UNIVERSIDADE FEDERAL DE JUIZ DE FORA
PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL

FILTRAGEM ROBUSTA DE SNPS UTILIZANDO REDES NEURAIS EM DNA GENÔMICO COMPLETO

Bruno Zonovelli da Silva

Juiz de Fora
Junho de 2013

Bruno Zonovelli da Silva

Filtragem robusta de SNPs utilizando redes neurais em DNA genômico completo

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional, da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre em Modelagem Computacional.

Orientador: Prof. D.Sc. Carlos Cristiano Hasenclever Borges

Coorientador: Prof. D.Sc. Wagner Antonio Arbex

Juiz de Fora

2013

Silva, Bruno Zonovelli da

Filtragem robusta de SNPs utilizando redes neurais em DNA genômico completo/Bruno Zonovelli da Silva. – Juiz de Fora: UFJF/MMC, 2013.

XV, 101 p.: il.; 29, 7cm.

Orientador: Carlos Cristiano Hasenclever Borges

Coorientador: Wagner Antonio Arbex

Dissertação (mestrado) – UFJF/MMC/Programa de Modelagem Computacional, 2013.

Referências Bibliográficas: p. 94 – 101.

1. Bioinformática. 2. DNA Genômico. 3. Filtragem de SNP. 4. Aprendizado de Máquina. 5. Inteligência Computacional. 6. Rede Neural. I. Borges, Carlos Cristiano Hasenclever *et al.*. II. Universidade Federal de Juiz de Fora, MMC, Programa de Modelagem Computacional.

Bruno Zonovelli da Silva

Filtragem robusta de SNPs utilizando redes neurais em DNA genômico completo

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Computacional, da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do grau de Mestre em Modelagem Computacional.

Aprovada em 25 de Junho de 2013.

BANCA EXAMINADORA

Prof. D.Sc. Carlos Cristiano Hasenclever Borges - Orientador
Universidade Federal de Juiz de Fora

Prof. D.Sc. Wagner Antonio Arbex - Coorientador
Empresa Brasileira de Pesquisa Agropecuária

D.Sc. Marcos Vinícius Gualberto Barbosa da Silva
Empresa Brasileira de Pesquisa Agropecuária

Prof. D.Sc. Raul Fonseca Neto
Universidade Federal de Juiz de Fora

Dedico este trabalho a minha esposa Débora e a minha filha Bruna Karla.

AGRADECIMENTOS

Agradecimentos, sim, essa parte tão importante do trabalho aonde nos lembramos daqueles que ficaram ao nosso lado, durante a construção desse trabalho. Pessoas essas especiais, merecedoras de mais que simples linhas nessa humilde trabalho, porém, pela falta de como fazer tamanho agradecimento, fica aqui registrados, os nomes das pessoas que foram a base para que hoje eu pudesse escrever essas poucas linhas.

Muitas são as pessoas a quem desejo agradecer. Primeiramente agradeço a meu Deus, por guiar meu caminho até aqui. Sendo meu guia e meu companheiro em todos os momentos, matérias e escolhas, sendo sempre meu refugio nos momentos de dúvidas e angustias.

A minha esposa Débora Cristina, mulher e companheira, minha inspiração para prosseguir a cada passo dado. Sempre ao meu lado, desde o começo até hoje me incentivando a prosseguir. Débora Te AMO, mais do que a mim. E obrigado pelo meu presentinho lindo, que é a minha filhinha Bruna Karla, que mesmo tão pequenina, ocupa um lugar imenso no meu coração.

Os amigos, sim, eles, pessoas especiais, que te acompanham, te ajudam, e te escutam. Quero agradecer a todos. Todos que nesses 2 anos, me ouviram falar somente do mestrado. Mais em especial aos companheiros Marcelo, Acaccio, Daiana e Denise. Que foram mais que amigos nesses 2 anos caminharam comigo os longos percursos para a conquista do tão sonhado título. Ouviram-me, e como me ouviram, me ajudaram, e principalmente me inspiraram. Hoje levo um pedaço de cada um de vocês junto comigo, pois a perseverança, vontade e garra de cada um me motivou a prosseguir mesmo quando o obstáculo parecia impossível.

Não poderia deixar de citar outros nomes, como Bruno Novaes, sendo sempre prestativo, e me ajudando nas mais variadas dúvidas sobre C. Ao grande mestre Fabrizzio, pessoa que aprendi a respeitar, não por sua inteligência, que, diga-se de passagem, é grande, mais por seu caráter e disposição, obrigado pelos conselhos. Ao Vinícius, pela ajuda, e pelas dicas sobre biologia.

A todos os outros que não citei, saibam que não é por esquecer, pois guardo todos em minhas melhores lembranças. E que Deus possa retribuir a cada um todas as ajudas que me deram.

Aos orientadores, Carlos Cristiano e Wagner Arbex. Pela confiança a mim depositada, por me ouvirem, orientarem, mostrando o caminho a ser seguido, porém, deixando livre para traçar o meu próprio caminho, sempre se posicionando mais como conselheiros do que autoridades. Obrigado, hoje eu sei o tamanho da responsabilidade que é assumir um aluno, assinar por ele e dizer que ele irá concluir uma tarefa, e obrigado por confiar em mim.

A Fernanda Almeida pela paciência e auxilio na correção do trabalho, pelas orientações e sugestões sempre pertinentes e importantes.

Aos professores, pela paciência e dedicação oferecidas aos alunos, e pela disposição em me atenderem e me instruírem, nos mais variados assuntos, e por muitas vezes repetidamente. Mesmo assim estavam sempre dispostos. Em especial, queria deixar meus agradecimentos a Priscila por todos os conselhos, dicas, ajudas e conversas que tive com ela.

A CAPES, pelo financiamento da minha pesquisa, ajuda essa sem a qual não seria possível nem começar esse trabalho, quem dera escrever esses agradecimentos.

A UFJF e ao PGMC pela oportunidade a mim oferecida.

Deixo aqui registrado meus agradecimentos, a todos que direta ou indiretamente, me ajudaram nessa conquista. Mesmo que essa seja uma página pouco lida, deixo registro os nomes daqueles que durante essa etapa da minha vida, foram de alguma forma importantes.

"Bem-aventurado o homem que acha sabedoria, e o homem que adquire conhecimento; Porque é melhor a sua mercadoria do que artigos de prata, e maior o seu lucro que o ouro mais fino. Mais preciosa é do que os rubis, e tudo o que mais possas desejar não se pode comparar a ela."

Provérbios 3:13-15

RESUMO

Com o crescente avanço das plataformas de sequenciamento genômico, surge a necessidade de modelos computacionais capazes de analisar, de forma eficaz, o grande volume de dados disponibilizados. Uma das muitas complexidades, variações e particularidades de um genoma são os polimorfismos de base única (single nucleotide polymorphisms - SNPs), que podem ser encontrados no genoma de indivíduos isoladamente ou em grupos de indivíduos de alguma população, sendo originados a partir de inserções, remoções ou substituições de bases.

Alterações de um único nucleotídeo, como no caso de SNPs, podem modificar a produção de uma determinada proteína. O conjunto de tais alterações tende a provocar variações nas características dos indivíduos da espécie, que podem gerar alterações funcionais ou fenotípicas, que, por sua vez, implicam, geralmente, em consequências evolutivas nos indivíduos em que os SNPs se manifestam.

Entre os vários desafios em bioinformática, encontram-se a descoberta e filtragem de SNPs em DNA genômico, etapas de relevância no pós-processamento da montagem de um genoma. Este trabalho propõe e desenvolve um método computacional capaz de filtrar SNPs em DNA genômico completo, utilizando genomas remontados a partir de sequências oriundas de plataformas de nova geração. O modelo computacional desenvolvido baseia-se em técnicas de aprendizado de máquina e inteligência computacional, com o objetivo de obter um filtro eficiente, capaz de classificar SNPs no genoma de um indivíduo, independente da plataforma de sequenciamento utilizada.

Palavras-chave: Bioinformática. DNA Genômico. Filtragem de SNP. Aprendizado de Máquina. Inteligência Computacional. Rede Neural.

ABSTRACT

With the growing advances in genomic sequencing platforms, new developments on computational models are crucial to analyze, effectively, the large volume of data available. One of the main complexities, variations and peculiarities of a genome are single nucleotide polymorphisms (SNPs). The SNPs, which can be found in the genome of isolated individuals or groups of individuals of a specific population, are originated from inserts, removals or substitutions of bases.

Single nucleotide variation, such as SNPs, can modify the production of a protein. Combination of all such modifications tend to determine variations on individuals characteristics of the specie. Thus, this phenomenon usually produces functional or phenotypic changes which, in turn, can result in evolutionary consequences for individuals with expressed SNPs.

Among the numerous challenges in bioinformatics, the discovery and filtering of SNPs in genomic DNA is considered an important steps of the genome assembling post-processing. This dissertation has proposed and developed a computational method able to filtering SNPs in genome, using the genome assembled from sequences obtained by new generation platforms. The computational model presented is based on machine learning and computational intelligence techniques, aiming to obtain an efficient filter to sort SNPs in the genome of an individual, regardless of the sequencing platform adopted.

Keywords: Bioinformatics. Genomic DNA. SNP Filtering. Machine Learning. Computational Intelligence. Neural Network.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

Lista de Algoritmos

1 INTRODUÇÃO

Com o crescente avanço das plataformas de sequenciamento genômico, surge a necessidade de modelos computacionais capazes de analisar, de forma eficaz, o grande volume de dados disponibilizados. A maior parte do genoma entre os indivíduos de uma mesma espécie é idêntica, porém, existe a variabilidade genética, que são as diferenças encontradas em algumas regiões do genoma (BRONDANI; BRONDANI, 2004). A variabilidade pode surgir devido a alteração nas sequências de bases ao longo do DNA, ocorrendo por: substituição, ausência ou duplicação de bases e, os polimorfismos de base única (*single nucleotide polymorphisms* - SNPs). Os SNPs são diferenças pontuais entre pares de bases de diferentes sequências alinhadas, sendo o tipo mais comum de variabilidade genética (CONSORTIUM, 2003). Assim, tais diferenças são importantes no estudo da variabilidade das espécies, pois, podem provocar alterações funcionais ou fenotípicas, que, por sua vez, podem implicar em consequências evolutivas ou bioquímicas nos indivíduos em que os SNPs se manifestam.

As aplicações mais comuns relacionadas ao estudo e à identificação de SNPs são encontradas nos trabalhos que objetivam correlacionar genótipo e fármacos, a definição de marcadores de predisposição a determinadas patologias e de sensibilidade a diferentes tratamentos. Contudo, atualmente, outras ciências não muito próximas da genética ou da bioinformática também utilizam as ferramentas de estudo, identificação e análise de SNPs, empregando os resultados em áreas como medicina forense, antropologia molecular, evolução, genética de populações, conservação e manejo de fauna.

A correta identificação dos SNPs é um importante passo para seu uso em outros estudos, porém, para sua correta identificação pode ser necessário um processo de filtragem. A filtragem de SNPs em dados provenientes de plataformas de nova geração se apresenta como uma linha de pesquisa onde existe a necessidade de novos desenvolvimentos. Especificamente, filtros baseados em estratégias de aprendizado de máquina e inteligência computacional, que basicamente não são explorados, sendo esta uma das metas deste trabalho. Para isto, apresenta-se, neste capítulo, algumas informações preliminares e a definição de conceitos biológicos necessários para entendimento do processo de sequenciamento genômico e posterior filtragem de SNPs.

1.1 Considerações Preliminares

No final da década de 70, foram desenvolvidos dois métodos clássicos de sequenciamento do DNA, o método de degradação química ou procedimento de Maxam e Gilbert (1977) e o método de degradação enzimática ou procedimento de Sanger (SANGER; NICKLEN; COULSON, 1977). Tais técnicas empregam processos químicos para identificar e determinar a ordem das bases nitrogenadas no DNA de um organismo. Mas, devido a facilidade de interpretação dos dados provenientes do método desenvolvido por Frederick Sanger, sua técnica foi amplamente utilizada pelos grupos interessados no sequenciamento do DNA. Entretanto, o alto custo e o baixo rendimento inerente desse método se tornou um fator limitante para os projetos que visam o sequenciamento genômico em larga escala (CHEN et al., 2013).

A partir de 2005, as tecnologias de sequenciamento sofreram um considerável avanço, redução de custos e aumento da capacidade de sequenciamento. Hoje, as novas plataformas de sequenciamento conhecidas como sequenciamento de nova geração (*next-generation sequencing* - NGS), se tornaram opções eficazes para a utilização rotineira em projetos de sequenciamento e ressequenciamento de genomas individuais (SERVICE, 2006; GUPTA, 2008). Essas plataformas representam uma alternativa poderosa para a detecção de variações entre o genoma-alvo e o de referência, para os estudos de genômica estrutural e funcional (MARDIS, 2008; MOROZOVA; MARRA, 2008). São capazes de gerar informações sobre milhões de sequências (*reads*) em uma única corrida (ZHANG et al., 2011; CHEN et al., 2013). Nesse sentido, existe a exigência da aplicação de algoritmos robustos para a montagem do genoma de interesse.

O sequenciamento do genoma constitui uma importante etapa para o desenvolvimento de pesquisas genômicas mais detalhadas, que podem envolver uma diversidade de estudos, tais como: associação de doenças, filogenéticos, de assinaturas genômicas, dentre outros. Neste aspecto, a investigação de SNPs, destina-se a entender se a diferença pontual entre o genoma de dois indivíduos (o *mismatch*) ocorreu de um erro de leitura proveniente do sequenciamento, de um erro no alinhamento, ou de uma mutação ou SNP (ARBEX, 2009). Assim, uma das etapas de um projeto de sequenciamento de um genoma é a etapa de descoberta de SNPs.

A descoberta de SNPs por algoritmos computacionais é uma prática bastante difundida e, nessa área, destacaram-se, pelo amplo uso, os programas Polyphred (NICKERSON;

TOBE; TAYLOR, 1997) e Polybayes (MARTH et al., 1999), que foram amplamente utilizados quando o método Sanger era uma tecnologia de sequenciamento de uso corrente. Contudo, as plataformas de NGSs possuem seus próprios recursos para investigação de SNPs, onde cada empresa disponibiliza ferramentas e recursos de computação específicos para a identificação de SNPs, levando o Polyphred e Polybayes ao desuso. Todavia, ressalta-se que os recursos disponibilizados pelas plataformas de NGS são proprietários, "fechados" e restringem-se às sequências produzidas pelas mesmas.

É sabido que, em cada etapa destinada ao sequenciamento do DNA um erro pode ser introduzido, mesmo que em porções pequenas. Entretanto, tais erros podem ocasionar a identificação equivocada de um SNP. Para solucionar esse problema, filtros para identificação de SNPs vêm sendo construídos, vinculados ou não a *software* de alinhamento e mapeamento de sequências, que são utilizados na montagem do genoma de um determinado organismo. Dentro desse cenário, destaca-se o software MAQ (*Mapping and Assembly with Quality*), considerado um dos principais programas destinados ao alinhamento de genomas disponíveis atualmente. Tal programa visa o mapeamento e a montagem de genomas completos sequenciados por meio de plataformas NGS (LI; RUAN; DURBIN, 2008), além de possuir um filtro de SNPs acoplado.

A Embrapa Gado de Leite desenvolve trabalhos voltados para todas as dimensões do agronegócio do leite e nos últimos anos parte dos trabalhos de melhoramento genético animal baseiam-se em estudos de genômica para avaliação e seleção de animais com características de interesse econômico. Entre esses estudos, encontram-se o projeto "Seleção Genômica em Raças Bovinas Leiteiras no Brasil - GENOMILK" e suas ações e atividades relacionadas.

O referido projeto faz parte da carteira de projetos da Embrapa, registrado no Sistema Embrapa de Gestão (SEG), sob o código SEG 02.09.07.008.00.00. Esse projeto encontra-se com várias ações já desenvolvidas e outras em desenvolvimento e, ainda, permite estabelecer uma rede de pesquisa com várias instituições de pesquisa e universidades, envolvendo dezenas de profissionais da Embrapa e das instituições parceiras, tal como, a Universidade Federal de Juiz de Fora.

Os estudos e trabalhos realizados para essa dissertação são parte das ações do GENOMILK, em específico, nas atividades do projeto "Modelos computacionais de mineração de dados para prospecção de SNP", onde são propostos métodos computacionais para a

investigação de SNPs, como marcadores moleculares de regiões do genoma onde podem ser encontradas informações sobre as características e o potencial genético desejáveis.

A proposta dessa dissertação foi desenvolvida sobre o genoma montado de um animal da raça Fleckvieh, utilizando como referência o genoma bovino bosTau4.0 (HGSC, 2007) e que, futuramente, será utilizado sobre a montagem do genoma do zebú leiteiro, para a identificação de marcadores específicos para as espécies e subespécies zebuínas.

Atualmente o número estimado de SNPs em genoma bovino está na casa de 12 milhões, porém, as diferenças pontuais entre dois genomas recém-montados podem ser de 3 a 4 vezes o número de SNPs antes da etapa de filtragem. O conhecimento do genoma dessas raças, aliado a ferramentas computacionais e de melhoramento genético, poderão gerar saltos de produtividade e de qualidade, contribuindo para o crescimento sustentável da pecuária de leite brasileira.

1.2 Conceitos Biológicos

O avanço nas pesquisas relativas à DNA abriram oportunidades, antes desconhecidas, de estudo em vários processos biológicos conhecidos, transformando a pesquisa, agropecuária, médica, agrícola, ecológica, médica legal entre tantas outras. A clonagem do DNA é definida como um dos principais desenvolvimentos na área de bioquímica e biologia molecular (LEHNINGER; COX, 2011).

A estrutura do DNA consiste em uma molécula com duas longas cadeias polipeptídicas conhecidas como cadeias ou fitas complementares de DNA, compostas por quatro subunidades ou bases, que pode ser: adenina (A), citosina (C), guanina(G) ou timina(T), chamadas de nucleotídeos. A sequência de nucleotídeos do código genético é traduzida e organizada em tripletos, conhecidos como códons, que codificam aminoácidos que serão traduzidos em proteína. A metionina, codificada pela sequência ATG, é o códon iniciador da síntese de uma proteína. Os códons (TAA, TAG, TGA) não produzem aminoácidos, pois são sinais de parada da síntese de uma proteína. A Tabela 1.1 mostra os aminoácidos possíveis a partir de uma sequência de DNA. Caso seja utilizado RNA, substitui-se a base T(timina) por U(uracila).

A série completa de informações do DNA, o genoma, contém tudo o que é necessário para a síntese de proteínas e moléculas durante toda a vida do indivíduo. Somente cerca

Tabela 1.1: Tabela de códons.

	T		C		A		G		
T	TTT	Fenilalanina (F)	TCT	Serina (S)	TAT	Tirosina (Y)	TGT	Cisteína (C)	T
	TTC		TCC		TAC		TGC		C
	TTA	Leucina (L)	TCA		TAA	Códon de parada	TGA	Códon de parada	A
	TTG		TCG		TAG		TGG	Triptofano (W)	G
C	CTT	Leucina (L)	CCT	Prolina (P)	CAT	Histidina (H)	CGT	Arginina (R)	T
	CTC		CCC		CAC		CGC		C
	CTA		CCA		CAA	Glutamina (Q)	CGA		A
	CTG		CCG		CAG		CGG		G
A	ATT	Isoleucina (I)	ACT	Treonina (T)	AAT	Asparagina (N)	AGT	Serina (S)	T
	ATC		ACC		AAC		AGC		C
	ATA		ACA		AAA	Lisina (K)	AGA	Arginina (R)	A
	ATG	Metionina (M)	ACG		AAG		AGG		G
G	GTT	Valina (V)	GCT	Alanina (A)	GAT	Ácido Aspártico (D)	GGT	Glicina (G)	T
	GTC		GCC		GAC		GGC		C
	GTA		GCA		GAA	Ácido Glutâmico (E)	GGA		A
	GTG		GCG		GAG		GGG		G

de 3% do genoma humano codifica proteínas, regiões conhecidas como "éxon", sendo o restante, parte não codificadora, conhecida como "íntron" (ALBERTS et al., 2010). O DNA é uma molécula, bastante longa, com alguns cromossomos humanos possuindo cerca de 5.10^8 pares bases (pb), o que torna o processo de identificação das sequências a primeira dificuldade enfrentada no projeto genoma humano (DIAS NETO, 2004).

Crick (1958) fez uma série de propostas teóricas, principalmente a de que a informação genética segue um fluxo determinado, o que ficou conhecido como dogma central da biologia, onde foram definidos três importantes processos: a transcrição, a tradução e a replicação. O processo conhecido como transcrição utiliza a informação presente no DNA para sintetizar a molécula de RNA, que é usada para a síntese de proteínas através do processo chamado de tradução. Outro processo abordado é a replicação ou "duplicação" da molécula de DNA (STANSFIELD; COLOMÉ; CANO, 1998), onde cada fita complementar atua como molde para a duplicação do DNA, que copia, com precisão, todas as informações presente para uma nova molécula de DNA. A taxa de erro presente é de uma base a cada replicação. Essa ação é responsável por transmitir as informações hereditárias de um indivíduo para um novo indivíduo, bem como a manutenção da vida do mesmo (ALBERTS et al., 2010).

Atualmente já são conhecidos outros fluxos como a transcrição reversa, a replicação do RNA e a tradução direta de DNA em proteína, conforme mostrado na Figura 1.1. O

processo de transcrição reversa consiste em passar a informação do RNA para o DNA, podendo ser feita por "retrovírus" como o HIV. A tradução direta do DNA em proteína, sem o processo de transcrição, ainda é pouco conhecido, mas já possível de ser feito em laboratório. Já o processo de replicação do RNA é detectado em alguns vírus e plantas (STANSFIELD; COLOMÉ; CANO, 1998). Esses processos foram adicionados ao dogma central da biologia, compondo-o como o conhecemos atualmente.

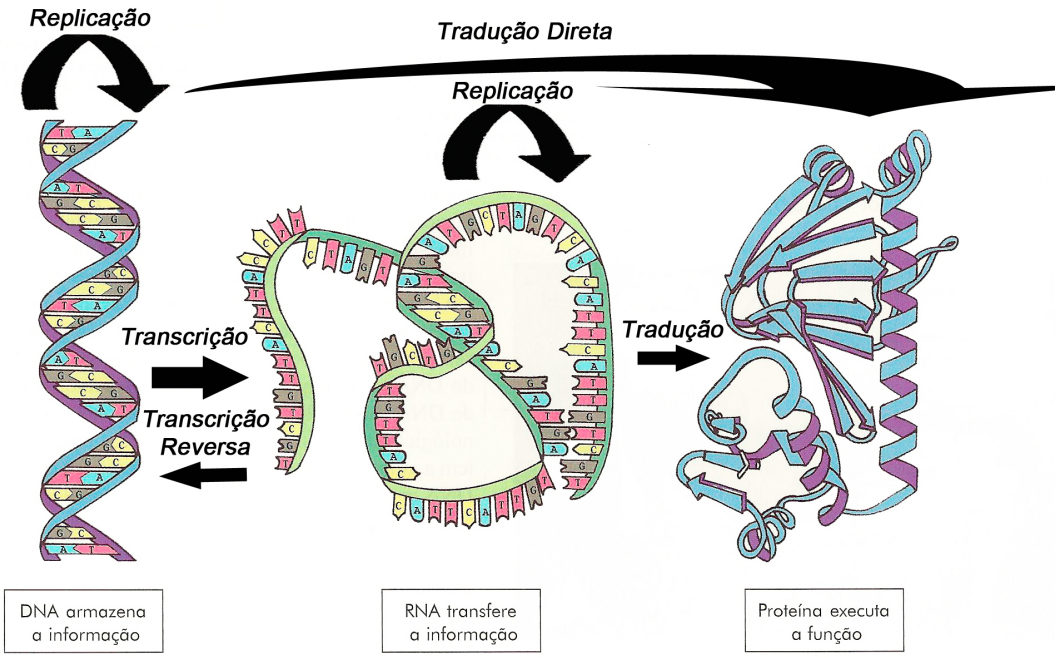


Figura 1.1: Dogma central da biologia atualizado - (**Fonte:** *Domínio publico)* .

A individualidade genética tem como uma das consequências as mutações, que constituem certamente uma das maiores descobertas dos projetos de sequenciamento, principalmente do Projeto Genoma Humano, pois nosso código genético se mostrou mais variado e complexo do que propriamente maior do que os das outras espécies. Outro fator de interesse reside no fato de que dois genomas humanos são $99,9\%$ iguais, porém a fração restante é que nos diferencia (DIAS NETO, 2004). Essa individualidade apresenta-se, em um contexto mais amplo, como objeto de interesse desse trabalho.

1.3 Objetivos

Esta dissertação visa o desenvolvimento de uma ferramenta computacional destinada a classificação de SNPs em genomas completos e já montados, advindos de quaisquer pla-

taformas NGS. Objetiva-se desenvolver um modelo, baseado em técnicas de aprendizado de máquina e inteligência computacional, capaz de classificar em candidatos "fortes" ou "fracos" os SNPs encontrados no genoma completo dos organismos de interesse. Desta forma, pretende-se melhorar a capacidade de classificação dos SNPs em relação aos filtros tradicionais.

Modelos de classificação supervisionada para filtragem de SNPs ainda não são explorados na literatura especializada. Entre os possíveis motivos está a dificuldade de se ter uma base de dados confiável, tanto para falsos positivos como para SNPs comprovados, para a obtenção da hipótese de generalização. Assim, qualquer tentativa de se utilizar classificação supervisionada para a filtragem de SNPs deve passar, necessariamente, pela definição de uma estratégia eficaz para a construção da base de treinamento e/ou determinação da classe das instâncias (BASHEER; HAJMEER, 2000).

A construção de um modelo de classificação supervisionada só é efetiva com a definição de boas estratégias no processo de treinamento do modelo. Nessa dissertação foram elaboradas três estratégias de treinamento, cada qual gerando um novo modelo de classificação supervisionada. Os modelos são baseados: i) na utilização de uma pré-filtragem para determinação das classes; ii) na construção de bases específicas para maximizar o poder de generalização de uma ferramenta de classificação supervisionada. iii) na construção de bases específicas utilizando algumas regras da pré-filtragem.

Cada um destes modelos será viabilizado por meio de redes neurais, ferramenta de inteligência computacional aplicada em problemas de classificação e/ou regressão. Tal escolha se deu pelo potencial das redes neurais na representação e generalização em problemas de aprendizado supervisionado (HAYKIN, 2001).

Tem-se, como objetivo final, a obtenção de uma estratégia para filtragem de SNPs, baseada em aprendizado de máquina e inteligência computacional, que seja competitiva com filtros tradicionais como, por exemplo, o filtro SNPfilter acoplado ao código do MAQ. Para isto, os resultados obtidos pelo programa MAQ são utilizados como referência para a comparação com os modelos desenvolvidos neste trabalho. Nos experimentos são utilizados os genomas de dois organismos, um bovino da raça taurina Fleckvieh (ECK et al., 2009) e da planta modelo *Arabidopsis thaliana* germoplasmas "BUR-0" e "TSU-1" (INITIATIVE, 2000). A seguir, apresenta-se como o trabalho foi estruturado visando uma melhor compreensão de seu desenvolvimento.

1.4 Organização do Trabalho

A definição de uma boa estratégia de treinamento do modelo gera a necessidade de se definir um bom conjunto de dados para esse processo. Porém, existe uma dificuldade na montagem de conjunto de dados com informação sobre SNPs, que em geral são obtidos após a etapa de montagem do genoma do indivíduo, sendo necessário também o seu entendimento. Para facilitar a compreensão do desenvolvimento do trabalho ele foi dividido em três etapas. A primeira etapa consiste em remontar os genomas de interesse para a obtenção dos arquivos necessários para a montagem dos conjuntos de dados utilizados na etapa de treinamento. A segunda etapa consiste em analisar o arquivo obtido na etapa de identificação dos SNPs de forma a extrair as informações necessárias para a construção do modelo de aprendizado de máquina. A terceira e última etapa consiste em construir o modelo, testá-lo e comparar os resultados obtidos. O trabalho desenvolvido nessa dissertação foi distribuído em sete capítulos.

O **Capítulo 1** apresenta uma introdução, os conceitos biológicos necessários para o entendimento do problema, assim como os objetivos a serem alcançados.

O **Capítulo 2** desenvolve a parte teórica e prática da primeira etapa de desenvolvimento dessa dissertação. A parte teórica do capítulo é a descrição de todo o processo de sequenciamento de DNA, da geração anterior e da nova, bem como os algoritmos utilizados para a montagem e alinhamento dessas sequências. A parte prática demonstra o processo de remontagem dos genomas de interesse, etapa essa de grande importância para o desenvolvimento do trabalho, pois, os arquivos obtidos servem de base para o modelo de aprendizado de máquina.

O **Capítulo 3** delineia a segunda etapa de desenvolvimento do trabalho, que consiste em analisar o arquivo obtido na etapa de identificação de SNPs. Nesse capítulo é definido o problema de classificação dos *mismatches*, mostrando os erros gerados nas diferentes etapas do processo de sequenciamento, apresentando os filtros disponíveis, com ênfase para o filtro desenvolvido pelo software de alinhamento utilizado nesse trabalho. A análise serviu de base a para a definição das estratégias de filtragem utilizadas para o treinamento do modelo de aprendizado.

No **Capítulo 4** são definidos os conceitos relativos à estratégia de aprendizado de máquina utilizada para a classificação dos *mismatches*. O **Capítulo 5** apresenta a terceira e última etapa, que é o desenvolvimento do modelo de aprendizado de máquina, além da

forma como foram montados os conjuntos de dados para o treinamento do modelo de aprendizado.

O **Capítulo 6** apresenta os resultados dos vários experimentos computacionais realizados. Finalmente, no **Capítulo 7** são delineadas algumas conclusões de interesse e diretrizes para futuros desenvolvimentos.

2 SEQUENCIAMENTO DE DNA E MONTAGEM DE GENOMAS COMPLETOS

A primeira etapa de desenvolvimento do trabalho consiste na remontagem dos genomas de interesse. Porém, é necessário entender o processo de sequenciamento do DNA, além do processo de montagem dos fragmentos para a obtenção da sequência completa do DNA. Esse capítulo apresenta a teoria para entender os processos de sequenciamento e montagem, bem como o próprio processo de remontagem.

O sequenciamento do DNA é um processo que determina a ordem dos nucleotídeos, em uma dada sequência, a partir de uma amostra biológica. Existem vários métodos disponíveis que visam o sequenciamento sendo, um dos mais utilizado, o Método de Sanger. Esse procedimento foi a alternativa metodológica empregada no projeto de sequenciamento do genoma humano.

O método de Sanger realiza o sequenciamento a partir de uma fita simples do DNA que servirá de molde para gerar uma fita complementar. Este processo ocorre pela desnaturação da "molécula nativa" do DNA de interesse, cada produto da reação contém uma marcação diferente, permitindo a identificação dos nucleotídeos no processo de análise.

Atualmente, as tecnologias que visam o sequenciamento do DNA sofreram grandes avanços e são capazes de gerar dados de milhões de pares de bases em uma única corrida. As NGSs são fundamentadas no método de Sanger e estão sendo amplamente empregadas por serem procedimentos menos custosos e mais velozes do que os métodos clássicos de sequenciamento.

2.1 Plataformas de Sequenciamento de Nova Geração

Este seção iniciará expondo brevemente três plataformas de NGS, são elas: 454, SOLEXA e SOLiD. Serão apresentados os principais fundamentos e aplicações de cada uma delas. Também será feita uma explanação sobre a aplicação de recursos computacionais

como solução para problemas biológicos, em especial, para problemas de alinhamento e montagem de genomas.

As tecnologias de sequenciamento de nova geração tiveram suas primeiras versões comercializadas a partir de 2005, e desde então continuaram a evoluir rapidamente. Essas tecnologias sequenciam o DNA em plataformas capazes de gerar dados de milhões de pares de bases em uma única corrida. Todas podem gerar informação em um volume muitas vezes maior que o sequenciamento Sanger, com grande economia de tempo e custo por base. Essa maior eficiência é resultado do uso da clonagem *in vitro* ou em sistemas de suporte sólido, permitindo que milhares de leituras possam ser produzidas de uma só vez.

2.1.1 A Plataforma 454

A plataforma 454 foi a primeira a ser comercializada. Seu sequenciamento utiliza a síntese por pirosequênciamento, que consiste em uma combinação enzimática, iniciada com a liberação de um pirofosfato, que ao ser convertido em ATP produz um sinal luminoso após ser oxidado. O sequenciamento pode ser dividido em três etapas: o preparo da amostra; a reação de polimerase em cadeia (*Polymerase Chain Reaction* - PCR) em emulsão; e o sequenciamento (RONAGHI; UHLÉN; NYRÉN, 1998).

O preparo consiste em fragmentar o DNA aleatoriamente e conectar adaptadores A e B em suas extremidades. Os fragmentos A e B são específicos para cada sequência. Os fragmentos são ligados às microesferas magnéticas por meio do pareamento com sequências curtas complementares presentes na superfície da microesfera. Apenas um único tipo de fragmento se liga a uma determinada microesfera. As microesferas são capturadas individualmente em gotículas oleosas onde a PCR em emulsão ocorre. Milhares de cópias do fragmento alvo são produzidas nessa fase. As microesferas ligadas às sequências alvo são capturadas individualmente em poços no suporte de sequenciamento. São fornecidos os reagentes para a reação de pirosequênciamento, e o sinal de luz emitido é identificado a cada base incorporada (MARGULIES et al., 2005).

A placa de sequenciamento é inserida no sistema óptico de leitura, onde são lidos a cada ciclo 1,6 milhões de poços paralelamente. A cada ciclo um nucleotídeo é adicionado a reação, se ele for incorporado a sequência em síntese, ocorre a emissão de um sinal de luz, a intensidade do sinal é reflexo do número de nucleotídeos incorporados a molécula.

Como o nucleotídeo que é adicionado a cada ciclo é conhecido, o sinal de luz emitido pode ser diretamente utilizado como a informação da sequência (RONAGHI, 2001).

Os *reads* produzidos possuem em torno de 400pb, um comprimento de leitura menor que o produzido pelo sistema de Sanger ($\approx 700pb$) (ROCHE, 2008). Com relação às demais tecnologias de sequenciamento da segunda geração, a plataforma 454 é a que produz os maiores *reads* (WICKER et al., 2009).

2.1.2 A Plataforma SOLEXA

O sequenciamento na plataforma Solexa é realizado por síntese usando DNA polimerase e nucleotídeos terminadores marcados, assim como o sequenciamento de Sanger. A inovação esta no fato de que a clonagem dos fragmentos é feita *in vitro*, ou seja, utiliza uma plataforma sólida de vidro, processo conhecido como PCR de fase sólida (FEDURCO et al., 2006; TURCATTI et al., 2008). Onde são afixados adaptadores a superfície de clonagem (*flow cells*), eles são fixados pela extremidade 5', deixando a extremidade 3' livre para servir de ponto de início da reação de sequenciamento, e são imobilizados no suporte por hibridização. O DNA então é aleatoriamente fragmentado, e ligado aos adaptadores A e B em ambas as extremidades. Os fragmentos ligados aos adaptadores permitem sua fixação, por afinidade, ao suporte de sequenciamento, que possui uma alta densidade de oligonucleotídios complementares aos adaptadores A e B (TURCATTI et al., 2008).

Na etapa de anelamento ocorre o primeiro ciclo de amplificação da PCR em fase sólida, onde o adaptador da extremidade livre da molécula aderida ao suporte encontra seu oligonucleotídio complementar, formando uma estrutura em ponte. Nucleotídeos não marcados são fornecidos para que haja a síntese da segunda fita do fragmento imobilizado no suporte. Uma vez fornecidos os reagentes necessários, é iniciada a PCR utilizando a extremidade 3' livre do oligonucleotídio como *primer*. Ao fim do ciclo de anelamento, ocorre a formação de uma estrutura em "ponte", do fragmento e sua fita complementar, na superfície de sequenciamento. O aumento da temperatura, na etapa de desnaturação, rompe as "pontes", separando e linearizando as fitas de DNA (SHENDURE; JI, 2008).

A etapa de anelamento é repetida, formando assim novas estruturas em "ponte" e iniciando um novo ciclo de amplificação. Esses ciclos são repetidos 35 vezes, gerando cerca de mil cópias de cada fragmento na fase de PCR sólida, formando um *cluster* de

sequenciamento. A alta densidade dos *clusters* de sequenciamento possibilita que o sinal de fluorescência gerado com a incorporação de cada um dos nucleotídeos terminadores tenha uma intensidade suficiente para garantir sua detecção (TURCATTI et al., 2008).

Com a excitação a laser dos nucleotídeos marcados, um sinal é gerado e captado por um dispositivo de leitura, sendo então interpretado como um dos quatro nucleotídeos possíveis. Esse processo é repetido para cada nucleotídeo que compõem a sequência. Até 50 milhões de *clusters* podem ser produzidos por linha, correspondendo a uma representação satisfatória da biblioteca. Em geral, leituras de 25-35 pb são obtidas de cada *cluster* (SHENDURE; JI, 2008).

2.1.3 A Plataforma SOLiD

O sistema SOLiD (MCKERNAN et al., 2011), difere dos outros, pois utiliza como catalisador uma DNA ligase, e não uma polimerase. O processo se inicia com a fragmentação mecânica do DNA-alvo, com 60-90pb para as bibliotecas de *tags* únicas, ou 1-10Kb para as bibliotecas de *tags* duplas (*mate-pair*), e a ligação de adaptadores universais(P1 e P2) em ambas as extremidades dos fragmentos. Ocorre então a PCR de emulsão, amplificando os fragmentos e permitindo sua ligação por hibridação a microesferas metálicas, que são ligadas a lâminas de vidros, sendo utilizadas duas lâminas por corrida, cada uma com capacidade para cem mil microesferas (MOROZOVA; MARRA, 2008).

O sequenciamento possui etapas distintas, que se iniciam com n bases na primeira etapa, sendo diminuída uma base a cada etapa até a quinta. A primeira e a segunda base de cada sonda são chamadas bases seletivas, as restantes são degeneradas. Por isso na primeira etapa, ocorre a adição do *primer* universal completo, com o anelamento exato. A sonda complementar se hibridizará com a sequência molde dentro do *pool* de sondas pela ação da ligase que se ligará ao *primer* universal. Essa plataforma produz *reads* de 35pb para as bibliotecas de *tag* única e de 50pb para as de *mate-pair* (GLENN, 2011).

Cada sinal de fluorescência indica um dinucleotídeo e não uma única base, a decodificação desses sinais é feita combinando-se os dados. Com o conhecimento das bases dos adaptadores P1, é possível identificar corretamente a primeira base do fragmento durante a segunda etapa. Os demais sinais de fluorescência são especificados pela única combinação possível de cores que inclui a base conhecida. Esse sistema de leitura é muito eficiente na detecção de polimorfismos (SNPs), que em outras plataformas podem ser confundidos

com erros de sequenciamento (MOROZOVA; MARRA, 2008). As leituras produzidas com o SOLiD apresentam acurácia superior às demais técnicas, sendo perfeitamente adequadas à identificação de polimorfismos genômicos reais (MARDIS, 2008).

2.2 Montagem e Alinhamento de sequências de DNA

A montagem do genoma a partir de sequências de DNA é uma tarefa exclusivamente computacional. Tendo seu início com a leitura dos arquivos originados das máquinas de sequenciamento, que após o tratamento correto, contêm as sequências de nucleotídeos e podem conter ou não as informações relativas a qualidade de sequenciamento, eles são conhecidos como FASTA, quando contém somente os nucleotídeos, e FASTQ quando contêm também a informação de qualidade.

O arquivo em formato FASTA foram desenvolvidos inicialmente para servirem de entrada para o software com mesmo nome desenvolvido por Pearson e Lipman (1988), se tornando padrão para algoritmos de alinhamento de sequência. Ele se inicia com a linha de descrição (*defline*) que possui o sinal de maior (">") como carácter iniciador, e na linha seguinte à sequência de nucleotídeos referente à descrição fornecida. Uma sequência de exemplo no formato FASTA, pode ser visto na Figura 2.1.

```
>E.coliK-12-MG1655
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC
TTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTATTGACTTAGGTCACTAAATACTTTAACCAA
TATAGGCATAGCGCACAGACAGATAAAAATTACAGAGTACACAACATCCATGAAACGCATTAGCACCACC
ATTACCACCACCATCACCATTACCACAGGTAACGGTGCGGGCTGACGCGTACAGGAAACACAGAAAAAAG
```

Figura 2.1: Exemplo de arquivo FASTA.

As sequências, em geral, são representadas no padrão IUB/IUPAC para nucleotídeos. São aceitas letras minúsculas e maiúsculas, porém, ambas são mapeadas como maiúsculas. Um único hífen ou traço pode ser usado para representar um *gap*, que é a diferença entre duas sequências de DNA. Os códigos do padrão IUB/IUPAC podem ser vistos na Tabela 2.1 (NCBI, 2007).

Tabela 2.1: Padrão IUB/IUPAC, de codificação de nucleotídeos.

A	adenina	C	citosina	G	guanina	T	timina
U	uracila	N	A/G/C/T (qualquer)	K	G/T (cetona)	S	G/C (forte)
Y	T/C (pirimidina)	M	A/C (amino)	W	A/T (fraco)	R	G/A (purina)
B	G/T/C	D	G/A/T	H	A/C/T	V	G/C/A
-	- *gap* com tamanho indeterminado						

O uso do software PHRED, que atribui um valor de qualidade para cada nucleotídeo presente nos *reads* utilizados na montagem, introduziu o índice de qualidade conhecido como PHRED *quality score* (PQS), que defini a probabilidade estimada de erro (EWING et al., 1998; EWING; GREEN, 1998). A probabilidade PQS é mostrada na equação (2.1):

$$Q_{phred} = -10 \times log_{10}(P_e) \tag{2.1}$$

O uso do índice de qualidade levou a introdução de um novo formato de arquivo, conhecido como QUAL ou FASTQ (Figura 2.2). Estes são como os arquivos FASTA, porém, contêm a pontuação PHRED de cada um dos nucleotídeos. Esse índice agora é um padrão de fato, sendo usado para representar a qualidade das sequências. Por exemplo, a plataforma 454 Roche permite a conversão de um formato binário *Flowgram Standard* (SFF) em arquivos FASTA e FASTQ. O índice PQS também é usado por: SAM (http://samtools.sourceforge.net/), *Staden Experiment* (BONFIELD; STADEN, 1996) e ACE (GORDON; ABAJIAN; GREEN, 1998).

```
@ERR000017.3191126 IL6_554:7:330:641:135
CAGCGCCGCAGGAATATTGGTGATCGACATCGAGAA
+
<>>:<::<<399-0<3<<':0:660/0,+3*&''&*
```

Figura 2.2: Exemplo de arquivo FASTQ.

O arquivo FASTQ possui quatro formatos de linha. A primeira se inicia com o marcador "@", a exemplo do FASTA que se inicia com o ">", seguida de um texto livre, de identificação do registro. Alguns centros ao executarem o sequenciamento das duas fitas, utilizam /1 e /2 no final de cada registro identificador e também no nome do arquivo FASTQ, nesse caso são usados dois arquivos um para cada fita. A segunda linha, como no FASTA, contém a sequência de nucleotídeos. A terceira linha se inicia com o marcador "+", podendo ou não ser seguido de uma descrição, que em muitos casos e a repetição do registro de identificação da linha 1. A quarta e última linha, contém a informação de qualidade, os valores numéricos são mapeados em um conjunto específico de caracteres da tabela ASCII (entre o código 33-126). A Figura 2.3, demonstra a distribuição de qualidade usada nos arquivos FASTQ (COCK et al., 2010).

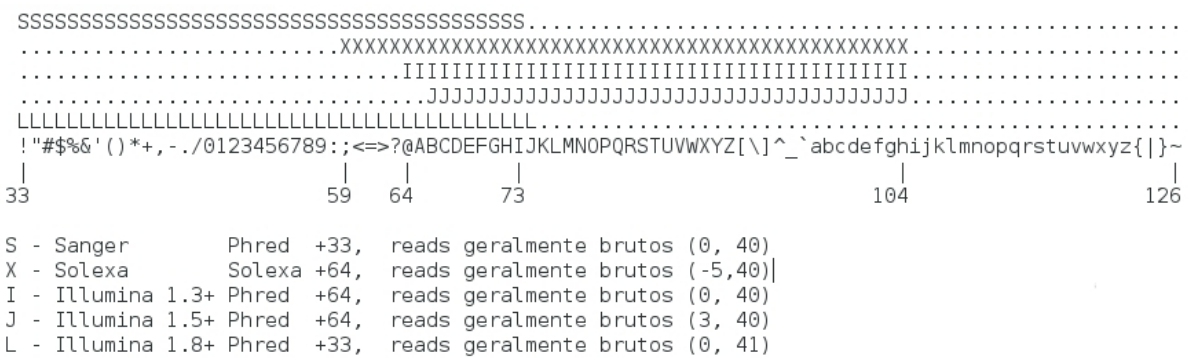


Figura 2.3: Codificação do valor de qualidade em caracteres utilizado nos arquivos FASTQ.

2.2.1 Abordagens Empregadas para o Alinhamento e Montagem de Genomas

Setubal (2004) apresenta várias linhas de pesquisas da bioinformática relacionada a problemas genômicos, entre elas esta o problema de montagem de sequências. Contudo o autor indicava que o avanço das plataformas de sequenciamento levaria a geração de *reads* mais longos, diminuindo assim a complexidade desse problema. Porém, esse avanço somente aumentou a complexidade do problema, pois atualmente as plataformas de NGSs produzem *reads* mais curtos que os originados na metodologia Sanger.

A montagem do genoma a partir de sequências de DNA genômico completo começa com a obtenção dos arquivos FASTA ou FASTQ com os fragmentos para sua posterior montagem. A etapa de montagem pode ou não utilizar um genoma completo como referência, quando isso não acontece o processo recebe o nome de *DE NOVO* (LIN et al., 2011). Os genomas utilizados nesse trabalho foram remontados utilizando um genoma completo como referência.

O processo de montagem de um genoma é divido em: montagem (validação e edição), *scaffolding*[1] e o fechamento dos *gaps* ou espaços entre os *contigs*. A montagem de fragmentos de DNA consiste em construir uma sequência de nucleotídeos continua, construída a partir de um conjunto de fragmentos sobrepostos, essa sequência é conhecida como *contig*. Se o número de fragmentos for muita grande, a resolução do problema será como resolver um quebra-cabeça, que possui uma característica fundamental: a colocação das peças nos locais corretos, e uma experiência com quebra-cabeças demonstrou que eles são

[1] O processo através do qual a informação de emparelhamento dos *reads* é utilizada para ordená-la e orientar os *contigs* ao longo do cromossomo

matematicamente difíceis de resolver (DEMAINE; DEMAINE, 2007). Por isso, uma das tarefas mais difíceis num projeto consiste na montagem dos fragmentos, principalmente quando se compara o tamanho dos mesmos com o do genoma completo.

O processo de montagem se inicia com o método de *shotgun*, que consiste em quebrar o genoma em pequenas frações (Figura 2.4), e posteriormente os fragmentos resultantes são sobrepostos gerando os *contigs*. Mesmo que a técnica utilize sequências mais longas ($\approx$ 1.000 pb), sequenciadas através do método de Sanger, qualquer genoma possui um número muito maior de nucleotídeos.

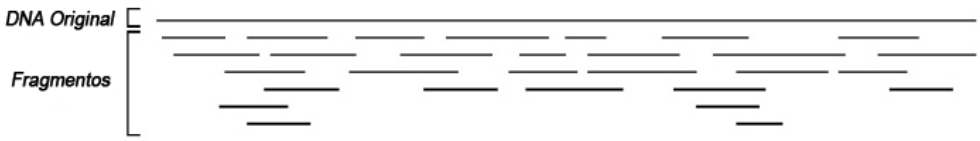


Figura 2.4: Fragmentação das sequências.

As sobreposições são alinhamentos (Figura 2.5), executados entre o fragmento e o genoma de referência, onde o número total de *reads* alinhados recebe o nome de profundidade. Em geral para se encontrar a melhor sobreposição para um *reads* é utilizado análise probabilística, sendo a mais comum o modelo de Lander e Waterman (1988). O processo é finalizado com a geração dos *contigs* e dos consensos, como mostra a Figura 2.6.

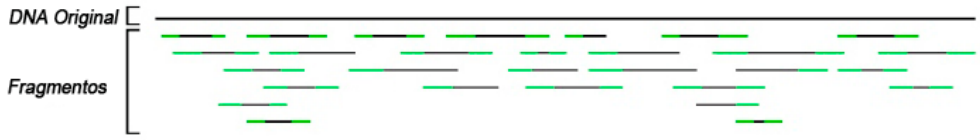


Figura 2.5: Alinhamento dos fragmentos.

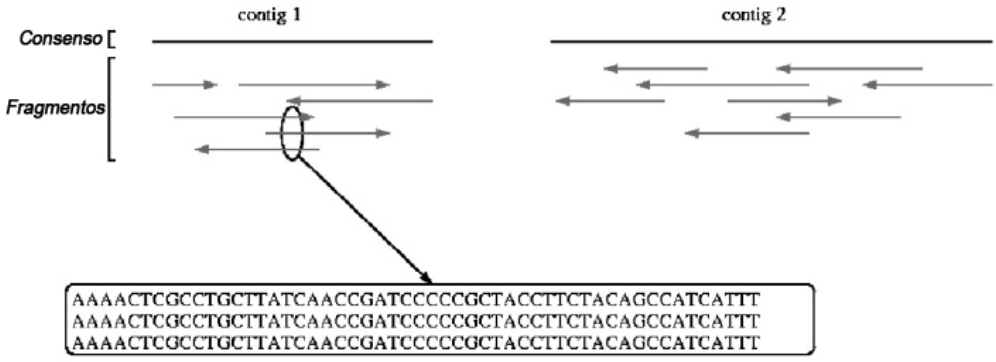


Figura 2.6: Montagem dos consensos.

A grande quantidade de dados gerados pelas plataformas de NGS bem como suas desvantagens (sequências curtas e propensas a erros), tem gerado grandes desafios aos profissionais de bioinformática. Sendo assim, a promessa esperada pelas NGSs só será concretizada quando os métodos computacionais para processar seu conjunto de dados forem eficientes e precisos (MILLER; KOREN; SUTTON, 2010).

Esses fatores dificultam a obtenção de sequências consensos com alta qualidade, mesmo com o uso de um genoma de referência o processo pode ser difícil. A solução encontrada é o uso de uma cobertura maior que a necessária para projetos que utilizam o método de sequenciamento de Sanger. Porém, o grande volume de dados exigem hardware e software compatíveis com a dimensão do genoma. A Tabela 2.2 contém uma lista dos softwares de alinhamento para dados de NGS (LEE; TANG, 2012).

Tabela 2.2: Softwares para montagem de genoma oriundos de plataformas de NGS.

Ferramentas	Plataforma
ELAND	Solexa
Soap	Solexa
ZOOM	Solexa e SOLiD
PASS	Solexa, SOLiD e 454
MOM	Solexa
Vmatch	Solexa
Bowtie	Solexa
CloudBurst	Solexa
BWA	Solexa
SHRiMP	Solexa e SOLiD
AB mapreads	SOLiD
MuMRescueLite	SOLiD
MAQ	Solexa e SOLiD
SeqMap	Solexa
RMAP	Solexa

Coberturas entre $8x - 10x$ são consideradas adequadas para sequências Sanger, porém, para sequências de NGS coberturas entre $30x - 40x$ podem ser necessárias (LEE; TANG, 2012). O artigo de Eck et al. (2009) utilizado como referência para esse texto utilizou uma cobertura entre $8x - 16x$ como satisfatória, porém, no conclusão do referido trabalho o autor sugere que coberturas maiores que $16x$ devam ser analisadas como maior rigor, demonstrando que o valor de cobertura pode sofrer variações entre diferentes projetos de sequenciamento.

Assim como a montagem de um quebra-cabeça a de fragmentos é de difícil resolução,

obrigando que algoritmos de montagem utilizem heurísticas diferentes (MYERS, 1995). No geral a abordagem escolhida recairá em uma das três principais categorias: a abordagem gulosa, a sobreposição layouts de consensos (*Overlap Layout Consensus* - OLC), e a aproximação por grafo de Bruijn. Uma descrição de cada método é feita a seguir.

A abordagem gulosa, foi a mais utilizada nos primeiros anos em que os software de montagem se desenvolveram, principalmente devido a seu fácil entendimento, sendo adotado por Green (1994), Huang e Madan (1999). O algoritmo tenta escolher a melhor solução disponível, em cada etapa do processo, se utilizando de alguma heurística, sendo a mais comum a par de sequências, que procura a região de maior similaridade entre o fragmento a ser montado e o genoma de referência.

O funcionamento típico da abordagem gulosa segue seguintes passos: (1) todos os *reads* são computados para identificar sobreposições; (2) cada *reads* formará um *contig* separado; (3) a definição da heurística gulosa se dá com a seleção de um par de *contig* com as melhores sobreposições; (4) é calculada a sequência consenso, que depois é utilizada para aumentar o *contig*; (5) o alinhamento entre os *contigs* novos e os existentes são atualizados. Os passos 3,4 e 5 são repetidos até que não haja mais pares de *contigs* se sobrepondo.

Embora a implementação dessa abordagem seja rápida e funcione bem para algumas amostras, a presença de regiões repetitivas pode dificultar o processo de montagem. A existência de regiões repetitivas permite que ocorra mais de um local no genoma aonde os *contigs* irão se encaixar. Porém, quando os *contigs* forem fundidos eles identificaram somente um região do genoma podendo gerar erros na montagem, conforme mostra a Figura 2.7.

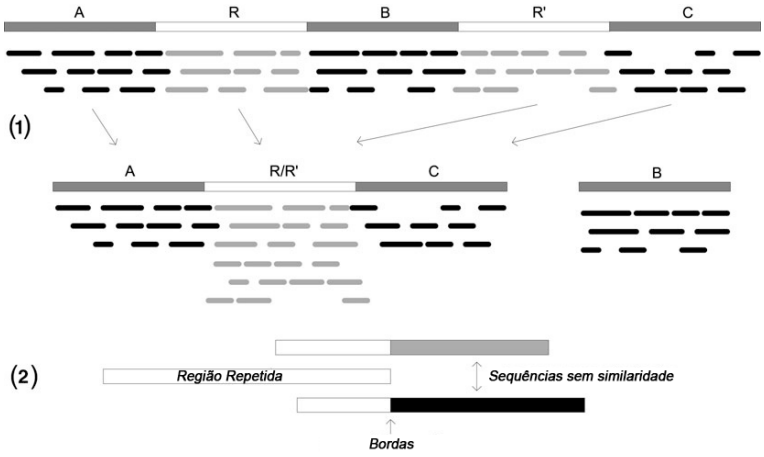


Figura 2.7: Regiões repetidas no genoma e seu problema durante a montagem.

A abordagem OLC foi adotada por muitos softwares de montagem, sendo uma das mais populares e bem sucedidas, ao oferecer diversas melhorias em relação a abordagem gulosa. Ela possui três passos principais: (1) A construção de um gráfico de sobreposição, através da sobreposição computacional de todos os *reads*, onde cada nó representa um *read*, e cada aresta representa a sobreposição entre eles; (2) a extração de um caminho, que corresponde a um *contig*, o resultado desejado é encontrar um caminho hamiltoniano[2], que visita um nó de cada vez; (3) a última etapa, é a sequência resultante do caminho encontrado na etapa anterior.

A abordagem com grafo de Bruijn difere da abordagem anterior, por utilizar grafos de Bruijn ao invés de gráficos de sobreposição. Na abordagem grafo de Bruijn, todas as k-tuplas contidas em cada *read* é utilizada, onde cada qual representa um vértice no grafo, somente ocorre a formação de arestas entre dois vértices se o sufixo $k-1$ da primeira k-tupla for idêntico ao prefixo $k-1$ da segunda k-tupla, formando um *read* continuo (ZERBINO; BIRNEY, 2008). O valor de k é definido, de forma a ser mais curto que o comprimento do *read*, porém, precisa ser grande o suficiente para que cada k-tupla seja única no genoma. A montagem do genoma pode ser feita encontrando um caminho euleriano[3], que passe em cada borda somente uma vez (IDURY; WATERMAN, 1995; PEVZNER; TANG; WATERMAN, 2001).

2.2.2 Alinhamento Local com o BLAST

Assim como a montagem de fragmentos, a busca por similaridade entre sequências, esta entre as atividades primárias, de um processo de sequenciamento. A atividade é tão básica, que é utilizada pelos softwares de montagem, para encontrar sobreposições (LI; RUAN; DURBIN, 2008; LI et al., 2008). O processo de busca por similaridade pode fornecer a primeira evidência de função de um gene sequenciado recentemente, sendo assim uma tarefa executada durante e após o processo de montagem de um genoma. Por isso em 1989, o *National Center For Biotechnology Information* (NCBI) apresentou a ferramenta de alinhamento local, *Basic Local Alignment Search Tool* (BLAST) (ALTSCHUL et al., 1990). A ferramenta permite a pesquisa entre sequências de nucleotídeos e de proteínas, bem como a tradução direta de nucleotídeo em proteína e posterior pesquisa. A tabela

[2]Um caminho hamiltoniano é um caminho que permite passar por todos os vértices de um grafo, não repetindo nenhum

[3]Um caminho euleriano é um caminho em um grafo que visita cada aresta apenas uma vez.

2.3, mostra os atuais comandos disponíveis no BLAST.

Tabela 2.3: Programas BLAST.

Programa	Sequência de consulta	Banco de dados	GAP
BLASTP	Proteína	Proteína	sim
BLASTN	O ácido nucleico	O ácido nucleico	sim
BLASTX	Ácido nucleico traduzido	Proteína	sim a cada quatro bases
TBLASTN	Proteína	Ácido nucleico traduzido	sim a cada quatro bases
TBLASTxc	Ácido nucleico traduzido	Ácido nucleico traduzido	não

Ao executar um alinhamento, o BLAST disponibiliza três informações importantes: *gap*, *match* e o *mismatch* que correspondem a inserções e deleções entre as sequências geralmente utilizando o caracter '-', bases idênticas e as bases diferentes, conforme Figura 2.8. Também é possível visualizar a pontuação dada para cada um dos itens. Essa pontuação serve para calcular o *score* de cada alinhamento, sendo que o mesmo é utilizado na escolha do melhor alinhamento.

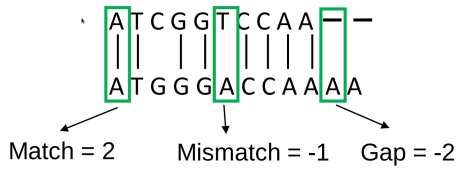


Figura 2.8: Alinhamento entre duas sequências.

2.2.3 Mapeamento e montagem de genoma com MAQ

O artigo de referência dessa dissertação utilizou para a montagem do genoma o MAQ, que é um software de montagem e alinhamento de sequências, que utiliza a informação de qualidade para alinha-las, e trabalha principalmente com dados gerados pela plataforma Solexa. Porém, possui funções para tratar dados sequenciados na plataforma ABI SOLiD (LI; RUAN; DURBIN, 2008).

O MAQ inicia o processo de montagem pelo alinhamento dos *reads* em relação ao genoma de referência, gerando em seguida os consensos. Na etapa de mapeamento ele executa o alinhamento, utilizando o algoritmo de Smith e Waterman (1981), sem a presença de *gap*. Para DNA de fita única o alinhamento aceita de 2 a 3 *mismatches* e de 1 a 2 para fita dupla. Entretanto, esses valores podem ser alterados por meio de parâmetros definidos durante o mapeamento. Na etapa de montagem cada consenso tem um valor

estatístico calculado. Esse valor é utilizado para maximizar a probabilidade posterior de cada posição do consenso.

Além das funções principais o MAQ também informa valores de inserções e deleções conhecidos como Indels, dados de SNPs, e um visualizador de alinhamentos. Na Figura 2.9 estão apresentadas todas as funções do programa MAQ, bem como o fluxograma de funcionamento. O filtro será descrito com mais detalhes no capítulo 3.

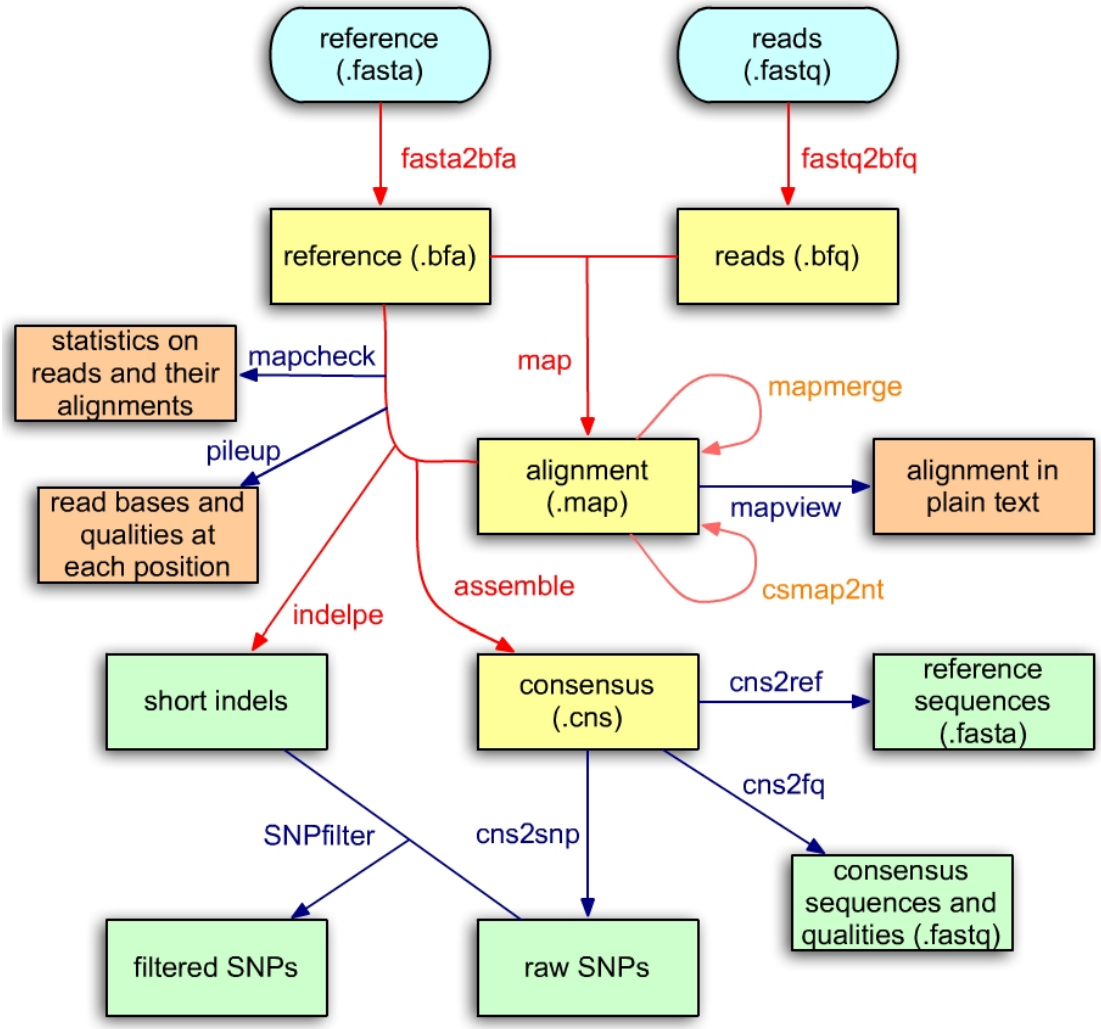


Figura 2.9: Fluxograma MAQ e suas funções - (**Fonte:** *Li (2008b)*) .

Atualmente o MAQ é utilizando em vários projetos de ressequenciamento, inclusive o *1000genome* humano, e também no projeto de genoma do câncer. Sendo distribuído sob licença *GNU Public License* (GPL), incluindo os códigos fontes e esta disponível em: `http://maq.sourceforge.net` (LI; RUAN; DURBIN, 2008).

2.3 Remontagem do Genoma para a Obtenção de Dados

Os processos de descoberta e filtragem são executados sempre após a montagem do genoma, por isso, a necessidade de se executar essa fase do projeto. Um projeto de montagem de um genoma pode ser extenso. Por isso, para que fosse possível a execução das etapas de descoberta e filtragem de SNPs, foram remontados os genomas de duas espécies distintas, utilizando o software MAQ.

O processo de remontagem visa obter a sequência completa do DNA do genoma de um indivíduo, anteriormente sequenciado em plataformas de NGS ou não. Os arquivos contendo as sequências são armazenados em repositórios, de forma que o processo se inicia com a obtenção desses arquivos que são em geral do tipo FASTQ. O próximo passo é a definição da montagem que será utilizada como referência, em seguida as sequências são alinhadas com o genoma de referência escolhido, obtendo assim o genoma consenso, ou genoma alvo. Após essa etapa de alinhamento, é possível a execução da etapa de descoberta, que gera o arquivo necessário para o estudo realizado no Capítulo 3. O filtro de SNPs do software MAQ, o SNPfilter, utiliza esse arquivo para executar a etapa de filtragem.

Foram utilizados nesta dissertação, as sequencias do genoma de duas espécies distintas, uma animal e outra vegetal. Este procedimento foi adotado, como uma tentativa de assegurar a eficiência da ferramenta implementada.

O genoma principal é o de um animal da espécie *bos taurus*, raça Fleckvieh, que foi sequenciado utilizando NGS (ECK et al., 2009). O desenvolvimento do filtro faz parte do projeto de descoberta de SNPs em genoma bovino completo, da EMBRAPA gado de leite, por isso, a escolha do primeiro genoma bovino completo sequenciado utilizando NGS.

Também foi remontado o genoma da *Arabidopsis thaliana*, escolhido, devido ao grande volume de informação disponível, e principalmente sequências de NGS, e também por ser uma planta bem estudada e com SNPs bem definidos, sendo o primeiro genoma de planta a ser sequenciado. A seguir será mostrado como foi o processo de remontagem de cada um desses genomas.

2.3.1 O genoma do Bos taurus

O genoma bovino é diplóide e com 30 pares de cromossomos homólogos, sendo 29 pares autossômicos e um sexual, sendo os machos heterogâmico XY e as fêmeas homogamética XX, e com aproximadamente 3 bilhões de pares bases (SEQUENCING et al., 2009).

O genoma remontado foi sequenciado utilizando a plataforma *Genome Analyzer II* da Solexa, gerando 24 giga bases de sequência, com tamanho de 36pb de *mate-pair* após a trimagem[4], resultando numa montagem com $7,4x$ de cobertura média. Foi utilizado como referência a montagem *bosTau4.0* do genoma bovino, sequenciado pelo *Baylor College of Medicine* e disponibilizado pela Universidade da Califórnia em Santa Cruz (HGSC, 2007).

A maioria dos SNPs presentes no dbSNPs, pertenciam a uma única raça, hereford. O trabalho de Eck et al. (2009), avaliou um segundo animal. No projeto, foram utilizadas amostra de sangue de um touro Fleckvieh para a extração do DNA seguindo os protocolos padrões. Os autores utilizaram informações do chip Illumina *BovineSNP50* e ferramentas de espectrometria de massa, para identificação de falsos positivos e falsos negativos. Estabeleceram a frequência alélica da população utilizando genótipos de 96 animais (48 Fleckvieh e 48 Braunvieh).

Os arquivos FASTQ com as sequências foram depositados no Arquivo Europeu de Nucleotídeos (ERA - *Europeu Read Archive*), com o código ERA000089. Os fragmentos foram distribuídos em 98 arquivos FASTQ, totalizando 43Gb de pares bases, e 125Gb de espaço em disco.

O processo de remontagem dos fragmentos, seguiu os mesmos passos utilizados por Eck et al. (2009) em seu artigo. A Figura 2.10, mostra os procedimentos executados, para a remontagem do genoma. As sequências foram remontadas com o software MAQ, versão 0.7.1, a etapa de mapeamento das sequências foi paralelizado, de forma a acelerar o processo de montagem, o processo todo utilizando o *cluster*, demorou 11 dias, 8 horas e 8 minutos. Cada processo executado no cluster, quando finalizado, informa o tempo de processamento gasto. Sendo que a soma de todo os processos em paralelo resultaram em 30 dias, 2 horas e 29 minutos. O ganho de um código paralelo é em geral calculado usando a lei de Amdahl's, que determina o potencial de aumento, e é calculado dividindo o tempo do código sequencial pelo tempo do mesmo código paralelizado (PACHECO, 2011). Logo o

[4]A trimagem consiste em retirar as sequências de adaptadores (*primers*), vetores, rRNAs e cauda poli-A das sequências obtidas.

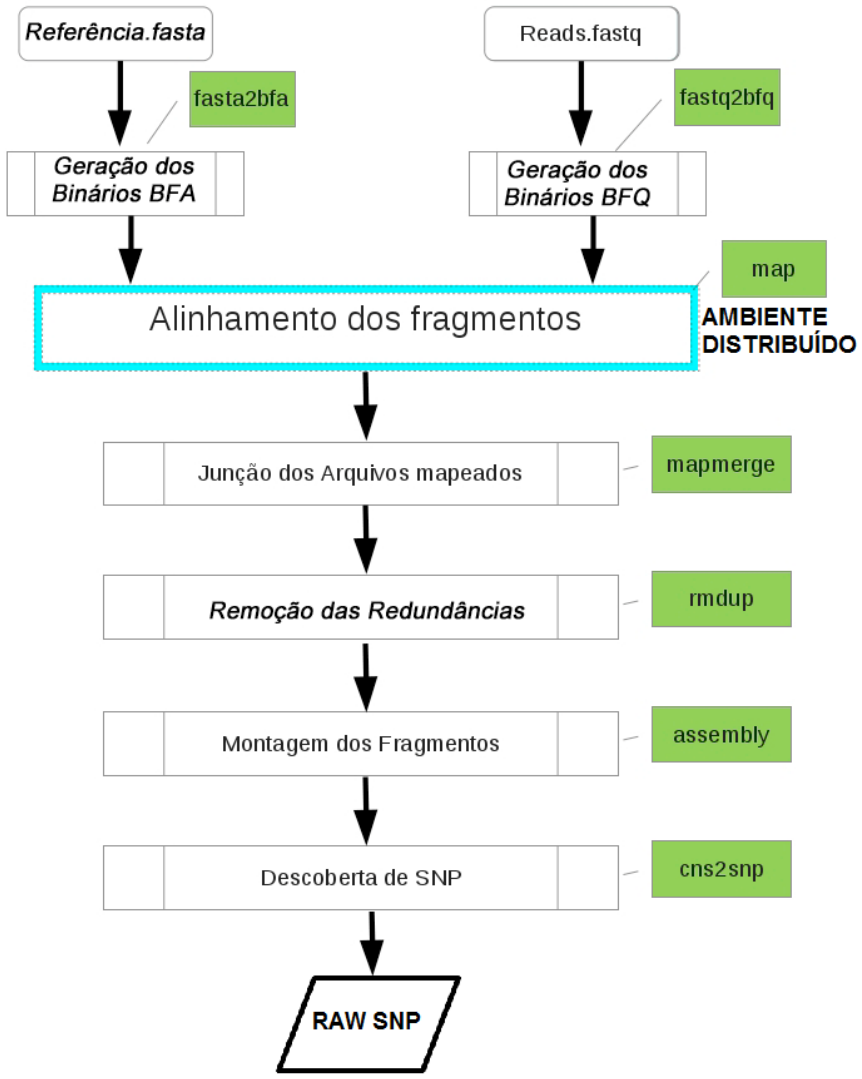


Figura 2.10: *Workflow* do processo de remontagem do *Bos taurus*.

uso do cluster permitiu que os processos de alinhamento e montagem obtivessem um ganho de 2,6 no tempo final de processamento. O processo de mapeamento, ou alinhamento, é o mais demorado de toda a remontagem, por isso, foi o único a ser paralelizado.

A etapa de descoberta de SNPs encontrou 10.652.208 SNPs putativos, após a retirada dos artefatos este número caiu para 6.869.797, e depois de filtrados foram reduzidos para 2.331.820 novos candidatos a SNPs. O artigo de Eck et al. (2009) encontrou valores diferentes na etapa de descoberta, sendo 7.102.734 SNPs putativos já sem artefatos e 2.444.637 após a execução do filtro. A diferença encontrada pode ser explicada, pois o autor trabalhou com o software MAQ versão 0.6.8, e nesse trabalho foi utilizada a versão 0.7.1. O autor do software MAQ informa no arquivo de *NEWS* presente no diretório do mesmo, que em relação à versão 0.6.8, a 0.7.1 recebeu melhorias nas etapas de alinhamento

e montagem, permitindo o uso de *reads* maiores que 63pb. Essa melhoria, segunda o autor, gerou mapeamentos melhores, o que pode ter resultado na diferença entre esse trabalho e o artigo de referência utilizado.

2.3.2 O genoma da Arabidopsis thaliana

A *Arabidopsis thaliana* foi o primeiro genoma de planta a ser sequenciado e atualmente possui um grande volume de pesquisa, Seu genoma é diplóide com cinco cromossomos e aproximadamente 125 milhões de pares bases. O trabalho de Ossowski et al. (2008) serviu de base para a remontagem do genoma, o autor avaliou três germoplasmas diferentes, o BUR-0 o COL-0 e o TSU-1, alinhados com o genoma de referência TAIR10[5].

O processo de montagem seguiu a mesma ordem utilizada para o *Bos Taurus* (Figura 2.10). Como o processo para esse genoma é relativamente rápido, e o número de SNPs é diferente entre eles, os três germoplasmas foram remontados. A tabela 2.4 exibe o tempo de remontagem, bem como o ganho de tempo final de processamento. Para cada germoplasmas remontado, foram obtidos os tempos gastos pela execução completa no cluster, e pelo somatório dos tempos de todos os processos executados. Sendo que o germoplasmas TSU-1 obteve um ganho maior devido a uma melhoria na forma como os mapeamentos foram distribuídos. A tabela 2.5 demonstra as diferenças entre os três germoplasmas e, o número de SNPs encontrado em cada uma delas.

Tabela 2.4: Tempo de remontagem.

Germoplasma	Tempo Gasto	Cluster	Ganho
BUR-0	02:16	01:18	1,74
COL-0	03:39	01:18	2,87
TSU-1	02:48	00:36	4,6

Tabela 2.5: SNPs encontrados nos genomas da *Arabidopsis thaliana*.

Germoplasmas	Putativos	Filtrados
BUR-0	1.135.193	544.881
COL-0	287.397	44.262
TSU-1	1.025.908	460.140

[5]TAIR10, disponível em:`ftp://ftp.arabidopsis.org/home/tair/Sequences/whole_chromosomes/`

2.4 Considerações

Os processos de descoberta e filtragem de SNP são importantes etapas de pós-processamento de um projeto de sequenciamento de DNA, sendo os estudos com SNPs por vezes mais laboriosos que o próprio sequenciamento. Com o avanço das plataformas de sequenciamento, o tempo para finalizar o sequenciamento e montagem de um genoma diminuiu de forma considerável. No entanto o tempo e os custos gastos em pesquisa de pós-processamento não sofreram redução. Por isso, se um falso positivo for escolhido como SNP alvo, a pesquisa sofrerá com perda de tempo e investimento. Neste sentido, fica evidente a importância na criação de filtros eficientes.

3 POLIMORFISMO DE BASE ÚNICA E FALSOS POSITIVOS

Esse capítulo descreve a segunda etapa do desenvolvimento do trabalho que consiste em entender o processo de identificação dos SNPs a partir de sequências de DNA genômico completo, bem como o surgimento dos falsos positivos. Para isso é feito uma apresentação da parte teórica de SNPs além da análise dos falsos positivos, e do filtro utilizado pelo software MAQ.

3.1 Definição de SNPs

Os projetos de sequenciamento de genomas trouxeram muitas revelações para a ciência, uma delas foi a descoberta, por meio do Projeto Genoma Humano, de que o código genético humano mostrou-se mais variado e complexo do que propriamente maior, quando comparado ao de outras espécies.

Em geral, as "regras" que regem o estudo do genoma podem ser aplicadas a qualquer espécie viva, com diferenças apenas entre organismos procariotos e eucariotos. Uma das muitas variações e particularidades do genoma, humano ou de qualquer espécie, são os SNPs, modificações de um único nucleotídeo, em uma dada sequência, quando comparada a outra. Ou seja, SNPs são pares de bases em uma única posição no DNA genômico, que se apresentam com diferentes alternativas nas sequências, isto é, alelos, e podem ser encontrados no genoma de indivíduos normais em algumas populações ou grupos de indivíduos.

O que difere um indivíduo dos demais da sua espécie é o código genético, ou seja, em sua essência, as sequências de nucleotídeos que formam as moléculas e sequências de DNA, RNA e proteínas, que, por sua vez, interagem e formam as células, as quais também, por sua vez, interagem e formam os tecidos, os orgãos até que, finalmente, formam os indivíduos. A organização do código genético pode ser comparada à de um livro. O genoma seria o próprio livro, os cromossomos seriam os capítulos, os genes seriam as histórias, enquanto que os éxons interrompidos por íntrons os códons e os nucleotídeos

corresponderiam, respectivamente, aos parágrafos, palavras e letras.

Desta forma, se cada ser vivo fosse um livro, as diferenças entre os indivíduos de uma espécie começariam nas letras, mais especificamente, na ordem em que as letras formam as palavras. Ou seja, no código genético, as diferenças se iniciam na ordem em que os nucleotídeos se apresentam para, posteriormente, após um complexo processo que envolve transcrição e tradução, originarem as proteínas. Essa é a importância dos polimorfismos de base única, pois, em síntese, a alteração de um único nucleotídeo, uma única base, em uma dada sequência, pode alterar a produção de certa proteína e, se for o caso, o conjunto dessas alterações pode provocar variações nas características dos indivíduos da espécie.

A maior parte do genoma entre os indivíduos de uma mesma espécie é idêntica, porém, existe a variabilidade genética, que são as diferenças encontradas em algumas regiões do genoma (BRONDANI; BRONDANI, 2004). A variabilidade consiste na alteração nas sequências de bases ao longo do DNA e ocorre por substituição, ausência ou duplicação de bases e, os SNPs, essas diferenças pontuais entre pares de bases de diferentes sequências alinhadas, são o tipo mais comum de variabilidade genética (CONSORTIUM, 2003).

Assim, tais diferenças são importantes no estudo da variabilidade das espécies, pois podem provocar alterações funcionais ou fenotípicas, que, por sua vez, podem implicar em consequências evolutivas ou bioquímicas nos indivíduos em que os SNPs se manifestam.

Os SNPs evoluem de forma lenta sendo também responsável pela formação de alelos, que são as diferentes variações para um mesmo gene, ou seja, as diferentes formas com que um gene pode se apresentar. Tais formas podem ser bi, tri ou tetra-alélicas, ou seja, possuírem duas, três ou quatro formas distintas (Figura 3.1). A forma bi alélica é a mais comum de ser encontrada, sendo quase absoluta (BROOKES, 1999).

Figura 3.1: Exemplos hipotéticos de polimorfismos bi, tri e tetra-alélicos, respectivamente. A primeira linha, em negrito, representa a sequência consenso e as bases sublinhadas, os polimorfismos.

Segundo Arbex (2009) o estudo de polimorfismo busca basicamente esclarecer as seguintes questões:

i) Como identificar um polimorfismo de base única em uma sequência?

ii) Como comprovar se o nucleotídeo "trocado", que caracteriza a sequência como polimórfica, é realmente um caso de polimorfismo, já que uma "base diferente" pode ser um falso positivo?

iii) O polimorfismo provocara alteração na sequência de bases a ponto de alterar a conformação de uma proteína, formando uma "nova" proteína?

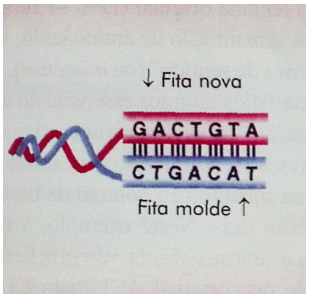
iv) A nova proteína, se esta realmente foi formada, quando combinada com as demais, provocará ou suprimirá a manifestação de alguma característica especifica no individuo?

A individualidade consequente da expressão do código genético é o que define a importância dos SNPs, pois, em síntese, a alteração de um único nucleotídeo, em uma sequência em particular, pode alterar a formação de proteínas e o conjunto dessas alterações pode "sinalizar" ou provocar variações nas características dos indivíduos.

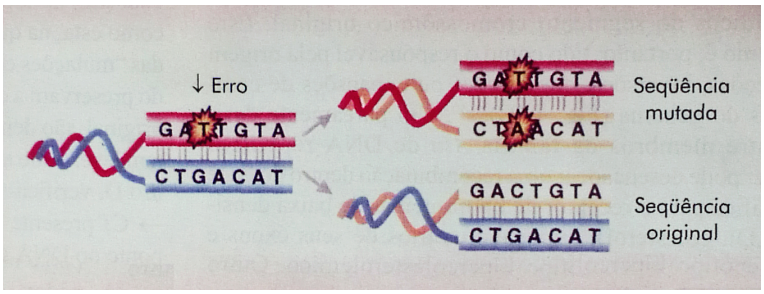
3.1.1 Polimorfismo e mutação

As mutações podem ser divididas em inserções, deleções e SNPs. Tendo sua origem através de três mecanismos básicos: erros na replicação do DNA, danos físico-químicos ao DNA, e pareamento desigual entre duas sequências. A replicação é um processo extremamente fiel, porém, com uma taxa de erro de 10^{-10} por cada base no processo de divisão celular, essas mutações originam os SNPs (Figura 3.2a). A mutação gerada pela exposição a agentes físico-químicos são em geral espontâneas, pois logo após o fim da exposição, as mesmas são reparadas pelos mecanismos de correção do DNA. Entretanto, as mutações que oferecerem alguma vantagem evolutiva aos organismos, são em geral incorporadas ao DNA, gerando uma mutação permanente (Figura 3.2b). O pareamento desigual, ocorrido por meio da recombinação de sequências mal pareadas ou pelo processo de *crossing-over* podem gerar mutações do tipo: inserção; deleção; inversão; ou duplicação (Figura 3.2c).

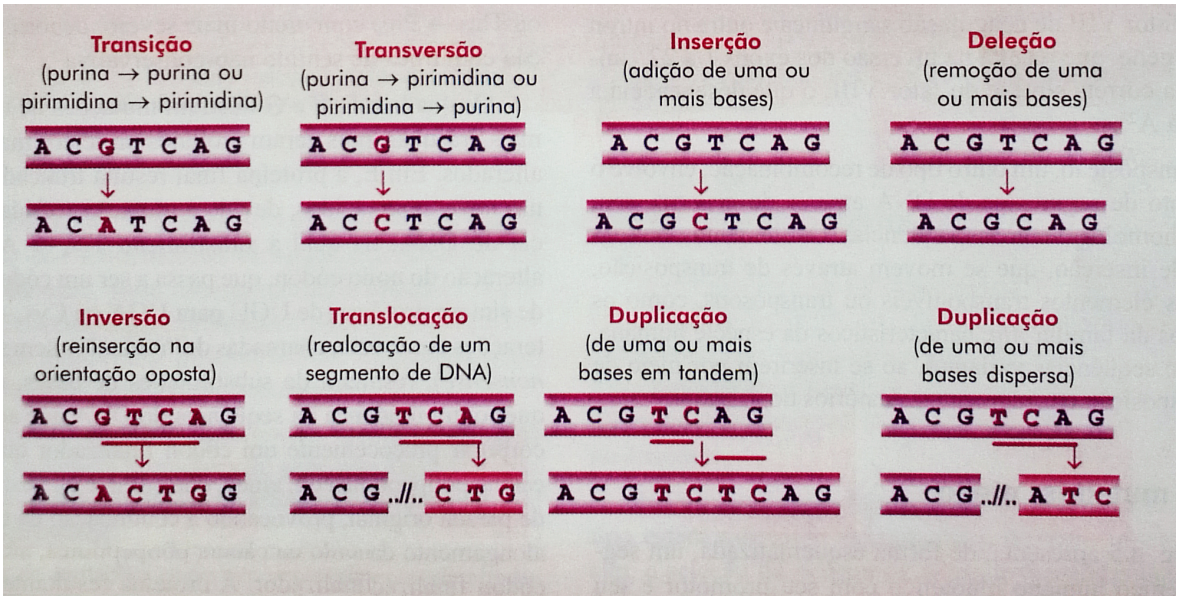
As mutações presentes nas regiões não traduzidas ou íntron, só irão comprometer a função gênica se estiverem localizadas em regiões repetidas ou em elementos reguladores de transcrição ou de processamento do RNA mensageiro (RNAm). Porém, quando estão presente em regiões traduzidas, ou que produzem proteína, conhecidas como éxon, podem gerar algum efeito no processo de expressão gênica.



(a) Replicação.



(b) Erro de incorporação.



(c) Diferentes tipos de mutação.

Figura 3.2: Diferentes classes de mutações. - (**Fonte:** *Alho (2004) pag.79)*

Podem ocorrer três tipos de mutações: sinônimas ou silenciosas, onde a presença dessa mutação não altera o aminoácido gerado pelo novo códon; mutações com sentido trocado ou incorreto (*missense*), onde sua presença gera um novo aminoácido, podendo modificar a estrutura da proteína e consequentemente sua função; e mutações sem sentido(*nonsense*) onde sua presença gera um códon de parada prematuro, interrompendo o processo de tradução da proteína, podendo gerar uma deficiência total ou parcial da mesma (PASSOS-BUENO; MOREIRA, 2004).

Mutações silenciosas, não devem ser desprezadas, pois estudos demonstram que apesar de não alterarem o aminoácido gerado, elas podem modificar o processamento do RNAm com geração de códons de parada prematuros associados a rápida degradação dos transcritos (PASSOS-BUENO; MOREIRA, 2004).

As mutações tem sua importância, pois sua presença de forma única, ou em conjunto, podem definir a existência ou não de uma doença ou de determinada característica fe-

notípica. Por isso, a correta identificação dos polimorfismos na sequência de DNA de interesse, é de grande importância, sendo o primeiro passo para a identificação de marcadores moleculares.

Entretanto existe uma diferença entre SNPs e mutações. Os polimorfismos de base única são modificações que se manifestam naturalmente e podem ocorrer devido à substituição de uma base ou por "edição do RNA", que pode causar a inserção ou exclusão de uma base. Entretanto essas manifestações são, em geral, erroneamente desconsideradas (BROOKES, 1999). O exemplo apresentado na Figura 3.3 mostra a alteração por substituição de um nucleotídeo em uma sequência de 10 bp.

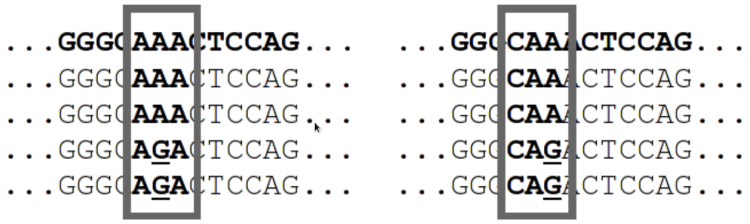


Figura 3.3: Exemplos hipotéticos de um SNP não-sinônimo e de SNP sinônimo.

Tais modificações, contudo, também poderiam ser vistas como mutações. Em termos gerais, a diferença entre o que é um SNP e o que é uma mutação é determinada em função do número de ocorrências de alterações de base, mais especificamente, em função da frequência alélica. Caso uma alteração de base, em uma determinada população, ocorra com frequência superior a 1%, fica caracterizada a ocorrência de SNP, caso contrário, a alteração caracteriza uma mutação (BROOKES, 1999; GUIMARÃES; COSTA, 2002; BARNES, 2007). Entretanto, essa definição apresentada para mutação vem sendo negligenciada e as alterações de base com frequência menor do que 1% estão sendo chamadas de "variações de baixa frequência", enquanto o termo mutação está sendo utilizado para denominar variações genômicas que estejam relacionadas a doenças no indivíduo (BARNES, 2007).

3.1.2 Importância

As aplicações mais comuns relacionadas ao estudo e à identificação de SNPs são encontradas nos trabalhos que objetivam correlacionar genótipo e fármacos como, por exemplo, as interações entre drogas e uma proteína em particular, a identificação de resistência ou susceptibilidade de indivíduos em relação a certas doenças, a definição de marcadores

de predisposição a determinadas patologias e de sensibilidade a diferentes tratamentos (BALDI et al., 2001; GUIMARÃES; COSTA, 2002; CONSORTIUM, 2003; SUAREZ-KURTZ, 2004; CONSORTIUM, 2005; LESK, 2008).

Contudo, atualmente, outras ciências não muito próximas da genética ou da bioinformática também utilizam as ferramentas de estudo, identificação e análise de SNPs, empregando os resultados em áreas como medicina forense, antropologia molecular, evolução, genética de populações, conservação e manejo de fauna (PENA et al., 2000; GUIMARÃES; COSTA, 2002; BRUMFIELD et al., 2003; LESK, 2008), entre outras.

Como exemplo, podem ser citados estudos antropológicos e sociológicos que podem utilizar as alterações de bases em sequências genéticas na determinação do padrão genético de populações, do indicativo de séries históricas de variação de seu tamanho e dos seus padrões de migração (PENA et al., 2000; BRUMFIELD et al., 2003; LESK, 2008).

Além de se conhecerem o mecanismo e a velocidade da evolução desse tipo de polimorfismo, é possível estabelecer períodos prováveis em que uma determinada população manifestou ou perdeu SNPs. Sob essa circunstância, como exemplo para tal investigação, reporta-se a existência de estudos que indicam 94% de probabilidade de que uma população venha a perder um SNP, ou mesmo uma mutação, em 10 gerações, cerca de 200 anos. Como consequência, uma vez estabelecido o período em que a sequência polimórfica acompanhou a população e sabendo que a sequência está restrita à mesma, é possível, com os dados e as ferramentas corretos, mapear a população que se quer estudar (BARNES, 2007).

De maneira geral, SNPs podem promover *splicing* alternativo, alterar o padrão de expressão de genes, como no caso de alterações em sequências de promotores, gerar ou suprimir códons de terminação e alterar códons de iniciação de tradução e, embora SNPs sinônimos não alterem a sequência protéica, podem modificar a estrutura e a estabilidade do RNA mensageiro, afetando, como consequência, a quantidade de proteína produzida (GUIMARÃES; COSTA, 2002; KRISHNAN; WESTHEAD, 2003).

3.2 Identificação de Falsos Positivos

Como visto no Capítulo 2, a tarefa de conclusão da montagem de um genoma, passa pelos processos de sequenciamento, alinhamento e montagem dos *reads*. Em cada uma dessas

etapas, existe uma taxa de erro, que na etapa de descoberta poderá ser interpretado como um SNP.

A descoberta de SNPs consiste na comparação base a base entre o genoma alvo ou consenso e o genoma de referência. Nessa etapa, qualquer diferença entre as sequências é um *mismatches*, alguma dessas diferenças são SNPs outras não. Apesar do nucleotídeo no genoma alvo ser diferente do nucleotídeo no genoma de referência, na mesma posição, o polimorfismo encontrado não ocorre normalmente na natureza. Apesar de ser computacionalmente uma diferença, definir quando esse *mismatch* é ou não um SNP é uma tarefa complexa, ficando essa definição a critério da etapa de filtragem. No que tange as plataformas NGS é sabido que são introduzidos erros na faixa de $0,1\%$ a 1% conforme Tabela 3.1 (GLENN, 2011).

Tabela 3.1: Taxas de erro das plataformas de sequenciamento.

Plataforma	Erro *single-pass* (%)	taxa de erro final (%)
Sanger (capilar)	0,1 - 1	0,1 - 1
Roche 454	1	1
SOLiD	≈ 5	$> 0,01$
Illumina	$> 0,1$	$> 0,1$

O processo de montagem dos fragmentos utilizando um genoma de referência consiste basicamente no alinhamento dos *reads*. Quando a sobreposição das sequências acontece, pode ocorrer uma variação de bases em uma mesma posição genoma. A Figura 3.4 demonstra o correto alinhamento entre fragmentos, gerando um SNP verdadeiro.

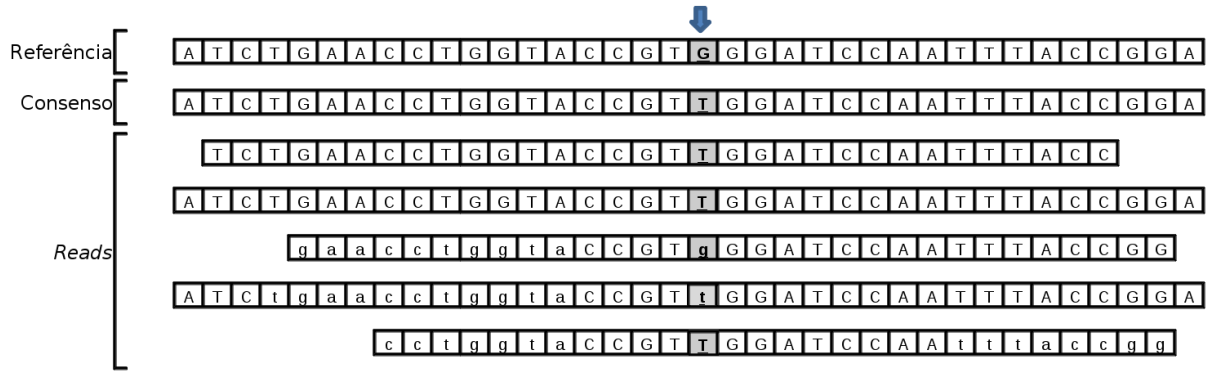


Figura 3.4: SNP verdadeiro gerado pela etapa de alinhamento.

Ao alinhar duas sequências com o genoma de referência para gerar o consenso, o software de alinhamento e montagem, identifica um *mismatch* nas primeiras posições do fragmento. Porém, esse pode ser o melhor alinhamento para o dado fragmento. Essa

situação geralmente ocorre quando os *reads* utilizados são curtos, o que é usual quando se emprega plataformas de NGS. O *mismatch* gerado por esse alinhamento pode ser resultado de um erro na etapa de sequenciamento, ou um SNP. A Figura 3.5 demonstra o exemplo de um *mismatch* gerado por um erro de alinhamento, resultante do erro de sequenciamento (MALHIS; JONES, 2010). A Figura 3.6 mostra outro exemplo de alinhamento correto, sem janelas, porém, os *reads* possuem uma qualidade baixa.

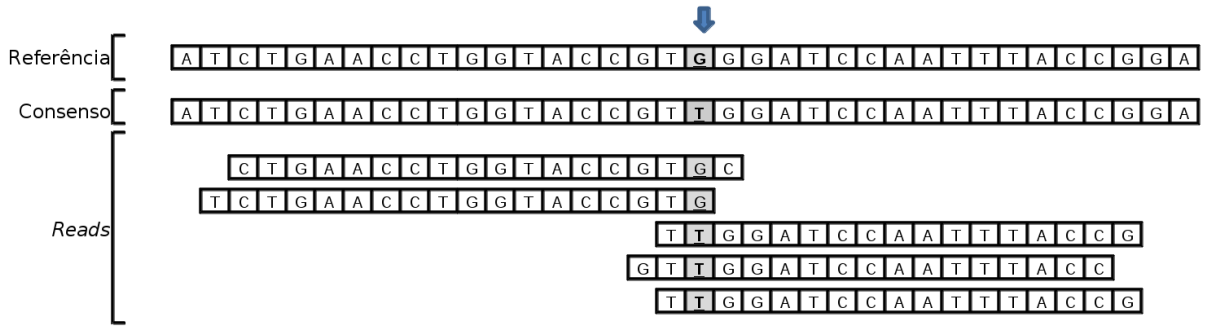


Figura 3.5: Falso positivo gerado pela etapa de alinhamento.

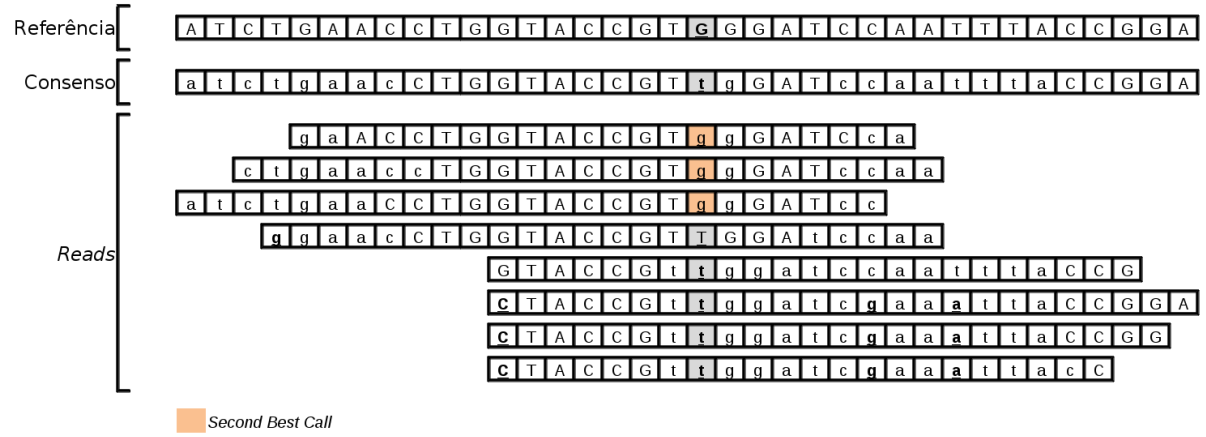


Figura 3.6: Falso positivo gerado por baixa qualidade.

Como visto um *mismatch* pode ser um SNP ou um erro. A tarefa do filtro é classificar os *mismatches* entre SNPs e erro, quando um erro é classificado como SNPs tem-se o falso positivo. Apesar do erro das plataformas de NGSs serem baixos (entre 0,1% a 1%), a dimensão de um genoma em geral é grande, ou seja, o erro em valores relativos é baixo, mas em valores absolutos é alto. Por isso, a necessidade de se construir filtros que sejam capazes de identificá-los.

3.3 Filtros Empregados na Identificação de Falsos Positivos

A Tabela 2.2 contém uma lista de softwares de alinhamento e montagem, porém, somente dois implementam filtros de SNPS, sendo que ambos foram desenvolvidos pelos mesmos autores (LI; RUAN; DURBIN, 2008; LI et al., 2008), de forma que os filtros possuem características próximas. Filtros de SNPs independentes das plataformas e dos software de alinhamento e montagem são encontrados podendo-se citar: o trabalho de Pongpanich, Sullivan e Tzeng (2010) que utiliza técnicas de análise de componentes principais e análise de *clusters* para filtrar SNPs, o trabalho apresentado em Genomics (2011) desenvolveu um filtro utilizando programação genética e algoritmos genéticos, e por último o trabalho de Koboldt et al. (2009), que desenvolveu um filtro de SNPs baseado em heurísticas e estatísticas. Cada filtro apresentado obteve segundo seus autores, tanto boa performance quanto resultados nos testes executados por eles, porém, nenhum deles utilizou redes neurais. Como o software MAQ foi utilizado para a remontagem dos genomas somente o filtro implementado por ele será utilizado e explicado a seguir.

3.3.1 SNPfilter

O SNPfilter é o filtro de SNPs acoplado ao software MAQ, que será descrito em relação ao seu funcionamento é ao arquivo de saída gerado. A Figura 2.9, mostra o fluxograma de funcionamento do software MAQ, onde é possível ver, que após as etapas de mapeamento e montagem do consenso, o software permite uma série de análises com o genoma montado, entre elas está a etapa de *SNP-Calling*[1], que consiste na comparação base a base entre o genoma consenso e o genoma de referência, onde todas as "diferenças" ou *mismatches* são tratadas como SNPs Li (2008a). A saída do SNPfilter é um arquivo com 12 colunas, conforme apresentado na Figura 3.7 que contém um exemplo de saída da etapa de descoberta de SNPs.

A 1ª coluna refere-se a *defline* do arquivo FASTA utilizado no mapeamento, esse arquivo contém o genoma de referência. Em geral a descrição da *defline* é a informação de qual cromossomo a sequência representa. Contudo, se o genoma montado é o de uma bactéria com genoma circular, essa descrição, pode ser o nome da bactéria, ou o código

[1] É executada pelo MAQ através do comando cns2snp

1ª	2ª	3ª	4ª	5ª	6ª	7ª	8ª	9ª	10ª	11ª	12ª
chr1	42	T	A	1	4	66.50	0	2	T	26	W
chr1	171	T	A	1	4	0.00	0	2	T	26	W
chr2	82	C	S	255	255	23.06	63	20	C	150	G
chr2	86	G	R	255	255	23.81	63	20	G	22	A
chr3	10574	A	R	16	14	1.38	63	36	G	94	A
chr3	10777	G	R	94	19	1.00	63	54	A	27	G
chr4	15316	C	M	4	17	1.00	63	0	C	116	A
chr4	16269	G	C	6	6	1.00	63	4	G	5	S
chr5	3032	G	R	29	22	1.56	63	62	G	48	A
chr5	3260	A	M	27	27	6.56	63	62	A	105	C
chr5	3717	T	Y	31	21	19.56	63	62	C	18	T
chr29	51992550	T	G	9	4	1.00	63	38	K	1	T

Figura 3.7: Arquivo de saída do comando cns2snp, colunas: (1)Cromossomo; (2) Posição; (3) Nucleotídeo de referência; (4) Nucleotídeo consenso; (5) *Phred-like consensus quality*; (6) Profundidade; (7) A média do número de acertos do *read* cobrindo a posição; (8) A maior qualidade de mapeamento do *read* cobrindo a posição; (9) A menor qualidade no consenso, olhando uma janela de 6bp com um flanco de 3bp para cada lado; (10) Segundo melhor alinhamento; (11) Qualidade média entre o segundo e o terceiro melhor alinhamento; (12) terceiro melhor alinhamento.

do projeto de sequenciamento.

A 2ª coluna refere-se a posição, do nucleotídeo dentro do genoma montado. Como o processo de *SNP-Calling* é a simples comparação entre duas bases na mesma posição, logo a posição é a mesma nos dois genomas.

A 3ª coluna refere-se ao nucleotídeo presente no genoma de referência, e a 4ª coluna, ao nucleotídeo no genoma montado. Em ambos os casos as bases são representadas pelos códigos da tabela IUB/IUPAC. O genoma montado ou consenso utiliza a tabela como recurso, principalmente quando os fragmentos utilizados na construção dos *contigs* possuem nucleotídeos diferentes na mesma posição e ambos com mesma qualidade. Nesse caso é utilizado a letra que faz referência as duas bases de forma simultânea.

A 5ª coluna refere-se ao a qualidade do nucleotídeo que é calculada como a qualidade *PHRED*, recebendo o nome de *PHRED-like*. Essa coluna é considerada o critério chave para a classificação de um *mismatch* em SNP ou erro, sendo analisada de forma isolada pelo filtro do software MAQ. Assim, quando os outros parâmetros são alterados, visando o aumento na restrição, o filtro passa a selecionar os SNPs somente com base nesse valor.

A 6ª coluna refere-se à profundidade, ou seja, ao número de fragmentos que foram utilizados para se obter aquela região do genoma. O valor ideal pode variar de um projeto de sequenciamento para outro. Porém, o *read* pode ser utilizado em outras regiões do mesmo genoma, com a mesma qualidade. O correto alinhamento de um *read* recebe o

nome de *hit*. A média dos *hits* dos *reads* que foram utilizados para montar a região onde o *mismatch* se encotra é informada pela coluna 7, caso esse valor seja alto, tem-se o indicativo que *mismatch* é não confiável ou está em uma região repetitiva do genoma.

As colunas 8 e 9 fazem referência a informações da qualidade de mapeamento. Essa qualidade, é calculada pelo MAQ visando encontrar o melhor alinhamento para os fragmentos. A 8ª coluna, refere-se a maior qualidade de mapeamento dos *reads* que cobrem a posição, esse valor é similar ao *Phrap quality score* (PQS) utilizado em genomas da geração de sequenciamento anterior (GREEN, 1994). A 9ª coluna, refere-se a menor qualidade no consenso, olhando uma janela de 6pb com um flanco de 3pb para cada lado. Esse valor indica a qualidade dos vizinhos, de forma a permitir que filtros possam identificar erros de alinhamentos, sendo inspirada na ideia de *Neighborhood Quality Standard* (NQS) definida por (ALTSHULER et al., 2000).

As colunas 10,11 e 12 são valores definidos pelos autores do software MAQ, e visam facilitar à etapa de filtragem. A coluna 10 refere-se ao segundo melhor alinhamento, ou seja, na conclusão da montagem do genoma, o software MAQ escolhe o nucleotídeo com maior valor de qualidade para compor o *contig*. Quando mais de um nucleotídeo tem o mesmo valor, é então escolhido um valor da tabela IUB/IUPAC que seja composto pelos nucleotídeos possíveis. Por isso, o software MAQ, registra essas três colunas, a 10ª coluna contém o segundo melhor alinhamento, a 11ª coluna contém a média de qualidade entre o segundo e o terceiro melhor alinhamentos e a 12ª o terceiro melhor alinhamento.

Ao comparar a 10ª coluna com 2ª é verificado se elas são iguais, o que significa que os fragmentos utilizados para a montagem do genoma, possuem variações e que uma delas é igual ao genoma de referência. Assim é necessário avaliar o valor de qualidade, se o valor for alto, então é possível que o polimorfismo seja real, contudo, não permanente, e caso o valor seja baixo, o erro pode ter sido gerado por um erro na etapa de sequenciamento.

O SNPfilter pode ser personalizado de acordo com a necessidade do usuário, permitindo uma flexibilidade no uso do mesmo. As opções possíveis e seus valores padrões estão descritos na Tabela 3.2. O filtro consiste basicamente num conjunto de três regras booleanas simples, a primeira utiliza as opções padrões relativas a qualidade do mapeamento, a segunda utiliza somente e informação do PHRED e a terceira compara a vizinhança utilizando conceitos de NQS

As regras utilizadas pelo filtro consistem num conjunto de três condicionais indepen-

Tabela 3.2: Opções do comando SNPFilter.

Opção	definição	valor padrão (%)
-d INT	Profundidade mínima	[3]
-D INT	Profundidade máxima	[256]
-Q INT	Qualidade de mapeamento mínima	[40]
-q INT	Qualidade mínima do consenso	[20]
-n INT	Qualidade mínima do consenso adjacente	[20]
-w INT	Tamanho da janela para potencial indels[2].	[3]
-F FILE	Arquivo de saída do comando de INDELPE	[null]
-f FILE	Arquivo de saída do comando de INDELSOA	[null]
-s INT	*score* mínimo para o soa-indel	[3]
-m INT	O número máximo mapeado através de uma soa-indel	[1]
-a	filtro alternativo para mapeamento de única fita	-

dentes, ou seja , as regras não são complementares, elas são exclusivas, de forma que se um *mismatch* satisfizer uma das regras ele é considerado um SNP. As regras são:

Primeira:

- profundidade ≥ 3,
- *hit* > -1,
- qualidade de alinhamento ≥ 40 e
- qualidade no flanco ≥ 20.

Segunda:

- PHRED-*like* ≥ 20.

Terceira:

- média entre o segundo e o terceiro melhor alinhamento ≥ 20,
- segundo melhor alinhamento $\neq$ nucleotídeo do genoma de referência.

As etapas de *SNP-Calling* e de filtro, implementadas pelo MAQ, são simples de serem executadas e entendidas, sendo utilizadas por pesquisadores em variados projetos. Porém, a estrutura das regras foi definida pelos autores, com base em conhecimento prático e testes (LI; RUAN; DURBIN, 2008). A expectativa é que possa-se construir um filtro que ao analisar todos os parâmetros em conjunto, e não de forma separada como o SNPfilter, obtenha melhores resultados, conseguindo classificar melhor os *mismatches*. É esperada assim uma robustez maior do filtro, pois o mesmo poderá contornar melhor

ruídos presentes nos conjuntos de dados. O SNPfilter, se mostra muito dependente da variável PHRED-*like*, pois possui uma regra onde somente ela é verificada. A expectativa é que ao analisar todas as variáveis o filtro tenha uma dependência menor de somente uma variável.

Apesar da simplicidade, e do vasto uso, o filtro implementado pelo MAQ é simples, consistindo em um conjunto de condicionais booleanas. O uso de estatística está na etapa de alinhamento. O autor do software MAQ acredita que ao melhorar o alinhamento entre sequências, irá reduzir o número de falsos positivos. A dificuldade encontrada na implementação de um filtro está no fato de que somente as variáveis de alinhamento e montagem são conhecidas logo, se um erro possuir boa profundidade e qualidade de mapeamento ele poderá ser considerado um SNP, gerando assim um falso positivo. O uso de técnicas de inteligência computacional pode vir a gerar bons filtros de SNPs, pois estas ferramentas demonstram excelente capacidade de classificação.

4 FILTRAGEM DE SNPs UTILIZANDO REDE NEURAL

As redes neurais artificiais compreendem um recurso computacional frequentemente utilizado para a solução dos mais variados problemas, incluíndo os problemas biológicos e de bioinformática como os de Tomita et al. (2004), Heidema et al. (2006), Curtis (2007), Ren et al. (2009), Long et al. (2009), Bridges et al. (2011). Porém, nas pesquisas realizadas não foram encontradas referências do uso de redes neurais para o filtro de SNPs em DNA genômico completo, sequenciado em plataformas de NGS. A rede neural foi a técnica de inteligência computacional escolhida, pois a capacidade de classificação é uma das suas principais características, podendo assim ser utilizada na montagem de um filtro que nada mais é do que um classificador.

Nas pesquisas não foram encontrados artigos que utilizem redes neurais para o filtro de SNPs em DNA genômico completo, sequenciados através de plataformas de NGS. Logo é necessário saber se é possível desenvolver um filtro eficiente de SNPs utilizando redes neurais, se sim, qual a vantagem em substituir os filtros atuais por novos. Essas são algumas questões tratadas nesse trabalho.

Os artigos citados a seguir demonstram o poder computacional das redes neurais aplicadas à solução de problemas biológicos. Todos os trabalhos são executados após as etapas de descoberta e filtragem de SNPs, e sofrem com a presença dos falsos positivos nas amostras estudadas. Porém, nenhum artigo utilizando redes neurais para a filtragem de SNPs foi encontrado. Atualmente somente os softwares de alinhamento e montagem como o MAQ, ou as plataformas de sequenciamento, possuem esse tipo de filtro.

Tomita et al. (2004) desenvolveu um trabalho que buscava a associação de marcadores do tipo SNPs com o desenvolvimento de asma alérgica na infância, sendo aplicado uma população de 334 japoneses. Para o trabalho de associação o autor utiliza redes neurais, combinadas com um método de redução de parâmetros, e consegue obter resultados promissores. É o primeiro trabalho a selecionar automaticamente SNPs relacionados ao desenvolvimento de uma doença multifatorial.

O trabalho desenvolvido por Heidema et al. (2006), aborda o desafio da identificação

de SNPs envolvidos no desenvolvimento de doenças, identificando que apesar do grande volume de dados disponíveis, muitos pesquisadores não estão familiarizados com os métodos necessários para avaliar a associação entre os SNPs e as doenças. O trabalho utiliza algumas técnicas, entre elas as redes neurais. O autor conclui que as redes neurais, por lidarem somente com um limitado número de variáveis, são menos úteis que outros métodos não paramétricos. Porém, assim como outros métodos pode ter seu poder preditivo aumentado quando associado a outras técnicas.

O estudo de associação em escala genômica (*Genome-wide association studies* - GWAS) consiste em identificar as variantes causais no genoma de muitos indivíduos e sua associação com os fenótipos de interesse e posterior investigação de suas funções biológicas. Curtis (2007) em seu artigo, utiliza técnicas de inteligência computacional, para encontrar associação entre doenças e um conjunto de marcadores do tipo SNP. O autor do artigo comparou o desempenho de uma rede neural em relação a análises baseada em alótipos e a análises baseadas em lócus. A rede neural foi mais poderosa que a análise baseada em alótipos, além de, no seu trabalho, obter uma significância estatística maior.

O artigo desenvolvido por Ren et al. (2009), utilizou dados obtidos de amostras de variáveis discriminantes de genótipos, através de espectros de infravermelho próximo (*near-infrared spectra* - NIRS), sendo então desenvolvido um modelo computacional baseado em aprendizado de máquina utilizando redes neurais. Como exemplo, foi utilizado o SNP (857G $>$ A) da N-acetiltransferase 2 (NAT2), as amostras foram genótipadas em pares (GG, AA, GA). O objetivo da rede neural desenvolvida no referido artigo era classificar os SNPs como pertencendo a um dos três genótipos definidos. A rede obteve uma predição robusta quando apresentada a amostras desconhecidas. Ren et al. (2009), define a rede neural como um método simples, rápido e de baixo custo.

O trabalho desenvolvido por Long et al. (2009), utilizou métodos de classificação multicategoria na detecção da mortalidade de frangos de corte, associada a um conjunto de SNPs. Para isso o autor do trabalho utilizou três algoritmos de classificação: um classificador de Bayes, uma rede bayesiana e uma rede neural. Cada um dos algoritmos de classificação utilizado foi melhor em uma determinada característica procurada pelo autor do artigo, sendo que o classificador de Bayes e a rede neural foram os que obtiveram os melhores resultados no geral.

Outro desafio em genética é determinar se duas populações candidatas podem ser di-

ferenciadas com base em suas estruturas genéticas. O trabalho desenvolvido por Bridges et al. (2011) utiliza essa temática. A primeira etapa é detectar as estruturas presentes nas populações candidatas. O método tradicional utilizado é a análise de componentes principais (*Principal component analysis* - PCA). Bridges et al. (2011) utilizou dois métodos (redes neurais e máquinas de vetores de suporte - SVM) para a detecção de diferenças genéticas entre três populações: duas da Escócia e uma da Bulgária. A rede neural foi utilizada como técnica de aprendizado supervisionado, e a máquina de vetores de suporte (*support vector machine* - SVM) como técnica de aprendizado não supervisionado. Ambas exibiram uma sensibilidade consideravelmente maior que a atingida pela PCA, sendo capaz de distinguir entre duas populações da Escócia, onde o PCA não foi capaz. O autor do artigo conclui que uma abordagem de aprendizado supervisionado deva ser entre os métodos estudados, o escolhido para classificar os indivíduos em populações pré-definidas, em especial quando os estudos envolverem grandes genomas e populações.

4.1 Teoria das Redes Neurais

O cérebro humano adquire conhecimento através das "experiências" vivídas em situações anteriores. Seu funcionamento serviu de inspiração para que diversos pesquisadores tentassem simulá-lo, principalmente o processo de aprendizado por experiência, a fim de desenvolver sistemas capazes de executar tarefas simples para o nosso cérebro, como por exemplo, a classificação, o reconhecimento de padrões e o processamento de imagens. O modelo de neurônio artificial surgiu como resultado dessa pesquisa, que resultou na geração das redes neurais artificiais, que consistem num conjunto de neurônios artificiais interligados Haykin (2001).

O neurônio biológico é a unidade básica do cérebro humano. É especializado na transmissão e recepção de informação, que na realidade são impulsos elétricos. Sinais captados por receptores nervosos geram um impulso ou estímulo que são propagados ao longo do neurônio. O neurônio é constituído por três partes principais: o corpo celular, de onde se originam duas ramificações os dendritos e uma mais longa conhecida como axônio. Na extremidade dos axônios estão os nervos terminais, responsáveis por realizar a transmissão da informação para outros neurônios, processo conhecido como sinapse (ARBIB, 2002).

4.1.1 Neurônio Matemático

Vários pesquisadores tentaram simular o funcionamento do neurônio biológico, porém, o modelo mais bem aceito foi proposto por McCulloch e Pitts (1943), que desenvolveram um neurônio artificial conhecido como perceptron. No modelo proposto, os impulsos elétricos recebidos, são definidos como sinais de entrada (x_j), onde nem todos os estímulos excitarão o neurônio receptor na mesma proporção. À medida que define a intensidade do estímulo é representada no modelo de McCulloch e Pitts através dos pesos sinápticos (ω_{kj}), onde k representa o índice do neurônio e j o terminal de entrada da sinapse.

O corpo celular é composto por dois módulos, o somatório das entradas multiplicado pelo peso sináptico, e a função de ativação (FA). A função de ativação define a saída do neurônio com base no resultado do somatório. A saída (y_k) por sua vez representa o axônio Haykin (2001). O peso sináptico pode ser negativo ou positivo, fazendo com que o estímulo seja inibitório ou excitatório, respectivamente. A Figura 4.1 apresenta o modelo proposto por McCulloch e Pitts (1943).

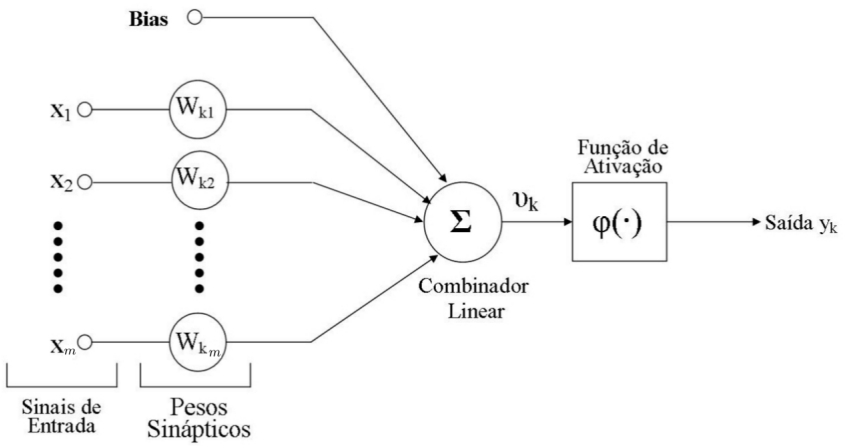


Figura 4.1: Neurônio de McCulloch e Pitts.

Por acreditar que o funcionamento do cérebro possui um caráter binário, McCulloch desenvolveu seu modelo matemático para o perceptron de forma que os sinais de entrada e saída fossem valores binários. Característica esta referenciada como propriedade "tudo ou nada" (HAYKIN, 2001).

O modelo do neurônio artificial pode ser matematicamente representado pela Equação 4.1, mostrada a seguir:

$$u_k = \sum_{j=1}^{m} \omega_{kj} \cdot x_j \tag{4.1}$$

onde m representa o número de entradas de um determinado neurônio k. Por sua vez, a saída y_k é dada pela função de ativação $\varphi(u_k)$, ou seja:

$$y_k = \varphi(u_k) \tag{4.2}$$

onde a função de ativação implementada por McCulloch e Pitts (1943) consistia numa função degrau, definida pela Equação 4.3:

$$\varphi(u_k) = \begin{cases} 1 & \text{se } u \geq 0; \\ 0 & \text{se } u < 0. \end{cases} \tag{4.3}$$

Um conjunto de outras funções de ativação são apresentadas na literatura, e segundo Haykin (2001) as funções do tipo sigmóide são as mais utilizada na construção de redes neurais artificiais, pois possuem um comportamento entre o linear e o não linear.

O objetivo da construção do perceptron era a aprendizado, porém, o primeiro modelo de aprendizado supervisionada foi apresentado por Rosenblatt (1958) e consistia numa rede de perceptron de camada única, sendo esta, a forma mais simples de uma rede neural artificial, usada para classificar padrões linearmente separáveis.

4.1.2 Rede Neural

A rede neural consiste basicamente na interligação de um conjunto de neurônios que se auto influenciam, e possui a capacidade de adquirir conhecimento através da observação de exemplos, podendo, após o treinamento, realizar a decisão sobre novas situações apresentadas. Em geral podem ser apresentadas como um grafo orientado, onde os neurônios são os vértices e as sinapses as arestas, e a direção informa o sentido dos dados.

O aprendizado da rede pode ser supervisionado ou não supervisionado. No aprendizado supervisionado, uma situação de exemplo é previamente apresentada, no outro tipo de aprendizado isso não ocorre. O conhecimento obtido é armazenado na forma de pesos das conexões sinápticas, que são ajustadas a fim de que a rede tome a decisão correta, quando apresentada a novas entradas (HAYKIN, 2001).

O ajuste dos pesos das conexões sinápticas é de responsabilidade dos algoritmos de aprendizado. Entre os vários algoritmos apresentados na literatura o *backpropagation* é o mais utilizado (RUMELHART; HINTON; WILLIAMS, 1986).

O processo de construção de uma rede neural é composto de três etapas: a definição

da topologia, a estratégia para aprendizado e a determinação da função de ativação que se apresente mais adequada.

4.1.2.1 Topologia

A topologia de uma rede neural define a forma como os neurônios estão dispostos e pode ser dividida em três classes: *feed-forward network*; redes recorrentes e; as redes competitivas. Para esse trabalho somente o entendimento das redes *feed-forward network* será necessário.

As redes *feed-forward* são organizadas em camadas, com cada uma possuindo um conjunto de neurónios ordenados sequencialmente. O fluxo da informação ou impulso é sempre da camada de entrada para a camada de saída. Essas redes podem ser de camada única ou de múltiplas camadas, Figura 4.2

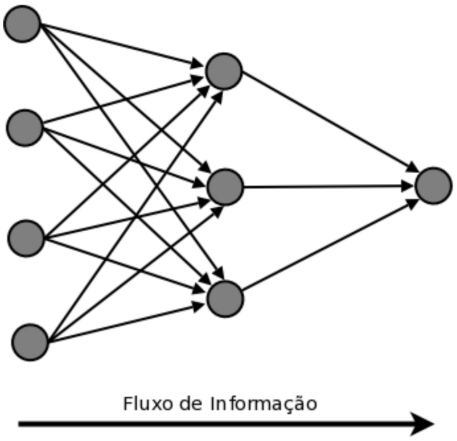


Figura 4.2: Rede neural apresentada como um grafo orientado.

As principais características de uma rede *feed-forward* são: i) O sinal de entrada é recebido na camada inicial, e o resultado é informado pela camada de saída. Podendo ou não possuir camadas intermediárias que são chamadas de camadas ocultas; ii) Cada neurônio de uma camada é conectado com todos os neurônios da camada seguinte; iii) Não há conexão entre os neurônios de uma mesma camada.

4.1.2.2 Aprendizado

O processo de aprendizado de uma rede neural é sua principal característica. O aprendizado de uma rede consiste no ajuste da representação interna em resposta ao estímulo externo, visando desempenhar uma tarefa específica (HAYKIN, 2001). O ajuste da re-

presentação interna ocorre através da correção dos pesos sinápticos entre os neurônios, com as regras de aprendizado definindo como a rede efetuará a correção. Haykin (2001) identifica quatro tipos de aprendizado:

1. **Aprendizado por correção de erro**: o erro consiste na diferença entre o valor da saída e o valor esperado pela rede, esta técnica ajusta os pesos sinápticos visando diminuir o erro. Ela é utilizada em treinamento supervisionado.

2. **Aprendizado Hebbiana**: esse modelo tem por base o postulado de Hebb (1949) que afirma: "se dois neurônios em ambos os lados de uma sinapse são ativados síncrona e simultaneamente, então a força daquela sinapse é seletivamente aumentada". O ajuste dos pesos é feito localmente durante o treinamento, de acordo com a atividade de cada neurônio.

3. **Aprendizado de Boltzmann**: esse modelo de aprendizado utiliza as ideias da mecânica estatística, sendo utilizado no processo de aprendizado não supervisionado, pois modela a distribuição de probabilidade especifica de cada neurônio. Possuindo dois estados possíveis, ligado (+1) e desligado (-1).

4. **Aprendizado Competitivo**: nesse modelo, ocorre uma competição entre os neurônios, pois somente um deles será ativo e os pesos dos outros, próximos a eles, terão seus valores ajustados.

4.1.3 Multilayer Perceptron

As redes *Multilayer Perceptron* (MLP) são redes *feed-forward* com aprendizado por correção de erro, possuindo uma ou mais camadas ocultas (LIPPMANN, 1987). Essa característica permite com que as redes MLPs consigam classificar padrões não lineares.

O desenvolvimento de uma rede MLP, ficou durante muitos anos limitado devido à falta de um algoritmo de treinamento adequado, porém, Rumelhart, Hinton e Williams (1986), desenvolveram o algoritmo de retro propagação de erro(*backpropagation*) o que permitiu o desenvolvimento de redes com múltiplas camadas.

O algoritmo possui basicamente duas fases: a fase de propagação que transmite os valores da entrada até a saída passando pelos neurônios da camada oculta. E a fase de retro propagação, que ajusta os pesos sinápticos com base no erro encontrado na saída. A Figura 4.3 mostra uma rede MLP.

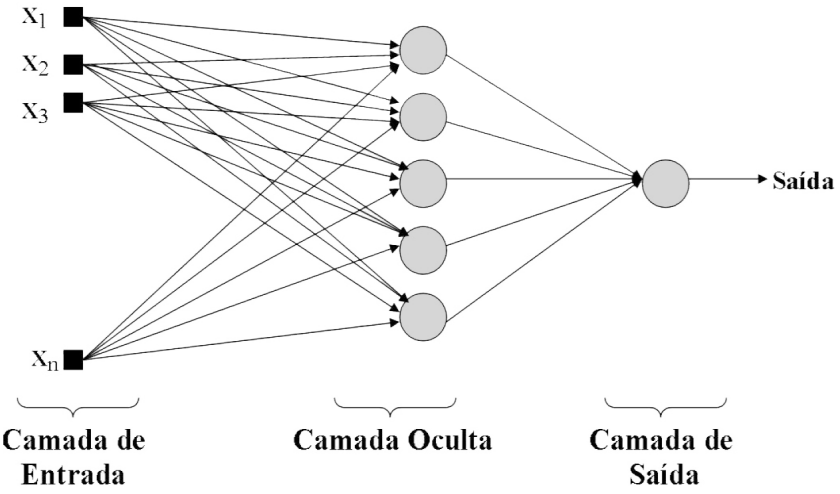


Figura 4.3: Arquitetura de uma rede MLP.

O algoritmo 1, mostra o pseudocódigo do *backpropagation*, que minimiza a função custo na direção contrária ao gradiente do erro (LIPPMANN, 1987).

Algoritmo 1: Pseudocódigo do *backpropagation*.

```
1 Atribuição dos valores iniciais;
2 repita
3 |   Apresentação à rede dos padrões de entrada e as saídas desejadas;
4 |   Cálculo dos valores de saída dos neurônios ocultos;
5 |   Cálculo dos valores de saída dos neurônios de saída (resposta real da rede);
6 |   Cálculo do erro (diferença entre a resposta da rede e o valor esperado);
7 |   Ajuste dos pesos sinápticos;
8 |
até condição de parada não satisfeita;
```

O código se inicia na linha 1 com a atribuição aleatória dos valores iniciais dos pesos sinápticos, o intervalo [0,1] é geralmente escolhido. Na linha 3 os dados são apresentados à rede, bem como os valores esperados. Os cálculos dos valores de saída são realizados, nas linhas 4 e 5, através da aplicação da função de ativação (HAYKIN, 2001).

O valor da saída para um neurônio j na camada l na iteração n é dado pela equação 4.4 :

$$v_j^{(l)}(n) = \sum_{i=1}^{r+1} w_{ji}^{(l)}(n) y_i^{l-1}(n) \tag{4.4}$$

onde $y_i^{l-1}(n)$ é à saída do neurônio i na camada $l-1$, na iteração n. $w_{ji}^{(l)}$ é o peso sináptico do neurônio j da camada l. A variável r é o número de neurônios da camada anterior

$(l-1)$. O uso de $r+1$ é devido ao bias que é representado como um neurônio. O bias equivale a: $y_{r+1}^{l-1}(n) = +1$

O valor da saída de uma neurônio j na camada l é dado pela função de ativação $\varphi(.)$. A equação 4.5 define a saída com base na função de ativação.

$$v_j^{(l)} = \varphi(v_j(n)) \tag{4.5}$$

O erro da rede na iteração n, é calculado na linha 6, e é dado pela equação 4.6, onde d_j é a j-ésima resposta desejada é y_j é a j-ésima resposta da rede.

$$e_j(n) = d_j(n) - y_j(n) \tag{4.6}$$

A grande vantagem desse algoritmo é a sua capacidade de ajustar os erros da camada oculta, ajuste feito na linha 7 do algoritmo, sendo executado da camada oculta para a camada de entrada. Qualquer camada l, com pesos $w_{ji}^{(j)}$ na iteração n, terá seus pesos ajustados com base na iteração anterior $n-1$. Esse ajuste é dado pela equação 4.7

$$w_{ji}^{(j)}(n) = w_{ji}^{(j)}(n-1) + \Delta w_{ji}^{(j)}(n) \tag{4.7}$$

onde $\Delta w_{ji}^{(j)}(n)$, consiste na correção aplicada, determinada pela regra delta modificada, definida na equação 4.8

$$\Delta w_{ji}^{(j)}(n+1) = \eta \delta_j^{(l)} y_i^{(l-1)} + \mu \Delta w_{ji}^{(j)}(n) \tag{4.8}$$

onde η é a taxa de aprendizado que define o tamanho do passo de atualização, $\delta_j^{(l)}$ é o gradiente local, μ a constante de momento (CM) que é utilizado para que o método possa fugir de mínimos locais na superfície de erro, e $y_i^{(l-1)}$ é a saída do neurônio i na camada anterior $l-1$.

Porém, o valor do gradiente é computado de maneira diferente entre os neurônios da camada de saída (L) e os da camada oculta, pois o gradiente do neurônio j, na iteração n é calculado através da equação 4.9.

$$\delta_j^{(L)}(n) = e_j^{(L)}(n)\delta'(v_j^{(L)}) \tag{4.9}$$

onde $\delta'(.)$ é a derivada da função de ativação, definida na equação 4.10:

$$\delta_j'^{(l)}(n) = \delta'(v_j^{(l)}) \left(\sum_{i=1}^{r+1} \delta_i^{l+1} w_{ij}^{l+1} \right) \tag{4.10}$$

onde l é uma camada oculta qualquer.

A linha 8 é o critério de parada, que segundo Basheer e Hajmeer (2000) pode ser determinado através: (i) do erro de treinamento ($e < \epsilon$), (ii) do gradiente do erro menor que um δ' ou (iii) utilizando técnica de validação cruzada.

A implementação de uma rede MLP de forma a obter bons resultados, necessita de que a mesma possua boas configurações. Basheer e Hajmeer (2000) definem a forma de montagem de boas redes MLP, porém, alguns parâmetros ainda são determinados por tentativa e erro. Entre esses parâmetros destacam-se: a taxa de aprendizado e a quantidade de camadas ocultas, bem como o número de neurônios dessa camada.

Para minimizar esse problema, foram desenvolvidos vários algoritmos, sendo as redes resilientes uma opção que apresenta resultados interessantes. A próxima seção explica o funcionamento de uma rede resiliente, que foi a rede implementada nesse trabalho.

4.1.3.1 Rede Resiliente

O conceito de resiliência, ou resiliente, pode ser definido como alguém ou alguma coisa, com capacidade de se adequar a uma situação inesperada, sendo assim flexível. Aplicando esse conceito à rede, uma rede resiliente, possui a capacidade de se adaptar, da melhor forma, aos dados apresentados. Redes resilientes não necessitam que a taxa de aprendizado seja informada, pois a mesma é atualizada pelo algoritmo de aprendizado desenvolvido, resolvendo assim uma das principais dificuldades apresentadas por Basheer e Hajmeer (2000), que é a definição da taxa de aprendizado.

O algoritmo RPROP (***R****esiliente back****prop****agation*), implementado por Riedmiller e Braun (1993), utilizou o conceito de resiliência para atualizar os valores dos pesos $w_{ji}^{(j)}$, e da taxa de aprendizado η, obtendo melhores resultados que o *backpropagation* tradicional nos testes realizados. Este algoritmo atualiza os valores dos pesos de acordo com o sinal da derivada parcial do erro $e_j(n)$ em relação ao peso $w_{ji}^{(j)}$ na $n-$ésima iteração.

Para que isso seja possível, cada peso $w_{ji}^{(j)}$ possui um valor de atualização Δ_{ji}, que é

aplicado utilizando a seguinte regra:

$$\Delta_{ji}^{n} = \begin{cases} \eta^{+}.\Delta_{ji}^{n-1} & \text{se } \frac{\partial e}{\partial w_{ji}}^{(n-1)} \cdot \frac{\partial e}{\partial w_{ji}}^{(n)} > 0; \\ \eta^{-}.\Delta_{ji}^{n-1} & \text{se } \frac{\partial e}{\partial w_{ji}}^{(n-1)} \cdot \frac{\partial e}{\partial w_{ji}}^{(n)} < 0; \\ \Delta_{ji}^{n-1} & \text{se } \frac{\partial e}{\partial w_{ji}}^{(n-1)} \cdot \frac{\partial e}{\partial w_{ji}}^{(n)} = 0. \end{cases} \tag{4.11}$$

Se o valor da derivada muda de sinal entre a interação anterior e atual, significa que a atualização anterior foi muito alta, logo o fator η^{-} diminui o valor de Δ_{ji}^{n}. Porém, se o sinal é mantido o fator η^{+}, aumenta ligeiramente o valor de Δ_{ji}^{n} visando acelerar o processo de convergência. Atualizado os valores de Δ_{ji}^{n}, o próximo passo é a atualização do pesos $w_{ji}^{(j)}$, que é dada pela regra:

$$\Delta w_{ji}^{(n)} = \begin{cases} -\Delta_{ji}^{(n)} & \text{se } \frac{\partial e}{\partial w_{ji}}^{(n)} > 0; \\ +\Delta_{ji}^{(n)} & \text{se } \frac{\partial e}{\partial w_{ji}}^{(n)} < 0; \\ 0 & \text{se } \frac{\partial e}{\partial w_{ji}}^{(n)} = 0. \end{cases} \tag{4.12}$$

$$w_{ji}^{(n+1)} = w_{ji}^{(n)} + \Delta w_{ji}^{(n)} \tag{4.13}$$

Existe uma exceção para a regra acima, que ocorre quando a derivada parcial muda de sinal, ou seja, o passo anterior foi tão grande que o mínimo desaparece, então o peso é revertido (Equação 4.14).

$$\Delta w_{ji}^{(n)} = -\Delta w_{ji}^{(n-1)}, se \frac{\partial e}{\partial w_{ji}}^{(n-1)} \cdot \frac{\partial e}{\partial w_{ji}}^{(n)} < 0 \tag{4.14}$$

O algoritmo 2 descreve o pseudocódigo para o RPROP, como descrito acima.

Os parâmetros de inicialização utilizados pelo algoritmo, por indicação dos autores são: $\Delta_0 = 0,1$; $\Delta_{max} = 50,0$; $\Delta_{min} = 0,0000001$; $\eta_{+} = 1,2$; $\eta_{-} = 0,5$.

Algoritmo 2: Pseudocódigo do RPROP.

para cada $w_{ji}^{(n)}$ **faça**
  se $\frac{\partial e}{\partial w_{ji}}^{(n-1)} \cdot \frac{\partial e}{\partial w_{ji}}^{(n)} > 0$ **então**
    $\Delta_{ji}^{(n)} = min(\Delta_{ji}^{(n-1)} \cdot \eta^{+}, \Delta_{max})$;
    $\Delta w_{ji}^{(n)} = -sing(\frac{\partial e}{\partial w_{ji}}^{(n)}) \cdot \Delta_{ji}^{(n)}$;
    $w_{ji}^{(n+1)} = w_{ji}^{(n)} + \Delta w_{ji}^{(n)}$;
  senão se $\frac{\partial e}{\partial w_{ji}}^{(n-1)} \cdot \frac{\partial e}{\partial w_{ji}}^{(n)} < 0$ **então**
    $\Delta_{ji}^{(n)} = max(\Delta_{ji}^{(n-1)} \cdot \eta^{-}, \Delta_{min})$;
    $w_{ji}^{(n+1)} = w_{ji}^{(n)} - \Delta w_{ji}^{(n-1)}$;
    $\Delta w_{ji}^{(n)} = 0$;
  senão se $\frac{\partial e}{\partial w_{ji}}^{(n-1)} \cdot \frac{\partial e}{\partial w_{ji}}^{(n)} == 0$ **então**
    $\Delta w_{ji}^{(n)} = -sing(\frac{\partial e}{\partial w_{ji}}^{(n)}) \cdot \Delta_{ji}^{(n)}$;
    $w_{ji}^{(n+1)} = w_{ji}^{(n)} + \Delta w_{ji}^{(n)}$;
fim para cada

4.1.3.2 *Overtraining*

Uma das principais dificuldades levantadas por Basheer e Hajmeer (2000) está no número de ciclos ou épocas que a rede deve ser treinada. Se o número de ciclos for elevado a rede pode sofrer um super ajuste aos dados da base de treino, ocorrendo o *overtraining* (Figura 4.4), se o número for baixo, ele pode ser incapaz de ajustar todos os seus pesos para representar corretamente o conjunto de dados.

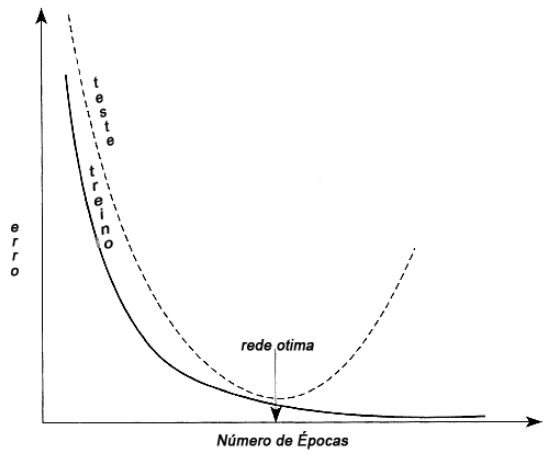


Figura 4.4: Fenómeno do *overtraining*. - (**Fonte:** *Basheer e Hajmeer (2000)*)

O *overtraining* pode ser evitado, com o uso correto da validação cruzada. Essa técnica consiste em avaliar o erro nas amostras de treino e de teste, e quando a rede obtém o valor ótimo no conjunto de teste, o treinamento é interrompido. Essa situação pode não ocorrer quando o valor dos dados são uniformes, ou quando já obteve o máximo do seu treinamento, e o erro passa a não variar mais (KOHAVI, 1995).

4.1.4 Considerações

A utilização de uma estratégia de aprendizado de máquina para a detecção de SNPs e também de falsos positivos apresenta características bem peculiares. Na prática, tem-se um problema de filtragem onde, dado um conjunto de possíveis SNPs pré-identificados, busca-se determinar os candidatos que apresentam maior possibilidade de realmente indicar um polimorfismo. Na literatura não se detectou adaptações de ferramentas de aprendizado de máquina para a filtragem de SNPs.

As estratégias que podem ser aplicadas no problema de filtragem de SNPs são: o aprendizado com classe única, classificação binária, e multiclasse. Desta forma, como primeiro estudo optou-se por utilizar a estratégia de classificação binária, para a determinação de um processo para filtragem de SNPs. O desenvolvimento de métodos de aprendizado de máquina para filtragem de SNPs será feito utilizando um procedimento tradicional de classificação binária, ou seja, busca-se determinar se a instância avaliada é um SNP ou apresenta características especificas de ser um falso positivo. Para isto, necessita-se de uma classificação prévia para construção de uma base de treinamento que será utilizada na determinação da hipótese de classificação. Ressalta-se que se pretende explorar outras possibilidades em trabalhos futuros.

No próximo capítulo serão apresentados os modelos desenvolvidos para geração da classe das instâncias e, de outra maneira, da própria construção de uma base especifica para ser utilizada no processo de treinamento do indutor.

5 IMPLEMENTAÇÃO DE UMA ESTRATÉGIA BASEADA EM REDES NEURAIS PARA DETECÇÃO DE SNPS

Neste capítulo é apresentado a terceira e última etapa de desenvolvimento do modelo de aprendizado de máquina, que será baseado em redes neurais. É apresentado o processo de estruturação e treinamento da rede neural, além da montagem dos conjuntos de dados utilizados para o seu treinamento. Essa etapa é importante, pois a estratégia utilizada para gerar o conjunto de treinamento, e posteriormente treinar a rede neural, interfere de forma direta na sua capacidade de classificação ou de filtragem.

5.1 Implementação de Filtro Utilizando Redes Neurais

Modelos de classificação supervisionada para filtragem de SNPs ainda não são explorados na literatura especializada. Entre os possíveis motivos estão a dificuldade de se ter uma base de dados confiável tanto para falsos positivos como para SNPs comprovados para a obtenção da hipótese de generalização. Assim, qualquer tentativa de se utilizar classificação supervisionada para a filtragem de SNPs deve passar necessariamente pela definição de uma estratégia para a construção da base de treinamento e/ou determinação da classe das instâncias. Objetiva-se, neste capítulo, apresentar estratégias para desenvolvimento de filtros de SNPs baseados em três modelos: i) utilização de uma pré-filtragem para determinação das classes; ii) construção de bases específicas para maximizar o poder de generalização de uma ferramenta de classificação supervisionada. iii) construção de bases específicas utilizando algumas regras da pré-filtragem.

Ressalta-se que estas estratégias sofrem com problemas relativos a ruído na determinação da classe (SNPs ou falso positivo) de cada candidato. Desta forma, busca-se avaliar o potencial de uma ferramenta de aprendizado supervisionada para contornar esta

característica do problema de detecção de SNPs. A técnica de aprendizado de máquina e inteligência computacional escolhida no desenvolvimento do modelo computacional para filtragem de SNPs em DNA genômico completo, foi a rede neural.

A biblioteca *Fast Artificial Neural Network* (FANN), de autoria de Nissen (2005) foi utilizada para a codificação do modelo. A biblioteca permite a criação de uma rede utilizando um variado número de linguagens de programação, sendo que a linguagem C foi à escolhida para esse trabalho. O processo de montagem de um modelo computacional para filtragem de SNPs pode ser descrito nas seguintes etapas:

1. Montar um conjunto de dados para o treino e para o teste;
2. Treinamento de várias redes com o conjunto de dados fornecido;
3. Análise dos resultados obtidos com os treinamentos e escolha da melhor rede;
4. O programa de filtro, faz a leitura da rede escolhida e filtra os SNPs;
5. Análise dos resultados fornecidos pelo filtro;
6. Quando necessário, refazer o processo.

A implementação de uma rede neural de forma que ela obtenha bons resultados, necessita de que a mesma possua boas configurações. Basheer e Hajmeer (2000) define que alguns parâmetros como: a taxa de aprendizado, a quantidade de camadas ocultas, bem como o número de neurônios dessa camada, e a função de ativação. São em geral definidos por tentativa e erro.

A definição da estrutura da rede, passa pela escolha da função de ativação que melhor se adapte ao problema apresentado. Entre as várias funções de ativação disponíveis, foram escolhidas quatro: a logística devido a seu extenso uso, a gaussiana por ser a função de uso geral, a Elliot por ser uma função com uma complexidade matemática menor (ELLIOTT, 1993), de forma que se espera que a mesma seja mais rápida que a logística, bem como a Elliot simétrica.

A tabela 5.1 mostra as funções de ativação utilizadas no trabalho e suas derivadas, que são utilizadas pelo algoritmo de treinamento, onde η é a taxa de aprendizado, x é a entrada da função de ativação, y é a saída e d e a derivada.

Além das funções de ativação é necessária a definição do número de camadas e de neurônios de cada uma delas. Nessa etapa, foi utilizada como conjunto de dados para

Tabela 5.1: Funções de ativação utilizadas.

Nome	espaço	Função	Derivada
logística	$0 < y < 1$	$y = \frac{1}{1+\exp(-2\cdot\eta\cdot x)}$	$d = 2 \cdot \eta \cdot y \cdot (1 - y)$
gaussiana	$0 < y < 1$	$y = \exp(-\frac{x^2\cdot\eta^2}{1^2})$	$d = -2 \cdot x \cdot y \cdot \eta^2$
Elliot	$0 < y < 1$	$y = \frac{\left(\frac{(x\cdot\eta)}{2}\right)}{(1+\lvert x\cdot\eta\rvert)+0{,}5}$	$d = \eta \cdot \frac{1}{(2(1+\lvert x\cdot\eta\rvert)(1+\lvert x\cdot\eta\rvert))}$
Elliot simétrica	$0 < y < 1$	$y = \frac{x\cdot\eta}{1+\lvert x\cdot\eta\rvert}$	$d = \eta \cdot \frac{1}{((1+\lvert x\cdot\eta\rvert)(1+\lvert x\cdot\eta\rvert))}$

treinamento a saída do software MAQ, que é um arquivo contendo os SNPs identificados na etapa de descoberta. O arquivo possui doze colunas, uma para cada um dos SNPs encontrados, sendo duas utilizadas para identificação, por isso, somente as outras dez foram utilizadas como entradas da rede neural.

A topologia utilizada consistiu em uma rede com dez neurônios na camada de entrada, uma camada oculta com vinte neurônios, sendo esse valor escolhido por meio de testes preliminares. A camada de saída com um neurônio, inicialmente binário, classificando os SNPs em 0 ou 1, simulando o comportamento do filtro do software MAQ. Para treinamento foi utilizado o algoritmo RPROP, fazendo com que não fosse necessária a definição de várias taxas de aprendizado. O conjunto de dados utilizado foi o genoma remontado do *Bos Taurus*. A rede foi implementada seguindo os padrões de código informados no manual do FANN (NISSEN, 2005). Apresenta-se, a seguir, o primeiro modelo.

5.1.1 Primeiro Modelo

O primeiro modelo, baseado em aprendizado de máquina, a ser apresentado para a filtragem de SNPs enquadra-se na linha de classificação com ruído, onde, basicamente, os SNPs apresentam-se poluídos por ruídos. Este ruído é introduzido por uma pré-classificação necessária de ser utilizada para o enquadramento dos candidatos a SNPs. O ruído é proviente das falhas presentes na pré-filtragem.

Neste trabalho, é natural que se use a filtragem obtida pelas expressões lógicas do filtro do MAQ como primeira avaliação dos SNPs e dos falsos positivos. Desta forma, pode-se formar uma base de treinamento com a saída das instâncias sendo definidas pelo resultado do filtro MAQ. O ruído se dá pelas instâncias que são erroneamente classificadas pelo filtro e dificultam assim, o processo de aprendizado da estratégia utilizada, no caso, redes neurais.

A expectativa é que o uso de variáveis adicionais, não utilizadas pelo SNPfilter, bem

como o potencial de uma rede neural com uma ou mais camadas internas representar funções não-lineares, possa gerar uma hipótese de classificação que consiga generalizar adequadamente a filtragem, minimizando o efeito do ruído nas classes.

O conjunto de dados utilizado foi extraído dos arquivos de saída de duas etapas distintas do software MAQ. O primeiro arquivo é oriundo da etapa de descoberta e o segundo da etapa de filtro. Nessas etapas foi utilizado o genoma remontado do *Bos Taurus*, logo o primeiro arquivo possuía $\approx$ 7 milhões de SNPs e o segundo $\approx$ 2 milhões. O arquivo utilizado para treinamento automático da rede neural pela biblioteca FANN, possui uma formatação específica. Por isso, foi desenvolvido um *script* em PHP que varre o primeiro arquivo selecionando aleatoriamente 4.000 SNPs por cromossomo para o conjunto de treino e 2.000 para o conjunto de testes. No processo de montagem do conjunto de treino caso o SNP selecionado esteja presente no arquivo de saída da etapa de filtragem ele era indicado como 1, caso contrário como 0. Ficando dessa forma definido 1 como SNP e 0 como erro.

A definição dos valores de 4.000 e 2.000 SNPs para as amostras de treino e testes foram escolhidos após testes iniciais com vários valores diferentes. Os valores testados, foram desde 2/3 e 1/3 do total, até somente 100 e 50 SNPs por cromossomo. Os valores de 4.000 e 2.000 ficaram muito próximos dos resultados obtidos pelos valores de 2/3 e 1/3 do total.

O algoritmo de treinamento utilizado, o RPROP, não necessita da definição da taxa de aprendizado. Porém, para evitar que o algoritmo sofresse com mínimos locais, foram escolhidas três constantes de momento, sendo elas: $0,1$; $0,5$ e $0,9$. Como a constante de momento vária entre 0 e 1 foram escolhidos valores próximos dos extremos e o meio, de forma a avaliar o comportamento das funções de ativação, de acordo com cada constante de momento.

Após as primeiras análises, foram montados 12 cenários diferentes, utilizando as quatro funções de ativação, com as três constantes de momentos. Cada cenário foi executado dez vezes, variando somente os pesos sinápticos iniciais da rede. Essas variações geram resultados diferentes para uma mesma rede. Assim, após as dez execuções é possível saber o comportamento médio de cada cenário. Cada execução utilizou dois critérios de parada, a saber: erro $< 0,0001$ ou 50.000 épocas. Retira-se uma amostra, para a construção dos gráficos de treino e teste, a cada 50 épocas.

A tabela 5.2, mostra o resultado geral das dez execuções de cada cenário, apresentando a média (M) e o desvio padrão (DP), visando possibilitar a observação do comportamento médio de cada cenário diferente. Essa análise serve para definir o cenário padrão a ser utilizado. Como pode ser visto na tabela 5.2 o cenário composto pela função Elliot com constante de momento igual a $0,5$, obteve menor erro e desvio padrão. Ressalta-se a grande influência na escolha da função de ativação (FA) assim como da constante de momento (CM) na qualidade da filtragem.

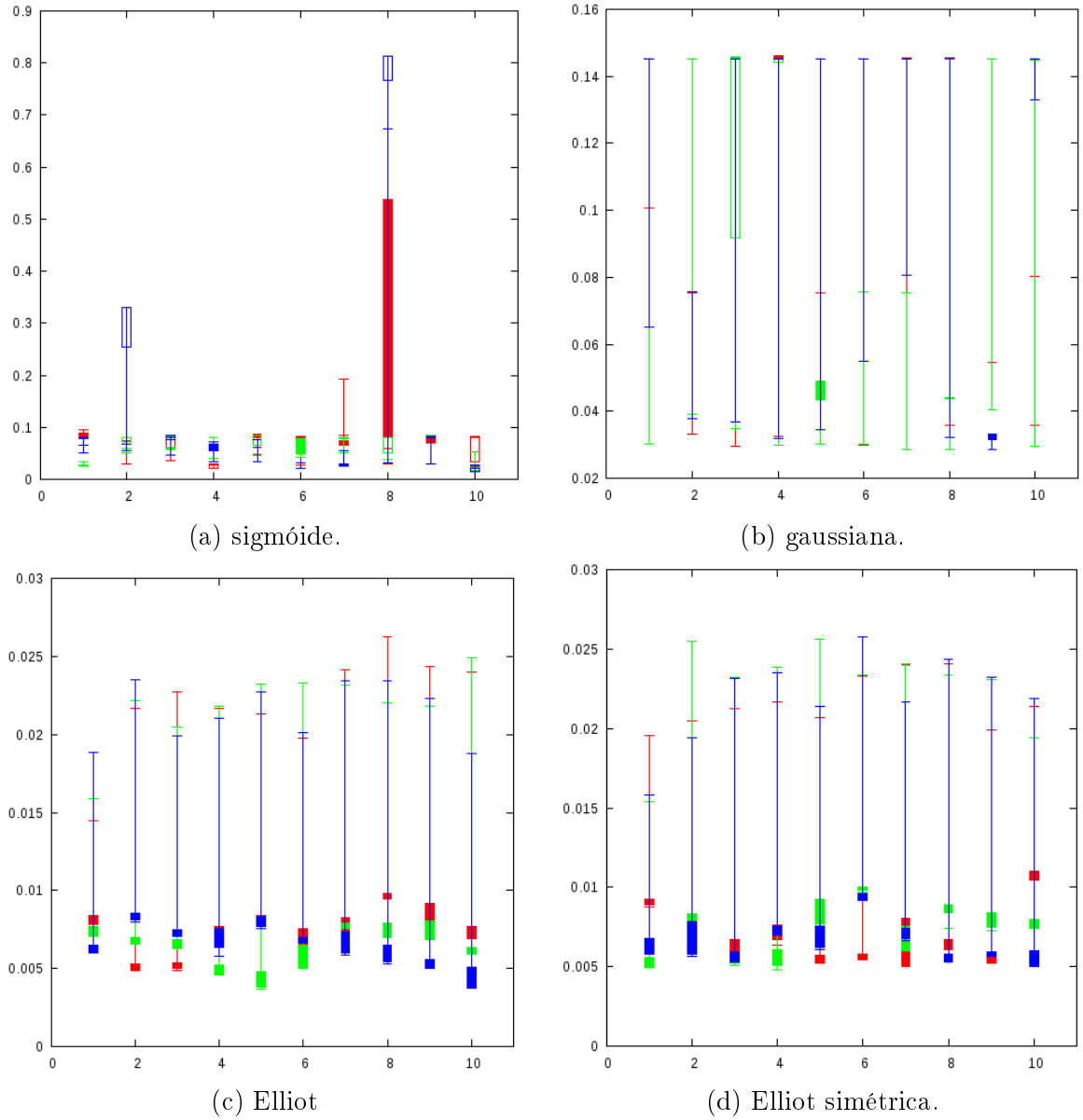
Tabela 5.2: Resultados do erro na primeira etapa.

Treino						
CM / FA	$0,1$		$0,5$		$0,9$	
	M	DP	M	DP	M	DP
sigmóide	$18,81\%$	$11,30\%$	$24,78\%$	$11,71\%$	$23,19\%$	$18,66\%$
gaussiana	$18,00\%$	$06,00\%$	$14,63\%$	$05,71\%$	$14,55\%$	$04,13\%$
Elliot	$01,43\%$	$00,33\%$	$01,23\%$	$00,24\%$	$01,26\%$	$00,28\%$
Elliot simétrica	$01,35\%$	$00,37\%$	$01,41\%$	$00,33\%$	$01,26\%$	$00,26\%$
Teste						
CM / FA	$0,1$		$0,5$		$0,9$	
	M	DP	M	DP	M	DP
sigmóide	$41,95\%$	$23,75\%$	$48,70\%$	$22,29\%$	$38,84\%$	$20,35\%$
gaussiana	$36,26\%$	$11,31\%$	$29,51\%$	$11,19\%$	$29,52\%$	$08,32\%$
Elliot	$01,79\%$	$00,62\%$	$01,38\%$	$00,47\%$	$01,45\%$	$00,55\%$
Elliot simétrica	$03,78\%$	$07,44\%$	$01,65\%$	$00,55\%$	$01,46\%$	$00,41\%$

A Figura 5.1 exibe o gráfico de vela, resultante das dez execuções. Nesse gráfico é possível avaliar o comportamento geral de cada função de ativação. Os gráficos estão agrupados por função de ativação, com uma variação de cor para cada constante de momento. É possível verificar que a função sigmóide, variou pouco entre o valor de erro inicial e os valores de erro médio e final, além de em muitos casos possuir um valor maior de erro final.

O gráfico da função gaussiana (Figura 5.1b), possui uma variação maior entre o erro inicial e o final, porém, não é possível ver o corpo da vela. Quando a vela não possui corpo, significa que o valor obtido no meio é igual ao valor inicial ou ao valor final.

Os gráficos das funções Elliot (Figura 5.1c) e Elliot simétrica (Figura 5.1d), possuem um comportamento similar. Em ambos, a haste da "vela" é longa, e com presença de um corpo, indicando que o valor inicial do erro cai rapidamente em relação ao valor médio.



(a) sigmóide.

(b) gaussiana.

(c) Elliot

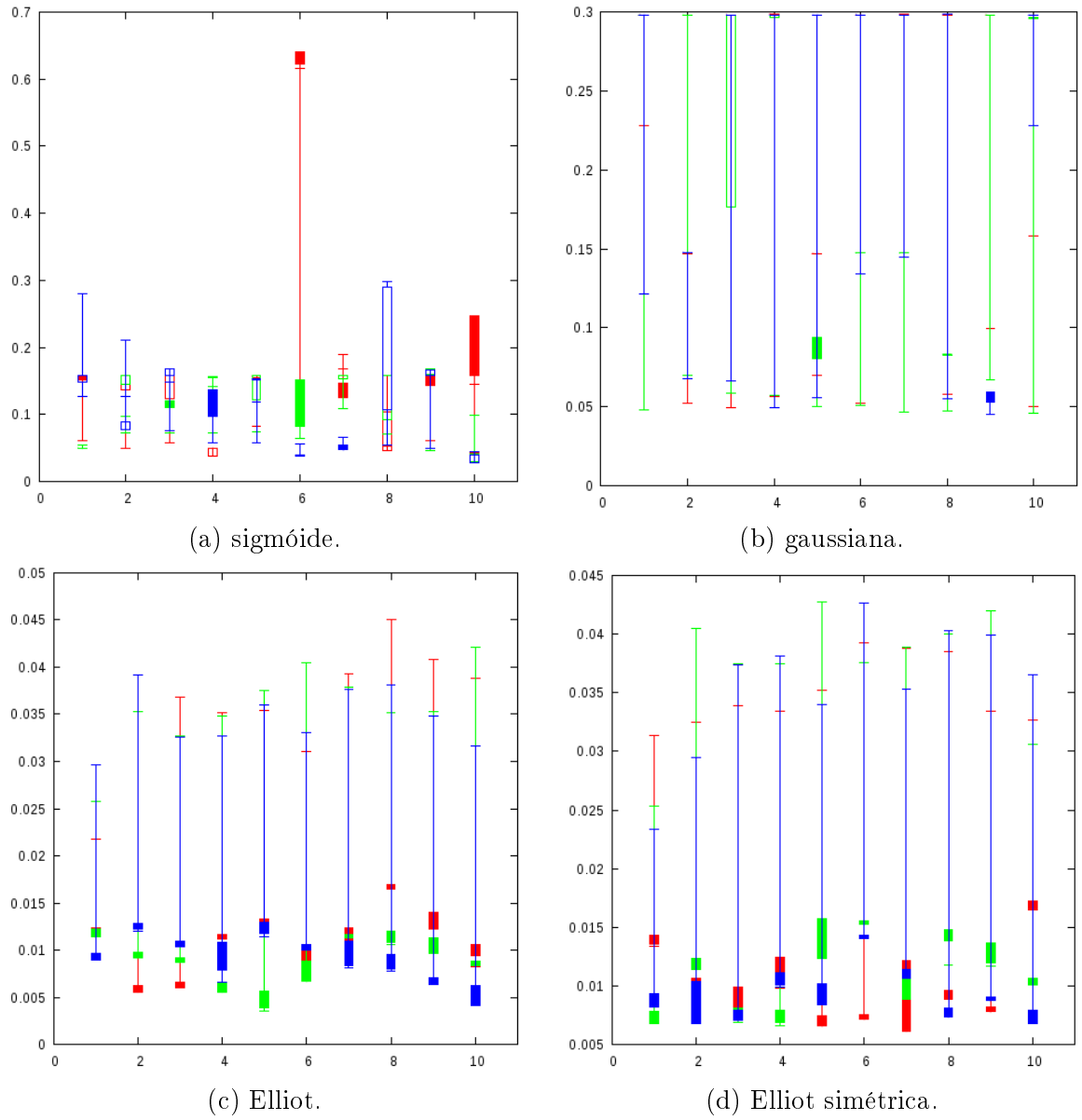
(d) Elliot simétrica.

Figura 5.1: Gráficos com todos os treinamentos de cada cenário. As constantes de momento são: $0,1$ em vermelho; $0,5$ em verde e $0,9$ em azul.

A queda é mantida, pois há presença do corpo e de uma haste inferior. Outro fator importante, é que as "velas" ou execuções são similares, conforme é possível verificar pela pequena variação do erro na Tabela 5.2.

A Figura 5.2 mostra os resultados obtidos pelos cenários durante os testes. Os resultados são próximos aos obtidos pelos treinos, com comportamentos iguais para as diferentes funções de ativações utilizadas.

Como é possível verificar a função de ativação Elliot foi a que obteve os melhores resultados. Yonaba, Anctil e Fortin (2010) em seu trabalho comparou três diferentes funções de ativação sigmóides: a tangente hiperbólica, a bi-hiperbólica e a Elliot, concluindo que



(a) sigmóide.

(b) gaussiana.

(c) Elliot.

(d) Elliot simétrica.

Figura 5.2: Gráficos com todos os testes de cada cenário As constantes de momento são: $0,1$ em vermelho; $0,5$ em verde e $0,9$ em azul.

para as tarefas de reconhecimento de padrão e classificação a função Elliot era mais rápida e apresentava melhores resultados. Os resultados obtidos nesta dissertação corroboram com os resultados obtidos por Yonaba, Anctil e Fortin (2010).

Após a etapa de treinamento, a rede está apta a classificar novos dados. Porém, cada treinamento gera uma rede diferente, com características diferentes, que podem trazer vantagens ou desvantagens na ação de filtrar SNPs. Após concluída a etapa de treinamento, uma rede deve ser definida para a tarefa de filtragem.

A tabela 5.3 mostra o pior e o melhor resultado de cada função de ativação. É possível identificar que todas as funções de ativação conseguem atingir um erro mínimo satisfatório

no melhor caso. Porém, quando comparamos a diferença entre o melhor e o pior resultado, as funções gaussiana e sigmóide, apresentam uma maior diferença se comparado com as funções Elliot e Elliot simétrica.

Tabela 5.3: Melhor e pior resultado de cada função de ativação.

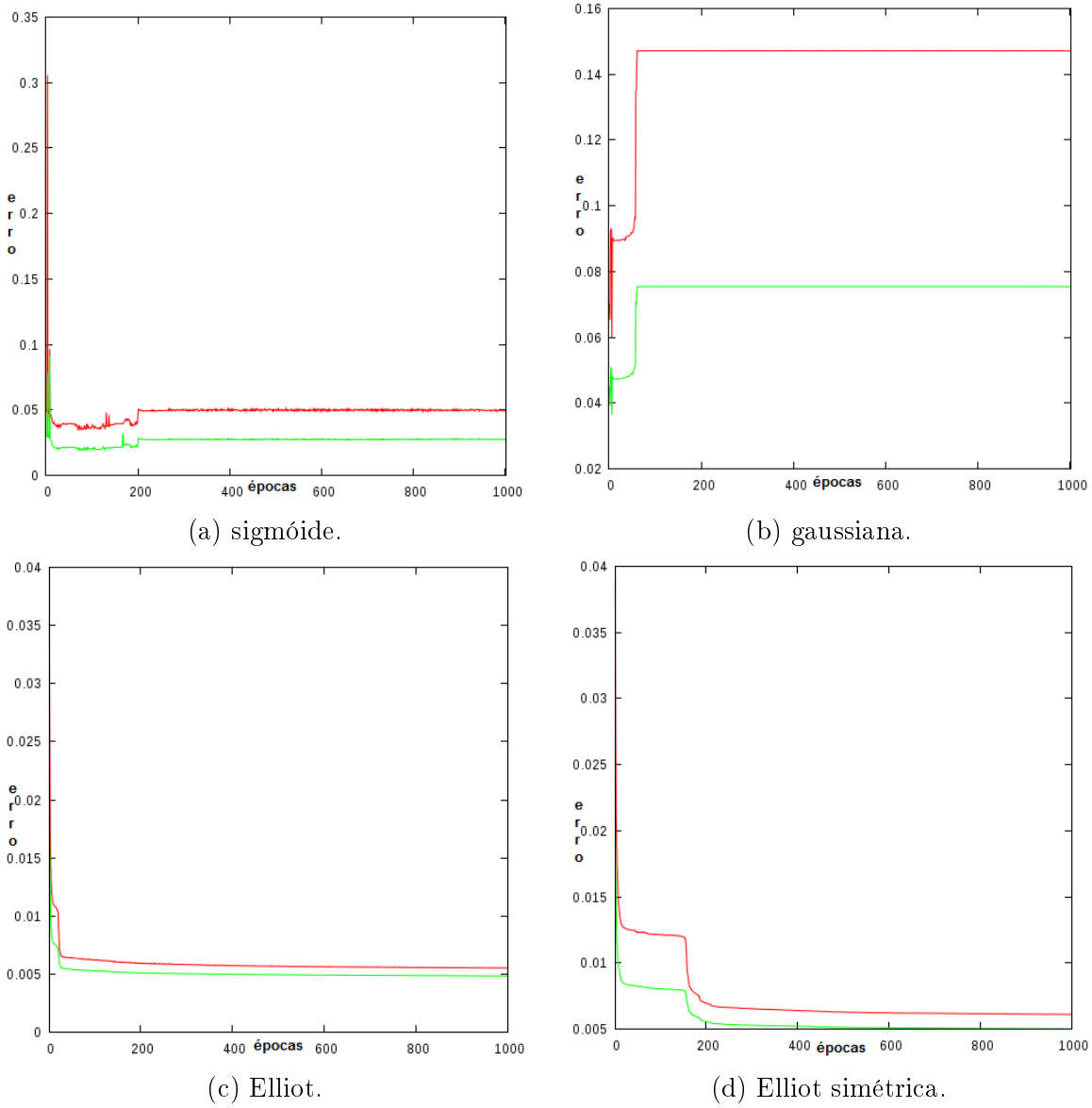
Treino				
	Melhor		Pior	
	Erro	CM	Erro	CM
sigmóide	03,30%	0,5	67,57%	0,9
gaussiana	05,89%	0,9	31,55%	0,1
Elliot	00,82%	0,9	01,82%	0,9
Elliot simétrica	00,92%	0,9	02,16%	0,1
Teste				
	Melhor		Pior	
	Erro	CM	Erro	CM
sigmóide	05,82%	0,5	62,40%	0,5
gaussiana	10,78%	0,9	62,33%	0,1
Elliot	00,55%	0,5	03,00%	0,1
Elliot simétrica	00,80%	0,1	24,89%	0,1

A função Elliot obteve o melhor resultado entre as quatro funções testadas, tanto na etapa de treino como na de teste, além de ser a que obteve o menor erro quando comparado com os piores resultados das outras funções, se mostrando mais robusta para esse problema.

Uma das dificuldades encontradas na montagem de uma rede MLP, é o critério de parada. Uma das técnicas mais utilizadas para definir o melhor ponto de parada, consiste em analisar o erro obtido no conjunto de treinamento e no conjunto de teste, verificando o surgimento do fenômeno de *overtraining* bem explicado por Basheer e Hajmeer (2000). Porém, na rede treinada pelo conjunto de dados oriundo do MAQ esse fenômeno não foi percebido.

A Figura 5.3, apresenta os gráficos obtidos para os melhores resultados em cada função de ativação, com a constante de momento igual a 0,1. Apesar da constante de momento ser baixa, é possível ver que a função de ativação sigmóide, Figura 5.3a, converge rapidamente. Porém, no decorrer das épocas sofre oscilações. A função gaussiana, Figura 5.3b, estabilizou, mas o erro final se mostrou maior que o inicial.

As funções Elliot (Figura 5.3c) e Elliot simétrica (Figura 5.3d), possuem comportamentos similares. Contudo, ao analisarmos o erro mínimo final, é possível verificar que a função Elliot consegue alcançá-lo antes da função Elliot simétrica. Ambas as funções



(a) sigmóide.

(b) gaussiana.

(c) Elliot.

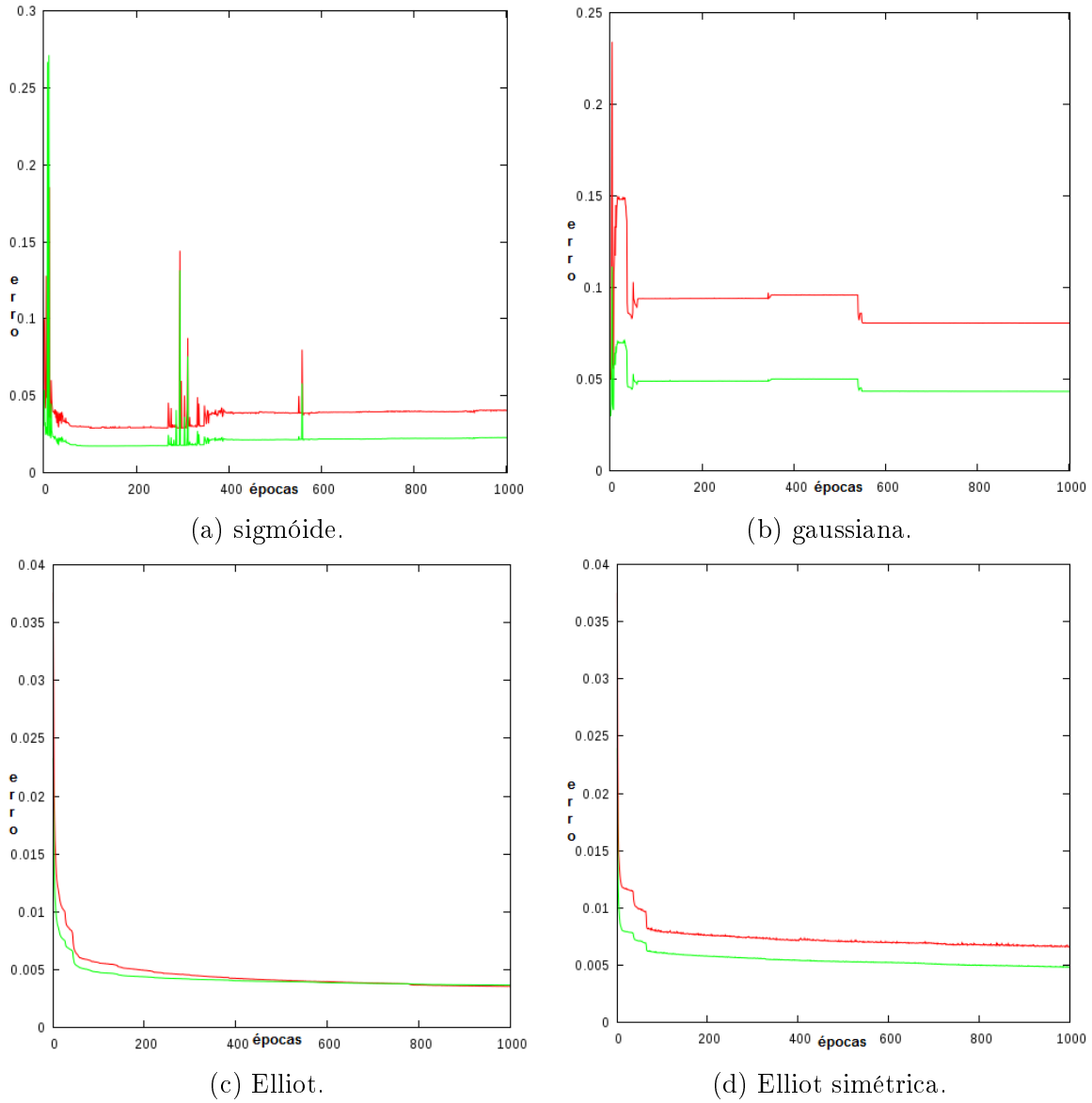
(d) Elliot simétrica.

Figura 5.3: Gráficos dos melhores resultados para cada função de ativação com constantes de momento igual a $0,1$. A linha verde faz referência ao treino, a vermelha ao teste.

estabilizam com erro próximo a $0,005\%$.

A Figura 5.4 mostra o gráfico dos melhores resultados de cada função de ativação, com a constante de momento igual a $0,5$. A função sigmóide, Figura 5.4a, assim como ocorreu na taxa anterior de $0,1$, estabiliza depois de um alto número de épocas. Verifica-se que quando ocorre um aumento no erro de treinamento, o fenômeno se repete no teste, indicando uma padronização nos dados.

A função Elliot com constante de momento igual a $0,5$ (Figura 5.4c), foi dentre todos os cenários o melhor resultado, obtendo um erro menor que todas as outras redes. A função Elliot simétrica (Figura 5.4d), apesar do comportamento similar, possui um erro



(a) sigmóide.

(b) gaussiana.

(c) Elliot.

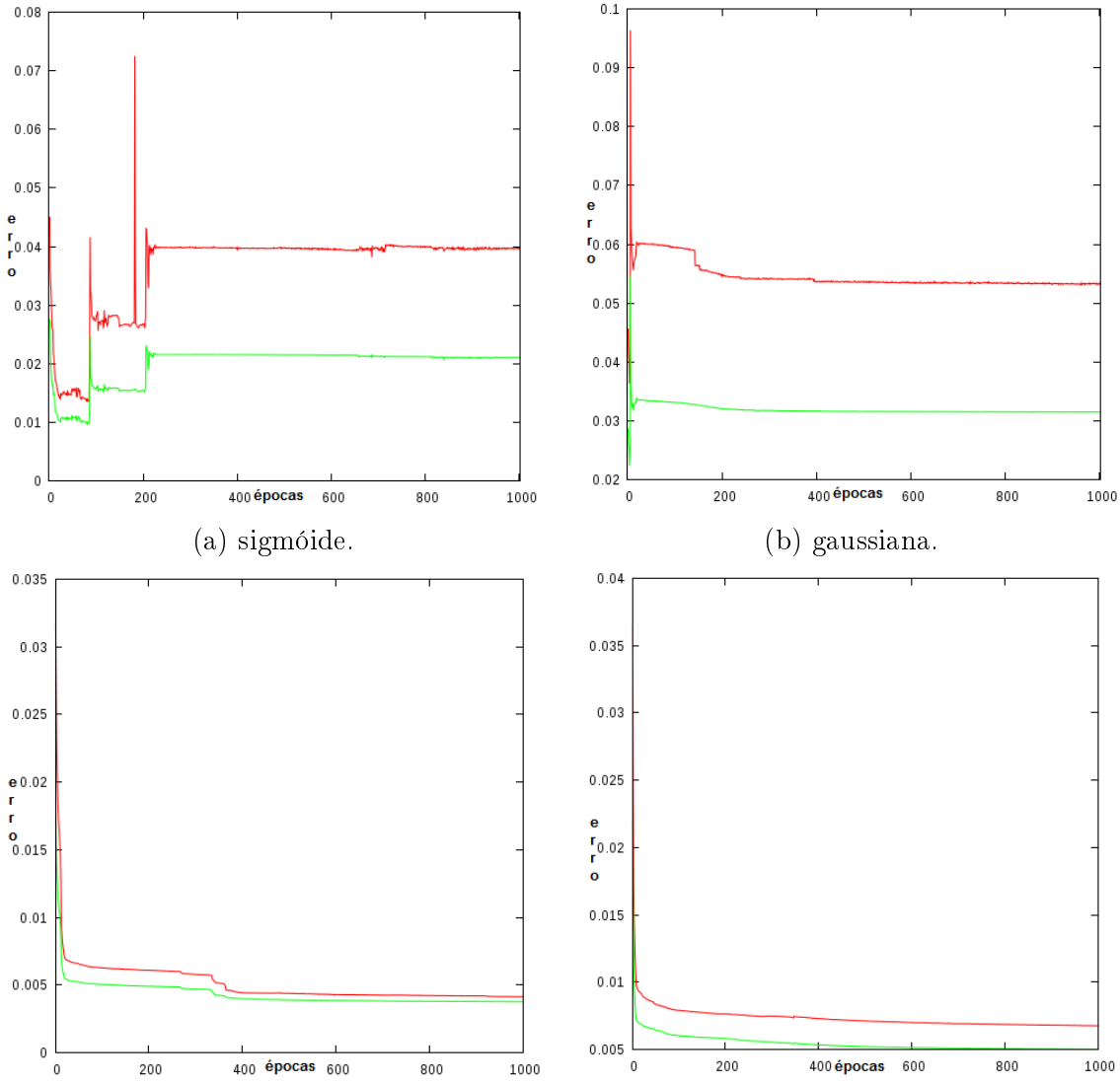
(d) Elliot simétrica.

Figura 5.4: Gráficos de treino e de teste da primeira etapa com constante de momento igual a $0,5$. A linha verde faz referência ao treino, e a vermelha ao teste.

final maior.

A Figura 5.5 apresenta o gráfico dos melhores resultados de cada função de ativação, com a constante de momento igual a $0,9$. Assim como ocorreu nas etapas anteriores, as funções sigmóide (Figura 5.5a) e gaussiana (Figura 5.5b), estabilizam depois de um número maior de épocas em relação às funções Elliot (Figura 5.5c) e Elliot simétrica (Figura 5.5d). Porém, verifica-se a similaridade entre as funções Elliot e Elliot simétrica, mesmo com o aumento da constante de momento.

Como visto, em nenhum momento ocorre o fenômeno de *overtraining* em sua forma padrão. Segundo Basheer e Hajmeer (2000) e Haykin (2001) esse fenômeno pode não



(a) sigmóide.

(b) gaussiana.

(c) Elliot.

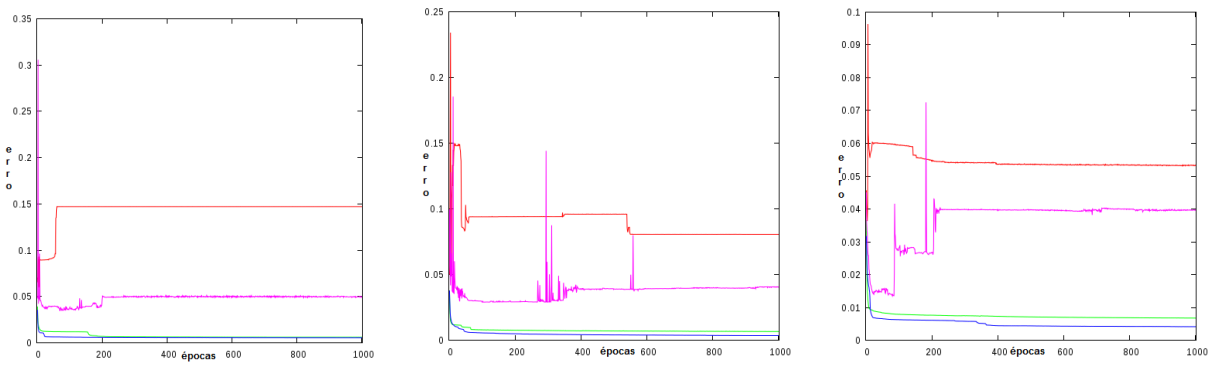
(d) Elliot simétrica.

Figura 5.5: Gráficos de treino e de teste da primeira etapa com constante de momento igual a $0,9$. A linha verde faz referência a treino, a vermelha a teste.

ocorrer se o conjunto de dados for uniforme ou se a rede já tiver obtido o melhor treinamento possível. Porém, visando avaliar se o comportamento atípico da rede estava correto, foi utilizado uma versão *trial* do software Neuro Solution (NEURODIMENSION, 2013), programa já bem estabelecido para a montagem de redes neurais. Utilizando o software foram construídas redes com parâmetros similares aos das redes neurais desenvolvidas nesse trabalho, bem como o uso de um algoritmo de treinamento resiliente. O conjunto de dados utilizados para o treinamento e o teste do NeuroSNP foram submetidos as redes neurais construídas no Neuro Solution. Os resultados obtiveram comportamentos similares aos gráficos de saída das redes implementadas nesse trabalho (Figura 5.5c).

A Figura 5.6, mostra a comparação dos melhores resultados de cada função de ati-

vação. Nesses gráficos é possível ver que a função Elliot, além de obter um erro menor, ela converge mais rápido que as outras funções. Como a função Elliot na sua composição não se utiliza a função exponencial espera-se que sua complexidade seja menor, diminuindo o esforço para o calculo, e tornando a convergência mais rápida (ELLIOTT, 1993; YONABA; ANCTIL; FORTIN, 2010).



(a) Constante de momento igual a $0,1$.

(b) Constante de momento igual a $0,5$.

(c) Constante de momento igual a $0,9$.

Figura 5.6: Gráficos de comparação entre as funções de ativação. Função gaussiana em vermelho, sigmóide em rosa, Elliot em verde e Elliot simétrica em azul.

O primeiro modelo indicou que é possível utilizar técnicas de inteligência computacional, para a filtragem de SNPs. Como visto os erros no treinamento e no teste foram, em geral, baixos. Porém, essa primeira estratégia considera o resultado do filtro do MAQ como correto, e sofreu com a presença de ruídos no conjunto de dados utilizados, que foram gerados pelos erros da filtragem do MAQ. A expectativa é que com a geração de novos conjuntos de dados, com base no conhecimento obtido sobre o funcionamento dos programas de alinhamento e montagem, ocorra a diminuição desse ruído.

5.1.2 Segundo Modelo

Apesar do primeiro modelo ter apresentado resultados interessantes, no que tange ao processo de classificação utilizando classes poluídas, os experimentos iniciais indicaram que a rede neural não apresentou uma capacidade de generalização que obtivesse uma melhor determinação dos SNPs e dos falsos positivos em relação aos determinados pelo filtro MAQ de referência.

Porém, tem-se o indicativo que uma ferramenta de aprendizado de máquina com treinamento adequado pode ser competitiva em relação a filtros tradicionais. Neste segundo

modelo, pretende-se aprimorar a capacidade de generalização do modelo de aprendizado de máquina através de tentativas de minorar a influência do ruído advindo da pré-filtragem utilizada.

Assim, busca-se substituir o uso do filtro MAQ na pré-filtragem por "regras" mais rigorosas para a determinação de SNPs e, principalmente, dos falsos positivos. A nova forma de filtragem, nestes moldes, será menos sensível a casos limítrofes. Assim, estas regras devem criar uma base de treinamento com maior definição principalmente dos falsos positivos. Espera-se que, desta forma, tenha-se benefícios no processo de generalização, com uma maior facilidade no aprendizado e discriminação de novas instâncias.

A determinação das regras para geração da classe da base de treinamento, assim como qualquer filtro, estará sujeita a ruídos. A expectativa é que as mesmas sejam mais rígidas na detecção de falsos positivos que, para este problema especifico, representa a maioria dos dados. Desta maneira, espera-se um reflexo melhor nos resultados, com a generalização facultando uma filtragem mais acurada.

Os conjuntos de dados utilizados foram construídos baseados em duas regras. A primeira regra define um grupo de SNPs com alta confiança, e a segunda um grupo com baixa confiança, onde, por confiança se entende o quanto um *mismatch* poderia ser considerado um SNP. O objetivo é definir um SNP com alta confiança como sendo verdadeiro, e um SNP com baixa confiança como sendo um erro. Os SNPs que não estiverem em nenhum dos dois grupos, serão classificados pela rede. Com base nos testes anteriores, a mesma topologia foi definida para o segundo modelo, utilizando a função de ativação sigmóide Elliot, com constante de momento de $0,5$. Assim, como na etapa anterior, foi utilizado o genoma remontado do *Bos Taurus*.

5.1.2.1 Geração dos Conjuntos de Dados

A primeira regra para a geração do conjunto de dados utilizado no treinamento é a escolha dos SNPs com alta confiança. A regra foi definida após a análise dos parâmetros e do genoma em estudo. Os parâmetros são as doze colunas do arquivo de saída do software MAQ, apresentado anteriormente. As duas primeiras colunas identificam o SNPs, por isso, não são utilizadas nos filtros. As outras 10 possuem informações diversas, sendo que quatro delas informam os nucleotídeos presentes no genoma de referência e no genoma consenso, por isso, essas colunas não são consideradas no momento da seleção dos SNPs

de alta confiança. Os valores de média entre a segunda e a terceira melhor chamada não foi utilizado, por ser uma informação característica do software MAQ. O valor de *hit* não foi utilizado, pois segundo Li, Ruan e Durbin (2008), essa variável pode gerar dúvida no momento do filtro, por isso, está entre os parâmetros apresentado a rede neural, porém, mas não foi considerada na seleção dos SNPs para a montagem do cojunto de dados.

A escolha dos SNPs com alta confiança seguiu os seguintes critérios: profundidade maior ou igual a 6 (o genoma bovino possui profundidade média de $6,98$ por isso, a escolha dos SNPs que estão proximos à ou acima dela); *Phred-like* maior ou igual 20; qualidade de mapeamento e qualidade no flanco de 6 maior ou igual a 50, esse valor foi o mesmo utilizado por Liu et al. (2012) em seu trabalho. Desta forma, foram encontrados 429.078 SNPs no conjunto total, originados do arquivo de descoberta do software MAQ antes do filtro, que satisfaziam estes critérios.

A construção do grupo de SNPs com baixa confiança utilizou os mesmo parâmetros definidos no grupo com alta confiança. O critério para a determinação dos SNPs de baixa confiança, consiste na retirada do conjunto de dados total, os SNPs que possuem pelo menos um dos parâmetros igual a 0, onde 1.821.527 SNPs satisfizeram o critério.

O conjunto de dados montado é constituído de um arquivo de treino com 116.000 entradas e um arquivo de teste com 58.000, constituído de forma balanceada, ou seja, metade oriunda do conjunto de dados com alta confiança e a outra metade do conjunto com baixa confiança. Além disto, utilizou-se o mesmo número de SNPs para cada um dos 29 cromossomos presentes no genoma bovino estudado.

5.1.3 Terceiro Modelo

Neste terceiro modelo, pretende-se, assim como no segundo, aprimorar a capacidade de generalização do modelo de aprendizado de máquina. A diferença entre o segundo e o terceiro modelo, consiste na regra de escolha dos SNPs com alta confiança, que nesse modelo é menos restritiva. A diferença está também na não consideração de SNPs que tenham algum parâmetro nulo. O conjunto de dados foi construído com base em duas regras que serão descritas a seguir. Assim como nos casos anteriores, será utilizada a mesma topologia definida para o primeiro e o segundo modelos. Também como na etapa anterior, foi utilizado o genoma remontado do *Bos Taurus*.

5.1.3.1 Geração dos Conjuntos de Dados

A escolha dos SNPs com alta confiança seguiu os seguintes critérios: profundidade maior ou igual a 6; *Phred-like* maior ou igual a 20; Qualidade de mapeamento maior ou igual a 40 e qualidade no flanco maior ou igual a 20. Nota-se que, os valores utilizados são iguais ao do filtro do MAQ, exceto pelo valor de profundidade.

Os SNPs de baixa confiança, não possuem critérios diferentes para o segundo e o terceiro modelo. Ou seja, os mesmos SNPs considerados de baixa confiança foram utilizados na montagem dos conjuntos de dados dos dois modelos, o segundo e o terceiro.

Os dois novos conjuntos de dados possuem conteúdos diferentes, porém, foram montados de forma idêntica. Cada conjunto de dados é constituído de um arquivo de treino com 116.000 entradas e um arquivo de teste com 58.000. Como no segundo modelo as bases são balanceadas, com metade oriunda do conjunto de dados com alta confiança e a outra metade do conjunto com baixa confiança. Mantém-se a mesma representatividade nas bases para os 29 cromossomos presentes no genoma bovino estudado.

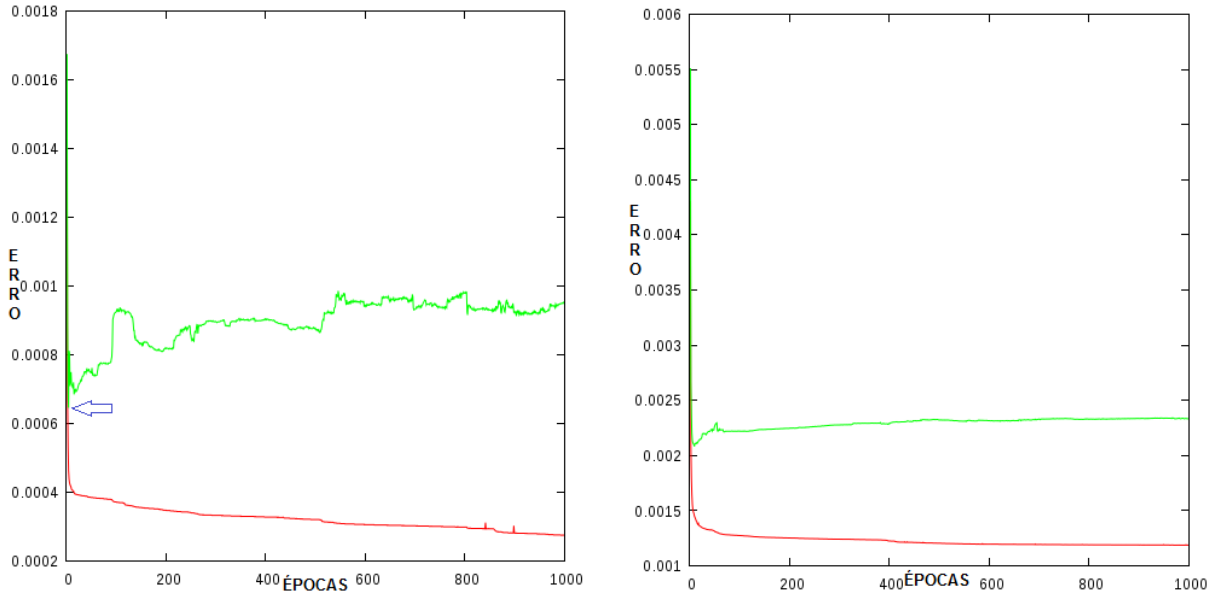
5.1.4 Treinamento do Segundo e do Terceiro Modelos

Os dois novos modelos foram treinados, cada um com seu conjunto de dados específico. Realizou-se dez treinamentos, selecionando o melhor para a construção do gráfico de treino e teste. A Figura 5.7a apresenta o resultado da etapa de treinamento do segundo modelo. Diferente do gráfico do primeiro modelo, é possível ver um pequeno aumento na curva de teste, enquanto a curva de treino continua diminuíndo. Esse fenômeno é o indicativo de parada do processo de treinamento. A função Elliot, convergiu rapidamente para um resultado ótimo, assim como esperado.

O algoritmo implementado para o treinamento salva a rede com seus respectivos pesos sinápticos, quando o erro no teste for menor do que o erro anterior, momento esse indicado pela seta azul. Desta forma, os parâmetros da rede neural de melhor desempenho é armazenado para posterior uso. Mesmo que o algoritmo não pare o treinamento, a rede armazenada é aquela que obteve o menor erro no teste.

A Figura 5.7a mostra o gráfico da etapa do treinamento do segundo modelo. Como é possível ver, o erro aumenta no conjunto de testes, enquanto diminui no conjunto de treino. A Figura 5.7b, mostra o gráfico de treinamento do terceiro modelo, onde também é possivel observar um aumento do erro no teste, porém, em menor escala se comparado

ao segundo modelo.



(a) Segundo modelo.

(b) Terceiro modelo.

Figura 5.7: Gráfico do treinamento do segundo e do terceiro modelo, treinamento em vermelho e teste em verde.

Os resultados na etapa de treinamento dos três modelos, utilizando bases de dados diferentes, indicam um comportamento bastante diferenciado entre os mesmos. Não é trivial identificar qual modelo apresenta resultados de maior qualidade. Pretende-se aplicar as redes neurais geradas para cada um dos modelos em genomas completos podendo, assim, obter um melhor indicativo em relação a qualidade das bases utilizadas na geração das redes neurais. A seguir, apresenta-se a construção do filtro que utiliza as redes neurais treinadas, para posterior aplicação em genomas completos.

5.1.5 Implementando o filtro NeuroSNP

Após a conclusão da etapa de treinamento, o próximo passo está relacionado ao desenvolvimento do filtro propriamente dito. O filtro consiste em um algoritmo que lê os parâmetros de uma rede previamente treinada, reconstruindo-a. Visa a filtragem de novas bases de dados de *mismatches*, obtidas no processo de montagem de genomas em geral. O objetivo da aplicação do filtro NeuroSNP nessas bases de *mismatches* é a filtragem desses dados identificando, principalmente, falsos positivos.

Com todos os testes iniciais finalizados, o filtro baseado em técnicas de inteligência computacional, chamado de NeuroSNP, foi finalizado e a chamada do mesmo agora ne-

cessita de quatro parâmetros, explicados na tabela 5.4

Tabela 5.4: Parâmetros do NeuroSNP

Parâmetro	Descrição
-n <arquivo>	Arquivo de saída do treinamento da rede. Esse arquivo contém a estrutura da rede treinada.
-d <arquivo>	Arquivo de origem dos SNPs, por padrão é o arquivo de saída do Software MAQ.
-r <restrição>	Restrição (0 - BAIXA, 1 - ALTA, 2 - MÉDIA).
-o <arquivo>	Arquivo de saída, os SNPs considerados positivos são salvos nesse arquivo.

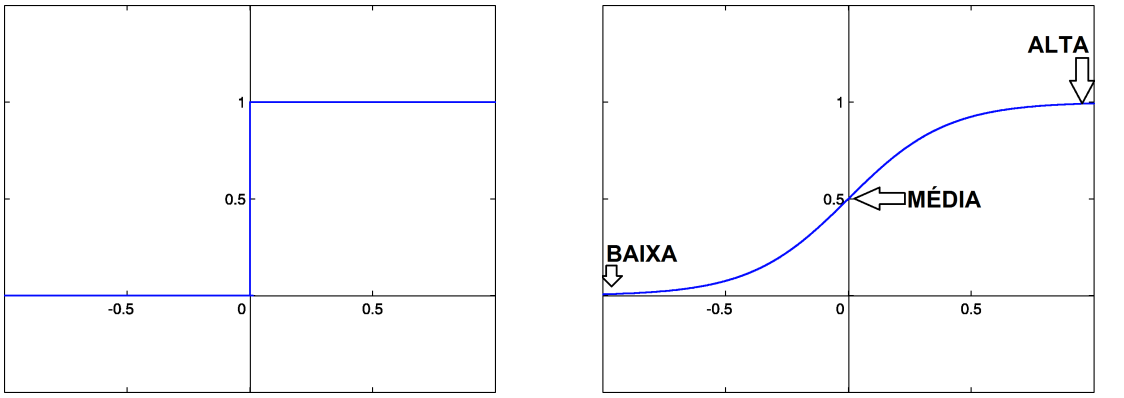
O NeuroSNP recebe os parâmetros da tabela 5.4 no momento da sua chamada. A primeira ação do filtro é remontar a rede neural, com seus pesos. O parâmetro $-n$, contém o caminho para o arquivo de saída da etapa de treinamento da rede, arquivo utilizado para remontar a rede. Em seguida o filtro, inicia a leitura do arquivo com os SNPs, parâmetro $-d$. Cada SNP contido no arquivo possui 12 colunas com sua identificação e características de montagem e alinhamento. O filtro faz a leitura das colunas identificando e apresentando os dados de cada instância a serem utilizados para o processamento da rede. A rede retorna um valor de saída que, se satisfizer a restrição pré-definida, parâmetro $-r$ (que será explicado a seguir) a instância será considerada um SNP, sendo armazenada no arquivo de saída, parâmetro $-o$. A seguir é possível ver o pseudocódigo da NeuroSNP, definido no algoritmo 3.

Algoritmo 3: Pseudo-código da NeuroSNP.

Entrada: Arquivo com os pesos da rede, arquivo de candidatos a SNP, restrição
Saída: Arquivo com os SNPs filtrados

```
1  data = abrir(Arquivo de SNPs);
2  saída = abrir(Arquivo de saída com SNPs filtrados);
3  ann = criar_rede_apartir_arquivo(Arquivo com os pesos da rede);
4  enquanto nao_fim_arquivo(data) faça
5      linha = extrair(data);
6      input = linha;
7      output = 0;
8      zerar_erro_MSE(ann);
9      processar_entrada_rede(ann, input, output);
10     se erro_obtido(ann) > restricao então
11         salvar_inst(linha , saída);
12
13
   fim enquanto
```

O filtro do software MAQ é binário, de forma que classifica os SNPs como 0 ou 1, ou seja, falso ou verdadeiro positivo. Sendo assim a função característica é uma função degrau, (Figura 5.8a). Contudo a rede implementada e treinada obteve melhores resultados com a função sigmóide, que classifica os SNPs no intervalo entre [0 , 1], sendo esse padrão aproveitado para a implementação de uma importante característica do filtro, definida aqui como restrição. A Figura 5.8b mostra a função com as respectivas restrições.



(a) Função degrau. (b) Função sigmóide e suas restrições.

Figura 5.8: funções de saída.

Foram definidas três restrições, BAIXA, MÉDIA e ALTA. A restrição BAIXA permite que qualquer instância seja classificada como SNP caso a saída da rede seja um valor maior que zero. A restrição MÉDIA se comporta como a função degrau, ou seja, todo candidato a SNP classificado com valor acima de $0,5$ é considerado verdadeiro. A restrição ALTA, somente classifica o candidato a SNP como verdadeiro caso possua o valor 1 como saída.

5.2 Considerações

Construir um filtro de SNPs utilizando redes neurais apresentou-se como o desafio a ser investigado nesse trabalho. Nesta tarefa, foram definidos três modelos para geração das bases de dados utilizadas na construção das redes neurais para filtragem de SNPs. Cada modelo apresenta características específicas na tentativa, principalmente, de minorar a obtenção de falsos positivos. Todos os modelos foram treinados de forma similar, com avaliação das propiedades do melhor treinamento de cada modelo. Os resultados preliminares indicaram que a construção de bases de dados com regras mais rígidas, em relação a filtros padrões, podem ser mais efetivas no processo de generalização quando se utiliza

uma ferramenta de classificação supervisionada, mais específicamente redes neurais. O Capítulo 6, analisa cada uma das redes, treinadas com os diferentes conjuntos de dados, comparando-as com o filtro do MAQ. Tais experimentos visam avaliar se as redes neurais podem ser adptadas para a utilização como filtros de forma eficiente e robusta.

6 EXPERIMENTOS COMPUTACIONAIS

Os resultados obtidos com os experimentos computacionais executados, serão apresentados a seguir. Serão descritos os procedimentos de obtenção das informações necessárias para a apuração dos resultados, bem como uma descrição das técnicas utilizadas para comparar os diferentes modelos computacionais desenvolvidos.

Liu et al. (2012), em seu artigo, define três métricas para a medição de acurácia de SNPs em DNA genômico de nova geração, a saber, a taxa dbSNP, a razão Ti/Tv e a genotipagem ou *Array* de SNPs. A taxa dbSNP consiste em verificar o número de alinhamentos positivos entre os SNPs encontrados no genoma estudado, e os existentes no banco de dados de SNPs do NCBI. A razão Ti/Tv é a medida comparativa entre transcrição (Ti) e a transversão (Tv), sendo que, o ideal é que esse número fique próximo a 2. A genotipagem é um processo executado por máquinas específicas, e utiliza material biológico para a obtenção dos marcadores genéticos. Como esse trabalho utiliza sequências em formato FASTA, não é possível a montagem dos *chips* de genotipagem. A razão Ti/Tv é calculada com nucleotídeos reais, porém, nas sequências remontadas existem bases no padrão IUB/IUPAC, o que dificulta o cálculo dessa razão. Por esses motivos, somente a taxa dbSNP será utilizada.

Quando tais testes foram realizados, constavam no NCBI 13.704.221 SNPs submetidos e 3.003 válidos para os animais da raça *Bos Taurus*, último acesso em 02/2013. Foram extraídas as sequências de nucleotídeos dos SNPs presentes na base de dados dbSNP, utilizadas para construir localmente a base de dados necessária para aplicação do software BLAST. A base BLAST local contém todos os SNPs do NCBI para animais da raça *Bos Taurus*.

Para a construção do banco de dados que o BLAST utiliza, foram montados *reads* com 120pb de tamanho, onde as variações polimórficas presentes nos SNPs geram cada uma um novo *read* com o SNP posicionado na base 60. Para a análise dos resultados, só foram aceitos os alinhamentos sem *gap* nem *mismatches*, com 100% de similaridade, sempre no sentido $5' \rightarrow 3'$ e com tamanho de 120pb.

No entanto, para comparar diferentes filtros, é comum o cálculo da acurácia. Esses cálculos são relevantes, mas pouco informativos sobre a capacidade do filtro. Isto ocorre porque, para uma dada amostra de SNPs, é encontrada uma certa quantidade de alinhamentos válidos utilizando o software BLAST. Essa amostra, depois de filtrada sofre uma redução de tamanho, bem como uma diminuição no número de alinhamentos encontrados. Ou seja, se o objetivo é encontrar o filtro que mantenha o maior número de alinhamentos possíveis, então a amostra não filtrada é a melhor.

Para solucionar esse problema, foi utilizada uma medida estatística definida por Bland e Altman (2000) conhecida como *odds ratio* (OR) ou razão de chances. A OR indica em quantas vezes o filtro aumentou a chance de se encontrar um alinhamento dentro da amostra de SNPs. A aplicação do filtro em uma determinada amostra de SNPs gera uma redução em sua quantidade e, como consequência, uma redução no número de alinhamentos. Entretanto, o objetivo do filtro é eliminar os *mismatches*, mantendo somente os melhores candidatos a SNPs. Ou seja, ao se comparar duas amostras de SNPs, uma filtrada e outra não filtrada, e o número de alinhamentos encontrados, é possível verificar se a aplicação do filtro aumenta a chance de se encontrar um alinhamento válido na amostra de SNPs apresentada.

A medida OR indica a mudança de probabilidade de se encontrar um alinhamento válido dentro de uma amostra de SNPs filtrados, em comparação com outra amostra não filtrada. Por exemplo, seja:

$$\begin{aligned} A_t &= \text{Amostra de SNPs sem filtro, ou total.} \\ A_{ta} &= \text{Número de alinhamentos encontrados em } A_t \text{ na base dbSNP} \\ A_f &= \text{Amostra filtrada (SNPfilter ou NeuroSNP).} \\ A_{fa} &= \text{Número de alinhamentos (SNPs) encontrados em } A_f \text{ na base dbSNP} \end{aligned} \tag{6.1}$$

com a probabilidade de se encontrar um alinhamentos na amostra A_t é dada pela razão $r(A_t) = \frac{A_{ta}}{A_t - A_{ta}}$, e para a amostra A_f é dada por: $r(A_f) = \frac{A_{fa}}{A_f - A_{fa}}$. O cálculo da OR é a razão entre as duas probabilidades, definida pela Equação 6.2, dada a seguir:

$$OR = \frac{r(A_t)}{r(A_f)} \tag{6.2}$$

Bland e Altman (2000), em seu trabalho, orientam a calcular o intervalo de confiança (IC), pois o mesmo indica a precisão da OR encontrada. O valor adotado para a análise é

o intervalo de confiança de 95%, que por padrão é obtido através do conjunto de equações 6.3.

$$\begin{aligned} IC^{+} &= \exp\left(\ln(OR) + 1.96 \cdot \sqrt{\tfrac{1}{A_t - A_{ta}} + \tfrac{1}{A_{ta}} + \tfrac{1}{A_f - A_{fa}} + \tfrac{1}{A_{fa}}}\right) \\ IC^{-} &= \exp\left(\ln(OR) + 1.96 \cdot \sqrt{\tfrac{1}{A_t - A_{ta}} + \tfrac{1}{A_{ta}} + \tfrac{1}{A_f - A_{fa}} + \tfrac{1}{A_{fa}}}\right) \\ IC &= [IC^{-}, IC^{+}] \end{aligned} \tag{6.3}$$

A seguir, são apresentados os resultados obtidos com o filtro do MAQ, em comparação com o NeuroSNP, objetivando avaliar se o modelo computacional é capaz de minorar a taxa de falsos positivos encontradas. Para facilitar o entendimento do problema, serão utilizados os termos amostra e informação, com o primeiro fazendo referência ao total de SNPs encontrados ou filtrados e o segundo ao número de alinhamentos encontrados. O cálculo do total de falsos positivos é feito utilizando o valor da OR, se o número de alinhamento decair em quantidade menor que o total da amostra após o filtro, menor o número de falsos positivos. Desta forma, foi feito o cálculo da OR para cada filtro, visando determinar a chance de se encontrar um alinhamento válido. Assim, se houver uma redução no tamanho da amostra de SNPs, porém, se a informação for mantida, a expectativa é que o número de falsos positivos tenha sido reduzido.

A seguir serão descritos os principais resultados referente ao genoma bovino, com a seção seguinte apresentando o desempenho, através da medida OR, do filtro desenvolvido quando aplicado em outro genoma completo.

6.1 Genoma do *Bos Taurus*

Para a obtenção das medidas comparativas via BLAST, primeiro é necessário a montagem da base de dados que será utilizada pelo mesmo. Essa base foi montada localmente, seguindo critérios para a construção dos arquivos utilizados. Desta forma, foram obtidos os arquivos FASTA[1] com os dados de SNPs do NCBI.

A primeira etapa consiste em extrair as sequências de nucleotídeos e montar um arquivo no formato FASTA, para a geração da base de dados BLAST. Para isso foi desenvolvido um algoritmo em PHP que percorre os arquivos RS disponíveis no FTP do NCBI, e gera um novo arquivo FASTA, contendo as variações polimórficas. A Figura 6.1 mostra como é disponibilizada a informação no arquivo RS, e a Figura 6.2 apresenta a saída do

[1]Disponível em : `ftp://ftp.ncbi.nih.gov/snp/organisms/cow_9913/rs_fasta/`

programa em PHP.

```
>gnl|dbSNP|rs17870308 rs=17870308|pos=153|len=408|taxid=9913|mol="genomic"|class=1|alleles="A/G"|build=124|suspect=?
TCTGAGGGGT TGGTGCTTCT CCTGGGCCCT TGCTAAAGAT CAGTGTCTTT TCCAGGTTAC AATCCAGCAG AAGTTGGAGT
TGCAGGCCAG GGCTACCTCT GGAAAGCTGC AGATATGAAC ATGGAGGAAG AGGAGGAACT ACGGCAGAAT CT
R
TGGGGTGAGC TAGGCCTCAT TCCTGCCATG CACCATCTAA ATCAGATTAT TTTGAGGGAG TTTGAACACT TCAGAATCAG
ACTGGCAGTG GTttagtccc taagttgtgt ctgactgtta tggcccaatg gactgtagcc tgccaggctt ctctatccat
aggattctgc aggcaaaaat actggagtga gttgccattt tcttctccag gggatcAGAA TCAGACTGCT TGTTCATTAA
TTTGTTCATG CAGAC
```

Figura 6.1: Formato do arquivo RS disponível no NCBI.

```
>gnl|dbSNP|rs17870308 rs=17870308|Allelo=A
TACCTCTGGAAAGCTGCAGATATGAACATGGAGGAAGAGGAGGAACTACGGCAGAATCTATGGGGTGAGCTAGGCCTCATTCCTGCCAT
>gnl|dbSNP|rs17870308 rs=17870308|Allelo=G
TACCTCTGGAAAGCTGCAGATATGAACATGGAGGAAGAGGAGGAACTACGGCAGAATCTGTGGGGTGAGCTAGGCCTCATTCCTGCCAT
```

Figura 6.2: Arquivo FASTA gerado pelo código em PHP ou PERL.

Como é possível avaliar, cada SNP presente no arquivo FASTA do NCBI possui um valor *allele*, em destaque na Figura 6.1, contendo, pelo menos, duas variações. Por isso, no novo arquivo FASTA existem pelo menos duas sequências distintas, uma para cada alelo. O arquivo gerado serve para a montagem da base BLAST usada no cálculo da taxa dbSNP.

A entrada do programa **blastn** disponibilizado pela biblioteca BLAST, recebe como parâmetro outro arquivo FASTA, com as sequências que se desejam alinhar. Para a geração desse arquivo, foi desenvolvido um novo código em PERL, utilizando a biblioteca BioPerl, que percorre o arquivo de SNP e o arquivo FASTA, que contém o genoma completo montado pelo software MAQ, fazendo a leitura do cromossomo na posição onde o SNP foi encontrado. Busca 59 posições anteriores a posição onde o SNPs foi encontrado e 60 posições à frente, gerando assim uma sequência de nucleotídeos com 120pb com o SNP na posição 60. Somente foram considerados alinhamentos válidos, aqueles com tamanho de 120pb, sem *gap* e nem *mismatches*, ou seja, que tiveram 100% de similaridade com 100% de sobreposição.

Cada modelo foi executada 10 vezes, onde, cada execução obteve diferentes valores para o erro de treinamento. De forma a avaliar a interferência do ruido no treinamento, foram selecionadas duas redes neurais por modelo, a com o maior e a com o menor nível de erro no treinamento. Entretanto, com a nomenclatura de maior e menor entende-se a existência de uma ordem, contudo, não necessariamente, um erro maior gera uma rede menos eficaz. A definição da melhor rede passa pela avaliação dos resultados obtidos pela mesma. Desta forma para facilitar a análise dos resultados, as redes selecionadas

receberam os seguintes nomes: NeuroSNP1.A para rede com menor erro do primeiro modelo e NeuroSNP1.B para a com o maior erro. NeuroSNP2.A para rede com menor erro do segundo modelo e NeuroSNP2.B para o maior erro. E NeuroSNP3.A para a rede com menor erro do terceiro modelo e NeuroSNP3.B para o maior erro.

6.1.1 Resultados Obtidos pelo Primeiro Modelo

O primeiro modelo utilizou quatro funções de ativação diferentes com três constantes de momento. Porém, somente a melhor estrutura do primeiro modelo foi escolhida. A estrutura escolhida utiliza a função de ativação Elliot e a constante de momento igual a $0,5$. Os resultados obtidos pelo primeiro modelo estão dispostos na Tabela 6.1. As redes selecionadas nesse modelo, a NeuroSNP1.A e NeuroSNP1.B, obtiveram os seguintes erros de treinamento: 0,003560 e 0,011363.

Ao analisarmos a tabela 6.1 é possível ver que o valor 5,3746 obtido através do cálculo da OR para o SNPfilter, só foi ultrapassado pelo valor de 6,0669 do NeuroSNP1.B com restrição ALTA. Contudo, as restrições MÉDIA e BAIXA, apresentam valores de OR menores (4,8398 e 3,4714), indicando que o NeuroSNP1.B foi pouco eficiente. Pois o aumento do número de SNPs filtrados ou amostra, não gerou um aumento igual no número de alinhamentos válidos. O mesmo comportamento é observado para o NeuroSNP1.A, com um valor de OR de 5,0302 com restrição ALTA, e com valores menores para as restrições MÉDIA e BAIXA (4,3026 e 3,5126).

Tabela 6.1: Comparativo entre o SNPfilter e o Primeiro Modelo.

	SNPs	Alinhamentos	OR	IC
MAQ	6.599.143	2.162.709	-	-
SNPfilter	2.174.341 (32,95%)	1.573.706 (72,77%)	5,3746	5,3565 - 5,3929
NeuroSNP1.A				
ALTA	1.878.258 (28,46%)	1.334.174 (61,69%)	5,0302	5,0124 - 5,0480
MÉDIA	2.243.455 (34,00%)	1.519.172 (70,24%)	4,3026	4,2887 - 4,3166
BAIXA	2.725.354 (41,30%)	1.720.551 (79,56%)	3,5126	3,5022 - 3,5229
NeuroSNP1.B				
ALTA	1.557.915 (23,61%)	1.164.256 (53,83%)	6,0669	6,0429 - 6,0910
MÉDIA	2.001.787 (30,33%)	1.405.903 (65,01%)	4,8398	4,8232 - 4,8565
BAIXA	2.809.366 (42,57%)	1.765.863 (81,65%)	3,4714	3,4613 - 3,4815

Outro fator a ser observado é o IC, que em todos os casos se manteve baixo, sendo quase nulo se o valor for arredondado para somente uma casa decimal. O IC pequeno

indica que a OR calculada é precisa, sendo extremamente significante.

A Figura 6.3, mostra a distribuição dos valores reais atribuídos pela rede no intervalo (0,1] aos candidatos, sendo que os que receberam valor nulo foram omitidos do gráfico para uma melhor visualização por representarem cerca de 70% dos candidatos. Dos restantes, observa-se que a maioria obteve como saída o valor unitário, provavelmente devido a estratégia utilizada na montagem das base de treinamento.

Desta forma, o uso desse primeiro modelo só se mostrou superior ao SNPfilter quando utilizada a restrição ALTA pois, o valor de OR obtido pelo NeuroSNP1.A é próximo ao do SNPfilter, e do NeuroSNP1.B é superior. Logo, a rede treinada utilizando a classe de cada instância como sendo a saída do SNPfilter não se mostra promissora para a classificação de *mismatches*. Porém, a expectativa é que as redes neurais sejam capazes de executar essa tarefa de forma satisfatória, sendo necessário somente o treinamento com bases mais promissoras.

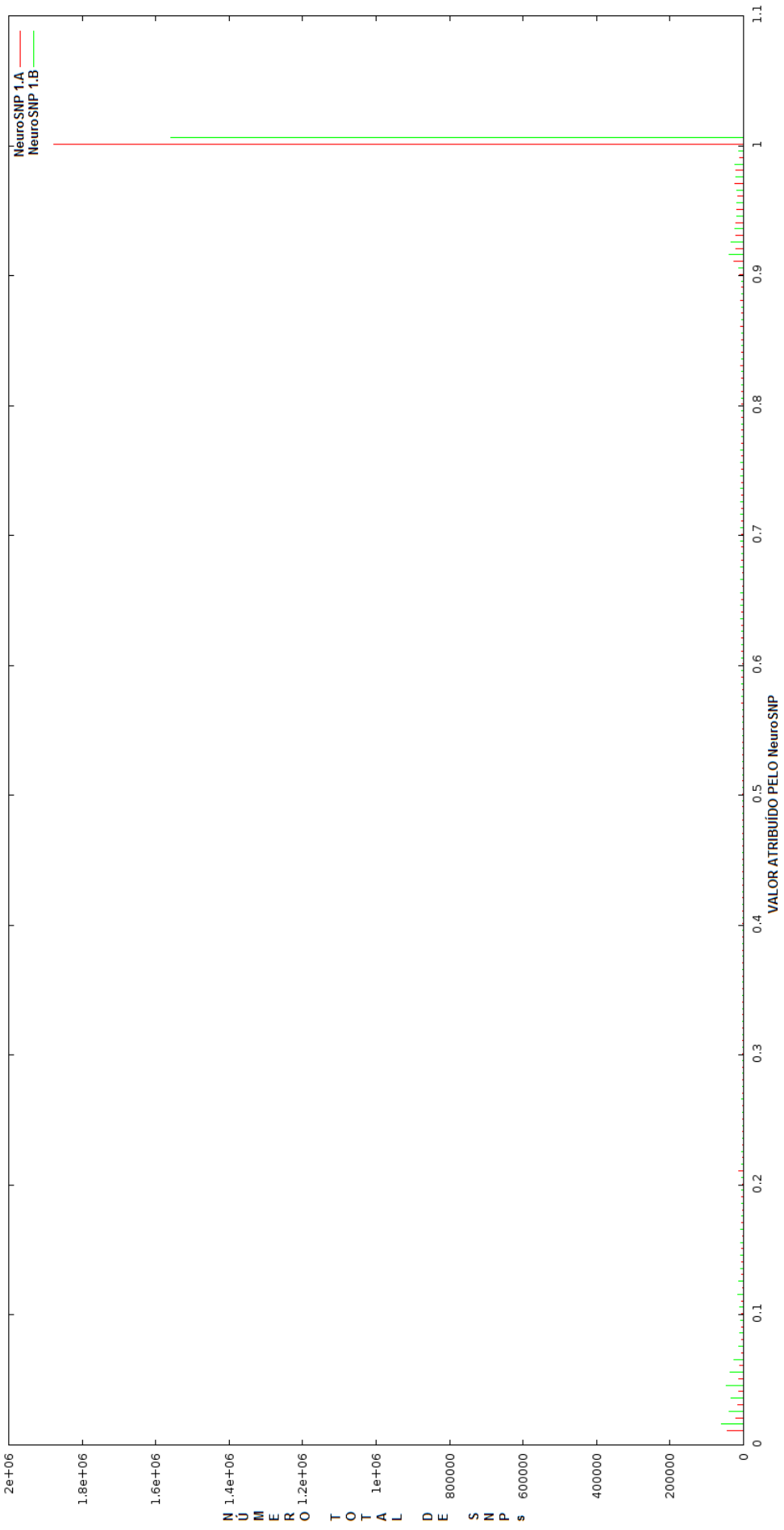


Figura 6.3: Distribuição da classificação calculada pela rede.

6.1.2 Resultados Obtidos pelo Segundo Modelo

O segundo modelo, foi treinado com um conjunto de dados montado a partir das regras apresentada na seção 5.1.2.1. A tabela 6.2 mostra o resultado comparativo entre o segundo modelo e o SNPfilter. As redes selecionadas obtiveram os seguintes erros: 0,000646 para o NeuroSNP2.A e 0,000791 para o NeuroSNP2.B. Como é possível observar os dois valores são muito próximos. Analisando a tabela 6.2, é possível verificar que ambas as redes conseguem classificar os *mismatches* de forma muito eficiente, obtendo valores de ORs superiores ao do SNPfilter nas três restrições, e principalmente com valores de ORs próximos entre as restrições, demonstrando que o segundo modelo é estável.

Como se pode observar, a correta classificação dos *mismatches* é uma tarefa difícil, pois duas redes com erros próximos possuem resultados finais bem diferentes. O espaço de busca percorrido pela rede na otimização do erro pode possuir muitos mínimos locais, possivelmente próximos ao mínimo global, gerando assim redes com erros baixos, porém, com variações na etapa de classificação. Outra hipótese é que a proximidade entre os candidatos seja grande, fazendo com que duas redes, com erros muito próximos, venham a ter comportamentos diferentes para um mesmo conjunto de dados. Para esse análise basta observar o tamanho da amostra, que possui uma variação moderada na NeuroSNP2.A, e uma variação maior na NeuroSNP2.B. Assim como no primeiro modelo, as redes do segundo modelo possuem ICs pequenos, demonstrando que a OR calculada é precisa.

Tabela 6.2: Comparativo entre o SNPfilter e o Segundo Modelo.

	SNPs	Alinhamentos	OR	IC
MAQ	6.599.143	2.162.709	-	-
SNPfilter	2.174.341 (32,95%)	1.573.706 (72,77%)	5,3746	5,3565 - 5,3929
NeuroSNP2.A				
ALTA	209.875 (3,18%)	164.320 (7,60%)	7,3993	7,3220 - 7,4774
MÉDIA	398.005 (6,03%)	308.975 (14,29%)	7,1191	7,0649 - 7,1736
BAIXA	658.551 (9,98%)	507.243 (23,45%)	6,8769	6,8359 - 6,9180
NeuroSNP2.B				
ALTA	81.797 (1,24%)	61.480 (2,84%)	6,2074	6,1092 - 6,3072
MÉDIA	408.590 (6,19%)	314.781 (14,55%)	6,8834	6,8321 - 6,9350
BAIXA	1.143.865 (17,33%)	853.942 (39,48%)	6,0420	6,0148 - 6,0694

A Figura 6.4, mostra a distribuição do valor atribuído pela rede no intervalo [0,1]. É possível observar, uma diferença significativa entre as duas redes neurais. A NeuroSNP2.B possui uma distribuição mais uniforme, apesar da diferença entre o tamanho da amostra

obtida com a aplicação das restrições ALTA e BAIXA. A NeuroSNP2.A possui a grande parte dos SNPs classificados como positivos com o valor de saída da rede igual a 1, fazendo com que a maioria da amostra não filtrada seja selecionada mesmo com o uso da restrição ALTA. Pode-se observar que quase 1/3 da amostra total, conforme pode ser visto na tabela 6.2, apresenta esta característica. Na NeuroSNP2.B a amostra filtrada com restrição ALTA corresponde a menos de 10% da amostra filtrada com BAIXA restrição. O uso das restrições aplicadas ao NeuroSNP, pode ser de interesse para outras frentes de pesquisas, que utilizem SNPs como fonte de informação. Ou seja, uma amostra de SNPs menor, porém, mais informativa, pode ser mais significante em etapas da pesquisa, onde se necessite de resultados rápidos.

O uso do segundo modelo se mostrou mais promissor do que o primeiro. Ressalta-se, inclusive, que o segundo modelo apresenta resultados superiores aos obtidos com o uso do SNPfilter. Porém, como o erro observado nas duas redes é próximo, a determinação de qual rede é a melhor para a etapa de classificação não é tarefa trivial. Caso a necessidade seja uma amostra mais controlada e com maior número de informação a NeuroSNP2.A é melhor, e no caso de uma amostra menor que mantenha o grau de informação presente a NeuroSNP2.B se mostra mais promissora.

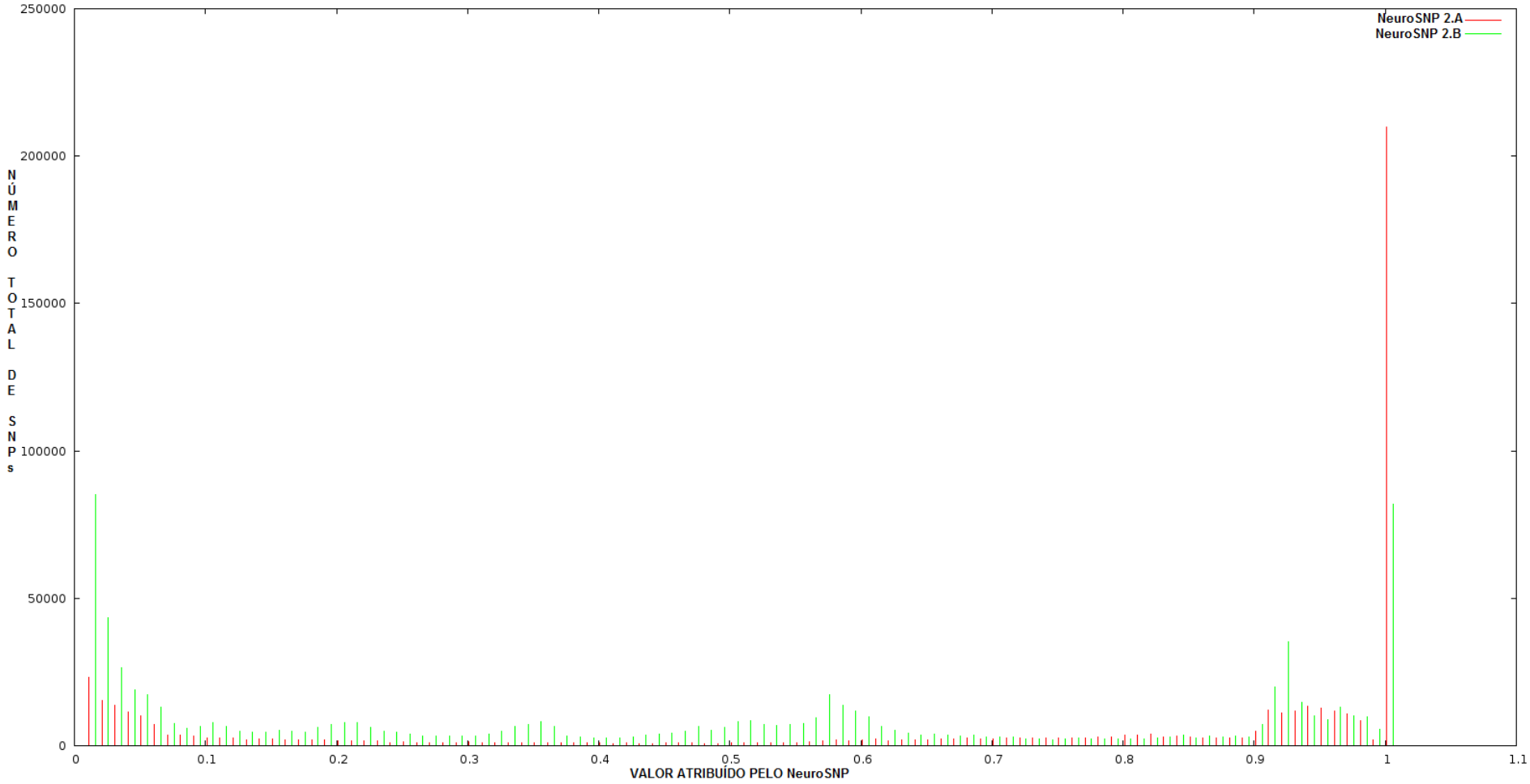


Figura 6.4: Distribuição da classificação das redes do Segundo Modelo.

6.1.3 Resultados Obtidos pelo Terceiro Modelo

O terceiro modelo, foi treinado com uma base montada nas regras apresentadas na seção 5.1.3.1. A Tabela 6.3 mostra o resultado comparativo entre as redes neurais em relação ao SNPfilter. Após o treinamento, foi obtido os seguintes valores de erros: 0,002003 para NeuroSNP3.A e 0,002167 para a NeuroSNP3.B. Como é possível observar, novamente os dois valores são muito próximos.

Ao analisar a tabela 6.3 nota-se um comportamento muito próximo entre esse modelo e o primeiro, no que tange a capacidade de classificação dos SNPs. Ambos os modelos são pouco informativos, como indicado pelo valor da OR que oscila com o aumento no tamanho da amostra. É importante observar, que apesar de possuir uma OR maior que do SNPfilter na restrição ALTA da NeuroSNP3.A (6, 9419), e na restrição MÉDIA da NeuroSNP3.B (5, 8454), os valores das ORs não possuem um padrão, mostrando que a rede não esta sendo eficiente no processo de classificação. Entretanto, possui um comportamento semelhante ao do segundo modelo em relação a variação no tamanho da amostra. Os ICs obtidos nas redes do terceiro modelo, assim como nos modelos anteriores, foram pequenos, mostrando que as ORs calculadas são extremamente precisas.

Tabela 6.3: Comparativo entre o SNPfilter e as redes do Terceiro Modelo.

	SNPs	Alinhamentos	OR	IC
MAQ	6.599.143	2.162.709	-	-
SNPfilter	2.174.341 (32,95%)	1.573.706 (72,77%)	5,3746	5,3565 - 5,3929
NeuroSNP3.A				
ALTA	545.227 (8,26%)	420.862 (19,46%)	6,9419	6,8967 - 6,9874
MÉDIA	952.373 (14,43%)	660.245 (30,53%)	4,6363	4,6148 - 4,6579
BAIXA	2.740.161 (41,52%)	1.715.832 (79,34%)	3,4361	3,4261 - 3,4463
NeuroSNP3.B				
ALTA	75.242 (1,14%)	46.722 (2,16%)	3,3605	3,3111 - 3,4107
MÉDIA	680.492 (10,31%)	503.720 (23,29%)	5,8454	5,8124 - 5,8785
BAIXA	1.938.537 (29,38%)	1.362.622 (63,01%)	4,8535	4,8366 - 4,8704

A Figura 6.5, mostra a distribuição do valor atribuído pela rede no intervalo [0,1] e o número de *mismatches* classificados para o dado valor, o gráfico possui discretização de 0, 01. É possível observar que as duas redes do terceiro modelo, possuem um comportamento próximo, pois a distribuição observada na NeuroSNP3.A, também pode ser visualizada na NeuroSNP3.B, em seções diferentes do gráfico, mais com um comportamento qualitativo muito próximo.

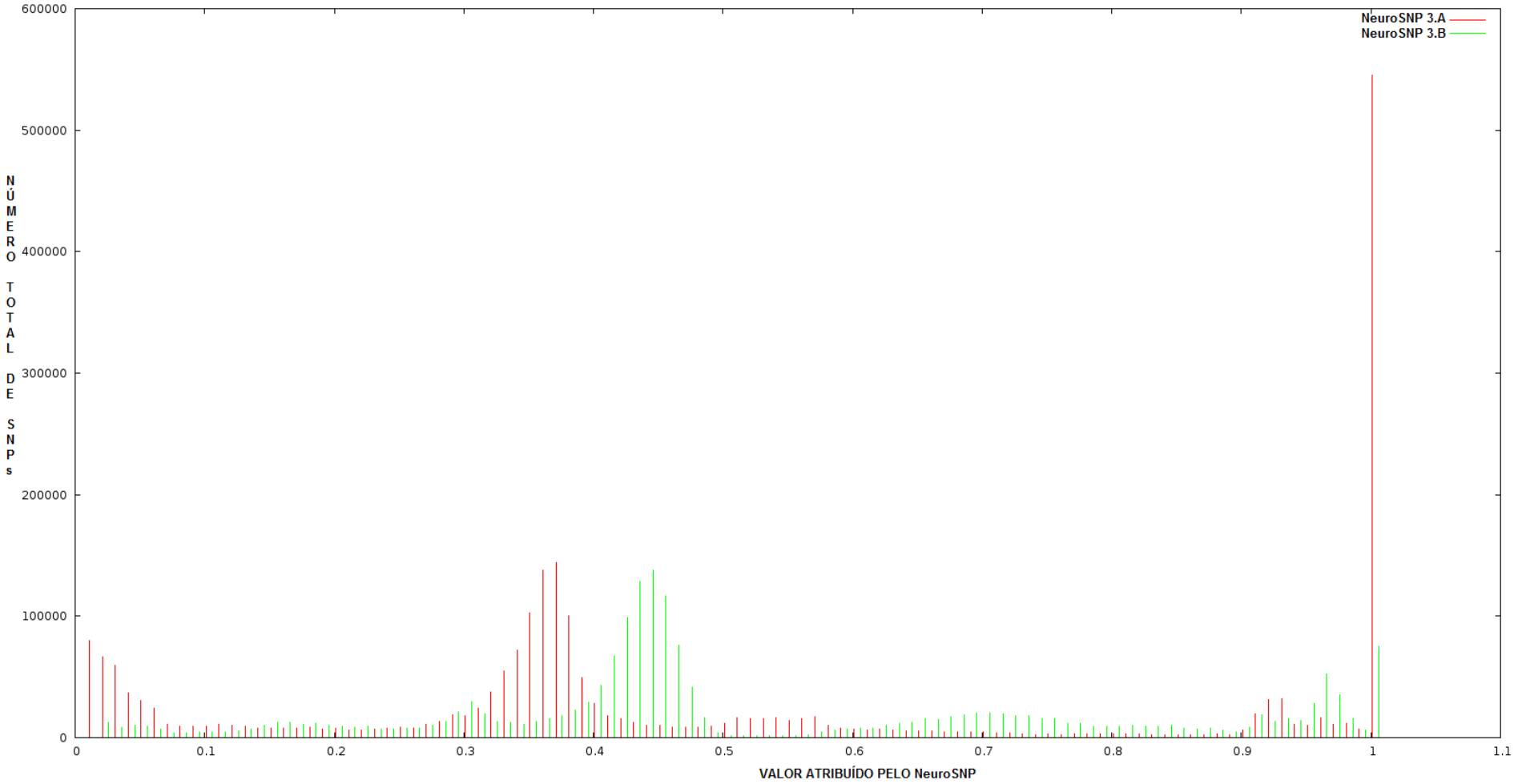


Figura 6.5: Distribuição da classificação calculada pelas redes do Terceiro Modelo.

De qualquer forma, a classificação supervisionada mostrou ser uma ferramenta viável na complexa tarefa de detecção de SNPs. Expectativas em relação a universalização de seu uso em genomas diferentes ou seja, para os quais a rede neural não foi especificamente treinada, serão avaliadas em experimentos seguintes.

6.2 Genoma da *Arabidopsis Thaliana*

Os modelos de redes neurais implementados utilizaram sempre o genoma bovino como fonte dos dados. As bases geradas para o treino e teste das redes, foram originadas do arquivo de SNPs descobertos no genoma remontado do *Bos Taurus*, como explicado anteriormente.

Nesta seção, busca-se verificar se estes modelos, treinados e testados para o genoma bovino manterão seu comportamento, quando apresentada a novos genomas. Para responder a essa pergunta, os modelos foram testados em dois germoplasmas diferentes da planta da espécie *Arabidopsis Thaliana*. O objetivo é mostrar que os modelos podem ser utilizados por outros genomas com resultados similares aos encontrados no genoma bovino. Os germoplasmas montados são identificados como BUR-0 e TSU-1, remontados observando o trabalho de Ossowski et al. (2008).

Foram baixados os arquivos FASTA[2] com as sequências de nucleotídeos contendo os SNPs da *Arabidopsis Thaliana*, para a montagem da base de dados BLAST. Quando estes experimentos foram realizados [3], constavam no NCBI 6.798 SNPs submetidos.

Em seguida, o mesmo script PHP utilizado para extrair as sequências dos arquivos de SNPs bovinos foi utilizado para extrair as sequências dos arquivos da planta. Porém, 1/3 das sequências possuía tamanho inferior a 120pb por isso, a base foi montada com 30pb, com o SNP localizado na posição 15.

A montagem dos arquivos FASTA utilizados no comando **blastn** seguiu a mesma linha, sequências com 30pb e com o SNP na posição 15. Para montagem desse arquivo foi utilizado o mesmo script em PERL do genoma bovino.

A seguir, são apresentados os resultados de cada germoplasmas, em comparação com a base de dados BLAST, montada a partir das sequências de nucleotídeos presentes nos arquivos de SNPs do NCBI. Da mesma forma como foi contabilizado para o genoma

[2]Disponível em: `ftp://ftp.ncbi.nih.gov/snp/organisms/arabidopsis_3702/rs_fasta/`
[3]os dados foram baixados do NCBI em 02/2013

bovino, ou seja, sendo aceitos somente alinhamentos com 30pb e sem *gap* nem *mismatches*, ou seja, que tiveram 100% de similaridade com 100% de sobreposição.

6.2.1 Germoplasma BUR-0

O primeiro germoplasmas a ser analisado é o BUR-0. Depois de remontado foram executadas as etapas de descoberta e filtragem de SNPs, seguindo os mesmos passos executados para o genoma bovino. Foram encontrados 1.135.193 SNPs na etapa de descoberta, restando 544.881 após a aplicação do filtro.

6.2.1.1 Resultados Obtidos pelo Primeiro Modelo

A tabela 6.4 mostra os resultados obtidos pelo primeiro modelo, onde é possível observar que nenhuma das duas redes, conseguiu superar o próprio SNPfilter. O comportamento do primeiro modelo se manteve similar ao observado no genoma bovino, ou seja, o modelo não sofreu alteração quando apresentado a um novo conjunto de dados. Os ICs obtidos pelas respectivas ORs são maiores do que o valor observado no genoma bovino, porém, o intervalo ainda é pequeno, o que demonstra que as ORs se mantiveram precisas.

Tabela 6.4: Comparativo entre o SNPfilter e as redes do Primeiro Modelo.

	SNPs	Alinhamentos	OR	IC
MAQ	1.135.193	921	-	-
SNPfilter	544.881 (48,00%)	832 (90,34%)	1,8834	1,7148 - 2,0686
NeuroSNP1.A				
ALTA	284.015 (27,68%)	455 (51,82%)	1,8733	1,6726 - 2,0980
MÉDIA	429.476 (41,86%)	658 (74,94%)	1,7914	1,6191 - 1,9820
BAIXA	582.256 (56,76%)	780 (88,84%)	1,5660	1,4220 - 1,7247
NeuroSNP1.B				
ALTA	135.773 (13,23%)	165 (18,79%)	1,4205	1,2027 - 1,6777
MÉDIA	459.968 (44,84%)	716 (81,55%)	1,8201	1,6490 - 2,0091
BAIXA	648.088 (63,17%)	765 (87,13%)	1,3797	1,2522 - 1,5202

A variação no valor da ORs é menor no segundo genoma, porém, ainda é inconstante como no genoma bovino. Apesar de o primeiro modelo não ser o melhor entre os três, a manutenção do comportamento demonstra que o mesmo é robusto, mesmo não sendo o mais eficaz.

O aumento do IC é explicado pela diferença de tamanho entre a quantidade de SNPs e o total de alinhamentos encontrados. No genoma bovino o total de SNPs era de 2 a

3 vezes maior que o número de alinhamentos, enquanto que no germoplasma da BUR-0 esse valor é de 600 a 850 vezes maior.

6.2.1.2 Resultados Obtidos pelo Segundo Modelo

A tabela 6.5, mostra o resultado obtido pelas redes do segundo modelo. Assim como observado no genoma bovino, esse modelo manteve um comportamento estável, obtendo valores de OR superiores ao do SNPfilter, com exceção NeuroSNP2.B com restrição BAIXA, que obteve um valor de OR um pouco abaixo. O comportamento do modelo ficou próximo ao obtido no genoma bovino, mostrando que o mesmo pode ser utilizado como filtro de SNPs em outros genomas, com a mesma eficiência obtida no processo de desenvolvimento da ferramenta.

Tabela 6.5: Comparativo entre o SNPfilter e as redes do Segundo Modelo

	SNPs	Alinhamentos	OR	IC
MAQ	1.135.193	921	-	-
SNPfilter	544.881 (48,00%)	832 (90,34%)	1,8834	1,7148 - 2,0686
NeuroSNP2.A				
ALTA	295.959 (26,07%)	576 (62,54%)	2,4016	2,1639 - 2,6653
MÉDIA	416.194 (36,66%)	767 (83,28%)	2,2738	2,0659 - 2,5026
BAIXA	454.620 (40,05%)	785 (85,23%)	2,1302	1,9367 - 2,3432
NeuroSNP2.B				
ALTA	142.681 (12,57%)	265 (28,77%)	2,2916	1,9988 - 2,6274
MÉDIA	302.030 (26,61%)	545 (59,17%)	2,2263	2,0024 - 2,4753
BAIXA	476.529 (41,98%)	681 (73,94%)	1,7625	1,5962 - 1,9462

Assim como observado no genoma bovino, o resultado da aplicação do filtro NeuroSNP2.B possui uma variação amostral maior que a aplicação do filtro NeuroSNP2.A, porém, com uma OR menor. A variação no tamanho da amostra pode ser uma característica interessante do filtro baseado em redes neurais. Assim como observado no primeiro modelo, os ICs possuem uma variação maior que a do genoma bovino, novamente podendo ser explicada pela diferença entre o tamanho da amostra de SNPs e o número de alinhamentos, que nesse modelo é de 500 a 650 vezes maior.

O segundo modelo se mostra novamente muito eficiente, mesmo quando apresentado a dados oriundos de um novo genoma. As redes do segundo modelo se mostram tanto informativa quanto restritiva, pois, variações na restrição geraram amostras com tamanhos diferenciados, mas, com valores de ORs próximos. Essas características demonstram que o segundo modelo é robusto e eficaz.

6.2.1.3 Resultados Obtidos pelo Terceiro Modelo

Apresentam-se, agora, os resultados obtidos pelas redes do terceiro modelo. Assim como observado nos modelos anteriores, as redes do terceiro modelo mantêm um comportamento similar ao encontrado no genoma bovino. A tabela 6.6 apresenta os resultados obtidos com a aplicação desse modelo. O NeuroSNP3.A, com restrição ALTA, obteve uma OR de $2,7$, que para esse germoplasma foi o maior valor, porém, na restrição BAIXA o valor foi de $1,4$, mostrando que o modelo não possui um comportamento estável. Outro ponto a ser observado, é que o NeuroSNP3.B com restrição ALTA obteve uma OR de $0,2$ ou seja, a chance de se encontrar um alinhamento válido na amostra é menor do que se a mesma não tivesse sido filtrada.

Tabela 6.6: Comparativo entre o SNPfilter e as redes do Terceiro Modelo.

	SNPs	Alinhamentos	OR	IC
MAQ	1.135.193	921	-	-
SNPfilter	544.881 (48,00%)	832 (90,34%)	1,8834	1,7148 - 2,0686
NeuroSNP3.A				
ALTA	112.399 (9,90%)	247 (26,82%)	2,7124	2,3567 - 3,1218
MÉDIA	347.281 (30,59%)	661 (71,77%)	2,3486	2,1251 - 2,5955
BAIXA	723.895 (63,77%)	879 (95,44%)	1,4973	1,3650 - 1,6423
NeuroSNP3.B				
ALTA	11.237 (0,99%)	2 (0,22%)	0,2192	0,0547 - 0,8781
MÉDIA	147.931 (13,03%)	232 (25,19%)	1,9345	1,6749 - 2,2343
BAIXA	594.448 (52,37%)	832 (90,34%)	1,7261	1,5716 - 1,8959

O terceiro modelo manteve o mesmo comportamento observado no genoma bovino, ou seja, apresenta uma variação amostral na aplicação do filtro NeuroSNP3.B, contudo, o modelo é instável possuindo variações no valor da OR. Assim como ocorreu com os modelos anteriores, os ICs são maiores que os do genoma bovino, e novamente a diferença entre o tamanho da amostra de SNPs e o número total de alinhamentos é ALTA, chegando nesse modelo a ser 5619 vezes maior.

6.2.1.4 Considerações

O segundo modelo obteve os melhores padrões de filtragem entre os três modelos estudados, se mostrando a melhor alternativa de filtro de SNPs. Mesmo quando apresentado a um novo genoma, os modelos mantiveram o comportamento observado com sua aplicação na sua construção, mostrando que é possível a universalização do seu uso.

6.2.2 Germoplasma TSU-1

O segundo germoplasma a ser analisado é o TSU-1, que assim como a variação anterior, seguiu os passos padrões executados para o genoma bovino e o germoplasma BUR-0. Foram encontrados 1.025.908 SNPs na etapa de descoberta, restando 460.140 após a aplicação do filtro.

6.2.2.1 Resultados Obtidos pelo Primeiro Modelo

A Tabela 6.7 apresenta o resultado obtido com as redes do primeiro modelo, e assim como ocorreu nos genomas anteriores o modelo obteve ORs abaixo do próprio SNPfilter, contudo, o comportamento geral se manteve similar.

Tabela 6.7: Comparativo entre o SNPfilter e as redes do Primeiro Modelo.

	SNPs	Alinhamentos	OR	IC
MAQ	1.025.908	878	-	-
SNPfilter	460.140 (44,85%)	750 (85,42%)	1,9060	1,7289 - 2,1012
NeuroSNP1.A				
ALTA	284.015 (27,68%)	455 (51,82%)	1,8733	1,6726 - 2,0980
MÉDIA	429.476 (41,86%)	658 (74,94%)	1,7914	1,6191 - 1,9820
BAIXA	582.256 (56,76%)	780 (88,84%)	1,5660	1,4220 - 1,7247
NeuroSNP1.B				
ALTA	135.773 (13,23%)	165 (18,79%)	1,4205	1,2027 - 1,6777
MÉDIA	459.968 (44,84%)	716 (81,55%)	1,8201	1,6490 - 2,0091
BAIXA	648.088 (63,17%)	765 (87,13%)	1,3797	1,2522 - 1,5202

O comportamento do primeiro modelo se manteve similar nos três genomas analisados, isso demonstra que apesar de o mesmo não ser o mais eficaz, ele é robusto. Os ICs calculados possuem valores maiores que os do genoma bovino, porém, assim como ocorreu com o germoplasma BUR-0, a diferença entre o tamanho da amostra de SNPs e o número de alinhamentos é ALTA, sendo nesse modelo de 600 a 830 vezes maior.

6.2.2.2 Resultados Obtidos pelo Segundo Modelo

O próximo modelo analisado tem seus resultados apresentados na Tabela 6.8. É possível observar que assim como nos genomas anteriores, esse modelo obteve resultados superiores em ambas as redes, com exceção da NeuroSNP2.B com restrição ALTA. O modelo manteve o padrão de comportamento, bem como a variação amostral entre a NeuroSNP2.A e a NeuroSNP2.B, mostrando-se mais restritivo é informativo que o SNPfilter.

Tabela 6.8: Comparativo entre o SNPfilter e as redes do Segundo Modelo.

	SNPs	Alinhamentos	OR	IC
MAQ	1.025.908	878	-	-
SNPfilter	460.140 (44,85%)	750 (85,42%)	1,9060	1,7289 - 2,1012
NeuroSNP2.A				
ALTA	267.469 (26,07%)	548 (62,41%)	2,3968	2,1541 - 2,6670
MÉDIA	364.580 (35,54%)	692 (78,82%)	2,2201	2,0095 - 2,4529
BAIXA	402.649 (39,25%)	703 (80,07%)	2,0419	1,8489 - 2,2550
NeuroSNP2.B				
ALTA	118.335 (11,53%)	233 (26,54%)	2,3032	1,9932 - 2,6615
MÉDIA	255.114 (24,87%)	473 (53,87%)	2,1686	1,9390 - 2,4253
BAIXA	401.684 (39,15%)	612 (69,70%)	1,7814	1,6066 - 1,9753

Os valores dos ICs obtidos pelas redes do segundo modelo, são maiores que os do genoma bovino, mas mantêm um comportamento similar ao encontrado no germoplasma BUR-0. Da mesma forma a diferença entre o tamanho da amostra de SNPs e o número de alinhamentos é alta, sendo nesse modelo na ordem de 500 a 650 vezes maior.

6.2.2.3 Resultados Obtidos pelo Terceiro Modelo

O terceiro modelo tem seus resultados apresentados na Tabela 6.9. Assim como ocorreu com os genomas anteriores, o modelo mantém um comportamento intermediário entre primeiro e o segundo. Tanto o maior quanto o menor valor das ORs de todas as redes estudadas, foram obtidos por esse modelo, sendo o maior (2,4215) obtido pela NeuroSNP3.A com restrição ALTA, e o menor (0,5173) pela NeuroSNP3.B com restrição ALTA. Isto indica que as redes do modelo não são estáveis, sendo que a NeuroSNP3.B se mostra muito restritiva e pouco informativa, e a NeuroSNP3.A pouco restritiva e muito informativa.

Tabela 6.9: Comparativo entre o SNPfilter e as redes do Terceiro Modelo.

	SNPs	Alinhamentos	OR	IC
MAQ	1.025.908	878	-	-
SNPfilter	460.140 (44,85%)	750 (85,42%)	1,9060	1,7289 - 2,1012
NeuroSNP3.A				
ALTA	79.714 (7,77%)	165 (18,79%)	2,4215	2,0502 - 2,8601
MÉDIA	268.004 (26,12%)	507 (57,74%)	2,2127	1,9834 - 2,4686
BAIXA	622.944 (60,72%)	809 (92,14%)	1,5181	1,3797 - 1,6704
NeuroSNP3.B				
ALTA	6.773 (0,66%)	3 (0,34%)	0,5173	0,1665 - 1,6076
MÉDIA	106.661 (10,40%)	165 (18,79%)	1,8088	1,5315 - 2,1363
BAIXA	514.642 (50,16%)	760 (86,56%)	1,7266	1,5667 - 1,9028

Assim como ocorreu com os modelos anteriores os valores dos ICs são maiores que os do genoma bovino, sendo que a diferença entre o tamanho da amostra de SNPs e o número de alinhamentos é alta, onde, nesse modelo, chega a ser 2.250 vezes maior.

6.2.2.4 Considerações

Assim como nos genomas anteriores, o segundo modelo foi o que obteve os melhores resultados na filtragem. A manutenção do comportamento dos três modelos indica que as regras utilizadas para a geração das bases de treinamento dos modelos, refletem diretamente nas características das filtragens obtidas, porém, somente o segundo modelo mostrou um nível adequado de eficiência na classificação de SNPs. A diferença entre o tamanho da amostra de SNPs e o número total de alinhamentos encontrados, aumentou o IC de todos os modelos aplicados no germoplasmas da TSU-1, entretanto, apesar do aumento a precisão das ORs ainda é alta (BLAND; ALTMAN, 2000).

6.3 Considerações

Os experimentos computacionais indicaram, claramente, o potencial da ferramenta de aprendizado desenvolvida para a detecção de SNPs. Sua utilização de forma isolada ou em conjunto com filtros tradicionais apresenta-se como uma alternativa para a determinação robusta de SNPs em genomas distintos. A utilização da medida OR mostrou que a aplicação do filtro desenvolvido aumenta a chance de se encontrar um alinhamento positivo dentro da amostra de SNPs, com o indicativo que esse aumento reflita na diminuição dos falsos positivos.

Logicamente, a construção da base de treinamento pode ser aprimorada, principalmente em duas direções: por meio da definição de regras mais específicas, com prioridade para a determinação de falsos positivos; e pela utilização de SNPs comprovados biologicamente na construção da classe de verdadeiros positivos. De qualquer forma, a classificação supervisionada mostrou ser uma ferramenta viável na complexa tarefa de detecção de SNPs.

7 CONCLUSÕES

O aumento na capacidade das plataformas de NGS, que disponibilizam dados de milhões de pares de bases em uma única corrida, gera a necessidade de um constante avanço nos métodos computacionais que são utilizados na manipulação e análise desse grande volume de dados visando, de forma geral, uma maior compreensão biológica das espécies. Entre as análises possíveis, baseadas em material genético, pode-se destacar as pesquisas relativas a SNPs. Porém, para que essas pesquisas possam gerar conhecimento relevante, etapas prévias como a descoberta e a filtragem desses SNPs precisam ser realizadas de forma eficaz. Especificamente, para as plataformas NGS, os *reads* produzidos são curtos e propensos a erros, o que dificulta o processo de montagem, além de aumentarem o número de *mismatches* presentes na amostra que será utilizada na etapa de descoberta de SNPs. Diferenças no sequenciamento como as descritas, indicam a necessidade de adaptação de estratégias computacionais para o trato de sequências obtidas via NGS.

Neste trabalho, foi apresentada e desenvolvida uma estratégia computacional fundamentada em aprendizado de máquina e inteligência computacional, com capacidade de filtrar SNPs a partir de DNA genômico completo (NeuroSNP). No processo construtivo do NeuroSNP, foram utilizados três modelos diferentes, sendo cada um analisado e comparado com o filtro de referência do software MAQ, a saber, SNPfilter. Nos genomas avaliados, o NeuroSNP conseguiu resultados similar ou superior ao filtro do MAQ.

Em relação a cada modelo, foram geradas 10 redes neurais visando avaliar o comportamento do modelo em testes experimentais. O desempenho na filtragem das redes de cada modelo com maior e menor erro de treinamento, respectivamente, foi apresentado. Os resultados indicaram que cada par avaliado, apresentou características bem distintas na filtragem, demonstrando a dificuldade em se classificar os *mismatches*, encontrados em DNA genômico completo. O uso das chamadas faixas de restrição se mostrou uma alternativa viável, pois os modelos conseguiram abstrair do conjunto de parâmetros um valor numérico, entre [0,1], que indica a importância do SNP filtrado. Em geral, os testes realizados indicam que o segundo modelo obteve os melhores resultados, mostrando-se mais restritivo e informativo. Seu treinamento sofreu pouca ou nenhuma variação, pois, como visto, entre as 10 execuções as redes obtidas apresentavam erros de treinamento

muito próximos, na ordem de 10^{-4}.

Um fator a ser observado é que com o aumento da restrição o filtro do MAQ passa a selecionar os candidatos com base somente no PHRED, pois os condicionais presentes no filtro são do tipo **ou**, aceitando o SNP que satisfaça as altas restrições ou que tenha PHRED maior ou igual a 20. Nesse ponto, a rede apresenta uma solução mais eficiente para a classificação de *mismatches*, e como observado ela se mostra tanto restritiva quanto informativa. Com a aplicação da restrição, permite-se ao usuário reduzir a amostra de SNPs a ser estudada mantendo, contudo, a informação presente nela.

Como primeiro trabalho desenvolvido para filtrar SNPs oriundos de DNA genômico completo sequenciado por plataformas de NGS, a rede neural demonstrou potencial para que ferramentas baseadas em técnicas de aprendizado de máquina e inteligência computacional possam ser aplicadas em filtragem de SNPs. Por ter sido utilizado um método de aprendizado supervisionado, o resultado sofre influência do conjunto de dados gerado para a construção da hipótese de classificação. Esta característica foi amplamente explorada no segundo e terceiro modelos. Os resultados indicam que a exploração e o desenvolvimento de novos conjuntos de dados, baseados em novas regras podem incrementar a generalização do modelo. A utilização de outros genomas, com outras características, também podem trazer novos indícios visando aprimorar os filtros.

Outra perspectiva de trabalho futuro está no teste de novas funções e topologias para as redes neurais implementadas. O uso de outras técnicas de aprendizado, incluindo modelos baseado em classificadores de classe única *one class*, podem apresentar resultados complementares a classificação binária utilizada.

Além das redes neurais, outras técnicas de inteligência computacional podem ser aplicadas para a filtragem de SNPs, entre elas a lógica difusa. No caso da lógica difusa, sistemas híbridos como, por exemplo, sistemas neuro-fuzzy, podem trazer ganhos no processo de filtragem. Neste caso, a saída da rede pode servir de entrada para um sistema de lógica difusa ou o contrário. Em ambas as opções, a rede neural, aqui apresentada, teria seu resultado refinado através da construção de um conjunto de regras difusas.

REFERÊNCIAS

ALBERTS, B. et al. *Biologia molecular da célula*. 5. ed. Porto Alegre, BR: ARTMED, 2010. ISBN 978-85-363-2066-3.

ALHO, C. S. Projeto genoma humano. In: *Genômica*. 1. ed. São Paulo, Rio de Janeiro, Ribeirão Preto, Belo Horizonte, BR: ATHENEU, 2004. p. 71–103. Dinâmica dos genes e medicina genômica.

ALTSCHUL, S. F. et al. Basic local alignment search tool. *Journal of Molecular Biology*, 1990. v. 215, n. 3, p. 403 – 410, 1990. ISSN 0022-2836. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0022283605803602>.

ALTSHULER, D. et al. An snp map of the human genome generated by reduced representation shotgun sequencing. *Nature*, 2000. v. 407, n. 6803, p. 513–516, 2000. Disponível em: <http://dx.doi.org/10.1038/35035083>.

ARBEX, W. A. *Modelos Computacionais para Identificação de Informação Genômica Associada à Resistência ao Carrapato Bovino*. Tese (Doutorado) — UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2009.

ARBIB, M. A. (Ed.). *The Handbook of Brain Theory and Neural Networks*. 2nd. ed. Cambridge, MA, USA: MIT Press, 2002. ISBN 0262011972.

BALDI, P. et al. *Bioinformatics: the machine learning approach*. [S.l.]: The MIT Press, 2001.

BARNES, M. R. *Bioinformatics for geneticists*. [S.l.]: Wiley Online Library, 2007.

BASHEER, I.; HAJMEER, M. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 2000. v. 43, n. 1, p. 3 – 31, 2000. ISSN 0167-7012. Neural Computting in Micrbiology. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167701200002013>.

BLAND, J. M.; ALTMAN, D. G. The odds ratio. *British Medical Journal*, 2000. BMJ Publishing Group, v. 320, n. 7247, p. 1468, 2000. Disponível em: <http://www.ncbi-.nlm.nih.gov/pmc/articles/PMC1127651/>.

BONFIELD, J.; STADEN, R. Experiment files and their application during large-scale sequencing projects. *HARWOOD ACAD PUBL GMBH, C/O STBS LTD, PO BOX 90, READING, BERKS, ENGLAND RG1 8JL*, 1996. v. 6, n. 2, p. 109–117, 1996. ISSN 1042-5179.

BRIDGES, M. et al. Genetic classification of populations using supervised learning. *PLoS ONE*, 2011. Public Library of Science, v. 6, n. 5, p. e14802, 05 2011. Disponível em: <http://dx.doi.org/10.1371%2Fjournal.pone.0014802>.

BRONDANI, R. P. V.; BRONDANI, C. Germoplasma: base para a nova agricultura. *Ciência Hoje*, 2004. v. 35, n. 207, p. 70–73, 2004.

BROOKES, A. J. The essence of snps. *Gene*, 1999. v. 234, n. 2, p. 177 – 186, 1999. ISSN 0378-1119. Disponível em: <http://www.sciencedirect.com/science/article/pii-/S037811199900219X>.

BRUMFIELD, R. T. et al. The utility of single nucleotide polymorphisms in inferences of population history. *Trends in Ecology & Evolution*, 2003. Elsevier, v. 18, n. 5, p. 249–256, 2003.

CHEN, F. et al. The history and advances of reversible terminators used in new generations of sequencing technology. *Genomics, Proteomics & Bioinformatics*, 2013. v. 11, n. 1, p. 34 – 40, 2013. ISSN 1672-0229. Disponível em: <http://www.sciencedirect-.com/science/article/pii/S1672022913000077>.

COCK, P. J. A. et al. The sanger fastq file format for sequences with quality scores, and the solexa/illumina fastq variants. *Nucleic Acids Research*, 2010. v. 38, n. 6, p. 1767–1771, 2010. Disponível em: <http://nar.oxfordjournals.org/content/38/6/1767.abstract>.

CONSORTIUM, I. H. The international hapmap project. *Nature*, 2003. v. 426, n. 6968, p. 789 – 96, 2003. Disponível em: <http://dx.doi.org/10.1038/nature02168>.

CONSORTIUM, T. I. H. A haplotype map of the human genome. *Nature*, 2005. v. 437, n. 7063, p. 1299–1320, 2005. Disponível em: <http://www.nature.com/nature/journal-/v437/n7063/suppinfo/nature04226_S1.html>.

CRICK, F. On the protein synthesis. *Symposia of the Society for Experimental Biology*, 1958. Cambridge University Press, v. 12, p. 138–163, 1958.

CURTIS, D. Comparison of artificial neural network analysis with other multimarker methods for detecting genetic association. *BMC Genetics*, 2007. v. 8, n. 1, p. 49, 2007. ISSN 1471-2156. Disponível em: <http://www.biomedcentral.com/1471-2156/8/49>.

DEMAINE, E. D.; DEMAINE, M. L. Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. *Graph. Comb.*, 2007. Springer-Verlag, Springer-Verlag, Tokyo, Japan, v. 23, n. 1, p. 195–208, feb 2007. ISSN 0911-0119. Disponível em: <http://dx.doi.org/10.1007/s00373-007-0713-4>.

DIAS NETO, E. Projeto genoma humano. In: *Genômica*. 1. ed. São Paulo, Rio de Janeiro, Ribeirão Preto, Belo Horizonte, BR: ATHENEU, 2004. p. xli–lviii. Introdução.

ECK, S. et al. Whole genome sequencing of a single bos taurus animal for single nucleotide polymorphism discovery. *Genome Biology*, 2009. v. 10, n. 8, p. R82, 2009. ISSN 1465-6906. Disponível em: <http://genomebiology.com/2009/10/8/R82>.

ELLIOTT, D. L. A better activation function for artificial neural networks. *Institute for Systems Research Technical Reports*, 1993. 1993. Disponível em: <http://hdl.handle.net-/1903/5355>.

EWING, B.; GREEN, P. Base-calling of automated sequencer traces usingphred. ii. error?probabilities. *Genome Research*, 1998. v. 8, n. 3, p. 186–194, 1998. Disponível em: <http://genome.cshlp.org/content/8/3/186.abstract>.

EWING, B. et al. Base-calling of automated sequencer traces usingphred. i. accuracy?assessment. *Genome Research*, 1998. v. 8, n. 3, p. 175–185, 1998. Disponível em: <http://genome.cshlp.org/content/8/3/175.abstract>.

FEDURCO, M. et al. Bta, a novel reagent for dna attachment on glass and efficient generation of solid-phase amplified dna colonies. *Nucleic Acids Research*, 2006. v. 34, n. 3, p. e22, 2006. Disponível em: <http://nar.oxfordjournals.org/content/34/3/e22-.abstract>.

GENOMICS, C. for P. H. *SNP Filter GA/GP*. 2011. Disponível em: <http://cphg-.virginia.edu/mackey/projects/sequencing-pipelines/snp-filter-gagp/>.

GLENN, T. C. Field guide to next-generation dna sequencers. *Molecular Ecology Resources*, 2011. Blackwell Publishing Ltd, v. 11, n. 5, p. 759–769, 2011. ISSN 1755-0998. Disponível em: <http://dx.doi.org/10.1111/j.1755-0998% -.2011.03024.x>.

GORDON, D.; ABAJIAN, C.; GREEN, P. Consed: A graphical tool for sequence?finishing. *Genome Research*, 1998. v. 8, n. 3, p. 195–202, 1998. Disponível em: <http://genome.cshlp.org/content/8/3/195.abstract>.

GREEN, P. *PHRAP documentation*. [S.l.], 1994. Acessado em:10/01/2013. Disponível em: <http://www.phrap.org/phredphrap/phrap.html>.

GUIMARÃES, P.; COSTA, M. Snps: sutis diferenças de um código. *Biotecnol. Cienc. Desenvolv*, 2002. v. 26, p. 24–27, 2002.

GUPTA, P. K. Single-molecule DNA sequencing technologies for future genomics research. *Trends in Biotechnology*, 2008. v. 26, n. 11, p. 602 – 611, 2008. ISSN 0167-7799. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167779908002047>.

HAYKIN, S. *Redes Neurais: princípios e prática*. 2. ed. [S.l.]: Porto Alegre: Bookman, 2001.

HEBB, D. O. *The Organization of Behavior: A Neuropsychological Theory*. New edition. New York: Wiley, 1949. Hardcover. ISBN 0805843000. Disponível em: <http://www.worldcat.org/isbn/0805843000>.

HEIDEMA, A. G. et al. The challenge for genetic epidemiologists: how to analyze large numbers of snps in relation to complex diseases. *BMC Genetics*, 2006. v. 7, n. 1, p. 23, 2006. ISSN 1471-2156. Disponível em: <http://www.biomedcentral.com/1471-2156/7-/23>.

HGSC, B. C. o. M. *UCSC, Genome Bioinformatics*. 2007. Disponível em: <http:/-/hgdownload.soe.ucsc.edu/goldenPath% -/bosTau4/bigZips/>.

HUANG, X.; MADAN, A. Cap3: A dna sequence assembly program. *Genome Research*, 1999. v. 9, n. 9, p. 868–877, 1999. Disponível em: <http://genome.cshlp.org/content/9-/9/868.abstract>.

IDURY, R.; WATERMAN, M. A new algorithm for dna sequence assembly. *Journal of Computational Biology*, 1995. v. 2, n. 2, p. 291–306, 1995.

INITIATIVE, T. A. G. Analysis of the genome sequence of the flowering plant arabidopsis thaliana. *Nature*, 2000. v. 408, n. 6814, p. 796–815, 2000. ISSN 0028-0836. Disponível em: <http://dx.doi.org/10.1038/35048692>.

KOBOLDT, D. C. et al. Varscan: variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics*, 2009. v. 25, n. 17, p. 2283–2285, 2009. Disponível em: <http://bioinformatics.oxfordjournals.org/content/25/17/2283-.abstract>.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995. (IJCAI'95), p. 1137–1143. ISBN 1-55860-363-8. Disponível em: <http://dl.acm.org/citation.cfm?id=1643031% -.1643047>.

KRISHNAN, V. G.; WESTHEAD, D. R. A comparative study of machine-learning methods to predict the effects of single nucleotide polymorphisms on protein function. *Bioinformatics*, 2003. Oxford Univ Press, v. 19, n. 17, p. 2199–2209, 2003.

LANDER, E. S.; WATERMAN, M. S. Genomic mapping by fingerprinting random clones: A mathematical analysis. *Genomics*, 1988. v. 2, n. 3, p. 231 – 239, 1988. ISSN 0888-7543. Disponível em: <http://www.sciencedirect.com/science/article/pii/0888754388900079>.

LEE, H.; TANG, H. Next-generation sequencing technologies and fragment assembly algorithms. *Methods in Molecular Biology*, 2012. v. 855, March 2012. Disponível em: <http://www.springerprotocols.com/Abstract/doi/10.1007/978-1-61779-582-4_5>.

LEHNINGER, D.; COX, M. M. *Princípios de Bioquímica de Lehninger*. 5. ed. Porto Alegre, BR: ARTMED, 2011. ISBN 978-85-363-2418-0.

LESK, A. M. *Introdução à Bioinformática*. 2. ed. Porto Alegre, BR: ARTMED, 2008. ISBN 978-85-363-1104-3.

LI, H. *Manual Reference Pages - MAQ (1)*. [S.l.], 2008. Acessado em:15/06/2012. Disponível em: <http://maq.sourceforge.net/maq-manpage.shtml>.

LI, H. *Maq: Mapping and Assembly with Qualities*. 2008. Disponível em: <http://maq-.sourceforge.net/>.

LI, H.; RUAN, J.; DURBIN, R. Mapping short dna sequencing reads and calling variants using mapping quality scores. *Genome Research*, 2008. v. 18, n. 11, p. 1851–1858, 2008. Disponível em: <http://genome.cshlp.org/content/18/11/1851.abstract>.

LI, R. et al. Soap: short oligonucleotide alignment program. *Bioinformatics*, 2008. v. 24, n. 5, p. 713–714, 2008. Disponível em: <http://bioinformatics.oxfordjournals.org-/content/24/5/713.abstract>.

LIN, Y. et al. Comparative studies of de novo assembly tools for next-generation sequencing technologies. *Bioinformatics*, 2011. v. 27, n. 15, p. 2031–2037, 2011. Disponível em: <http://bioinformatics.oxfordjournals.org/content/27/15/2031.abstract>.

LIPPMANN, R. An introduction to computing with neural nets. *ASSP Magazine, IEEE*, 1987. v. 4, n. 2, p. 4 –22, apr 1987. ISSN 0740-7467.

LIU, Q. et al. Steps to ensure accuracy in genotype and snp calling from illumina sequencing data. *BMC Genomics*, 2012. v. 13, n. Suppl 8, p. S8, 2012. ISSN 1471-2164. Disponível em: <http://www.biomedcentral.com/1471-2164/13% -/S8/S8>.

LONG, N. et al. Comparison of classification methods for detecting associations between snps and chick mortality. *Genetics Selection Evolution*, 2009. v. 41, n. 1, p. 18, 2009. ISSN 1297-9686. Disponível em: <http://www.gsejournal.org/content/41/1/18>.

MALHIS, N.; JONES, S. J. M. High quality snp calling using illumina data at shallow coverage. *Bioinformatics*, 2010. v. 26, n. 8, p. 1029–1035, 2010. Disponível em: <http://bioinformatics.oxfordjournals.org/content/26/8/1029.abstract>.

MARDIS, E. R. The impact of next-generation sequencing technology on genetics. *Trends in Genetics*, 2008. v. 24, n. 3, p. 133 – 141, 2008. ISSN 0168-9525. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0168952508000231>.

MARGULIES, M. et al. Genome sequencing in open microfabricated high density picoliter reactors. *Nature*, 2005. v. 437, n. 7057, June 2005. DOI: 10.1038/nature03959,acessado em: 17/11/2012 14:25. Disponível em: <http://www.nature.com/nature/journal/v437-/n7057/full/nature03959.html>.

MARTH, G. T. et al. A general approach to single-nucleotide polymorphism discovery. *Nature Genetics*, 1999. v. 23, n. 4, p. 452–456, 1999.

MAXAM, A. M.; GILBERT, W. A new method for sequencing dna. *Proceedings of the National Academy of Sciences of the United States of America*, 1977. v. 74, n. 2, p. 560–4, 1977. Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/265521>.

MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 1943. Kluwer Academic Publishers, v. 5, p. 115–133, 1943. ISSN 0007-4985. Disponível em: <http://dx.doi.org/10.1007-/BF02478259>.

Kevin Mckernan, Alan Blanchard, Lev Kotler e Gina Costa. *REAGENTS, METHODS, AND LIBRARIES FOR BEAD-BASED SEQUENCING*. 2011. 20110077169A1.

MILLER, J. R.; KOREN, S.; SUTTON, G. Assembly algorithms for next-generation sequencing data. *Genomics*, 2010. v. 95, n. 6, p. 315 – 327, 2010. ISSN 0888-7543. Disponível em: <http://www.sciencedirect.com/science/article/pii-/S0888754310000492>.

MOROZOVA, O.; MARRA, M. A. Applications of next-generation sequencing technologies in functional genomics. *Genomics*, 2008. v. 92, n. 5, p. 255–264, 2008. ISSN 0888-7543. Disponível em: <http://www.sciencedirect.com/science/article/pii-/S0888754308001651>.

MYERS, E. Toward simplifying and accurately formulating fragment assembly. *Journal of Computational Biology*, 1995. v. 2, p. 275–290, 1995. Disponível em: <http://online.liebertpub.com/doi/abs/10% -.1089/cmb.1995.2.275>.

NCBI, B. *Query Input and database selection*. [S.l.], 2007. Acessado em:10/01/2013. Disponível em: <http://www.ncbi.nlm.nih.gov/blast/blastcgihelp.shtml>.

NEURODIMENSION. *Neuro Solutions*. 2013. Disponível em: <http://www-.neurosolutions.com/index.html>.

NICKERSON, D. A.; TOBE, V. O.; TAYLOR, S. L. Polyphred: automating the detection and genotyping of single nucleotide substitutions using fluorescence-based resequencing. *Nucleic Acids Research*, 1997. v. 25, n. 14, p. 2745–2751, 1997. Disponível em: <http://nar.oxfordjournals.org/content/25/14/2745.abstract>.

NISSEN, S. Neural networks made simple. *Software 2.0 magazine*, 2005. n. 2, p. 14–19, 2005. Disponível em: <http://fann.sf.net/fann_en.pdf>.

OSSOWSKI, S. et al. Sequencing of natural strains of arabidopsis thaliana with short reads. *Genome Research*, 2008. v. 18, n. 12, p. 2024–2033, 2008. Disponível em: <http://genome.cshlp.org/content/18/12/2024.abstract>.

PACHECO, P. *An Introduction to Parallel Programming*. 1st. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN 9780123742605.

PASSOS-BUENO, M. R. d. S.; MOREIRA, E. d. S. Ferramentas bássicas da genética molecular humana. In: *Genômica*. 1. ed. São Paulo, Rio de Janeiro, Ribeirão Preto, Belo Horizonte, BR: ATHENEU, 2004. cap. 3, p. 43–70.

PEARSON, W. R.; LIPMAN, D. J. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 1988. v. 85, n. 8, p. 2444–2448, 1988. Disponível em: <http://www.pnas.org/content/85/8/2444% -.abstract>.

PENA, S. D. et al. Retrato molecular do brasil. *Ciência hoje*, 2000. v. 27, n. 159, p. 16–25, 2000.

PEVZNER, P. A.; TANG, H.; WATERMAN, M. S. An eulerian path approach to dna fragment assembly. *Proceedings of the National Academy of Sciences*, 2001. v. 98, n. 17, p. 9748–9753, 2001.

PONGPANICH, M.; SULLIVAN, P. F.; TZENG, J.-Y. A quality control algorithm for filtering snps in genome-wide association studies. *Bioinformatics*, 2010. v. 26, n. 14, p. 1731–1737, 2010. Disponível em: <http://bioinformatics.oxfordjournals.org/content/26-/14/1731.abstract>.

REN, L. et al. Typing snp based on the near-infrared spectroscopy and artificial neural network. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 2009. v. 73, n. 1, p. 106 – 111, 2009. ISSN 1386-1425. Disponível em: <http://www-.sciencedirect.com/science/article/pii/S1386142509000560>.

RIEDMILLER, M.; BRAUN, H. A direct adaptive method for faster backpropagation learning: the rprop algorithm. *IEEE International Conference on Neural Networks*, 1993. Ieee, v. 1, n. 3, p. 586–591, 1993. Disponível em: <http://ieeexplore.ieee.org/lpdocs-/epic03/wrapper.htm?arnumber=298623>.

ROCHE. *System features for GS FLX Titatnium series*. [S.l.], 2008. Acessado em:15/12/2013. Disponível em: <http://www.454.com/products/gs-flx-system% -/index.asp>.

RONAGHI, M. Pyrosequencing sheds light on dna sequencing. *Cold Spring Harbor Laboratory Press*, 2001. v. 11, n. 3-11, 2001. DOI: 10.1101/gr.150601 , acessado em: 19/11/2012 10:12. Disponível em: <http://genome.cshlp.org/content/11/1/3.long>.

RONAGHI, M.; UHLÉN, M.; NYRÉN, P. A sequencing method based on real-time pyrophosphate. *Science*, 1998. v. 281, n. 5375, July 1998. DOI: 10.1126/science.281.5375.363,acessado em: 15/11/2012 13:52. Disponível em: <http://www.sciencemag.org/content/281/5375/363.full>.

ROSENBLATT, F. The percepton: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 1958. MIT Press, Cambridge, MA, USA, v. 65, n. 6, p. 386 – 408, 1958.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, 1986. v. 323, n. Oct, p. 533–536, 1986.

SANGER, F.; NICKLEN, S.; COULSON, A. R. Dna sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America*, 1977. v. 74, n. 12, p. 5463–5467, 1977.

SEQUENCING, T. B. G. et al. The genome sequence of taurine cattle: A window to ruminant biology and evolution. *Science*, 2009. v. 324, n. 5926, p. 522–528, 2009. Disponível em: <http://www.sciencemag.org/content/324/5926/522.abstract>.

SERVICE, R. F. The race for the $1000 genome. *Science*, 2006. v. 311, n. 5767, p. 1544–1546, 2006. Disponível em: <http://www.sciencemag.org/content/311/5767/1544.short>.

SETUBAL, J. C. Bioinformática. In: *Genômica*. 1. ed. São Paulo, Rio de Janeiro, Ribeirão Preto, Belo Horizonte, BR: ATHENEU, 2004. cap. 6, p. 105–118.

SHENDURE, J.; JI, H. Next-generation dna sequencing. *Nature Biotechnology*, 2008. v. 26, n. 10, p. 1134–1145, 2008. Disponível em: <http://www.nature.com/nbt/journal/v26/n10/full/nbt1486.html>.

SMITH, T.; WATERMAN, M. Identification of common molecular subsequences. *Journal of Molecular Biology*, 1981. v. 147, n. 1, p. 195 – 197, 1981. ISSN 0022-2836. Disponível em: <http://www.sciencedirect.com/science/article/pii/0022283681900875>.

STANSFIELD, W. D.; COLOMÉ, J. S.; CANO, R. J. *Biologia molecular e Celular*. Portugal: McGraw-Hill, 1998. ISBN 972-8298-97-8.

SUAREZ-KURTZ, G. F. a genética dos medicamentos. *Ciência Hoje*, 2004. v. 35, p. 208–27, 2004.

TOMITA, Y. et al. Artificial neural network approach for selection of susceptible single nucleotide polymorphisms and construction of prediction model on childhood allergic asthma. *BMC Bioinformatics*, 2004. v. 5, n. 1, p. 120, 2004. ISSN 1471-2105. Disponível em: <http://www.biomedcentral.com/1471-2105/5/120>.

TURCATTI, G. et al. A new class of cleavable fluorescent nucleotides: synthesis and optimization as reversible terminators for dna sequencing by synthesis. *Nucleic Acids Research*, 2008. v. 36, n. 4, p. e25, 2008. Disponível em: <http://nar.oxfordjournals.org/content/36/4/e25.abstract>.

WICKER, T. et al. A whole-genome snapshot of 454 sequences exposes the composition of the barley genome and provides evidence for parallel evolution of genome size in wheat and barley. *The Plant Journal*, 2009. Blackwell Publishing Ltd, v. 59, n. 5, p. 712 –722, 2009. ISSN 1365-313X. Disponível em: <http://dx.doi.org/10.1111/j.1365-313X% -.2009.03911.x>.

YONABA, H.; ANCTIL, F.; FORTIN, V. Comparing sigmoid transfer functions for neural network multistep ahead streamflow forecasting. *Journal of Hydrologic Engineering*, 2010. v. 15, 2010.

ZERBINO, D. R.; BIRNEY, E. Velvet: Algorithms for de novo short read assembly using de bruijn graphs. *Genome Research*, 2008. v. 18, n. 5, p. 821–829, 2008. Disponível em: <http://genome.cshlp.org/content/18/5/821.abstract>.

ZHANG, J. et al. The impact of next-generation sequencing on genomics. *Journal of Genetics and Genomics*, 2011. v. 38, n. 3, p. 95–109, 2011. ISSN 1673-8527. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1673852711000300>.