms. **Statistics and Computing 21**: 261. doi:10.1007/s11222-009-9166-3

ANEXO 6 - Manual LGEMA: fastSTRUCTURE

Tradução por Ana Beatriz da Silva Bueno (ana_bia_bueno@hotmail.com) das páginas citadas a seguir (consulta: 19/09/2016) acrescentada de dados.

Página da *internet* do fastStructure (manual incluso):
https://rajanil.github.io/fastStructure/
Página da *internet* para *download* do programa fastStructure:
https://github.com/rajanil/fastStructure

1. Introdução

A seguir apresento uma breve descrição do que o *software* realiza:
fastSTRUCTURE (Raj, 2014) é um algoritmo para inferir a estrutura populacional baseada em grande quantidade de dados genotípicos de SNPs. Baseia-se em um estrutura Bayesiana variacional para realizar inferência posterior e é escrito em Python 2.x. Neste manual é explicado como configurar este pacote de *software*, compilar os *scripts* C e Cython (ver a seguir) e executar o algoritmo em um teste simulado com conjunto de dados genotípicos.

2. Instalando os pacotes necessários

2.1. Numpy e Scipy

No terminal:
$ sudo dnf install numpy scipy

2.2. Cython

Cython é um compilador otimizado estático tanto para a linguagem de programação Python e a linguagem de programação Cython estendida (com base no Pyrex). Escreve extensões C para Python tão facilmente quanto o próprio Python.

Baixar no *site* https://pypi.python.org/pypi/Cython/
o arquivo "Cython-0.24.1.tar.gz (md5, pgp)"

No terminal:
$ tar -vzxf Cython-0.24.1.tar.gz
$ cd Cython-0.24.1

$ sudo python setup.py install

2.3. GNU Scientific Library

O GNU Scientific Library (GSL) é uma biblioteca numérica para programadores C e C ++. É *software* livre sob a GNU *General Public License*.

A biblioteca fornece uma ampla gama de rotinas matemáticas, tais como geradores de números aleatórios, funções especiais e de mínimos quadrados de montagem. Existem mais de 1000 funções no total, com um extenso conjunto de testes.

Baixar no *site* http://gnu.c3sl.ufpr.br/ftp/gsl/
o arquivo "gsl-latest.tar.gz"

No terminal:

$ tar -vzxf gsl-latest.tar.gz

$ cd gsl-2.2.1

$ ./configure

$ make

$ make check

$ make install

3. Adquirindo o código fonte de fastStructure

Para adquirir o pacote, digitar no terminal:

$ git clone https://github.com/rajanil/fastStructure

Para atualizar o pacote instalado, no terminal:

$ git fetch

$ git merge origin/master

O pacote também pode ser instalado pela seguinte linha de comando:

$ wget --no-check-certificate https://github.com/rajanil/fastStructure/archive/master.tar.gz

4. Construindo extensões Python

Antes de construir extensões Python, é importante identificar o caminho para os arquivos libgsl.so e libgslcblas.so e arquivo de cabeçalho GSL/gsl_sf_psi.h que fazem parte da sua

instalação GSL. Para uma instalação padrão do GSL, as bibliotecas (arquivos .so) são normalmente encontradas nos diretórios (.h) em /local/include/ e /local/lib. Neste caso, estas linhas podem ser adicionas ao seu arquivo .bashrc em seu diretório *home* usando as seguintes linhas de comando:

```
$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib
$ export CFLAGS="-I/usr/local/include"
$ export LDFLAGS="-L/usr/local/lib"
```

Em seguida, execute a seguinte linha de comando para definir essas variáveis de ambiente:

```
$ source ~/.bashrc
```

Cada configuração irá criar alguns arquivos .c e .so (objeto compartilhado). Se forem indicados erros que indicam que a compilação falhou, pode ser porque o compilador errado está sendo usado ou variáveis de ambiente (como LD_LIBRARY_PATH) estão definidos incorretamente. Para usar um compilador gcc específico, você pode digitar:

```
$ CC=/Cython-0.24.1/Cython/Compiler python setup.py build_ext  - -inplace
```

5. Executando o código

Para executar o fastSTRUCTURE, o principal *script* que você vai precisar é *structure.py.* Para visualizar as opções de linha de comando que precisam ser passados para o *script*, você pode digitar:

```
$ pyhton structure.py
```

E a seguinte mensagem aparecerá no terminal:

```
Here is how you can use this script

Usage: python structure.py
    -K <int>   (number of populations)
    --input=<file>   (/path/to/input/file)
    --output=<file>   (/path/to/output/file)
    --tol=<float>   (convergence criterion; default: 10e-6)
    --prior={simple,logistic}   (choice of prior; default: simple)
    --cv=<int>   (number of test sets for cross-validation, 0 implies no CV step; default: 0)
    --format={bed,str} (format of input file; default: bed)
```

--full (to output all variational parameters; optional)

--seed=<int> (manually specify seed for random number generator; optional)

Traduzindo:

Uso do 'pyhton structure.py'

-K <int> (número de populações)

--input=<file> (/caminho/para/arquivo/entrada)

--output=<file> (/caminho/para/arquivo/saída)

--tol=<float> (critério de convergência; padrão: 10e-6)

--prior={simple,logistic} (escolha de prioridade; padrão: simples)

--cv=<int> (número de conjuntos de teste para validação cruzada, 0 implica em nenhum CV; padrão: 0)

--format={bed,str} (formato do arquivo de entrada; padrão: bed)

--full (saída de todos os parâmetros variacionais; opcional)

--seed = <int> (especificar manualmente semente para o gerador de números aleatórios; opcional)

fastStructure realiza a inferência do modelo de mistura (*admixture model)* mais simples, baseado em locos independentes, com duas opções de *priors* que podem ser especificados usando o identificador '--prior'. Assim, ao contrário do STRUCTURE, fastStructure não requer os arquivos 'mainparams' e 'extraparam'. O algoritmo de inferência utilizado por fastStructure é fundamentalmente diferente daquele do STRUCTURE e exige o estabelecimento de muito menos opções. Todas as opções para fastStructure podem ser passadas através dos identificadores listados acima.

5.1. Opções obrigatórias

As opções obrigatórias que devem constar nos *scripts* são o arquivo de entrada, o arquivo de saída e o número de populações. Supondo que o arquivo de entrada seja nomeado genotypes.bed (deve ser incluído com os seguintes arquivos: genotypes.fam e genotypes.bim), o arquivo de saída seja nomeado genotypes_output e o número de populações a testar seja de 3, o algoritmo pode ser executado da seguinte forma no terminal:

```
$ python structure.py -K 3 --input=genotypes --output=genotypes_output
```

Isso gera um arquivo genotypes_output.3.log que controla como o algoritmo prossegue e

arquivos genotypes_output.3.meanQ e genotypes_output.3.meanP contendo a média posterior de proporções de mistura e frequências alélicas, respectivamente. As ordens de amostras e de SNPs nos arquivos de saída correspondem àquelas do arquivo '.fam' e '.bim' (arquivos de entrada), respectivamente. Observe que o nome de arquivo de entrada não deve incluir sufixos (por exemplo, .bed) e deve estar no diretório do projeto principal (a menos que um caminho completo seja fornecido).

5.2. Formato do arquivo de entrada

Dados no formato plink bed e no formato do STRUCTURE podem ser importados. Se os dados estão no formato plink, garantir que os arquivos bed, bim e fam do conjunto de dados estejam todos no mesmo caminho.

Enquanto o programa STRUCTURE original permitia um formato de entrada mais flexível, fastStructure necessita de um formato de entrada mais específico. Isto é, as linhas no arquivo de dados correspondem a amostras, com duas fileiras por amostra (note que apenas dados diplóides podem ser analisados por este *software*) e colunas correspondem aos SNPs. As primeiras seis colunas do arquivo serão ignoradas; estas geralmente incluem IDs, metadados, etc. Este *software* só lida com locos bi-alélicos. Os dois alelos em cada loco podem ser codificados como desejado. No entanto, dados faltantes devem ser codificados como '-9'.

6. Rodando dados (teste)

Um conjunto de dados simulados para teste é fornecido em test/testdata.bed com genótipos amostrados para 200 indivíduos com 500 SNPs independentes de diferentes locos. Os arquivos de saída em test/ foram gerados como se segue, no terminal:

```
$ python structure.py -K 3 --input=test/testdata --output=test/testoutput_simple --full --seed=100
$ ls test/testoutput_simple*
test/testoutput_simple.3.log  test/testoutput_simple.3.meanP  test/testoutput_simple.3.meanQ
test/testoutput_simple.3.varP  test/testoutput_simple.3.varQ

$ python structure.py -K 3 --input=test/testdata --output=test/testoutput_logistic --full --seed=100 --prior=logistic
$ ls test/testoutput_logistic*
```

test/testoutput_logistic.3.log test/testoutput_logistic.3.meanQ
test/testoutput_logistic.3.varQ
test/testoutput_logistic.3.meanP test/testoutput_logistic.3.varP

$ tail -n 3 test/testoutput_simple.3.log
Marginal Likelihood (avg over genotypes) = -0.9777044544
Total time = 4.7611 seconds
Total iterations = 160

Ao executar o código com os dados de teste fornecidos deve-se gerar um arquivo de log idêntico ao de test/, como uma verificação final de que o código-fonte foi baixado e compilado corretamente.

7. Escolhendo a complexidade do modelo

Recomenda-se a execução do fastStructure para várias opções de K para que a escolha do valor apropriado de K explique com maior precisão a estrutura populacional dos dados analisados. Gerando vários arquivos de saída, o teste do melhor valor pode ser feito com o *script* 'chooseK.py' conforme explicado abaixo.

Supondo que o algoritmo foi executado com o conjunto de dados de teste para K que varia de 1 a 10 e a sinalização de saída foi --output = test/testoutput_simple, a complexidade do modelo pode ser obtida, digitando no terminal:
$ python chooseK.py --input=test/testoutput_simple
Model complexity that maximizes marginal likelihood = 2
Model components used to explain structure in data = 4

8. Visualizando a proporção de mistura

Para visualizar as proporções de mistura genética inferidas pelo fastStructure, é fornecida uma ferramenta simples para gerar figura: o *script* distruct. As amostras serão agrupadas de acordo com os rótulos populacionais inferidos pelo fastStructure. No entanto, se pretende-se agrupar as amostras de acordo com alguma outra forma (por exemplo, baseado na localização geográfica), esses dados podem ser fornecidos como um identificador separado usando'--popfile'. A ordem dos rótulos neste arquivo (um rótulo por linha) deve coincidir com o fim de amostras nos arquivos de dados de entrada.

No terminal:

$ python distruct.py

Uso: python distruct.py

-K <int> (número de populações)

--input=<file> (caminho para o arquivo de entrada; mesmo arquivo de saída passado na linha de comando para rodar fastStructure - structure.py)

--output=<file> (caminho para o arquivo de saída)

--popfile=<file> (arquivo com localidades identificadas; opcional)

--title=<figure title> (título da figura; opcional)

Supondo que o algoritmo foi executado com o conjunto de dados de teste para K= 5 e o comando de saída foi --output = test/testoutput_simple, pode-se gerar uma figura usando o *script* distruct digitando:

$ python distruct.py -K 5 –input=test/testoutput_simple --output=test/testoutput_simple_distruct.svg

9. Análise dos meus dados

Para rodar meus dados utilizei a seguinte linha de comando:

$ python structure.py -K 1 –input=pyriglena_c85d6m4p3 –output=pyriglena --format=str –full

Nessa linha de comando variei o K de 1 até 10 e gerei 100 repetições para cada K para verificar se houve variação nas frequências. No meu caso não houve diferença de valores de frequência para um mesmo número K.

Para conferir qual o melhor modelo utilizei o seguinte comando:

$ python chooseK.py –input=pyriglena*

Model complexity that maximizes marginal likelihood = 1

Model components used to explain structure in data = 2

Com o resultado acima, conclui que o melhor K para meu conjunto de dados de dado é 2, mas mesmo assim plotei as matrizes Q de K= 2, K= 3, K= 4 e K= 5 (valores que foram testados no sNMF) para comparar os resultados. Utilizei o seguinte *script* em R (adaptado daquele utilizado para o sNMF):

```
# Lendo matriz Q
Q_pyriglena <- read.csv(file = 'pyriglena_Q_fastS.csv', header = TRUE)
head(Q_pyriglena)
#### k2 - [, 3:4]
#### k3 - [, 5:7]
#### k4 - [, 8:11]
#### k5 - [, 12:16]

# Plotando e criando arquivo pdf para os diferentes K (do menor para o maior) e localidades geográficas de norte para o sul
pdf('pyriglena_k2.pdf', )
barplot(t(as.matrix(Q_pyriglena[,3:4])), col= c('violetred' , 'turquoise1'), border=NA, axes = FALSE)
dev.off()

pdf('pyriglena_k3.pdf', )
barplot(t(as.matrix(Q_pyriglena[,5:7])), col= c('violetred' , 'turquoise1', 'yellow'), border=NA, axes = FALSE)
dev.off()

pdf('pyriglena_k4.pdf', )
barplot(t(as.matrix(Q_pyriglena[,8:11])), col= c('violetred' , 'turquoise1', 'yellow', 'green'), border=NA, axes = FALSE)
dev.off()

pdf('pyriglena_k5.pdf', )
barplot(t(as.matrix(Q_pyriglena[,12:16])), col= c('violetred' , 'turquoise1', 'yellow', 'green', 'orange'), border=NA, axes = FALSE)
dev.off()
```

Os diferentes valores de K resultaram em gráficos praticamente iguais, sendo que todos indicavam fortemente os mesmos 2 agrupamentos.

10. Dúvidas

Em casa de dúvidas, consulte os tutoriais oficiais do autor do programa (ver *link* nas referências) e consulte um grupo do Google específico para discussão relativo aos programas STRUCTURE e fastStructure: https://groups.google.com/forum/#!forum/structure-software.

11. Referências

fastSTRUCTURE:

Raj, A.; Stephens, M.; Pritchard, J.K. (2014) fastSTRUCTURE: variational inference of population structure in large SNP data sets. **Genetics 197**: 573-589.

Cython:http://cython.org/

http://docs.cython.org/en/latest/

http://docs.cython.org/en/latest/src/quickstart/install.html

GNU Scientific Library: http://www.gnu.org/software/gsl/

ANEXO 7 - Manual LGEMA: BayeScan e PGDSpider

Tradução por Ana Beatriz da Silva Bueno (ana_bia_bueno@hotmail.com) das páginas citadas a seguir (consulta: 09/04/2017) acrescentada de dados.

Página da *internet* do BayeScan:
http://cmpg.unibe.ch/software/BayeScan/
Página da *internet* para *download* do programa BayeScan:
http://cmpg.unibe.ch/software/BayeScan/download.html
Página do PGDSpider (conversor de arquivos):
http://www.cmpg.unibe.ch/software/PGDSpider/

1. Introdução

O programa BayeScan (Foll *et al.* 2008) busca identificar locos candidatos a estarem sob seleção natural baseado em diferenças de frequências alélicas entre populações. O BayeScan é baseado no modelo multinomial-Dirichlet. Um dos possíveis cenários mais simples é constituído por um modelo de ilha no qual as frequências alélicas das subpopulações são correlacionadas por meio de um *pool* de alelos compartilhados de diversas maneiras entre as subpopulações. A diferença na frequência alélica entre este conjunto de alelos em comum e a de cada subpopulação é medida por um coeficiente específico de F_{ST} de subpopulação. Portanto, esta estratégia permite considerar cenários ecológicos mais realistas nos quais o tamanho efetivo e a taxa de imigração podem diferir entre as populações.

Como o BayeScan realiza uma análise bayesiana, ele incorpora a incerteza sobre as frequências alélicas devido ao pequeno tamanho das amostras. Ou seja, a análise de um tamanho amostral pequeno, ainda que tendo baixo poder estatístico, não deve apresentar nenhum viés. As frequências alélicas são estimadas utilizando diferentes modelos estatísticos dependendo do tipo de marcador genético utilizado. No BayeScan, podem ser utilizados três tipos diferentes de dados: (i) dados codominantes (como SNPs ou microssatélites), (ii) dados binários dominantes (como AFLPs) e (iii) intensidade de banda de AFLP, que não são considerados nem dominantes nem codominantes. Como não utilizamos AFLP, somente foram abordados nessa tradução o que é relevante para análises de SNPs.

A seleção é detectada pela decomposição dos coeficientes de F_{ST} em um componente específico da população (beta) compartilhado por todos os locos e um componente específico do loco (alfa) compartilhado por todas as populações usando uma regressão logística. O

desvio da neutralidade em um dado loco é assumido quando o componente loco-específico é necessário para explicar o padrão observado de diversidade (alfa significativamente diferente de 0). Um valor positivo de alfa sugere seleção diversificadora, enquanto valores negativos sugerem seleção balanceadora ou purificadora. Isto leva a dois modelos alternativos para cada loco incluindo ou não o componente alfa para a seleção do modelo. Para cada loco, um MCMC de salto reversível explora modelos com e sem seleção (com o componente alfa presente ou ausente, respectivamente) e estima suas probabilidades posteriores relativas.

2. *Download*

O BayeScan é um *software* de código aberto baseado em linha de comando, sob a *GNU General Public License*, conforme publicado pela *Free Software Foundation*. O BayeScan é codificado usando C ++ padrão e pode ser facilmente compilado em qualquer sistema com um compilador C ++ disponível usando o "Makefile" fornecido junto com o código-fonte. Desde a versão 2.1, o BayeScan usa o OpenMP para implementar o cálculo paralelo em CPUs multicores, mas é necessário ter um compilador que suporte OpenMP.

São fornecidos os executáveis de linhas de comando prontos para ser usados em Microsoft Windows, Apple Mac OS X (processadores de Intel), Linux 32bits e 64bits. Também é fornecida uma versão gráfica da interface do usuário (GUI) com base no mesmo código que a versão da linha de comando, mas apenas para Microsoft Windows.

O BayeScan é distribuído como um único pacote contendo todos os arquivos necessários, incluindo manual, código-fonte, executáveis e *scripts* R; a versão atual é 2.1.

- *Download* do arquivo BayeScan2.1.zip em:
http://cmpg.unibe.ch/software/BayeScan/download.html
- Descompactar o arquivo BayeScan2.1.zip pelo terminal:
$ unzip BayeScan2.1.zip

3. Como usar o BayeScan

BayeScan pode ser compilado em qualquer sistema que tenha um compilador C++ usando o 'Makefile' disponível no código 'source' (digitando 'make' em um terminal dentro do diretório 'source' na pasta do 'BayeScan2.1' cria um arquivo executável).

- Primeiro conferir as instalações de C++
$ sudo dnf update

$ sudo dnf upgrade

$ sudo dnf install make automake gcc gcc-c++ kernel-devel

(na linha de comando acima estão os programas que substituem no Fedora o pacote 'build-essential' do Linux).

$ sudo dnf install glibc-devel

- Instalando o BayeScan

$ cd BayeScan2.1/source

$ make

-Para rodar o programa:

$ cd BayeScan2.1/source

$ ./bayescan_2.1

- Caso o comando 'make' não funcione, tente o seguinte:

$ cd BayeScan2.1/binaries

$ chmod +x BayeScan2.1_linux64bits

$ ./BayeScan2.1_linux64bits

O único arquivo obrigatório para o BayeScan é o arquivo de entrada descrevendo os dados genéticos multilocos (ver a seção a seguir sobre o preparo do arquivo de entrada). Não há limitação para o tamanho do conjunto de dados, só depende da memória do computador. Todas as outras opções tem valores padrões e podem ser colocados com outros valores nas linhas de comando se assim desejar. Ao iniciar o programa com a linha de comando descrita acima (sem argumentos, somente executando o programa) a mensagem a seguir aparecerá:

```
---------------------------
| BayeScan 2.0 uso:     |
---------------------------
-help        imprime essa ajuda
---------------------------
| Entrada               |
---------------------------
alleles.txt         nome do arquivo de entrada de dados genotípicos
-d discarded     arquivo de entrada opcional contendo lista para descarte
```

-snp usar matriz genotípica de SNPs

| Saída |

-od . diretório do arquivo de saída, o padrão é o mesmo do programa

-o alleles prefixo do arquivo de saída, o padrão é o do arquivo de entrada sem extensão

-fstat calcula o F_{ST} (sem seleção)

-all_trace MCMC escreve o arquivo '*trace file*' também para os parâmetros alpha (pode ser um arquivo longo)

| Parâmetros da cadeia |

-threads n número de núcleos usados, o padrão é a capacidade total do computador

-n 5000 número de iterações resultantes/de saída, o padrão é 5000

-thin 10 tamanho de intervalo que pode ocorrer a variação do erro, pode permitir uma maior ou menor variação, o padrão é 10

-nbp 20 número de corridas piloto, o padrão é 20

-pilot 5000 comprimento das corridas piloto, o padrão é 5000

-burn 50000 tamanho para o corte (*burnin*), o padrão é 50000

| Parâmetros do modelo |

-pr_odds 10 *prior odds* para o modelo neutro, o padrão é 10.

-lb_fis 0 limite inferior para *prior* uniforme em F_{IS} (dados dominantes), o padrão é 0

-hb_fis 1 limite superior para *prior* uniforme em F_{IS} (dados dominantes), o padrão é 1

-beta_fis *prior* beta opcional para F_{IS} (dados dominantes, m_fis e sd_fis precisam ser definidos)

-m_fis 0.05 média opcional para *prior* beta em F_{IS} (dados dominantes com -beta_fis)

-sd_fis 0.01 erro padrão opcional para *prior* beta em F_{IS} (dados dominantes com -beta_fis)

-aflp_pc 0.1 limiar para o genótipo recessivo como uma fração da intensidade máxima da banda, o padrão é 0.1

| Arquivos de saída |

-out_pilot arquivo de saída opcional de corridas piloto
-out_freq arquivo de saída opcional de frequências alélicas

- Minha linha de comando foi a seguinte:
$ ./BayeScan2.1_linux64bits pyriglenabayescan -all_trace -threads 30 -out_pilot -out_freq
Descrição:
./BayeScan2.1_linux64bits - executar o *software* em Linux 64bits
pyriglenabayescan – nome do meu arquivo de entrada (coloquei no mesmo diretório do *software*)
-all_trace – pedi para o arquivo '*trace file*' ser feito
-threads 30 – utilizei 30 núcleos do computador
-out_pilot – pedi o arquivo de saída opcional de corridas piloto
-out_freq – pedi o arquivo de saída opcional de frequências alélicas

4. Preparo do arquivo de entrada (PGDSpider)

O BayeScan usa seus próprios formatos de arquivo de entrada, que dependem do tipo de dados utilizados. Todos os arquivos de entrada devem estar em formato de texto, mas arquivos UNICODE não são reconhecidos (tenha cuidado ao exportar do Excel).

Introdução:

PGDSpider é uma poderosa ferramenta automatizada de conversão de dados para programas genéticos e genômicos de populações. Facilita as possibilidades de troca de dados entre programas para uma vasta gama de tipos de dados (ex. DNA, RNA, NGS, microssatélite, SNPs, RFLP, AFLP, dados multialélicos, frequência alélica ou distâncias genéticas). Além dos formatos convencionais de genética populacional, o PGDSpider integra os formatos de dados de genômica populacional comumente usados para armazenar e manipular dados de sequênciamento de segunda geração (*NGS*). Atualmente, PGDSpider não se destina a converter arquivos *NGS* muito grandes, pois ele carrega na memória todo o arquivo de entrada, cujo tamanho pode exceder o RAM disponível. No entanto, uma vez que PGDSpider permite converter subconjuntos específicos desses arquivos *NGS* em qualquer outro formato, pode-se usar esse recurso para calcular parâmetros ou estatísticas para regiões específicas e, assim, executar a análise de janela deslizante em grandes regiões genômicas.

O PGDSpider usa um formato recentemente desenvolvido de *Population Genetic Data* (PGD) como um passo intermediário no processo de conversão. PGD é um formato de

arquivo projetado para armazenar vários tipos de dados de genética populacional, incluindo diferentes tipos de dados genéticos e ploidias. O PGD é baseado no formato XML e é, portanto, independente de qualquer sistema de computador específico e extensível para necessidades futuras. PGDSpider usa PGD para conectar a genética populacional e programas de genômica como uma aranha tecendo uma teia.

PGDSpider é escrito em Java e é, portanto, independente da plataforma. É amigável devido à sua interface gráfica intuitiva. O PGDSpider permite que o usuário armazene suas configurações de conversão preferidas. Uma versão de linha de comando do PGDSpider também é fornecida, tornando possível incorporar PGDSpider em *pipelines* de análise de dados.

Requisitos:
- SUN Java 1.7 RE (ou uma versão mais recente) tem que estar instalado
- Java6 RE pode ser baixado no seguinte link:
http://www.oracle.com/technetwork/java/javase/downloads/index.html

Download e Instruções de Instalação
O primeiro passo é instalar o Java7RE
$ sudo dnf install java-1.7.0-openjdk
$ java -version

Em seguida:
Baixar o arquivo 'PGDSpider_2.1.1.0.zip' da seguinte página:
http://www.cmpg.unibe.ch/software/PGDSpider/#Download_and_Installation_Instructions
Descompactar o arquivo 'PGDSpider_2.1.1.0.zip':
$ unzip PGDSpider_2.1.1.0.zip
Executar o programa:
$ java -Xmx1024m -Xms512m -jar PGDSpider2.cli.jar

A versão de linha de comando do programa PGDSpider pode ser executada com as seguintes opções (a ordem que as opções são colocadas na linha de comando não influencia no funcionamento do *software*):
-? -h
mostra um texto de ajuda com as opções

-inputfile <file>
especifica o arquivo de entrada para o processo de conversão

-inputformat <format>
especifica o formato do arquivo de entrada; esta opção é obrigatória se não tem o arquivo 'spid'

-outputfile <file>
especifica o arquivo de saída para o processo de conversão

-outputformat <format>
especifica o formato do arquivo de saída; esta opção é obrigatória se não tem o arquivo spid

-spid <file>
especifica o arquivo SPID que contém perguntas de conversão pré-atendidas. O arquivo SPID pode ser gerado com o editor SPID (versão de interface gráfica). Como alternativa, um arquivo SPID modelo é gerado automaticamente se nenhum arquivo SPID for fornecido.

Minha linha de comando foi: VCF → BayeScan
$ java -Xmx1024m -Xms512m -jar PGDSpider2-cli.jar -inputfile /home/biabueno/pyrad/outfiles/pyriglena.vcf -inputformat VCF -outputfile /home/biabueno/pyriglenabayescan -outputformat GEST_BAYE_SCAN -spid /home/biabueno/PGDSpider_2.1.1.0/examples/template_VCF_GESTE_BAYE_SCAN.spid

Como eu rodei o BayeScan para 2, 3 e 4 populações, foi necessário acrescentar um *link* para o arquivo de texto contendo o número do indivíduo (sem espaço) e a população a qual ele pertence (tabulada), conforme a seguinte imagem:

```
15362.assembled pop_1
16545.assembled pop_1
16548.assembled pop_1
16550.assembled pop_1
16551.assembled pop_1
16552.assembled pop_1
16553.assembled pop_1
16554.assembled pop_1
16555.assembled pop_1
16556.assembled pop_1
16685.assembled pop_2
16702.assembled pop_2
16704.assembled pop_2
```

E meu arquivo SPID apresentou as seguintes configurações:

spid-file generated: Tue Apr 18 18:56:05 BRT 2017

VCF Parser questions

PARSER_FORMAT=VCF

Do you want to include a file with population definitions?

VCF_PARSER_POP_QUESTION=TRUE

Only input following regions (refSeqName:start:end, multiple regions: whitespace separated):

VCF_PARSER_REGION_QUESTION=

What is the ploidy of the data?

VCF_PARSER_PLOIDY_QUESTION=DIPLOID

Only output following individuals (ind1, ind2, ind4, ...):

VCF_PARSER_IND_QUESTION=

Output genotypes as missing if the read depth of a position for the sample is below:

VCF_PARSER_READ_QUESTION=

Take most likely genotype if "PL" or "GL" is given in the genotype field?

VCF_PARSER_PL_QUESTION=

Do you want to exclude loci with only missing data?

VCF_PARSER_EXC_MISSING_LOCI_QUESTION=

Select population definition file:

VCF_PARSER_POP_FILE_QUESTION= /home/biabueno/4pop.txt

OBS: só reconheceu meu arquivo de populações quando deixei ele na 'home'.

Only output SNPs with a phred-scaled quality of at least:

VCF_PARSER_QUAL_QUESTION=

Do you want to include non-polymorphic SNPs?

VCF_PARSER_MONOMORPHIC_QUESTION=

Output genotypes as missing if the phred-scale genotype quality is below:

VCF_PARSER_GTQUAL_QUESTION=

GESTE / BayeScan Writer questions

WRITER_FORMAT=GESTE_BAYE_SCAN

# Specify which data type should be included in the GESTE / BayeScan file (GESTE / BayeScan can only analyze one data type per file):
GESTE_BAYE_SCAN_WRITER_DATA_TYPE_QUESTION=SNP

5. Interpretando e plotando arquivos de saída

O prefixo dos arquivos de saída é: 'yourprefix'.

A maioria dos resultados que você precisará estará no arquivo 'yourprefix_fst.txt' criado no fim da análise. Cada linha do arquivo corresponde a um loco (eventualmente sem os marcadores listados e descartados no arquivo de entrada) e contém as seguintes informações nesta ordem:

- a identificação do loco correspondente à identificação no arquivo de entrada.
- a probabilidade posterior do modelo que assume a presença de seleção
- o logaritmo de *Posterior Odds* na base 10 para o modelo que assume a presença de seleção. Note que este valor é arbitrariamente fixado em 1000 quando a probabilidade posterior é 1 (deve ser infinito).
- o valor-q para o modelo que assume a presença de seleção (o valor-q é análogo FDR-*False Discovery Rate* do valor-p ; valor-q só é definido no contexto de vários testes, enquanto que um valor-p é definido em uma única prova).
- o coeficiente alfa estimado indica a força e direção da seleção. Um valor positivo de alfa sugere seleção diversificadora, enquanto que valores negativos sugerem seleção balanceadora ou purificadora.
- o coeficiente F_{ST} com médias calculadas pelas populações. Em cada população o F_{ST} é calculado com a média posterior usando um modelo médio.

Uma função para utilizar no *software* R é fornecida para plotar e identificar *outliers* usando diferentes critérios deste arquivo. A função é chamada 'plot_bayescan' e está localizado no arquivo 'plot_R.r'.

Abrindo o *software* R e carregando o arquivo 'plot_R.r':

```
$ R
R version 3.3.2 Patched (2017-01-18 r72012) -- "Sincere Pumpkin Patch"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
R é um software livre e vem sem GARANTIA ALGUMA.
```

Você pode redistribuí-lo sob certas circunstâncias.
Digite 'license()' ou 'licence()' para detalhes de distribuição.
R é um projeto colaborativo com muitos contribuidores.
Digite 'contributors()' para obter mais informações e
'citation()' para saber como citar o R ou pacotes do R em publicações.
Digite 'demo()' para demonstrações, 'help()' para o sistema on-line de ajuda,
ou 'help.start()' para abrir o sistema de ajuda em HTML no seu navegador.
Digite 'q()' para sair do R.
> source('plot_R.r')

O comando da função uma vez carregada no R e estando no diretório correto é:
> plot_bayescan ('yourprefix_fst.txt', FDR=0.05)

False Discovery Rate (FDR) é a taxa de falsos negativos que é permitida plotar no gráfico. O FDR é calculada pelo valor-q e geralmente é permitido 5% de falsos negativos. Com o comando acima serão listados os *outliers* com o valor-q mais baixos que 5% e uma figura será produzida. Mais detalhes sobre a função pode ser encontrada no começo do arquivo 'plot_R.r' quando aberto usando editor de texto.

A saída completa do algoritmo MCMC pode ser encontrada no arquivo "yourprefix.sel". Desde a versão 2.1, esses coeficientes listados abaixo são escritos somente se a opção "-all_trace" estiver habilitada. Cada linha corresponde a uma iteração do MCMC e contém os seguintes parâmetros nesta ordem (a primeira linha contém o que consta em cada coluna: logL Fst1):

- índice de iteração, iniciado após o piloto ser executado e *burn-in* (alíquota dos resultados que é desprezada),
- logaritmo da probabilidade,
- coeficiente F_{IS} para cada população no caso de dominante ou dados de intensidade de banda AFLP,
- coeficiente F_{ST} para cada população,
- coeficiente alfa para cada loco. Contando o valor nulo de alfa dá a probabilidade posterior para o modelo neutro.

Obter as distribuições posteriores e o intervalo de densidade de maior probabilidade (HPDI) para os diferentes parâmetros é fácil usando R e este arquivo. Primeiro deve-se carregar o arquivo (certifique-se que o R está buscando no diretório certo):

```
> mydata=read.table("yourprefix.sel",colClasses="numeric")
```

Em seguida pode-se escolher um parâmetro no qual você esteja interessado:

```
> parameter="Fst1"
```

O comando seguinte plota a distribuição posterior deste parâmetro:

```
> plot(density(mydata[[parameter]]), xlab=parameter,
main=paste(parameter,"posterior distribution"))
```

Se o pacote 'boa' estiver instalado, o comando abaixo disponibilizará o HPDI de 95%:

```
> boa.hpd(mydata[[parameter]],0.05)
```

Ao abrir o arquivo "yourprefix_Verif.txt" pode-se verificar se o arquivo de entrada foi lido corretamente.

Alguns arquivos de saída opcionais podem ser produzidos dependendo dos dados e da opção escolhida:

- O arquivo "yourprefix_prop.txt" fornece os resultados das execuções piloto.
- O arquivo "yourprefix_AccRte.txt" fornece a mudança da taxa de aceitação para os diferentes parâmetros do modelo.
- O arquivo "yourprefix_freq.txt" contém as frequências alélicas estimadas (média posterior) para cada marcador em cada população quando se utilizam marcadores dominantes ou dados de intensidade de banda de AFLP.

6. Dúvidas

Em casa de dúvidas, pode-se consultar os tutoriais oficiais do autor do programa (ver *link* nas referências) e no programa há um manual em PDF. Caso necessite entrar em contato com o autor: Matthieu Foll (matthieu.foll@iee.unibe.ch).

7. Referências

Foll M and OE Gaggiotti (2008) A genome scan method to identify selected loci appropriate for both dominant and codominant markers: A Bayesian perspective. **Genetics 180**: 977-993.

Foll M, Fischer MC, Heckel G and L Excoffier (2010) Estimating population structure from AFLP amplification intensity. **Molecular Ecology 19**: 4638-4647.

Fischer MC, Foll M, Excoffier L and G Heckel (2011) Enhanced AFLP genome scans detect local adaptation in high-altitude populations of a small rodent (*Microtus arvalis*). **Molecular Ecology 20**: 1450-1462.

Lischer HEL and Excoffier L (2012) PGDSpider: An automated data conversion tool for connecting population genetics and genomics programs. **Bioinformatics 28**: 298-299.

Copyright (c) 2007-2017, Heidi E.L. Lischer. All rights reserved.

PGDSpider is distributed under the BSD 3-Clause License. For the full text of the license, see the file LICENSE.txt. By using, modifying or distributing this software you agree to be bound by the terms of this license.

ANEXO 8 - Manual LGEMA: TreeMix

Tradução por Ana Beatriz da Silva Bueno (ana_bia_bueno@hotmail.com) das páginas citadas a seguir (consulta: 11/05/2017) acrescentada de imagens.

Página da *internet* do TreeMix:
https://bitbucket.org/nygcresearch/treemix/wiki/Home
Página da *internet* para *download* do programa TreeMix:
https://bitbucket.org/nygcresearch/treemix/downloads/

1. Introdução

TreeMix (Pickrell & Pritchard, 2012) é um programa para inferir relações e misturas entre linhagens baseado em dados de frequências alélicas. Ou seja, é produzida uma árvore de máxima verossimilhança (MV) com os eventos de mistura.

2. Instalação

O TreeMix pode ser executado em qualquer sistema Unix (por exemplo, Linux ou Mac OS X).
Requer a Biblioteca Científica GNU (http://www.gnu.org/s/gsl/) e a *Boost Graph Library* (http://www.boost.org/) versão 1.42 ou superior.
Certifique-se de que essas bibliotecas estão instaladas e, depois, baixe o código-fonte do *site*:
https://bitbucket.org/nygcresearch/treemix/downloads/
arquivo nomeado "treemix-1.13.tar.gz" (última versão do TreeMix)
Realizar os seguintes passos:

```
$ tar -xvf treemix-1.13.tar.gz
$ cd treemix-1.13
$ ./configure
$ make
$ make install
```

3. Arquivo de Entrada

SNPs

Por padrão, o TreeMix assume que os sítios são bialélicos. O arquivo de entrada é um arquivo gzip que consiste em um cabeçalho (com a lista dos nomes das populações separados por um

espaço) seguido por linhas contendo o número total de cada um dos dois alelos por SNP por população. Ou seja, cada linha mostra os dados dos dois alelos de um SNP separados por uma vírgula e os dados das populações separados por um espaço (ver exemplo abaixo). Supõe-se que a ordem dos SNPs no arquivo seja a sua ordem no genoma.

```
pop1 pop2 pop3 pop4
5,1 1,1 4,0 0,4
3,3 0,2 2,2 0,4
1,5 0,2 2,2 1,3
```

4. Opções

4.1. Criar a árvore de MV

O padrão do TreeMix é construir a árvore de MV das populações que constam no arquivo de entrada sob o pressuposto de que todos os sítios são independentes. Para isso, execute:

```
$ treemix -i inputfile.gz -o out_stem
```

4.2. Escolha a posição da raiz (-root)

Antes de adicionar eventos de migração a uma árvore, é importante definir a posição da raiz. Para construir a árvore e definir a posição da raiz, se o nome da população externa for "Outgroup", execute:

```
$ treemix -i inputfile.gz -root Outgroup -o out_stem
```

4.3. Agrupar os SNPs para explicar o desequilíbrio de ligação (-k)

Para incluir a informação de que os SNPs próximos não são independentes, agrupe-os em janelas de tamanho n SNPs usando a opção -k. A ordem dos SNPs no arquivo de entrada é assumida como sendo a ordem deles no genoma. Recomendamos usar um valor de n que excede em muito a extensão conhecida de desequilíbrio de ligação no organismo em questão (isso dependerá, obviamente, da densidade de SNP). Por exemplo, para a construção da árvore de MV usando blocos de 1000 SNPs, execute:

```
$ treemix -i inputfile.gz -k 1000 -o out_stem
```

4.4. Criar o gráfico com migração (-m)

Se deseja testar se há evidência de eventos de migração na árvore, use a opção -m seguido do número de eventos de migração a serem testados. Por exemplo, o comando a seguir criará a árvore de MV e, em seguida, adicionará dois eventos de migração:

$ treemix -i inputfile.gz -m 2 -o out_stem

4.5. Insira uma árvore/gráfico gerado anteriormente (-g)

Existem duas maneiras de inserir uma árvore/gráfico gerado anteriormente. O mais simples é a entrada do formato TreeMix usando a opção -g, que tem um arquivo de vértices e um arquivo de arestas como entradas. Por exemplo:

$ treemix -i inputfile.gz -m 2 -g out_stem.vertices.gz out_stem.edges.gz -o out_stem2

4.6. Gerar uma réplica de *bootstrap* (-bootstrap)

Para avaliar a confiança da topologia, é possível realizar *bootstrap*. O *bootstrapping* é realizado usando blocos de SNPs contíguos. O comando a seguir mostra como gerar uma única réplica de *bootstrap* por reamostragem de blocos de 500 SNPs:

$ treemix -i inputfile.gz -bootstrap -k 500 -o replicate1

4.7. Incorporar eventos de migração conhecidos (-cor mig e -climb)

A partir da versão 1.1, o TreeMix permite incorporar eventos de migração conhecidos. Para incluí-los, prepare um arquivo descrevendo os eventos conhecidos conforme descrito na seção sobre arquivos de entrada. Para incluir esses eventos, use a opção -cor mig. Isso permitirá construir uma árvore incluindo esses eventos. Além disso, recomendamos executar uma rodada de *hill-climbing* (pico de maior probabilidade) para otimizar as arestas de migração usando o identificador '-climb'. Para criar uma árvore incorporando eventos de migração conhecidos em um arquivo chamado, por exemplo, “4 eventos conhecidos”, use o seguinte comando:

$ treemix -i inputfile.gz -cor mig 4eventosconhecidos -climb -k 500 -o árvorecorrigida

4.8. Desativar a correção do tamanho da amostra (-noss)

Por padrão, quando o TreeMix calcula a matriz de covariância entre as populações, ele inclui uma correção para os efeitos do tamanho da amostra. Em alguns casos (por exemplo, com indivíduos isolados numa população), isto pode levar a uma sobrecorrecção. Se em sua análise há muitos ramos com comprimento zero, esse pode ser o problema. Para desativar a correção de tamanho de amostra, use o sinalizador -noss. Por exemplo:

$ treemix -i inputfile.gz -k 500 -noss -o árvorenãocorrigida

4.9. Comando 'treemix'

Digitando somente $ treemix no terminal aparecerá a seguinte mensagem:

TreeMix v. 1.13

$ Revisão: 231 $

Opções:

-h Exibe esta lista de ajuda

-i Arquivo de entrada [nome do arquivo]

-o [stem] Nome identificador dos arquivos de saída (será [stem] .treeout.gz, [stem] .cov.gz, [stem] .modelcov.gz)

-k [int] Número de SNPs por bloco para estimar a matriz de covariância (1)

-global Realiza rearranjos globais depois de adicionar todas as populações

-tf [nome do arquivo] Considera uma determinada topologia de árvore a partir de um arquivo, em vez de estimá-la

-m [int] Número de eventos de migração para adicionar (0)

-root [string] Lista de populações separadas por vírgulas para definir em um lado da raiz (para migração)

-g [nome do arquivo de vértices] [nome do arquivo de arestas] Le o gráfico de uma execução anterior do TreeMix

-se Calcula erros padrão de pesos de migração (computacionalmente caro)

-micro Dados de microsatélites

-bootstrap Executa uma única réplica de *bootstrap*

-cor_mig [arquivo] Lista de eventos de migração conhecidos para incluir (também use -climb)

-noss Desativa a correção do tamanho da amostra

-seed [int] Defini a semente para geração de números aleatórios

-n_warn [int] Exibe N avisos iniciais (N é um número inteiro e será escolhido se o identificador for colocado na linha de comando)

5. Arquivos de saída

TreeMix produzirá vários arquivos de saída. Se você usou o sinalizador -o para designar a haste de saída *out_stem*, estes serão seus arquivos de saída:

1. outstem.cov.gz. A matriz de covariância (Ŵ em Pickrell & Pritchard, 2012) entre populações estimadas a partir dos dados.
2. outstem.covse.gz. Os erros padrão para cada entrada na matriz de covariância.
3. outstem.modelcov.gz. A covariância ajustada (W em Pickrell & Pritchard, 2012) de acordo com o modelo.
4. outtem.treeout.gz. O modelo de árvore ajustado e os eventos de migração.
5. outstem.vertices.gz. Este e o arquivo seguinte (outstem.edges.gz) contêm a estrutura interna do gráfico inferido. A modificação desses arquivos causará problemas se você tentar analisar o gráfico novamente.
6. outstem.edges.gz.

A árvore inferida a partir dos dados estará em outstem.treeout.gz. A primeira linha deste arquivo é a árvore de MV no formato Newick [ex. ((a,b), c))], e as linhas restantes contêm as arestas dos eventos de migração. A primeira coluna apresenta o peso do evento de migração, seguida (opcionalmente) pela estimativa de *jackknife* do peso, a estimativa do erro padrão do *jackknife* e os valores p. Em seguida é apresentado o sentido da migração, ou seja, o ramo de origem da migração seguido do ramo de destino.

6. Visualização

6.1. Visualização de imagens

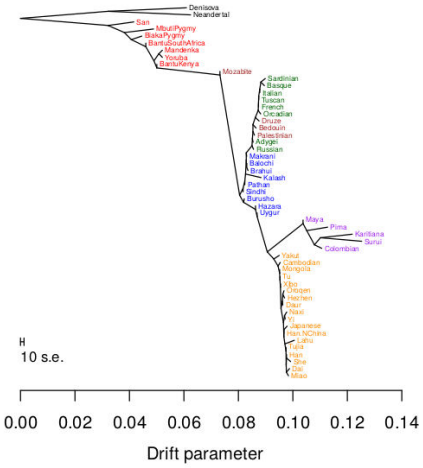
Para visualizar a imagem da árvore, use o *script* R 'plot_funcs.R', que está na pasta src / no tarball do código-fonte. A função é chamada de 'plot_tree'. Para usá-lo, a partir de R, execute:

```
$ R
R version 3.3.2 Patched (2017-01-18 r72012) -- "Sincere Pumpkin Patch"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
R é um software livre e vem sem GARANTIA ALGUMA.
Você pode redistribuí-lo sob certas circunstâncias.
Digite 'license()' ou 'licence()' para detalhes de distribuição.
R é um projeto colaborativo com muitos contribuidores.
Digite 'contributors()' para obter mais informações e
'citation()' para saber como citar o R ou pacotes do R em publicações.
Digite 'demo()' para demonstrações, 'help()' para o sistema on-line de ajuda,
ou 'help.start()' para abrir o sistema de ajuda em HTML no seu navegador.
```

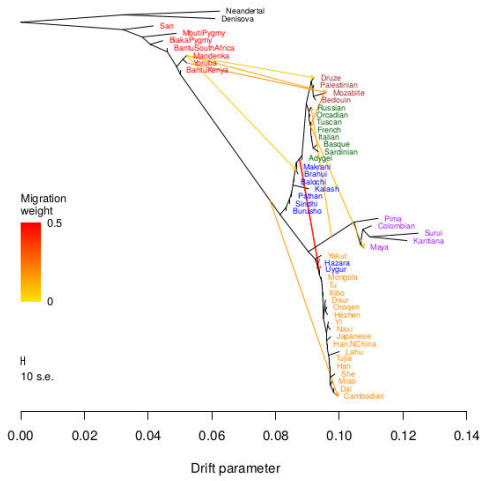
Digite 'q()' para sair do R.

```
> source ("src/plotting funcs.R")
> plot_tree ("outstem")
```

Isto resultará em uma figura como a apresentada abaixo:



Se houver eventos de migração na árvore, elas serão coloridas de acordo com seu peso, como:



Para plotar meus dados, realizei os seguintes passos

- Baixei o arquivo “RcolorBrewer_1.1-2.tar.gz” no site https://cran.r-project.org/web/packages/RColorBrewer/index.html (não possuía o pacote instalado ainda)

- Descompactei o arquivo no terminal:

$ tar -vxf RcolorBrewer_1.1-2tar. Gz

- Outra forma de instalar o RColorBrewer:

$ R

```
> if (!require("RColorBrewer")) {
install.packages("RColorBrewer")
library(RColorBrewer)
}
```

- Para abrir o R usei:

```
$ R
```

- Para carregar no R o local do diretório do RcolorBrewer digitei:

```
> source (“RcolorBrewer/”)
```

- Para carregar no R o local do *script* para plotar o TreeMix usei:

```
> source (“Documentos/Mestrado/TreeMix/plotting_funcs.R”)
```

- Dei o comando para plotar a árvore com a localização do seu *output*:

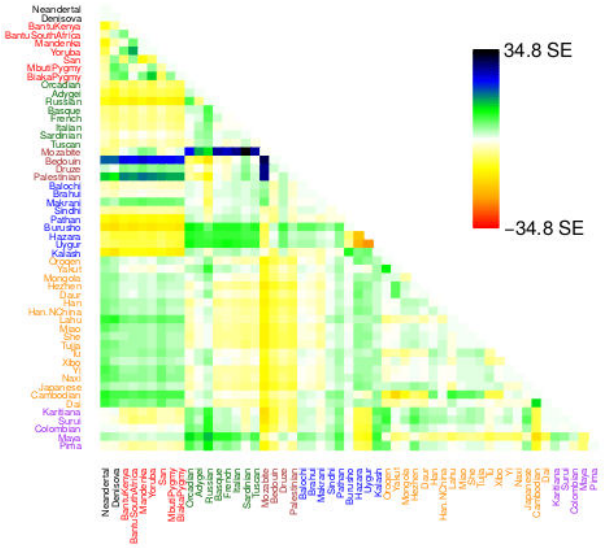
```
> plot_tree(“Documentos/Mestrado/TreeMix/Resultados/pyriglena”)
```

6.2. Visualização residual

É útil visualizar os resíduos do ajuste do modelo aos dados. Isso ajuda a identificar populações que não estão bem modeladas (devido, por exemplo, à migração adicional). Para exibir os resíduos, execute (novamente dentro de R) e depois de carregar o *script* de plotagem:

```
> plot_resid ("outstem", "poporder")
```

O arquivo "poporder" é uma lista dos nomes das populações na ordem em que você gostaria que fossem plotadas. Isto produzirá uma figura como a mostrada abaixo:



7. Dúvidas

Em casa de dúvidas, além desse tutorial traduzido e com minhas informações, consulte os tutoriais oficiais dos autores do programa (ver *link* nas referências).

8. Referências

Pickrell and Pritchard (2012). Inference of population splits and mixtures from genome-wide allele frequency data. **PLoS Genetics 8**(11): e1002967.