



Mendel,MD: Desenvolvimento e validação de uma ferramenta online para análise de exomas humanos e investigação de doenças Mendelianas.

Autor:

Raony GCCL Cardenas

Orientador:

Sérgio DJ Pena

*Documento apresentado em cumprimento aos requisitos para obtenção do
título de Doutor em Bioinformática*

no

Instituto de Ciências Biológicas

4 de março de 2016

U F *m* G

Universidade Federal de Minas Gerais

Instituto de Ciências Biológicas

Programa de Pós-Graduação em Bioinformática

**Mendel,MD: Desenvolvimento e validação de uma
ferramenta online para análise de exomas humanos e
investigação de doenças Mendelianas.**

Relatório final do projeto de tese apresentado
ao colegiado do curso de Doutorado em Bi-
oinformática como parte integrante dos re-
quisitos para obtenção do título de Doutor
em Bioinformática.

Aluno: Raony Guimarães Corrêa Do Carmo Lisbôa Cardenas

Orientador: Prof. Dr. Sérgio Danilo Junho Pena

Belo Horizonte

2015

Dedicado as minhas irmãs, Cecília, Duda e Isabel e ao meu irmão Pedro Henrique.

Agradecimentos

Gostaria antes de tudo de agradecer a minha família, em especial a minha querida avó Edna Guimarães Corrêa e a minha mãe Patrícia Do Carmo Lisboa Mouro que sempre fizeram de tudo para me dar a melhor educação possível.

A minha esposa Anna Kowalska Guimarães por todo amor e carinho que recebi e por sempre ter me apoiado muito quando decidi vir fazer meu Doutorado aqui em Belo Horizonte. Kocham Cię!

Ao Prof. Sérgio Danilo Junho Pena pela sua orientação e dedicação, por me ensinar os valores de um verdadeiro cientista e pelas valiosas lições que aprendi ao longo de meu Doutorado sob sua orientação.

Aos membros do Laboratório de Genômica Clínica Raquel Liboredo, Michele Pena, Tiago Magalhães, Lucas Santos, Natália Linhares, Maíra Freire e Fernanda Soardi e aos amigos do Laboratório de Genética e Bioquímica.

Aos pesquisadores Dr. Cascon Alberto, Dr. Christopher Carroll, Dr. Fowzan S. Alkuraya, Dr. Pia Ostergaard e Dr. Yaniv Erlich que contribuíram para este trabalho enviando seus exomas para serem analisados e validados pelo Mendel,MD. Sem essa contribuição esse trabalho não poderia ter sido realizado.

A pesquisadora Dra. Judith Conroy pela colaboração e as valiosas sugestões que foram feitas ao Mendel,MD.

Aos amigos do ICB em especial Thiago Mafra e Rondon Neto pelas valiosas discussões sobre Bioinformática que tivemos ao longo desse tempo.

Aos amigos do Hackerspace Área31 e Python User Group de Minas Gerais (PUG-MG) pela grande amizade e pelos encontros que tivemos nos bares de Belo Horizonte.

Aos meus amigos de infância Martielo Borelli e Miguel Rivero por terem me incentivado a estudar e a continuar aprendendo coisas novas.

Por último, mas não menos importante, gostaria de agradecer ao meu cachorro Einstein que sempre me acompanhou durante todo esse tempo nos passeios diários de reflexão e raciocínio.

Resumo

Com o advento dos métodos de sequenciamento de nova geração, o seqüenciamento de todo o exoma de um paciente tornou-se economicamente viável para realizar o diagnóstico clínico de doenças genéticas, incluindo as raras e complexas.

A estratégia para a identificação de variantes patogénicas é complexa, uma vez que em todos os exomas existem entre 40-50.000 variantes em comparação com o genoma humano de referência. Para simplificar este procedimento, filtros computacionais são aplicados de forma sequencial com o objetivo de eliminar variantes comuns e sinônimas, reduzindo o tamanho da amostra total.

Esse trabalho apresenta o desenvolvimento de uma ferramenta de Bioinformática chamada Mendel,MD, que é eficiente e sofisticada do ponto de vista computacional, e ao mesmo tempo, simples e amigável para ser utilizada por médicos e cientistas para auxiliar no diagnóstico de pacientes com doenças mendelianas.

A Análise de Filtros é um método que combina diferentes anotações, bases de dados e escores de patogenicidade, permitindo com isso reduzir o número de variantes e genes de cada caso clínico de milhares de candidatos para apenas algumas poucas dezenas.

Essa ferramenta foi validada com dados de 15 casos clínicos diferentes recebidos a partir de laboratórios especializados de diferentes países. Foi consistentemente possível identificar uma pequena lista de candidatos de genes causais, que incluiu o diagnóstico correto em todos os casos investigados.

Mendel, MD é uma ferramenta eficiente, segura e confiável na exploração de variantes de exomas de pacientes com Doenças Mendelianas, sofisticados do ponto de vista da bioinformática e ainda assim simples o suficiente para ser usado por médicos e cientistas para analisar rapidamente seus próprios dados genômicos.

Palavras-chaves: bioinformática, exoma humano, genoma humano, doença mendeliana, filtragem de variantes, anotação de variantes, análise de exomas, sequenciamento de nova geração.

Abstract

With the emergence of next-generation methodology, the sequencing of the whole exome of a patient has become economically viable to perform the clinical diagnosis of genetic diseases, including rare and complex.

The strategy for identification of pathogenic variants is complex, since in every exome there are between 40-50000 variants compared to the reference human genome. To simplify this procedure, computer filters are applied sequentially in order to remove common and synonymous variants, reducing the total sample size.

The study presents the development of a bioinformatics tool called Mendel,MD, which is efficient and sophisticated from the computational point of view, and at the same time, simple and friendly to be used by doctors and scientists to assist in the diagnosis of patients with Mendelian Disorders.

The Filter analysis is a method that combines hundreds of different annotations, databases and pathogenicity scores, thereby allowing for reducing the number of gene and variants in each clinical case of thousands of candidates for only a few tens.

This tool has been validated with data from 15 different clinical cases received from specialized laboratories in different countries. It was consistently able to identify a short list of candidates of causal genes, which included the correct diagnosis in all cases investigated. Mendel, MD is an efficient, safe and reliable tool in the exploration of exome variants of patients with Mendelian disorders, sophisticated from the bioinformatics point of view and yet simple enough to be used by physicians and scientists to quickly analyze their own genomic data .

Key-words: exome sequencing, exome analysis, Mendelian disorder, next generation sequencing, variant calling, variant annotation.

Listas de Figuras

Figura 1 – A estrutura de um Gene	4
Figura 2 – Processo de Filtragem de Variantes para identificação de uma doença Mendeliana	7
Figura 3 – Crescimento do número de artigos publicados com a palavra exome no Pubmed	8
Figura 4 – Representação gráfica de uma SNP	10
Figura 5 – Representação gráfica de uma Indel	11
Figura 6 – Medicina Personalizada	15
Figura 7 – Diagrama com a separação entre as camadas do Django	22
Figura 8 – Exemplo de uma leitura em formato FASTQ	24
Figura 9 – Exemplo de um arquivo no formato SAM	26
Figura 10 – Exemplo de um arquivo no formato VCF	27
Figura 11 – GATK Framework de análise de variantes	30
Figura 12 – Diagrama com o pipeline de análise de exomas	41
Figura 13 – Escore <i>Phred</i> de qualidade médio por base das sequências do indivíduo RMS	45
Figura 14 – Distribuição do Escore <i>Phred</i> em relação a todas as sequências do indivíduo RMS	46
Figura 15 – Conteúdo de sequência por base para as sequências do indivíduo RMS	47
Figura 16 – Distribuição do conteúdo de GC nas leituras	49
Figura 17 – Níveis de duplicação	50
Figura 18 – Perfis de K-mer	51
Figura 19 – Estatísticas do alinhamento antes de utilizarmos o GATK	53
Figura 20 – Qualidade média por base do alinhamento antes de utilizarmos o GATK	53
Figura 21 – Estatísticas do alinhamento depois de utilizarmos o GATK	54

Figura 22 – Qualidade média por base do alinhamento depois de utilizarmos o GATK	54
Figura 23 – Escore de Qualidade Empírico vs Reportado	57
Figura 24 – Número de Bases vs Escore de Qualidade	58
Figura 25 – Qualidade (Empírico - Reportado) vs Dinucleotídeos.....	59
Figura 26 – Dashboard - Interface para visualização dos indivíduos no sistema ...	67
Figura 27 – Interface para submissão dos indivíduos no sistema	68
Figura 28 – Interface do Celery	69
Figura 29 – Processo de anotação de variantes	70
Figura 30 – Programa desenvolvido para conversão entre arquivos VCF e CSV ..	72
Figura 31 – Framework de anotação de variantes.....	74
Figura 32 – Interface para visualizar métricas sobre os dados inseridos.....	75
Figura 33 – Interface para Busca de Genes	77
Figura 34 – Interface com grupos de genes adicionados	78
Figura 35 – Interface para Busca de Doenças	79
Figura 36 – Filtragem de Variantes - 1 ^a Etapa	81
Figura 37 – Filtragem de Variantes - 2 ^a Etapa	82
Figura 38 – Filtragem de Variantes - 3 ^a Etapa	83
Figura 39 – Family Analysis.....	86
Figura 40 – Resultado do Family Analysis	87
Figura 41 – Interface para comparação de indivíduos	89
Figura 42 – Processo de filtragem de variantes do caso LGC 2	94
Figura 43 – Visualização das variantes do caso LGC 2.....	95

Listas de Tabelas

Tabela 1 – Informações sobre a ancestralidade dos indivíduos que foram sequenciados pelo projeto <i>1000 Genomes</i>	14
Tabela 2 – Informações sobre o dbSNP 137.....	32
Tabela 3 – Estatísticas sobre os genes de acordo com o HUGO <i>Gene Nomenclature Committee</i> (HGNC)	34
Tabela 4 – Tabela com informações sobre o número de pares de bases de cada um dos cromossomos do genome build b37.	36
Tabela 5 – Tabela com as classes de impacto do programa SnpEff	39
Tabela 6 – Métricas sobre as Sequências <i>forward</i> e <i>reverse</i> do indivíduo RMS ..	44
Tabela 7 – Informações sobre o alinhamento após a remoção de leituras duplicadas	55
Tabela 8 – Resultado do processo de calling de variantes realizado usando o método UnifiedGenotyper para o indivíduo RMS.....	61
Tabela 9 – Informação sobre o número de registros armazenados em cada uma tabelas do sistema.	65
Tabela 10 – Processo de Filtragem utilizado para o paciente RMS	92
Tabela 11 – Descrição dos 12 exomas recebidos.....	96
Tabela 12 – Casos Recebidos para Validação	99

Listas de abreviaturas e siglas

AB3C	<i>Associação Brasileira de Bioinformática e Biologia Computacional</i>
BFAST	<i>Blat-like Fast Accurate Search</i>
BWA	<i>Burrows-Wheeler Aligner</i>
CADD	<i>Combined Annotation Dependent Depletion</i>
CENAPAD	Centro Nacional de Processamento de Alto Desempenho
CNV	<i>Copy Number variations</i>
CSV	<i>Comma Separated Values</i>
ESP	<i>Exome Sequencing Project</i>
FSS	Síndrome de Freeman-Sheldon (<i>Freeman-Sheldon Syndrome</i>)
GATK	<i>Genome Analisys ToolKit</i>
GL	<i>Genome Library</i>
HGMD	<i>Human Genome Mutation Database</i>
HGNC	<i>HUGO Gene Nomenclature Committee</i>
HUGO	<i>Human Genome Organisation</i>
INDEL	Inserção/Deleção (<i>Insertion/Deletion</i>)
LGC	Laboratório de Genômica Clínica
MVC	<i>Model View Controller</i>
NCBI	<i>National Center for Biotechnology Information</i>
NMD	<i>Nonsense-mediated decay</i>

OMIM	<i>Online Mendelian Inheritance in Man</i>
ORM	<i>Object-relational mapping</i>
PB	Par de Base
SAM	<i>Sequence Alignment/Map</i>
SBIS	<i>Sociedade Brasileira de Informática em Saúde</i>
SGBD	Sistema Gerenciador de Banco de Dados
SIFT	<i>Sort Intolerant From Tolerant</i>
SNP	Polimorfismo de Nucleotídeo Único (<i>Single Nucleotide Polymorphism</i>)
SQL	<i>Structured Query Language</i>
SV	Variante Estrutural (<i>Structural Variant</i>)
TF	<i>Transcription factor</i>
TFBS	<i>Transcription Factor Binding Sites</i>
UTR	<i>Untranslated Region</i>
VCF	<i>Variant Call Format</i>
VEP	<i>Variant Effect Predictor</i>

Índice

Resumo	iii
Lista de Figuras	v
Lista de Tabelas	vii
Lista de Abreviaturas e Siglas	viii
1 INTRODUÇÃO E JUSTIFICATIVA	1
1.1 Informática Biomédica.....	1
1.2 Bioinformática	2
1.3 O Exoma	3
1.4 Doenças Mendelianas.....	3
1.5 Análise de Exomas para o diagnóstico de doenças Mendelianas ..	5
1.6 SNPs e Indels.....	9
1.7 Projeto Genoma Humano	12
1.8 Projeto 1000 Genomas	12
1.9 Projeto de Sequenciamento do Exoma.....	13
1.10 Medicina Personalizada / Medicina de Precisão.....	13
2 OBJETIVOS	17
2.1 Objetivo Geral	17
2.2 Objetivos Específicos	17
3 MATERIAIS E MÉTODOS	19
3.1 Hardware	19
3.2 Sistema Operacional	19
3.3 Linguagens e Frameworks Utilizados	20
3.3.1 Bash	20
3.3.2 Python	20
3.3.3 Django	21

3.4	Formatos Utilizados	21
3.4.1	FASTQ	21
3.4.2	Sequence Alignment Map - Formato SAM	23
3.4.3	Variant Call Format - Formato VCF	25
3.5	Programas Utilizados	25
3.5.1	<i>FASTQC</i>	25
3.5.2	<i>SAMTOOLS</i>	28
3.5.3	<i>PICARD TOOLS</i>	28
3.5.4	<i>Burrows-Wheeler Aligner</i>	28
3.5.5	<i>BFAST</i>	28
3.5.6	<i>GATK</i>	29
3.5.7	<i>IGV</i>	29
3.6	Banco de Dados e Programas Utilizadas	31
3.6.1	1000Genomes	31
3.6.2	Exome Variant Server	31
3.6.3	dbSNP	31
3.6.4	<i>GATK Resource Bundle</i>	33
3.6.5	<i>OMIM - Online Mendelian Inheritance in Man</i>	33
3.6.6	HGNC - HUGO Gene Nomenclature Committee	33
3.6.7	HGMD	33
3.6.8	Genoma Humano de Referência b37	35
3.6.9	Escore PHRED	35
3.6.10	Escore SIFT	37
3.6.11	Escore Polyphen-2	37
3.6.12	Escore CADD	37
3.6.13	<i>SnpEff</i>	37
3.6.14	<i>ANNOVAR</i>	38
3.6.15	dbNSFP	38
3.7	Workflow de Análise de Exomas	40
3.7.1	Alinhamento dos dados de SOLID 5500xl	40

4	RESULTADOS	42
4.1	Alinhamento e Análise de Exomas	42
4.1.1	Controle de Qualidade sobre os dados	42
4.1.2	Alinhamento	48
4.1.3	GATK Genome Analysis ToolKit	48
4.1.3.1	Remoção de leituras duplicadas	52
4.1.3.2	Realignamento Local em regiões de Indels	52
4.1.3.2.1	Criação dos alvos para realinhamento	52
4.1.3.2.2	Aplicação da Recalibração	56
4.1.3.3	Recalibração do Escore de Qualidade da Base	56
4.1.3.4	Chamada de variantes	60
4.1.3.5	Recalibração do Escore de Qualidade da Variante	60
4.1.3.6	Processo de Filtragem de Indels.....	60
4.2	Anotação de Exomas	62
4.3	Mendel,MD - Construção da Ferramenta.....	62
4.3.1	Banco de Dados	63
4.3.2	Dashboard	64
4.3.3	Upload de Genomas	66
4.3.4	Agendador de Tarefas	66
4.3.5	Conversão dos dados para CSV	66
4.3.6	Anotação de Variantes	71
4.3.7	Controle de Qualidade sobre os dados	73
4.3.8	Genes	76
4.3.9	Doenças	76
4.3.10	Filtragem de Variantes	76
4.3.10.1	1-Click	84
4.3.10.2	Análise de Filtros para Famílias	85
4.3.10.3	Visualização de variantes	85
4.3.10.4	Exportação de resultados	85
4.3.11	Comparação de Exomas	88

4.4	Casos Clínicos	88
4.4.1	Caso Clínico LGC 1.....	88
4.4.2	Caso Clínico LGC 2.....	91
4.4.3	Outros casos clínicos	93
4.5	Mendel,MD - Validação	98
4.5.1	Caso 1	100
4.5.2	Caso 2	100
4.5.3	Caso 3	100
4.5.4	Caso 4	101
4.5.5	Caso 5	101
4.5.6	Caso 6	101
4.5.7	Caso 7	102
4.5.8	Casos 8 e 9	102
4.5.9	Caso 10	103
4.5.10	Caso 11	103
4.6	Validação da usabilidade da ferramenta	103
5	DISCUSSÃO	106
5.1	Sobre o armazenamento dos dados	107
5.2	Sobre a anotação de variantes	108
5.3	Sobre a filtragem de variantes	109
5.4	Variantes falso-positivas e falso-negativas	110
5.5	Exportação dos dados	111
5.6	Visualização dos Dados	112
5.7	Questões Éticas	113
5.7.1	Em relação ao número de pacientes	114
5.7.2	Validação Experimental	114
5.7.3	Sobre o futuro da análise de exomas e genomas	115
5.8	Custo do Sequenciamento e da Interpretação de Exomas	115
5.9	Serviços Online e Softwares Comerciais	115

6	CONCLUSÃO E PERSPECTIVAS	117
	REFERÊNCIAS	119
	ANEXO A – MODELO UTILIZADO PARA ARMAZENAR OS DADOS DE CADA INDIVÍDUO.....	125
	ANEXO B – MODELO UTILIZADO PARA ARMAZENAR OS DADOS SOBRE AS VARIANTES DE CADA IN- DIVÍDUO.....	128
	ANEXO C – BWA - SCRIPT DESENVOLVIDO PARA REALI- ZAR O ALINHAMENTO DOS ARQUIVOS EM FORMATO FASTQ.....	133
	ANEXO D – GATK- SCRIPT DESENVOLVIDO PARA EXE- CUTAR O PROGRAMA GATK.....	137
	ANEXO E – UNIFIED GENOTYPER - GATK - SCRIPT DE- SENVOLVIDO PARA REALIZAR O CALLING DE VARIANTES	154

1 Introdução e Justificativa

Desde a invenção dos primeiros computadores em 1950, já existia uma certa discussão sobre o potencial de sua aplicação em áreas como a Medicina e a Biologia. Em 1968, A. A. Robertson publicou um artigo com o título “*The use of computers in medicine with particular reference to general practice*”, porém, uma definição mais concreta sobre esta nova área da ciência que estava surgindo só começou a ser formulada em 1980 quando Edward Shortlife, Larry Fagan e Gio Wiederhold começaram a escrever o primeiro livro sobre Informática Médica para um curso de Medicina na Universidade de Stanford ([ADAMS, 1991](#)).

1.1 Informática Biomédica

De acordo com a terceira versão do livro “*Biomedical Informatics*” publicada em 2006 por Shortlife, essa área de conhecimento corresponde ao estudo da informação biomédica, conectando dados e conhecimento, sendo responsável pelo armazenamento e recuperação dos dados, e da utilização de computadores para a solução de problemas médicos e auxílio na tomada de decisão.

Inicialmente esta área foi chamada de Informática Médica, sendo que o termo Informática Biomédica só começou a surgir a partir dos anos 90, com a expansão do Projeto Genoma Humano e da investigação e análise de dados aplicados em Biologia, através da conscientização de que os métodos e processos utilizados também poderiam ser aplicados na Biomedicina de uma maneira mais ampla e geral ([KULIKOWSKI et al., 2012](#)).

Apesar de Informática Biomédica ser o termo mais adotado atualmente, ainda existe uma grande discussão sobre a real definição do seu significado e quais são as suas áreas de atuação. Um exemplo dessa discussão é o fato de que as associações representantes da área nos Estados Unidos ainda são chamadas de “informática médica” como a “*American Medical Informatics Association*” e “*International Medical Informatics Association*”.

ciation”. No Brasil a associação representante desta área chama-se Sociedade Brasileira de Informática em Saúde (SBIS)¹, essa associação foi fundada em 1986 pelo Prof. Renato Sabbatini com o objetivo de promover a troca de idéias e a apresentação de resultados nas áreas de Informática em Saúde, Telemedicina e Bioinformática.

De acordo com Bernstam (BERNSTAM; SMITH; JOHNSON, 2010) em seu artigo “*What is Biomedical Informatics*” publicado em 2010, a Informática Biomédica pode ser definida como a aplicação da Ciência da Informação e da Computação como dados e conhecimento para solução de problemas de interesse biomédicos.

1.2 Bioinformática

A Bioinformática é uma área multidisciplinar que integra os conhecimentos de diversas áreas diferentes como por exemplo: Biologia, Computação, Estatística, Matemática e Engenharia para realizar a análise de dados biológicos, além de construir ferramentas que permitam o armazenamento, organização, visualização e interpretação dos dados através do uso de um computador para realizar essas tarefas e gerar novo conhecimento.

A palavra Bioinformática foi utilizada pela primeira vez em um artigo publicado por Paulien Hogeweg em 1978 para descrever o estudo da informação em sistemas bióticos (HOGEWEG, 2011). Desde então este termo se tornou mais abrangente, ganhando popularidade entre 1990 e 2000 no período chamado de revolução genômica quando passou a ser utilizado para representar a construção, manutenção e interpretação de grandes bancos de dados biológicos impulsionados pelo sucesso do Projeto Genoma Humano e o aumento da quantidade de dados biológicos disponíveis.

Apesar de Informática Biomédica e da Bioinformática serem áreas distintas, a pergunta que sempre surge neste ponto é qual seria então a interseção entre as duas áreas e esse assunto já foi amplamente discutido em estudos anteriores (MAOJO; KULIKOWSKI, 2003; MARTIN-SANCHEZ et al., 2004). De maneira geral podemos dizer que a Bioinformática possui um foco maior na Biologia Molecular enquanto a Informática Biomédica possui um foco maior na área Médica.

¹ <<http://www.sbis.org.br/>>

A associação que representa a área de Bioinformática no Brasil chama-se Associação Brasileira de Bioinformática e Biologia Computacional (AB3C)².

1.3 O Exoma

Um éxon pode ser descrito como a parte do gene que fará parte do RNA maduro depois da remoção de todos os íntrons durante um processo de edição do RNA mensageiro que é chamado de *RNA splicing*. A figura 1 apresenta o modelo de uma estrutura gênica com as suas regiões de íntrons, éxons e UTRs (*Untranslated Regions*) delimitadas com as cores vermelho, azul e cinza respectivamente.

Um exoma pode ser definido como a coleção de todos os éxons de um genoma. Essas regiões correspondem a cerca de 1.5% do genoma humano (30 Megabases) e estão divididas entre 180,000 éxons. Apesar de constituir uma parte pequena do genoma, acredita-se que esta região contenha cerca de 85% das mutações associadas a doenças genéticas humanas (CHOI et al., 2009). Isso demonstra a importância do estudo e da compreensão dessa região para ajudar na compreensão de doenças genéticas. Até fevereiro de 2015, 2937 genes associados a 4163 doenças mendelianas foram descobertos, porém cerca de 50% das doenças mendelianas ainda não possuem um gene identificado. Portanto essa área ainda fornece grandes oportunidades para a descoberta de novos genes associados a doenças mendelianas (CHONG et al., 2015).

1.4 Doenças Mendelianas

Doenças Mendelianas são doenças genéticas que podem ser causadas por uma única mutação em um único gene. Além disso elas podem ser herdadas dos pais, ou seja, podem ser transmitidas de uma geração para outra de acordo com o modelo de herança proposto pelas leis de Mendel. Podemos citar como exemplos de doenças Mendelianas a anemia falciforme (OMIM:603903), a fibrose cística (OMIM:219700), a síndrome de

² <<http://ab3c.cebio.org/>>

A estrutura de um Gene



Figura 1 – Esta figura mostra a estrutura simplificada de um gene com suas regiões de íntrons, éxons e UTRs (regiões não traduzidas). Fonte: <<http://en.wikipedia.org/wiki/Exon>>

Marfan (OMIM:154700) e a doença de Huntington (OMIM:143100). Existem atualmente mais de 4 mil doenças Mendelianas conhecidas de acordo com o site OMIM.

1.5 Análise de Exomas para o diagnóstico de doenças Mendelianas

Em 2008 ([NG et al., 2008](#)) publicaram um artigo contendo o sequenciamento do primeiro exoma humano (do pesquisador J. Craig Venter), mas apenas em 2009, foi publicado o primeiro artigo que utilizou o sequenciamento de exomas e o método de filtragem de variantes como uma prova de conceito para realizar a identificação do gene *MYH3* (Gene ID: 4621) como sendo associado a Síndrome de Freeman-Sheldon (OMIM:193700) ([NG et al., 2009](#)). Para identificar este gene foi necessário realizar o sequenciamento de 12 exomas, sendo 4 dos indivíduos afetados pela doença não relacionados e 8 indivíduos normais do projeto HapMap que foram utilizados como controles no processo de filtragem de variantes. Desde então centenas de outros estudos foram publicados com a utilização do sequenciamento de exomas de pacientes para descoberta de novas associações entre genes e doenças Mendelianas que até então ainda eram desconhecidas ([BAMSHAD et al., 2011](#)).

A síndrome de Freeman-Sheldon (FSS) é uma doença genética causada por herança autossômica dominante, a forma mais conhecida pela literatura, e também por herança autossômica recessiva, ambas as formas são responsáveis por causar alterações ósseas e contraturas articulares levando a um fenótipo do paciente que é bastante característico.

A figura 2 mostra como foi o processo de filtragem de variantes que foi utilizado para a identificação do gene *MYH3*. Podemos observar através desta figura que ao filtrar apenas por genes com mutações não sinônimas que estivessem presentes nos quatro indivíduos foi possível reduzir o número de genes candidatos para 2479 genes, ao eliminarem as mutações que estavam presentes apenas na última versão do dbSNP conseguiram diminuir a lista de genes candidatos para 53, ao utilizarem os dados dos 8 exomas controles do projeto HapMap conseguiram reduzir a lista de genes para 21, e por fim ao removerem os dados do dbSNP e do HapMap ao mesmo tempo, conseguiram reduzir a lista de genes candidatos para apenas 1 único gene. É importante notar que este gene também poderia

ter sido encontrado utilizando outros critérios de filtragem como por exemplo o escore de SIFT e Polyphen-2 de cada variante. Este estudo mostrou que com a utilização de apenas 4 indivíduos não relacionados seria possível identificar o gene causador da doença Mendeliana investigada.

O gene *MYH3* já era conhecido como sendo o causador síndrome de Freeman-Sheldon porém o estudo serviu como uma prova de conceito de que a tecnologia e o método poderiam ser utilizados para investigação de outras doenças onde o gene responsável ainda fosse desconhecido.

Ainda em 2009, Ng e o seu grupo ([NG et al., 2010](#)) descobriram a associação de um novo gene chamado *DHODH* (Gene ID: 1723) como sendo o causador da Síndrome de Miller (OMIM%263750) utilizando o mesmo método que havia sido validado pelo estudo anterior. Esta foi a primeira vez que a análise de exomas foi utilizada para identificação de um novo gene responsável por causar uma doença Mendeliana onde até então sua causa ainda era desconhecida.

Um estudo publicado em Fevereiro de 2015 ([HU et al., 2015](#)) realizou o sequenciamento do cromossomo X de 405 famílias com casos clínicos que ainda não haviam sido solucionados por outros métodos, e como resultado deste estudo, eles conseguiram identificar 7 novos genes associados a deficiência intelectual.

A figura 3 mostra o crescimento do número de artigos publicados com a palavra “*exome*” em títulos ou abstracts desde o ano de 2008 de acordo com o site Pubmed.

Podemos observar que a partir de 2008 este número tem crescido a cada ano e que a previsão do número de artigos publicados em 2015 será ainda maior do que foi em 2014.

Esta técnica vem crescendo e se expandindo ao longo dos últimos anos e ainda deverá ser muito utilizada para realizar a investigação e o diagnóstico de doenças genéticas humanas. Apesar do sucesso comprovado deste método, ainda existem grandes desafios para que esta técnica seja completamente adotada e integrada dentro da prática clínica.

Atualmente existem poucas ferramentas desenvolvidas com o objetivo de auxiliar médicos e pesquisadores a analisarem este tipo de dado, e grande parte das ferramentas que existem, ainda exigem a execução de scripts e comandos manuais geralmente em um terminal Linux com o uso de parâmetros que possibilitem a exploração dos dados

Processo de Filtragem de Variantes para identificação de uma doença Mendelianas

	FSS24895	FSS24895 FSS10208	FSS24895 FSS10208 FSS10066	FSS24895 FSS10208 FSS10066 FSS22194	<u>ANY 3 OF 4</u>
SNPs não-sinônimas, em regiões de splicing ou indels	4,510	3,284	2,765	2,479	3,768
SNPs não presentes no dbSNP	513	128	71	53	119
SNPs não presentes em 8 exomas de indivíduos do HapMap	799	168	53	21	160
SNPs não presentes no dbSNP e em 8 exomas de indivíduos do HapMap	360	38	8	1 (MYH3)	22
SNPs preditas como damaging	160	10	2	1 (MYH3)	3

Figura 2 – Nesta imagem podemos verificar todas as etapas de filtragem que foram utilizadas para identificação do gene *MYH3* como causador da síndrome de Freeman-Sheldon. Fonte: ([NG et al., 2009](#))

Crescimento do número de artigos publicados com a palavra exome no Pubmed

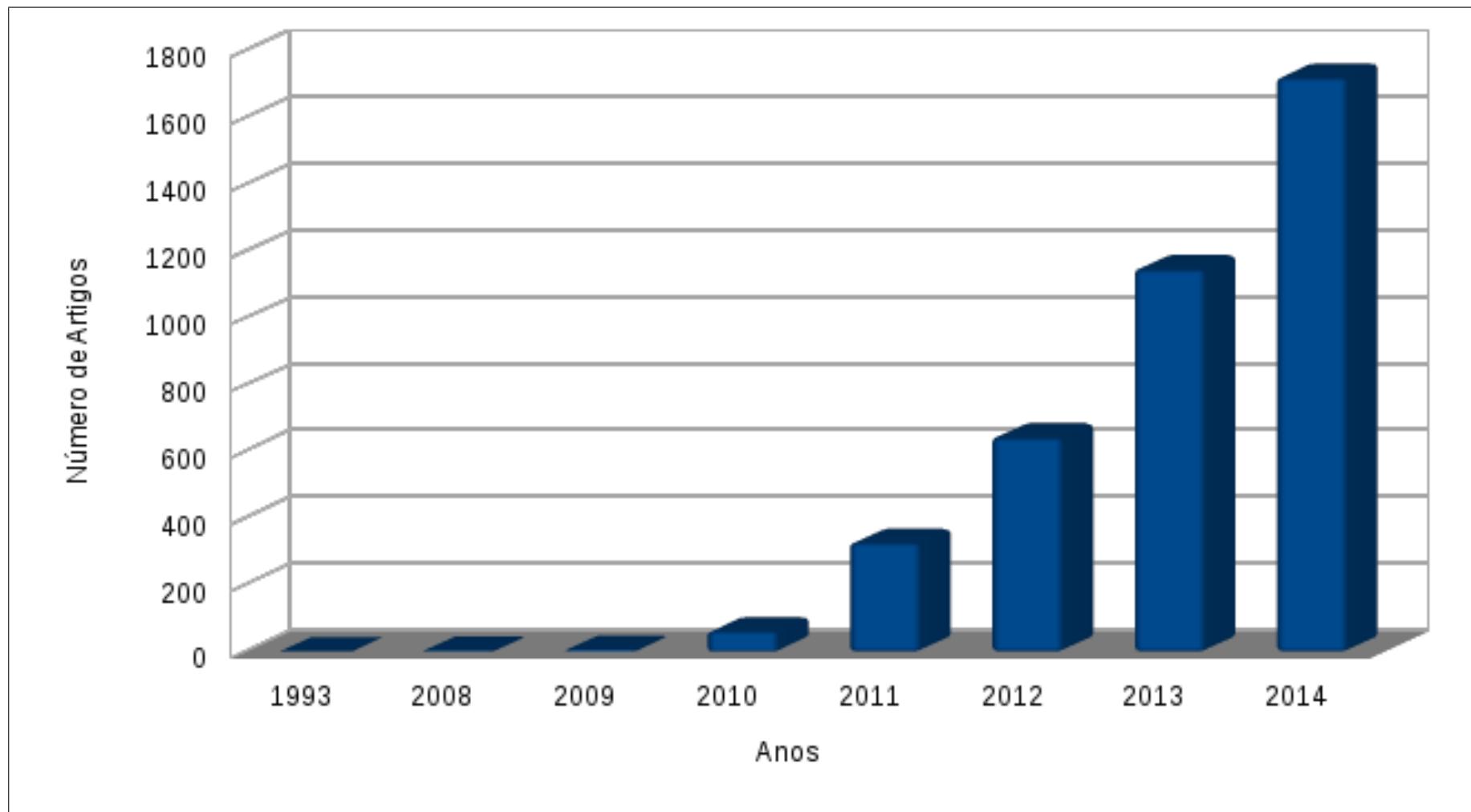


Figura 3 – Nesta imagem podemos verificar o crescimento do número de artigos publicados com a palavra “exome” de acordo com o Pubmed ao longo dos últimos anos.

e a filtragem de variantes. A ausência de uma interface simples e amigável dificulta o acesso aos métodos e as informações por pessoas que não possuem conhecimentos sobre programação ou computadores.

Para este método atingir sua plenitude é preciso que existam programas práticos e acessíveis, que possibilitem a exploração dos dados não apenas por cientistas, mas também por médicos e outros profissionais da área da Saúde.

É neste sentido que este trabalho foi iniciado, para ajudar traduzir os resultados obtidos a partir do sequenciamento de exomas e para tentar extrair informação útil deste tipo de dado para auxiliar no diagnóstico clínico de doenças Mendelianas.

1.6 SNPs e Indels

O Polimorfismo de Nucleotídeo Único (SNP) é uma variação genética causada pela alteração em um único par de base em uma posição específica do DNA.

A figura 4 mostra a representação gráfica de uma SNP e nela podemos observar a alteração de uma posição contendo GC para AT.

Os SNPs presentes nas regiões exônicas podem ser classificadas como sinônimas, ou seja, aquelas que não causam alteração na sequência de aminoácidos da proteína, e não-sinônimas, que são aquelas que causam alteração na sequência de aminoácidos da proteína. As mutações não-sinônimas podem ser classificadas como *missense* e *nonsense*. As mutações *missense* são aquelas que causam uma substituição de um aminoácido por outro, e as mutações *nonsense* são aquelas que causam a substituição de um aminoácido por um *stop* códon, o que pode causar a produção de uma proteína não funcional.

Indel é um polimorfismo que corresponde a adição ou remoção de uma pequena sequência de bases de DNA, sendo que a maioria ocorre em regiões de repetição em tandem.

Na figura 5 é apresentado a representação gráfica de uma indel. Podemos observar uma deleção de duas bases AC na parte de cima e da inserção de duas bases AC na parte de baixo. Existem muitas indels associadas a doenças humanas e este é um assunto já bastante discutido em estudos anteriores (TAYLOR; PONTING; COPLEY, 2004; HARAKSINGH;

Representação gráfica de uma SNP

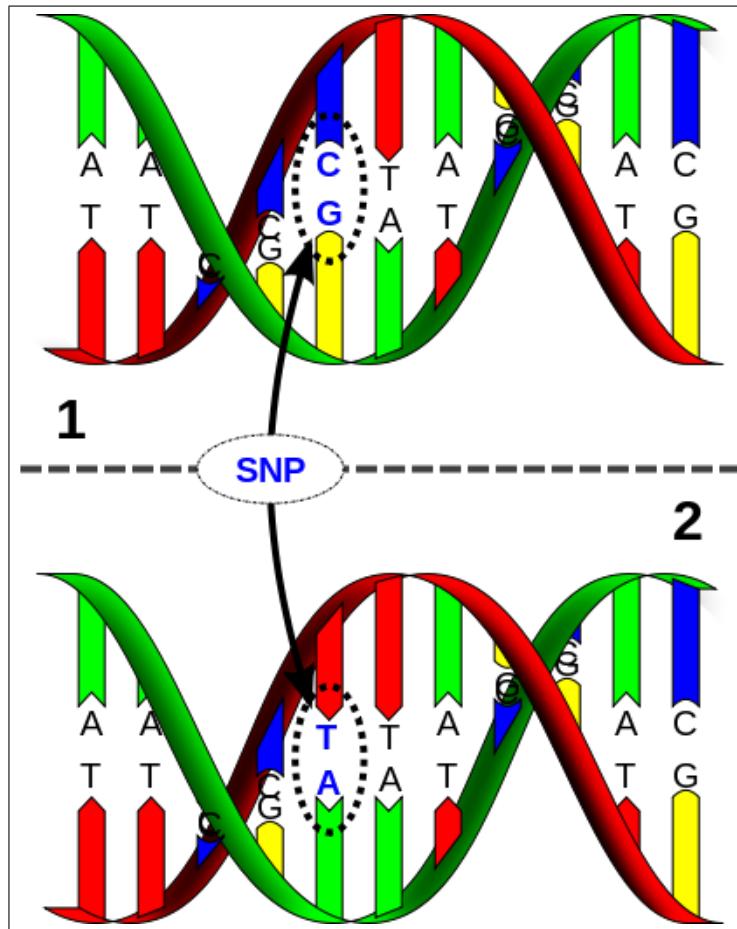


Figura 4 – Representação gráfica de uma SNP mostrando uma alteração de G->C para A->T. Fonte: <http://en.wikipedia.org/wiki/Single-nucleotide_polyorphism>

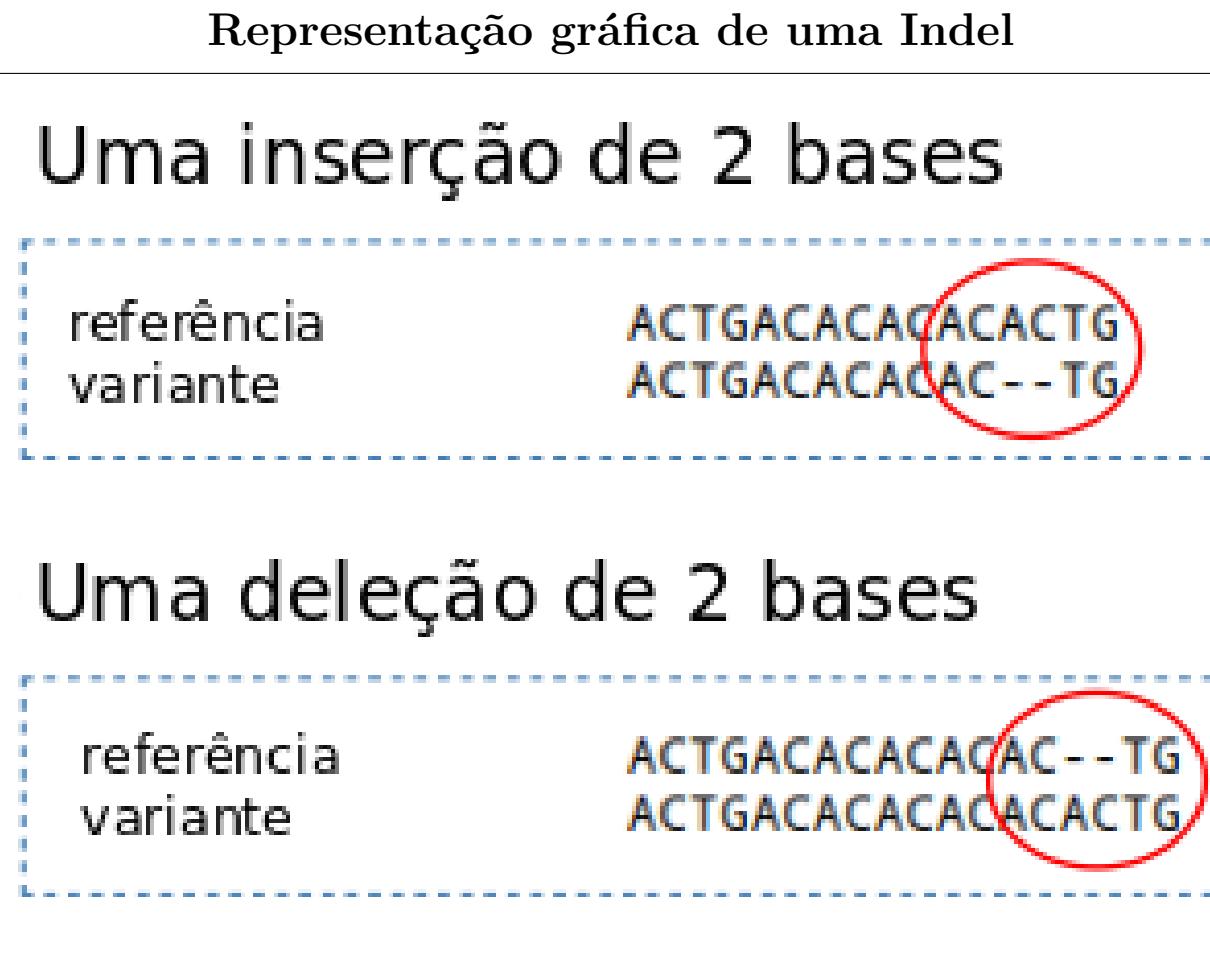


Figura 5 – Podemos observar uma deleção de duas bases AC na parte de cima e da inserção de duas bases AC na parte de baixo. Fonte: <<http://genome.sph.umich.edu/wiki/Indel>>

SNYDER, 2013).

1.7 Projeto Genoma Humano

O Projeto Genoma Humano (LANDER et al., 2001) foi um consórcio científico internacional coordenado pelo *National Institutes of Health* (NIH) com o objetivo de mapear todos os genes humanos e realizar o sequenciamento dos 3 bilhões de nucleotídeos do genoma humano. Esse projeto teve um custo aproximado de 3 bilhões de dólares e em 2001, 10 anos após o seu início do projeto, foi publicado o primeiro rascunho do genoma humano.

Em 1998 o pesquisador J. Craig Venter deu início a um projeto paralelo em sua empresa *Celera Genomics* para sequenciar o genoma humano usando uma técnica conhecida por *Shotgun Sequencing* (VENTER et al., 2001). Esta técnica permitiu o sequenciamento do genoma humano através da quebra do DNA em pequenos fragmentos de tamanho entre 100 e 1000 pares de base de maneira que a montagem do genoma pudesse ser realizada através do uso de programas de computador. Para isso foi necessário atingir uma cobertura mínima de 12X, ou seja, cada base do genoma foi sequenciada pelo menos 12 vezes para que essa montagem fosse possível.

Ambos os projetos foram publicados em 2001 pelas revistas *Nature* e *Science* (LANDER et al., 2001; VENTER et al., 2001).

1.8 Projeto 1000 Genomas

O “*1000 Genomes Project*” (DURBIN et al., 2010; ABECASIS et al., 2012) é um projeto de colaboração internacional criado em 2009 com o objetivo de produzir um catálogo abrangente sobre as variações genéticas humanas. Até o momento já foram sequenciados e genotipados 2504 indivíduos³, esses dados estão disponibilizados de forma totalmente pública na internet para qualquer pesquisador que tiver interesse em realizar o download para analisá-los. Isso passou a permitir o uso desses dados de forma a auxiliar

³ <<http://www.1000genomes.org/data>>

a clínica médica para realizar análises de exomas de pacientes e por exemplo para fazer a comparação de grupos de indivíduos normais presentes nesses bancos de dados com pacientes clínicos permitindo a eliminação de variantes controles que estivessem presentes no grupo de indivíduos de uma população.

Na tabela 1 apresentamos informações sobre o número dos indivíduos que foram sequenciados e os grupos de ancestralidade que foram definidos pelos pesquisadores do projeto *1000 Genomes*.

Site: <<http://www.1000genomes.org>>

1.9 Projeto de Sequenciamento do Exoma

O *Exome Sequencing Project* (ESP) (FU et al., 2013) realizou até o momento o sequenciamento de 6515 exomas e embora os genótipos dos indivíduos não estejam disponíveis para download, é possível obter informações sobre a frequência das variantes que foram encontradas e este valor pode ser utilizado no processo de filtragem de variantes de indivíduos que possuam uma Doença Mendeliana específica.

Site: <<http://evs.gs.washington.edu/EVS>>

1.10 Medicina Personalizada / Medicina de Precisão

O termo “Medicina Personalizada” começou a surgir a partir de 1999 quando Robert Langreth e Michael Waldholz publicaram no jornal “*The Oncologist*” o primeiro artigo com uma definição mais próxima da que temos atualmente sobre essa nova área da medicina (LANGRETH; WALDHOLZ, 1999). A Medicina Personalizada é um modelo de saúde que propõe a realização de práticas utilizando dados como o perfil genômico do paciente para a tomada de decisão médica, com o objetivo de direcionar o melhor tratamento específico para cada paciente de acordo essas informações.

Na figura 6 apresentamos uma imagem com a relação entre a Medicina Tradicional, a Genômica Personalizada e a Farmacogenômica. Essa área atualmente tem sido chamada de Medicina de Precisão (COLLINS; VARMUS, 2015; JAMESON; LONGO, 2015) por

Tabela 1 – Informações sobre a ancestralidade dos indivíduos que foram sequenciados pelo projeto *1000 Genomes*. Fonte: <<http://www.1000genomes.org/about>>

Populações	Número final de amostras
Ancestralidade Leste-Asiática	504
Ancestralidade Sul-Asiática	489
Ancestralidade Africana	661
Ancestralidade Européria	503
Ancestralidade Ameríndia	347
Total	2504

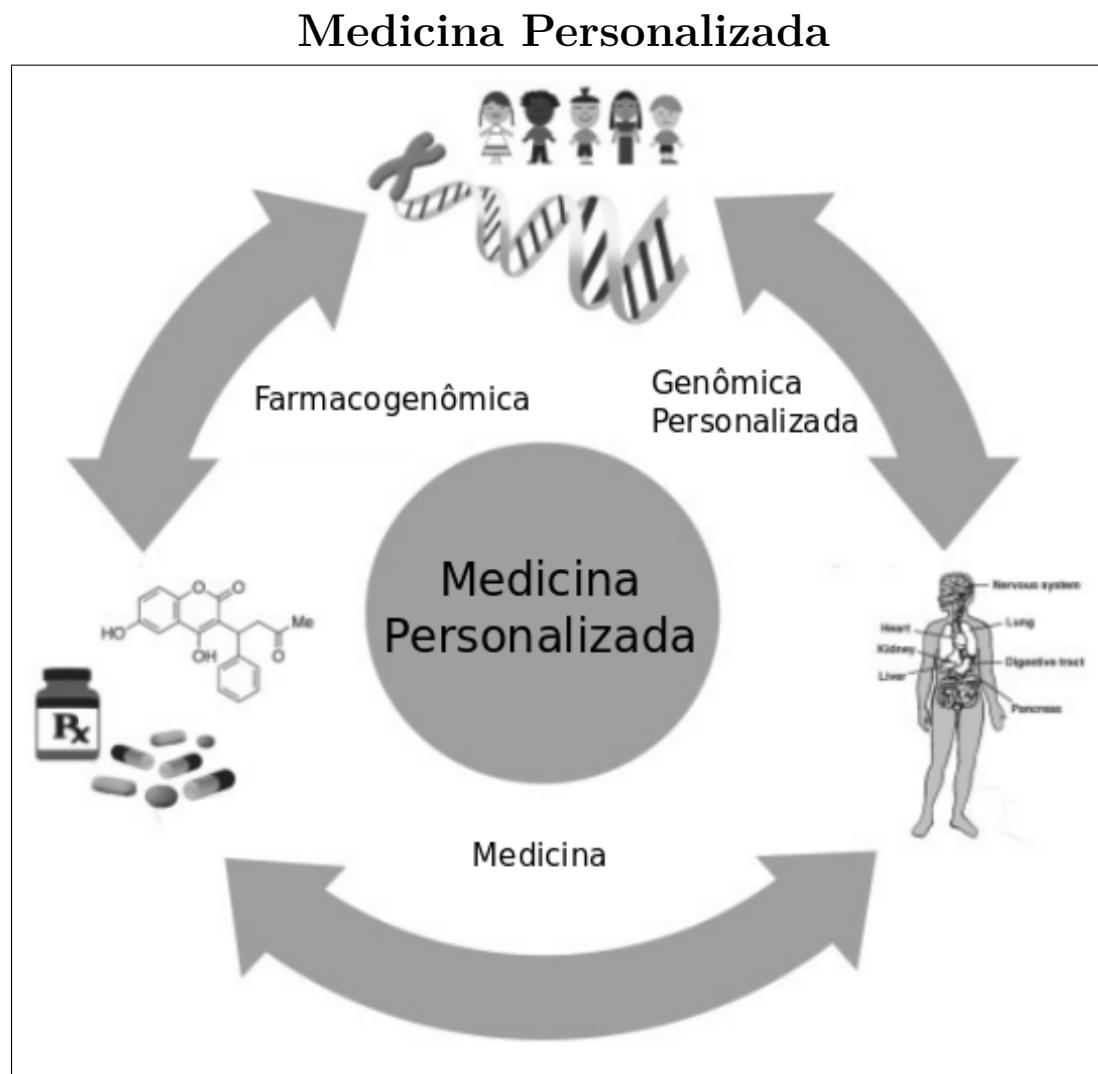


Figura 6 – Aqui podemos verificar que Medicina Personalizada é um termo que relaciona a Medicina Tradicional, a Genômica Personalizada e a Farmacogenômica. Fonte: ([FERNALD et al., 2011](#))

ser um termo que representa melhor a aplicação dos dados genômicos de uma forma direcionada para atender as necessidades clínicas de cada paciente.

2 Objetivos

2.1 Objetivo Geral

Estudar e desenvolver métodos e ferramentas que permitam o armazenamento e a análise de dados genômicos humanos possibilitando a investigação e o diagnóstico de casos clínicos de pacientes estudados pelo laboratório de uma maneira rápida e eficiente por médicos, cientistas e outros profissionais da área da Saúde. A seguir, apresentamos os objetivos específicos que foram definidos para serem realizados durante o desenvolvimento deste trabalho.

2.2 Objetivos Específicos

- Desenvolver uma ferramenta que permita a médicos e cientistas investigarem os dados genômicos de seus pacientes.
- Gerar métricas de qualidade sobre o sequenciamento, a genotipagem e a cobertura do sequenciamento para cada indivíduo inserido no sistema.
- Desenvolver um *pipeline* de anotação de variantes utilizando diferentes métodos e programas que ajudem na identificação de mutações que possam ser responsáveis por causar uma doença Mendeliana.
- Permitir a identificação de mutações em genes e doenças que já tiverem sido descritas pela literatura e de novas mutações e genes candidatos possivelmente associados com síndromes Mendelianas ainda não descritas pela literatura.
- Validar o programa desenvolvido com dados reais de pacientes do Laboratório de Genômica Clínica da Faculdade de Medicina da UFMG e com dados de exomas clínicos descritos anteriormente pela literatura.

- Estudar algoritmos de priorização de variantes como SIFT, Polyphen-2, CADD e Mutation Taster para estimar formas mais racionais de utilizar seus parâmetros para melhorar a análise dos dados.

3 Materiais e Métodos

3.1 Hardware

O desenvolvimento e as análises deste trabalho foram realizados em um computador Desktop com 500GB de espaço em disco, 4GB de RAM e 2 processadores Intel Pentium(R) Dual-Core E5400 @ 2.70GHz. Para o ambiente de produção foi utilizado um servidor Dell PowerEdge T-710 com 12TB de espaço em disco, 115GB de RAM e 24 cores Intel(R) Xeon(R) CPU X5660 @ 2.80GHz.

O pipeline de análise de exomas desenvolvido por este trabalho foi executado tanto no servidor local quanto no supercomputador do Centro Nacional de Processamento de Alto Desempenho de Minas Gerais (CENAPAD-MG) que possui 107 nós de computadores interligados através de uma rede rápida Infiniband com 8GB de RAM e 8 núcleos (Intel(R) Xeon(R) E5335 @ 2.00GHz) por máquina, totalizando 856 núcleos e 856GB de RAM disponíveis para serem utilizados para a execução de processos.

3.2 Sistema Operacional

O sistema operacional escolhido para o desenvolvimento deste projeto foi o Biolinux ([FIELD et al., 2006](#)) atualmente na versão 8.0. Este é um sistema baseado em Ubuntu que foi criado e mantido pelo “NERC Environmental Bioinformatics Centre” na Inglaterra desde 2002, e foi escolhido por possuir mais de 500 pacotes pré-instalados para a Bioinformática, todos configurados e prontos para serem utilizados como por exemplo: Galaxy, Biopython, Bioperl, Bioconductor entre muitos outros pacotes que estão disponíveis no repositório criado pelo projeto que podem ser acessados no link abaixo.

Site: <<http://nebc.nerc.ac.uk/tools/bio-linux/>>

3.3 Linguagens e Frameworks Utilizados

3.3.1 Bash

Bash é um ambiente “*shell*” de programação unix que oferece uma linha de comando interativa e permite a execução de scripts e comandos que são executados de uma maneira sequencial com o objetivo de automatizar o uso de programas utilizados para realizar a análise dos dados genômicos. Diversos scripts em Bash foram desenvolvidos ao longo deste trabalho, inclusive para executar o pipeline de análise de exomas desenvolvido por este trabalho no supercomputador do CENAPAD-MG da UFMG.

3.3.2 Python

O Python é uma linguagem de programação orientada a objetos que foi criada por Guido Van Rossum em 1991 ([ROSSUM, 1995](#)). Suas principais características são simplicidade, flexibilidade e elegância.

Todos os scripts deste projeto foram desenvolvidos utilizando Python 2.7 e atualmente estão sendo convertidos para Python 3. Esta linguagem foi escolhida por permitir a realização da análise dos dados genômicos e a construção da parte web utilizando para ambas as tarefas uma única linguagem de programação. Isso facilitou muito a integração das análises e o desenvolvimento do projeto, além de ter possibilitado o uso de algoritmos e técnicas modernas de programação de uma maneira estruturada, modular e orientada a objetos.

A linguagem Python tem sido cada vez mais adotada na Bioinformática para realizar a análise de dados biológicos, especialmente pela popularização de bibliotecas como ipython notebook, Biopython, Numpy, Scipy e Rpy2. Cada um desses projetos trouxe enormes benefícios para os projetos de pesquisa em que eles foram utilizados.

Site: <<http://www.python.org>>

3.3.3 Django

O Django é um “*web framework*” desenvolvido em Python que é bastante utilizado para o desenvolvimento de websites e aplicações web. Possui um padrão de arquitetura chamado de MVC (*Model, View e Controller*) que define uma separação entre essas camadas e controla o acesso ao banco de dados através de um processo conhecido como mapeamento de objetos relacionais (ORM). Este framework foi utilizado para construir toda a interface online deste trabalho e foi escolhido por ser rápido e estável além de permitir a mudança do tipo de banco de dados (Ex. MySQL, PostgreSQL, SQLite e etc) sem que para isso fosse preciso modificar uma única linha do código do sistema que foi desenvolvido.

A figura 7 mostra um diagrama com a separação entre as três camadas do Django. Podemos observar que essas camadas ficam localizadas entre o banco de dados e o servidor. O servidor web fica responsável por renderizar os resultados da cada página após consultar um banco de dados em retornar uma resposta em formato html para o navegador do usuário.

A camada de Visualização é responsável por encapsular toda a lógica de processamento de cada requisição que é feita pelo usuário ao Django, é nesta camada onde são aplicados todos os critérios de filtragem de variantes que foram definidos pelos usuário. A camada de Modelo e ORM é responsável pelo acesso ao banco de dados através da criação de consultas em SQL que possuem os critérios de filtragem definidos pelo formulário e finalmente a camada de Templates gera o HTML final com o resultado obtido para o usuário final utilizando HTML, CSS e Javascript.

Site: <<http://www.djangoproject.org>>

3.4 Formatos Utilizados

3.4.1 FASTQ

O formato FASTQ foi proposto pela primeira vez em 2009 por (COCK et al., 2009) e surgiu como um formato utilizado para descrever pequenas sequências de DNA

Diagrama com a separação entre as camadas do Django

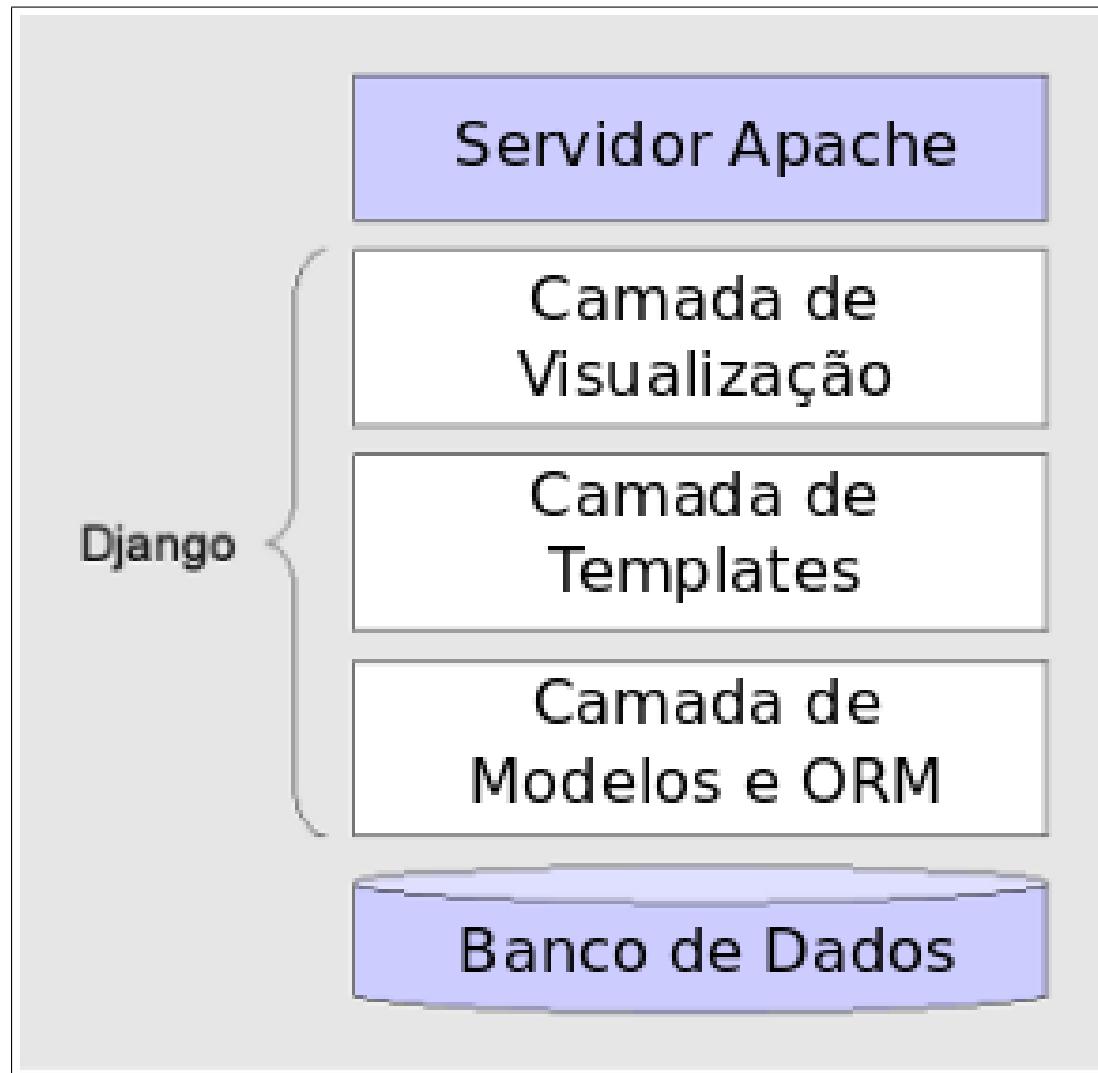


Figura 7 – Nesta figura podemos observar a separação interna do Django entre as camadas de Modelo, de Visualização e de Templates. Fonte:<<http://excess.org/article/2009/05/django-1-1-talk-text/>>

chamadas de leituras (*reads*) que são geradas pelos sequenciadores de DNA de nova geração. A característica principal deste formato é o armazenamento em um único arquivo da informação sobre as sequências de DNA que foram lidas e do escore de qualidade de cada base das sequências que estão presentes neste arquivo.

A figura 8 apresenta o exemplo de uma leitura em formato FASTQ. Podemos observar que cada sequência é iniciada pelo símbolo @, seguida de um ID único e com informações como o tamanho de cada leitura que foi lida pela máquina. O símbolo “+” significa indicar o início dos valores de qualidade de cada base na próxima linha para a leitura que foi lida anteriormente.

Existem algumas variações deste formato, como por exemplo em sequenciadores do modelo SOLID 5500XL são adicionadas informações sobre valores de qualidade de cada base em *color space*. Esta tecnologia foi criada pela Applied Biosystems e chamada de *2_base encoding*, de acordo com a empresa, ela permite a leitura de cada base duas vezes sem precisar para isso realizar o sequenciamento dos dados duas vezes.

3.4.2 Sequence Alignment Map - Formato SAM

O formato SAM (Sequence Alignment Map) é um arquivo de texto delimitado por tabulações que foi proposto por Heng Li e Richard Dublin em 2009 (LI; DURBIN, 2009) e foi criado para armazenar as informações sobre alinhamentos de sequências como resultado após o mapeamento dessas sequências contra um genoma de referência, como por exemplo, o Genoma Humano.

A figura 9 apresenta o exemplo de um arquivo SAM. O cabeçalho deste arquivo é delimitado pelo símbolo @ no início de cada linha, seguido de uma chave do tipo ID (Ex. HD, SQ, RG) com informações sobre como este arquivo foi gerado e os parâmetros utilizados durante o mapeamento. A segunda parte deste arquivo possui informações sobre todas as regiões mapeadas, contendo os valores de qualidade para cada base do alinhamento e um escore de qualidade médio para toda a região que foi mapeada.

O formato SAM também possui uma versão chamada de BAM, que converte os dados do arquivo SAM para uma linguagem binária (de máquina) ao invés de texto, o que

Exemplo de uma leitura em formato FASTQ

```
@SRR014849.1 EIXKN4201CFU84 length=93
GGGGGGGGGGGGGGCTTTTTGTTGGAACCGAAAGG
GTTTGAAATTCAAACCCCTTCGGTTCCAACCTTCCAA
AGCAATGCCAATA
+SRR014849.1 EIXKN4201CFU84 length=93
3+&$#""""""""""7F@71,'";C?,B;?6B;:EA1EA
1EA5'9B:?:#9EA0D@2EA5':>5?:%A;A8A;?9B;D@
/= <?7=9<2A8==
```

Figura 8 – Nesta figura podemos observar uma leitura que está presente em um arquivo no formato FASTQ. - Fonte: ([COCK et al., 2009](#))

ajuda a reduzir muito o tamanho do arquivo em relação ao seu original, sem que ocorra perda de informação durante essa conversão.

3.4.3 Variant Call Format - Formato VCF

O formato VCF (*Variant Call Format*) foi proposto por Danecek em 2011 (DA-NECEK et al., 2011) e atualmente está em sua versão 4.2. Ele foi criado para armazenar informações sobre SNPs, Indels e Variantes Estruturais encontrados nos indivíduos sequenciados pelo projeto *1000 Genomes*, juntamente com informações relevantes sobre cada variante de uma maneira compacta e indexada, usando para isso um programa desenvolvido em C chamado tabix¹. Este programa permite a recuperação rápida das informações genotípicas de cada indivíduo utilizando posições indexadas em relação a um genoma de referência.

A figura 10 apresenta um exemplo de um arquivo no formato VCF. Podemos verificar uma separação entre o cabeçalho do arquivo que é sempre iniciado com o símbolo # e o corpo do arquivo que contém uma linha para cada variante encontrada. Este arquivo permite o armazenamento da informações do genótipo de múltiplos indivíduos, adicionando uma coluna no final do arquivo com a informação sobre o genótipo para cada indivíduo daquela posição. Todos os campos do corpo deste arquivo são separados por tabulações.

3.5 Programas Utilizados

3.5.1 FASTQC

O *FASTQC* é um programa em Java que oferece uma interface gráfica para análise de dados genômicos em formato FASTQ ou BAM. Este programa permite a geração de relatórios em HTML com imagens que possuem métricas que permitem avaliar a qualidade do sequenciamento. Site: <<http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>>

¹ <<http://www.htslib.org/doc/tabix.html>>

Exemplo de um arquivo no formato SAM

```
@HD VN:1.0
@SQ SN:chr20 LN:62435964
@RG ID:L1 PU:SC_1_10 LB:SC_1 SM:NA12891
@RG ID:L2 PU:SC_2_12 LB:SC_2 SM:NA12891
read_28833_29006_6945 99 chr20 28833 20 10M1D25M = 28993 195 \
    AGCTTAGCTAGCTACCTATATCTTGGTCTTGGCCG <<<<<<<<<<<<:<9/,&,22;;<< \
    NM:i:1 RG:Z:L1
read_28701_28881_323b 147 chr20 28834 30 35M      = 28701 -168 \
    ACCTATATCTTGGCCTTGGCCGATGCGGCCTTGCA <<<<;<<<7;:<<<6;<<<<<<<<7<<< \
    MF:i:18 RG:Z:L2
```

Figura 9 – Exemplo de um arquivo no formato SAM proposto por Li. Fonte: (LI; DURBIN, 2009)

Exemplo de um arquivo no formato VCF

```

##fileformat=VCFv4.1
##FORMAT<ID=AO,Number=A,Type=Integer>Description="Alternate allele observation count">
##INFO<ID=DP,Number=1,Type=Integer>Description="Total read depth at the locus">
##FORMAT<ID=DP,Number=1,Type=Integer>Description="Read Depth">
##FORMAT<ID=GQ,Number=1,Type=Integer>Description="Genotype Quality, the Phred-scaled marginal (or unconditional) probability of the called genotype">
##FORMAT<ID=GT,Number=1,Type=String>Description="Genotype">
##contig=<ID=chr1,length=249250621>
##contig=<ID=chr10,length=135534747>
##contig=<ID=chr11,length=135006516>
##contig=<ID=chr12,length=133851895>
##contig=<ID=chr13,length=115169878>
##contig=<ID=chr14,length=107349540>
##contig=<ID=chr15,length=102531392>
##contig=<ID=chr16,length=90354753>
##contig=<ID=chr17,length=81195210>
##contig=<ID=chr18,length=78077248>
##contig=<ID=chr19,length=59128983>
##contig=<ID=chr2,length=243199373>
##contig=<ID=chr20,length=63025520>
##contig=<ID=chr21,length=48129895>
##contig=<ID=chr22,length=51304566>
##contig=<ID=chr3,length=198022430>
##contig=<ID=chr4,length=191154276>
##contig=<ID=chr5,length=180915260>
##contig=<ID=chr6,length=171115067>
##contig=<ID=chr7,length=159138663>
##contig=<ID=chr8,length=146364022>
##contig=<ID=chr9,length=141213431>
##contig=<ID=chrM,length=16569>
##contig=<ID=chrX,length=155270560>
##contig=<ID=chrY,length=59373566>
##reference=/results/referenceLibrary/tmap-f3/Hg19/Hg19.fasta
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT MM14001
chr1 88922 . G A 1674.13 PASS . A0=176;DP=178;FA0=176;FDP=176;FR=.;FR0=0;FSAF=78;FSAR=98;FSRF=0;FSRR=0;FWDB=-0.00762337;FXX=0.
0112353;HRUN=2;LEN=1;MLLD=170.834;QD=38.0483;RBI=0.00861651;REFB=-0.0194898;REVB=0.
00401601;R0=0;SAF=78;SAR=98;SRF=0;SRR=0;SSEN=0;SSEP=0;SSSB=0;STB=0.5;TYPE=snp;VARB=2.33004e-05;OID=.;OPOS=88922;OREF=G;OALT=A;OMAPALT=A;EFF=INTRON(
MODIFIER|||||FAM41C||NON_CODING||NR_027055|2) GT:GQ:DP:FDP:RO:FR0:A0:FA0:SAR:SAF:SRF:FSAR:FSAF:FSRF:FSRR 1/1:71:178:176:0:0:176:176:98:78:0:0:98:78:0:0
chr1 88928 . C T 1633.46 PASS . A0=172;DP=177;FA0=172;FDP=172;FR=.;FR0=0;FSAF=75;FSAR=97;FSRF=0;FSRR=0;FWDB=-0.0163608;FXX=0.
028247;HRUN=2;LEN=1;MLLD=82.2213;QD=37.9874;RBI=0.0167765;REFB=-0.0544412;REVB=0.00371158;R0=0;SAF=75;SAR=97;SRF=0;SRR=0;SSEN=0;SSEP=0;SSSB=5.80769e-
08;STB=0.5;TYPE=snp;VARB=-7.29791e-05;OID=.;OPOS=88928;OREF=C;OALT=T;OMAPALT=T;EFF=INTRON(MODIFIER|||||FAM41C||NON_CODING||NR_027055|2)
GT:GQ:DP:FDP:RO:FR0:A0:FA0:SAR:SAF:SRF:FSAR:FSAF:FSRF:FSRR 1/1:69:177:172:0:0:172:172:97:75:0:0:97:75:0:0
chr1 856041 . G A 77.44 PASS . A0=26;DP=75;FA0=26;FDP=75;FR=.;FR0=49;FSAF=19;FSAR=7;FSRF=24;FSRR=25;FWDB=-0.
0106165;FXX=0;HRUN=1;LEN=1;MLLD=135.306;QD=4.13036;RBI=0.0142518;REFB=-0.0209785;REVB=-0.
00950808;R0=48;SAF=19;SAR=7;SRF=23;SRR=25;SSEN=0;SSEP=0;SSSB=0.298059;STB=0.668731;TYPE=snp;VARB=0.
0313016;OID=.;OPOS=856041;OREF=G;OALT=A;OMAPALT=A;EFF=INTERGENIC(MODIFIER||||||),UPSTREAM(MODIFIER|||||LOC100130417||NON_CODING||NR_026874|)
GT:GQ:DP:FDP:RO:FR0:A0:FA0:SAR:SAF:SRF:FSAR:FSAF:FSRF:FSRR 0/1:77:75:75:48:49:26:26:7:19:23:25:7:19:24:25

```

Figura 10 – Exemplo de um arquivo no formato VCF podemos observar alguns campos do cabeçalho e do corpo.

3.5.2 SAMTOOLS

O *SAMTOOLS* é um programa de linha de comando que trabalha com arquivos SAM/BAM e VCF. Além de permitir a indexação e conversão desses arquivos, este programa também possui um algoritmo Bayesiano para identificação de variantes a partir de um arquivo SAM/BAM.

Site: <<http://samtools.sourceforge.net/>>

3.5.3 PICARD TOOLS

O *PICARD* é um programa em Java criado pelo Broad Institute que possui ferramentas de linha de comando para a manipulação de arquivos no formato SAM/BAM. Este programa permite ordenar e indexar arquivos SAM/BAM, remover leituras que forem duplicadas, extrair métricas do alinhamento entre muitas outras coisas tarefas.

Site: <<http://picard.sourceforge.net>>

3.5.4 Burrows-Wheeler Aligner

O programa *BWA* ([LI et al., 2009](#)) é um mapeador de leituras desenvolvido por Heng Li e Richard Durbin que utiliza a transformada de *Burrows-Wheeler* para indexar os genomas e depois realizar o processo de mapeamento das leituras contra um genoma de referência. Este alinhador foi publicado em 2009 e atualmente possui um método chamado *bwa mem* que é o mais utilizado para se alinhar leituras geradas por sequenciadores da empresa Illumina. Este alinhador foi utilizado para o alinhamento do primeiro exoma recebido pelo nosso laboratório.

Site: <<http://bio-bwa.sourceforge.net>>

3.5.5 BFAST

O *BFAST* ([HOMER; MERRIMAN; NELSON, 2009](#)) acrônimo de “BLAT-like Fast Accurate Search Tool” é um alinhador desenvolvido por Nils Homer da Universidade da Califórnia em Los Angeles. Este programa basicamente realiza o alinhamento das

leituras em duas etapas: Primeiro utilizando múltiplos índices de um genoma de referência ele identifica os locais de alinhamento candidatos para cada uma das leituras. Depois ele realiza um alinhamento com GAPs nos locais candidatos que foram identificados no passo anterior para definir aquele que possui a melhor correspondência com cada uma das leituras.

Este programa foi utilizado para a análise dos dados do sequenciador SOLID 5500XL e foi necessário a utilização de um *branch* do *BFAST* chamado de *BFAST+BWA* que foi recomendado pelo centro que realizou o sequenciamento dos dados, pois ele utiliza dois algoritmos diferentes para alinhar as leituras *forward* e *reverse* que possuem tamanhos diferentes 75 pb e 35 pb respectivamente. Site: <<http://bfast.sourceforge.net>>

3.5.6 GATK

O *Genome Analysis ToolKit* (*GATK*) (DEPRISTO et al., 2011) é um framework desenvolvido em Java pelo Broad Institute que possui diversos métodos utilizados para melhorar a qualidade da análise dos dados além de realizar a identificação das variantes gerando um arquivo VCF no final a partir dos arquivos SAM/BAM do alinhamento. Este programa foi utilizado para a análise de todos os dados recebidos pelo Laboratório de Genômica Clínica e também para realizar a análise dos dados do Projeto 1000Genomes.

Um diagrama do framework proposto pelo *GATK* para análise dos dados é apresentado na figura 11.

Site: <<http://www.broadinstitute.org/gatk/>>

3.5.7 IGV

O *IGV* é um Genome Browser desenvolvido pelo Broad Institute que permite a visualização dos dados genômicos de maneira integrativa permitindo a utilização de arquivos BAM e VCF para visualização de regiões genômicas. Este programa foi utilizado para investigação de algumas variantes candidatas identificadas por diferentes membros do nosso laboratório. Site: <<http://www.broadinstitute.org/igv/>>

GATK Framework de análise de variantes

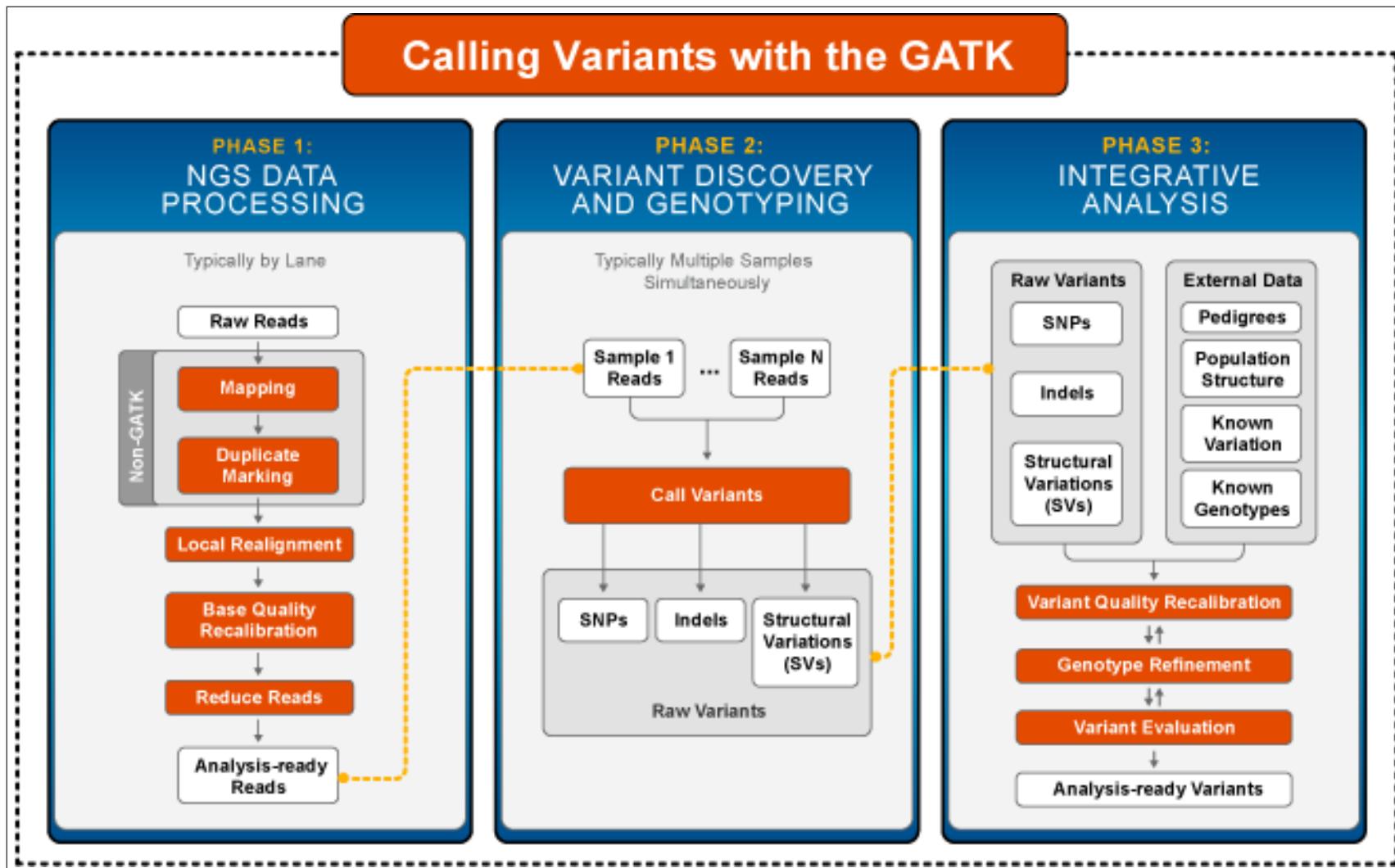


Figura 11 – Nesta figura podemos observar todas as etapas existentes do programa *GATK*. Source: <https://www.broadinstitute.org/gatk/>

3.6 Banco de Dados e Programas Utilizadas

3.6.1 1000Genomes

No site do projeto foi obtido um arquivo com 39.706.715 posições contendo 4 bilhões de genótipos que foram utilizadas como controle no processo de filtragem de variantes baseado em um limiar de frequência dessas variantes calculado sobre 2500 indivíduos.

Site: <<http://www.1000genomes.org/>>

3.6.2 Exome Variant Server

Esse projeto é importante pois já realizou o sequenciamento de mais de 6500 exomas de indivíduos. No site do projeto foi obtido um arquivo VCF com a frequência de 1.872.496 variantes, este arquivo é utilizado durante o processo de anotação de variantes para ajudar a filtrar variantes baseadas na sua frequência.

Site: <<http://evs.gs.washington.edu/>>

3.6.3 dbSNP

O dbSNP é um banco de dados público desenvolvido pelo National Center for Biotechnology Information (NCBI) que armazena informações sobre variações genéticas como SNPs, INDELS que foram encontradas em diferentes espécies como por exemplo: Humanos, Bovinos e Caninos. Os dados de humanos deste banco, atualmente na versão 142, foram utilizados ao longo deste projeto durante a análise dos dados e no processo de filtragem de variantes para ajudar a eliminar variantes comuns, por exemplo, baseado na frequência dessa variante que foi encontrada em um grupo de indivíduos.

Na tabela 2 apresentamos informações sobre o último release do dbSNP versão 137.

Podemos observar que existem atualmente 53,567,890 *reference clustered SNPIDs* (rsids), esses valores são ids únicos atribuídos a cada variante e são utilizados para anotação das frequências de cada variante.

Site: <<http://www.ncbi.nlm.nih.gov/projects/SNP/>>

Tabela 2 – Informações sobre o dbSNP 137

Organismo	Homo sapiens
dbSNP Build	137
Genome Build	37.4
Número de Submissões (ss#'s)	192.678.553
Número de RefSNP Clusters (rs#'s) (# validados)	53.567.890 (38.072.522)
Número de (rs#'s) em genes	22.450.743
Número de (ss#'s) com genótipo	75.115.460
Número de (ss#'s) com frequência	35.997.781

3.6.4 GATK Resource Bundle

O *GATK Resource Bundle* é um repositório de arquivos recomendados pelo *GATK* para serem utilizados durante a análise de genomas e exomas. Através deste repositório foram obtidos datasets com 292 exomas controles para serem utilizados no processo de filtragem de variantes.

Site: <<http://ftp.broadinstitute.org>>

3.6.5 OMIM - Online Mendelian Inheritance in Man

O Online Mendelian Inheritance in Man (OMIM) é um banco de dados público de genes associados a doenças Mendelianas. Este banco de dados em julho de 2015 possui 14.965 genes e 5.523 fenótipos associados a doenças Mendelianas. Essas informações foram obtidas e utilizadas para auxiliar na investigação de variantes candidatas. Por exemplo, o usuário pode buscar todos os genes associados a uma doença e visualizar as variantes desses genes presentes em cada um dos indivíduos inseridos no sistema.

Site: <<http://www.omim.org>>

3.6.6 HGNC - HUGO Gene Nomenclature Committee

Os dados sobre os genes incluídos no Mendel,MD foram obtidos a partir do site HUGO Gene Nomenclature Committee (HGNC) (GRAY et al., 2013) que é um banco de dados com “*gene symbols*” e nomes únicos para cerca de 37 mil regiões do genoma humano, sendo que mais de 19 mil dessas regiões são consideradas codificadoras de proteínas (genes).

A tabela 3 apresenta informações sobre os dados que foram obtidos no endereço <http://www.genenames.org/cgi-bin/hgnc_stats>.

Site: <<http://www.genenames.org>>

3.6.7 HGMD

O HGMD (STENSON et al., 2009) é um banco de dados comercial com mutações germinativas e nucleares que foram descritas e validadas pela literatura como sendo

Tabela 3 – Estatísticas sobre os genes de acordo com o HUGO
Gene Nomenclature Committee (HGNC)

Estatísticas sobre o Genome de Referência			
Grupo de Locus	Total por Grupo	Tipo	Total por Tipo de Locus
Gene codificador de proteína	19.067	gene com produto proteico	19.067
RNA não-codificador	4405	RNA Y	4
		cluster de RNA	125
		RNA longo não codificador	1690
		microRNA	1524
		RNA variado	3
		RNA ribossomal	34
		RNA pequeno citoplasmático	3
		RNA pequeno nuclear	71
		RNA pequeno nucleolar	414
		RNA de transferência	533
Fenótipos	703	apenas com fenótipos	703
Pseudogenes	11880	pseudogene receptor de células T	35
		pseudogene imunoglobulina	199
		pseudogene	11.646
Outros	1111	gene do receptor de células T	207
		constituinte de locus complexo	27
		retrovírus endógeno	73
		sítio frágil	117
		genes de imunoglobulina	224
		protocaderina	39
		leitura	86
		região	44
		elemento transponível	4
		desconhecidos	282
Total de símbolos aprovados			37.166

associadas a doenças humanas. Estes dados foram utilizados para permitir a busca de mutações que já estivessem descritas anteriormente pela literatura.

Site: <<http://www.hgmd.cf.ac.uk>>

3.6.8 Genoma Humano de Referência b37

Todas as análises deste trabalho foram realizadas usando a versão b37 do genoma humano, o mesmo arquivo que foi utilizado pelo projeto 1000genomes para suas análises e está disponível no repositório “*GATK Resource Bundle*” no FTP do Broad Institute. Ele seria o equivalente ao hg19 do genoma fornecido pelo UCSC.

A tabela 4 apresenta o número de pares de base em cada um dos 84 contigs referentes aos cromossomos além dos contigs que ainda não puderam ser incorporados nos cromossomos. Os contigs iniciados com GL (Genome Library) são sequências que ainda não puderam ser integradas aos contigs canônicos (ex. 1,2,3)

Site: <<ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/technical/reference/>>

3.6.9 Escore PHRED

O escore de qualidade chamado PHRED (Q) é um valor entre 4 e 60 que é dado para cada base de DNA que é lida de acordo com a sua qualidade utilizando uma escala negativa de probabilidade do log dos valores de qualidade. Esse valor é calculado de acordo com a seguinte fórmula:

$$Q = -10 \log_{10} P$$

Como exemplo, um phred escore de 10 significa que temos a probabilidade de 1 erro a cada 10 bases, ou seja 90% de acurácia para cada base que foi lida, já um Phred escore de 60 indica a probabilidade de erro de 1 em 1.000.000 milhão de bases ou seja 99.99999% de acurácia.

Site: <<http://www.phrap.com/phred/>>

Tabela 4 – Tabela com informações sobre o número de pares de bases de cada um dos cromossomos do genome build b37.

Contig	Pares de Base	Contig	Pares de Base
1	249.250.621	GL000208.1	92.689
2	243.199.373	GL000209.1	159.169
3	198.022.430	GL000210.1	27.682
4	191.154.276	GL000211.1	166.566
5	180.915.260	GL000212.1	186.858
6	171.115.067	GL000213.1	164.239
7	159.138.663	GL000214.1	137.718
8	146.364.022	GL000215.1	172.545
9	141.213.431	GL000216.1	172.294
10	135.534.747	GL000217.1	172.149
11	135.006.516	GL000218.1	161.147
12	133.851.895	GL000219.1	179.198
13	115.169.878	GL000220.1	161.802
14	107.349.540	GL000221.1	155.397
15	102.531.392	GL000222.1	186.861
16	90.354.753	GL000223.1	180.455
17	81.195.210	GL000224.1	179.693
18	78.077.248	GL000225.1	211.173
19	59.128.983	GL000226.1	15.008
20	63.025.520	GL000227.1	128.374
21	48.129.895	GL000228.1	129.120
22	51.304.566	GL000229.1	19.913
X	155.270.560	GL000230.1	43.691
Y	59.373.566	GL000231.1	27.386
MT	16.569	GL000232.1	40.652
GL000191.1	106.433	GL000233.1	45.941
GL000192.1	547.496	GL000234.1	40.531
GL000193.1	189.789	GL000235.1	34.474
GL000194.1	191.469	GL000236.1	41.934
GL000195.1	182.896	GL000237.1	45.867
GL000196.1	38.914	GL000238.1	39.939
GL000197.1	37.175	GL000239.1	33.824
GL000198.1	90.085	GL000240.1	41.933
GL000199.1	169.874	GL000241.1	42.152
GL000200.1	187.035	GL000242.1	43.523
GL000201.1	36.148	GL000243.1	43.341
GL000202.1	40.103	GL000244.1	39.929
GL000203.1	37.498	GL000245.1	36.651
GL000204.1	81.310	GL000246.1	38.154
GL000205.1	174.588	GL000247.1	36.422
GL000206.1	41.001	GL000248.1	39.786
GL000207.1	4.262	GL000249.1	38.502

3.6.10 Escore SIFT

O SIFT (NG; HENIKOFF, 2003) é um escore criado para predizer o impacto causado na função da proteína por mudanças na sua sequência de aminoácidos. Esse escore é bastante utilizado para ajudar a identificar boas variantes candidatas. Esse algoritmo utiliza homologia de sequências para predizer se uma substituição do aminoácido pode causar uma mudança na estrutura de uma determinada proteína. Apesar de seu primeiro artigo ter sido publicado em 2001 ainda é bastante utilizado na análise de exomas.

<http://sift.jcvi.org/>

3.6.11 Escore Polyphen-2

Esse é um escore de patogenicidade que também é bastante utilizado, porém sua diferença em relação ao SIFT é que quanto maior for o escore mais patogênica tende a ser a variante estudada (ADZHUBEI; JORDAN; SUNYAEV, 2013). Portanto, valores próximos de 1.0 seriam mais patogênicos de acordo com esse algoritmo.

<http://genetics.bwh.harvard.edu/pph2/>

3.6.12 Escore CADD

O CADD (*Combined Annotation Dependent Depletion*) é um escore que integra múltiplas anotações em uma única métrica utilizando mutações que sobrevivem a um processo de seleção natural aplicado sobre mutações simuladas ao longo do genoma (KIRCHER et al., 2014). Esse escore pode ser utilizado para encontrar mutações patogênicas, porém ao contrário dos escores de SIFT e Polyphen-2, ele não possui valores sugeridos para determinar se uma variante é ou não é patogênica, quanto maior o seu valor maior a chance da variante ser patogênica.

<http://cadd.gs.washington.edu/>

3.6.13 *SnpEff*

O *SnpEff* é uma ferramenta para anotação e predição do efeito causado por variantes presentes em diversas regiões de um genoma. Além de realizar a anotação de SNPs

e indels que estão localizados nos genes, ele também é capaz de predizer mutações em regiões de *splicing*, regiões com *frameshifts*, perda ou ganho de função entre outras que estão descritas no site da ferramenta.

Na tabela 5 podemos observar que a maioria dos efeitos das variantes que podem causar problemas pertencem as classes HIGH ou MODERATE classificadas por este programa. Estas duas categorias podem ser utilizadas como critério de filtragem durante a análise de exomas para identificação de possíveis genes e mutações candidatas.

Site: <<http://snpeff.sourceforge.net>>

3.6.14 ANNOVAR

O ANNOVAR ([WANG; LI; HAKONARSON, 2010](#)) é um software de anotação genômica que utiliza informações atualizadas de bancos de dados como o 1000Genomes, dbSNP137 e ESP6500 além de ferramentas como SIFT, Polyphen-2, Phylop e Mutation Taster de uma maneira integrada e rápida para anotar arquivos no formato VCF através de scripts desenvolvidos em Perl.

Site: <<http://www.openbioinformatics.org/annovar>>

3.6.15 dbNSFP

O dbNSFP ([LIU; JIAN; BOERWINKLE, 2011](#)) é um banco de dados que integra anotações funcionais de diversos algoritmos de predição de variantes e possui cerca de 75 milhões de SNPs não-sinônimas em sua base de dados. Ele utiliza escores de diferentes ferramentas de predição (Ex. SIFT, Polyphen-2, LRT, MutationTaster e MutationAssessor) e escores de conservação (PhyloP, GERP++ e SiPhy) para o processo de anotação de variantes.

Site: <<http://sites.google.com/site/jpopgen/dbNSFP>>

Tabela 5 – Tabela com as classes de impacto do programa SnpEff

Impacto	Termo Ontológico
HIGH	chromosome_number_variation
HIGH	exon_loss_variant
HIGH	frameshift_variant
HIGH	rare_amino_acid_variant
HIGH	splice_acceptor_variant
HIGH	splice_donor_variant
HIGH	start_lost
HIGH	stop_gained
HIGH	stop_lost
HIGH	transcript_ablation
MODERATE	3_prime_UTR_truncation +exon_loss
MODERATE	5_prime_UTR_truncation +exon_loss_variant
MODERATE	coding_sequence_variant
MODERATE	disruptive_inframe_deletion
MODERATE	disruptive_inframe_insertion
MODERATE	inframe_deletion
MODERATE	inframe_insertion
MODERATE	missense_variant
MODERATE	regulatory_region_ablation
MODERATE	splice_region_variant
MODERATE	TFBS_ablation
LOW	5_prime_UTR_premature_start_codon_gain_variant
LOW	initiator_codon_variant
LOW	splice_region_variant
LOW	start_retained
LOW	stop_retained_variant
LOW	synonymous_variant
MODIFIER	3_prime_UTR_variant
MODIFIER	5_prime_UTR_variant
MODIFIER	coding_sequence_variant
MODIFIER	conserved_intergenic_variant
MODIFIER	downstream_gene_variant
MODIFIER	conserved_intron_variant
MODIFIER	exon_variant
MODIFIER	feature_elongation
MODIFIER	feature_truncation
MODIFIER	gene_variant
MODIFIER	intergenic_region
MODIFIER	intragenic_variant
MODIFIER	intron_variant
MODIFIER	mature_miRNA_variant
MODIFIER	miRNA
MODIFIER	NMD_transcript_variant
MODIFIER	non_coding_transcript_exon_variant
MODIFIER	non_coding_transcript_variant
MODIFIER	regulatory_region_amplification
MODIFIER	regulatory_region_variant
MODIFIER	TF_binding_site_variant
MODIFIER	TFBS_amplification
MODIFIER	transcript_amplification
MODIFIER	transcript_variant
MODIFIER	upstream_gene_variant

3.7 Workflow de Análise de Exomas

Para análise dos exomas recebidos foi necessário o desenvolvimento de um workflow utilizando os programas: *BWA*, *SAMTOOLS*, *Picard* e *GATK*.

Este pipeline foi inicialmente desenvolvido em Bash e executado no supercomputador do CENAPAD-MG sendo posteriormente transformado em um script em python para permitir a sua execução no servidor do laboratório, facilitando a integração com os outros scripts desenvolvidos.

A seguir apresentamos um diagrama com o nosso pipeline implementado de acordo com o guia de melhores práticas oferecido pelo site do *GATK* no endereço: <<http://www.broadinstitute.org/gatk/guide/article?id=15>>

Na figura 12 podemos visualizar um diagrama com todas as etapas e os processos utilizados neste pipeline:

Além do *GATK* outros programas que aceitam arquivos BAMs como entrada foram utilizados para identificação de CNVs e Short Tandem Repeats (STRs) ([GYMREK et al., 2012](#); [KRUMM et al., 2012](#); [LI et al., 2012](#)).

3.7.1 Alinhamento dos dados de SOLID 5500xl

Para dos dados do sequenciador SOLID 5500xl foi necessário o desenvolvimento de um novo pipeline utilizando uma versão modificada do alinhador *BFAST* que possui uma implementação do *BWA* para alinhar leituras curtas. As leituras deste sequenciador possuem tamanhos diferentes sendo 75 pb para a sequência forward e 35 pb para a sequência reverse, portanto cada uma precisa ser alinhada com um algoritmo diferente. Este alinhador foi recomendado pelo Sick Children Hospital de Toronto que é o lugar onde os dados foram gerados.

Diagrama com o pipeline de análise de exomas

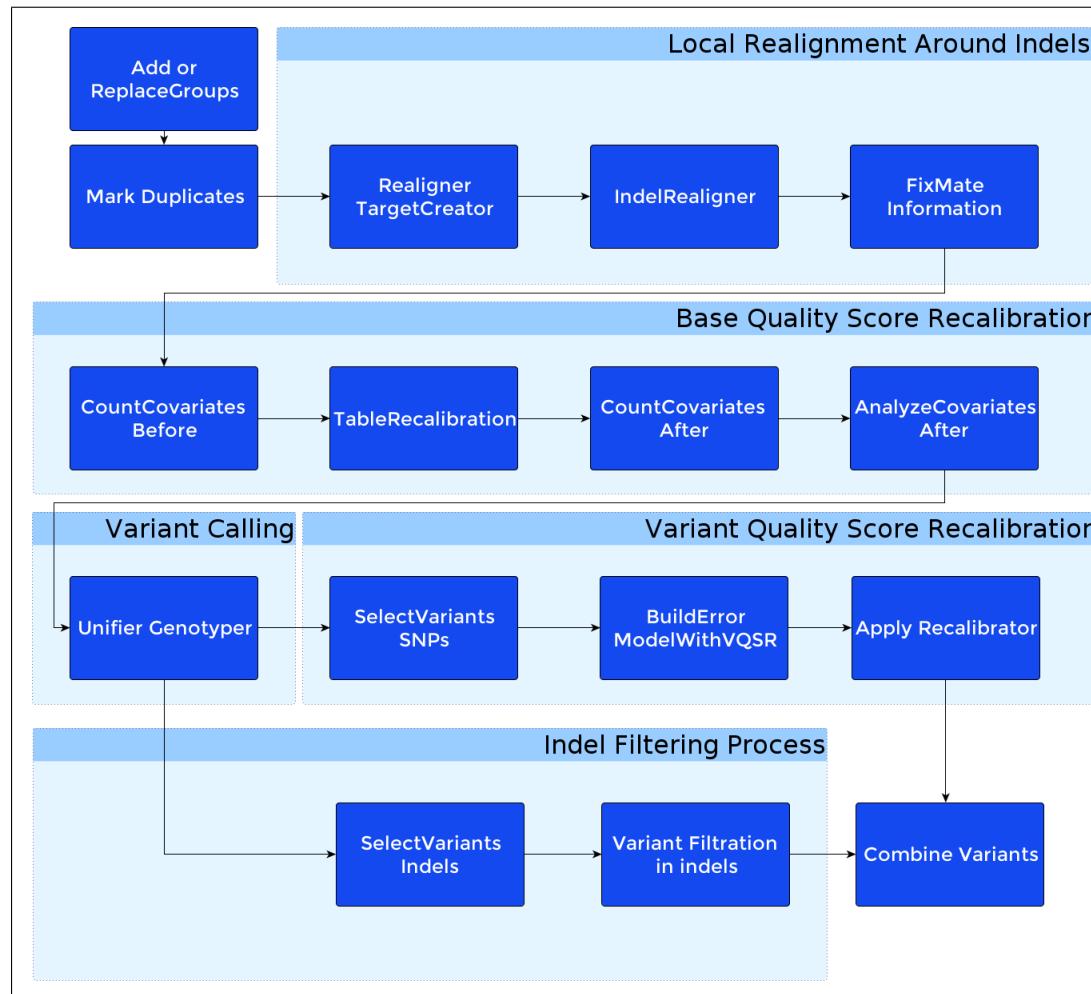


Figura 12 – Pipeline desenvolvido para realizar a análise de exomas utilizando o *GATK*. Nesta figura é possível observar que existem subetapas dentro de cada etapa proposta pelo GATK.

4 Resultados

4.1 Alinhamento e Análise de Exomas

No dia 6 de outubro de 2011 foi recebido o primeiro exoma para ser analisado pelo Laboratório de Genômica Clínica (LGC) na Faculdade de Medicina da UFMG em Belo Horizonte.

O sequenciamento desse exoma foi realizado utilizando um sequenciador modelo Illumina HiSeq2000, utilizando o kit de enriquecimento para captura das regiões exônicas desenvolvido pela Roche NimbleGen versão SeqCap EZ Human Exome Library v2 (44.1 Mbp) e foi dada uma garantia de cobertura mínima de pelo menos 30 vezes (30X) pela empresa Otogenetics que realizou o sequenciamento do DNA deste paciente.

Após o recebimento dos dois arquivos em formato FASTQ.GZ nós utilizamos o programa FASTQC para verificar a quantidade de leituras e obter métricas em relação aos valores de qualidade das leituras deste indivíduo.

4.1.1 Controle de Qualidade sobre os dados

Esses dois arquivos FASTQ continham 35.701.713 *leituras* cada um, totalizando 71.403.426 de “*leituras paired-end*”, essas leituras estavam codificadas com um escore phred de qualidade no formato Illumina 1.5+ e tiveram que ser convertidas para o formato Sanger de qualidade para se tornarem compatíveis com o programa GATK. Cada leitura possui 90 nucleotídeos de tamanho tanto para as sequências *forward* quanto as *reverse* e o conteúdo médio GC dessas sequências foi de 44%. Este valor está próximo do valor de 50% que seria o esperado para as regiões exônicas do genoma humano. Um valor muito alto ou muito baixo de GC poderia indicar que houve algum tipo de problema com o enriquecimento dessas regiões.

Na tabela 6 são apresentadas informações sobre as sequências forward e reverse que foram obtidas utilizando o programa FASTQC. Esses arquivos é o que chamamos de

dados brutos (“*raw data*”), pois são os dados que saíram diretamente do sequenciador e ainda não foram analisados ou alterados por nenhum tipo de programa.

A seguir são apresentadas imagens com informações sobre a qualidade das leituras. Podemos observar que esses dados estão com um valor ótimo de qualidade e não possuem nenhum desvio significativo em relação aos valores que seriam esperados.

Na figura 13 podemos observar que a qualidade das bases ao longo das leituras possui em média um *phred escore* acima de 30 e notamos que a partir do meio da leitura até o seu final existe uma leve redução dos valores de qualidade, isto geralmente acontece com dados de sequenciadores Illumina.

A figura 14 mostra a qualidade média das leituras e podemos observar que os valores de qualidade estão entre 37 e 38 tanto para a sequência forward quanto para a sequência reverse. Este resultado pode ser considerado bom, caso contrário esta imagem poderia indicar algum tipo de problema ocorrido durante o sequenciamento.

Na figura 15 apresentamos a distribuição dos nucleotídeos A, C, T e G ao longo das leituras e podemos observar que a quantidade de nucleotídeos A e T foi maior do que a quantidade de nucleotídeos G e C. Isso é esperado e pode ser considerado normal desde que a diferença entre esses dois grupos não ultrapasse 20%.

Na figura 16 podemos observar a distribuição do conteúdo GC para todas as sequências e que esse valor foi em média 44%. A linha em azul representa o modelo teórico de distribuição que seria esperado. Podemos afirmar que o valor real aproxima-se bastante do valor esperado e caso estas distribuições fossem diferentes isso poderia indicar um problema de contaminação da amostra com outros organismos.

Na figura 17 apresentamos os níveis de duplicação das sequências e podemos observar valores de duplicação de 39.79% e 38.73% para as sequências, o que pode ser considerado baixo. Um alto nível de duplicação indica que houve um enriquecimento de certas sequências, Um baixo nível de duplicação indica uma alta cobertura das regiões presentes no alvo.

Na figura 18 apresentamos os perfis de kmer das 6 sequências mais comuns com tamanho 5 nucleotídeos (5mer) ao longo das leituras. Podemos afirmar que não houve um enriquecimento relativo muito alto entre os 6 kmers mais frequentes das sequências.

Tabela 6 – Métricas sobre as Sequências *forward* e *reverse* do indivíduo RMS

(a) Forward		(b) Reverse	
Medida	Valor	Medida	Valor
Tipo de Arquivo	Conventional base calls	Tipo de Arquivo	Conventional base calls
Codificação	Illumina 1.5	Codificação	Illumina 1.5
Total de Sequências	35.701.713	Total de Sequências	35.701.713
Sequências Filtradas	0	Sequências Filtradas	0
Tamanho das Sequências	90	Tamanho das Sequências	90
%GC	44	%GC	44

Escore *Phred* de qualidade médio por base das sequências do indivíduo RMS

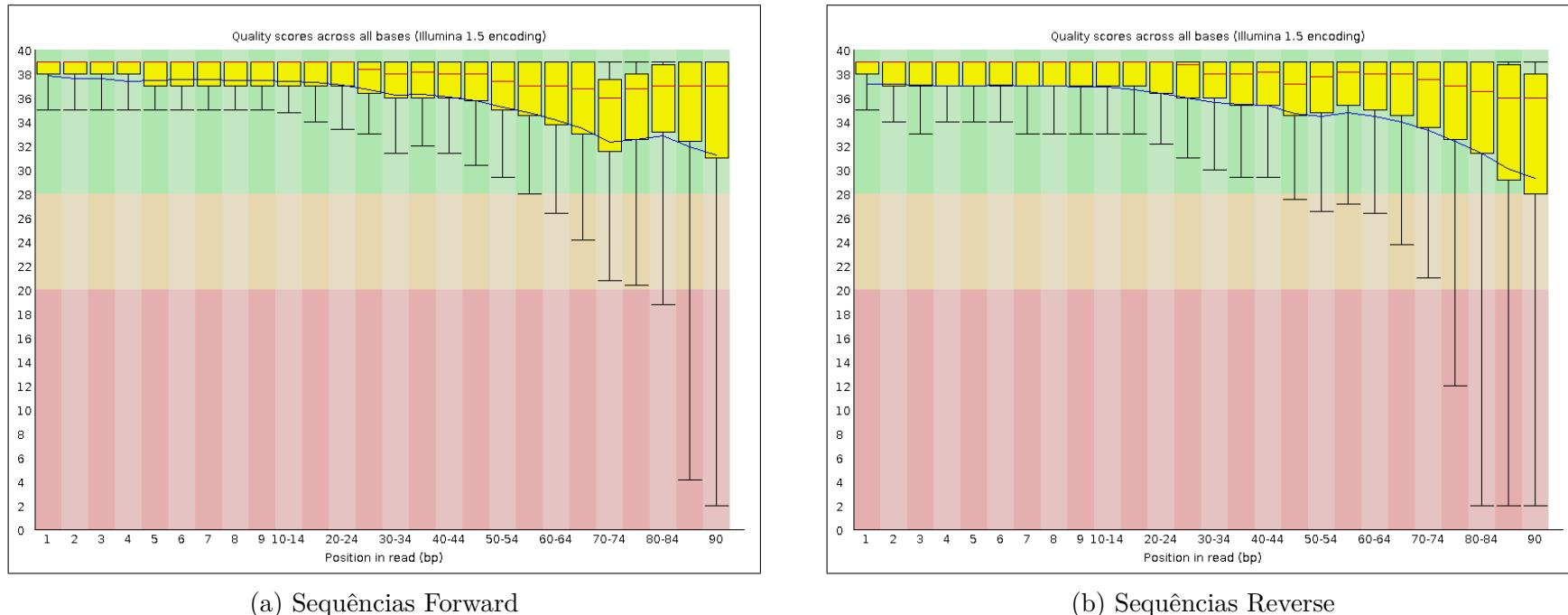
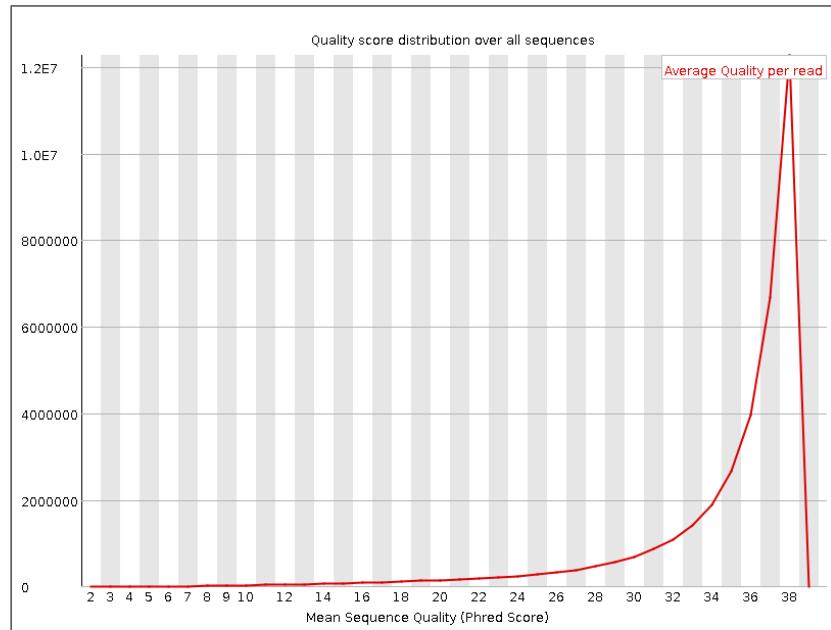
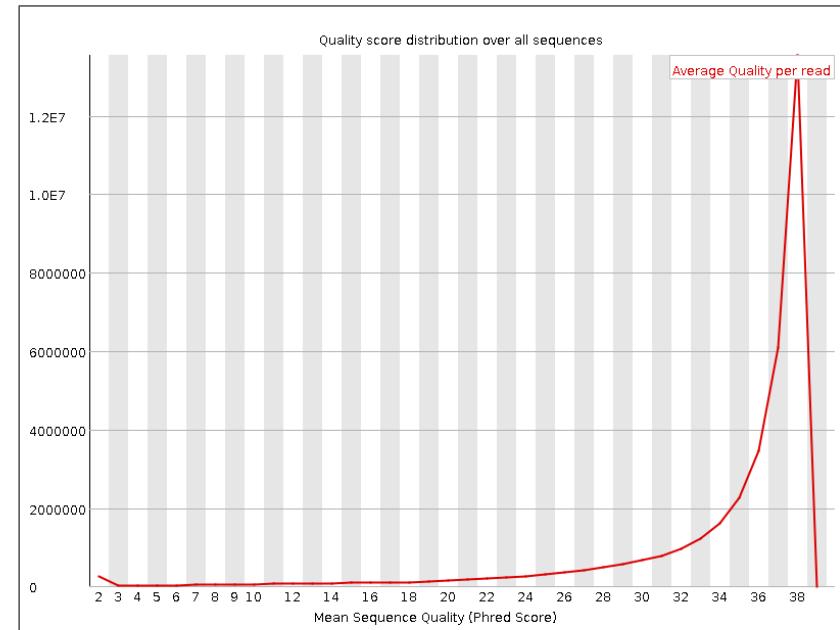


Figura 13 – Podemos observar que o score phred de qualidade das bases ficou sempre acima de 30, havendo uma pequena degradação no final das reads, o que é bastante característico de dados provenientes da plataforma illumina.

Distribuição do Escore *Phred* em relação a todas as sequências do indivíduo RMS



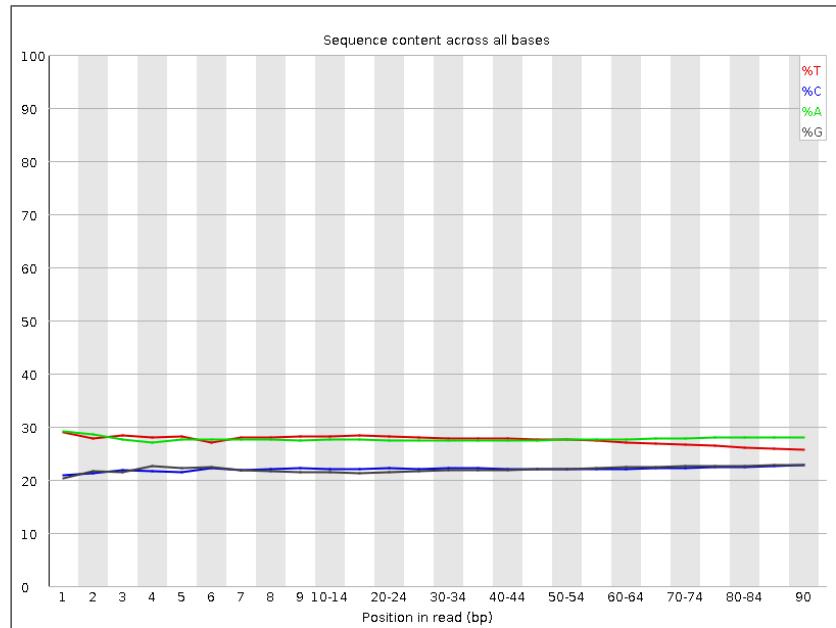
(a) Sequências Forward



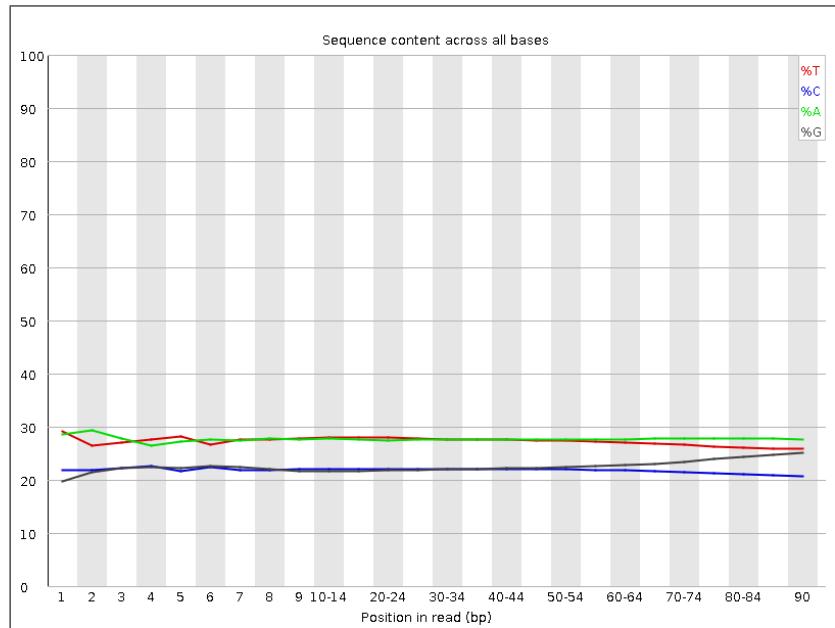
(b) Sequências Reverse

Figura 14 – Podemos observar que o phred escore médio de qualidade para as reads forward e reverse foram de 38.

Conteúdo de sequência por base para as sequências do indivíduo RMS



(a) Sequências Forward



(b) Sequências Reverse

Figura 15 – Nesta figura podemos observar a distribuição de cada uma das bases A, C, T, e G ao longo das reads.

Caso o enriquecimento de um kmer fosse por exemplo 3X maior em relação aos outros isto poderia indicar um problema com o sequenciamento.

Antes de realizar o alinhamento foi necessário converter o escore de codificação das sequências do formato Illumina 1.5 para o formato Sanger utilizando um script em perl fornecido pelo programa MAQ ([<http://maq.sourceforge.net/fq_all2std.pl>](http://maq.sourceforge.net/fq_all2std.pl)). Esta conversão é necessária para a análise dos dados utilizando o GATK.

4.1.2 Alinhamento

Após verificação da qualidade dos dados recebidos, foi feito um alinhamento utilizando o alinhador BWA que mapeou as leituras contra a versão b37 do genoma humano de referência.

Ao final deste mapeamento foi obtido um arquivo BAM como resultado e então foram extraídas métricas de qualidade sobre este alinhamento com o programa samstat. Este programa foi útil para ajudar a verificar métricas sobre a qualidade do alinhamento em arquivos do tipo BAM antes e depois de utilizarmos o GATK para melhorar a qualidade e corrigir alguns problemas nesses arquivos.

A seguir é apresentado um resumo do que aconteceu com os dados durante a execução do GATK dentro do pipeline de análise de exomas desenvolvido por este trabalho e apresentado na figura 12 na parte de Métodos.

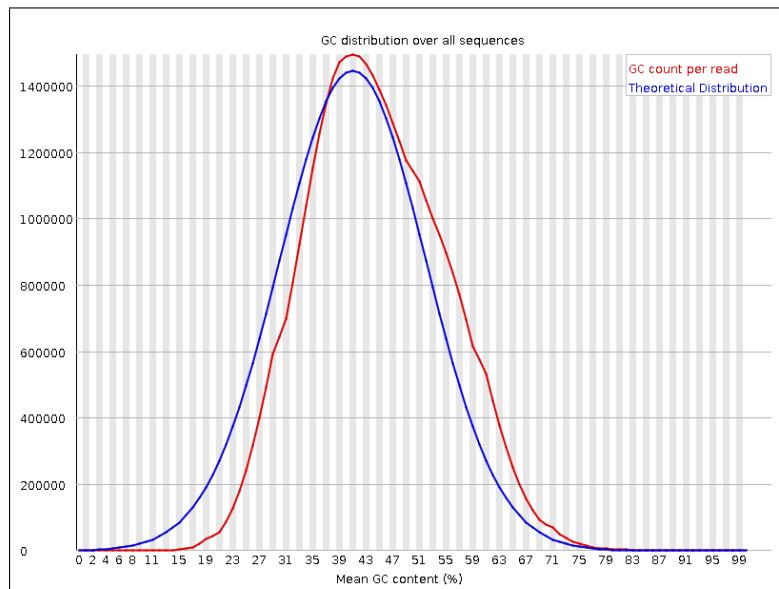
4.1.3 GATK Genome Analysis ToolKit

Após o alinhamento com o BWA nós utilizamos o programa GATK para melhorar a qualidade das leituras e realizar o processo chamado de “*calling das variantes*”. Para calcular a cobertura média do exoma, foi utilizado o programa bedtools¹ que procura por regiões definidas pelo kit de sequenciamento utilizado. O arquivo BED com os targets para este exoma foram obtidos no link: <http://www.nimblegen.com/products/seqcap/ez/v2/>

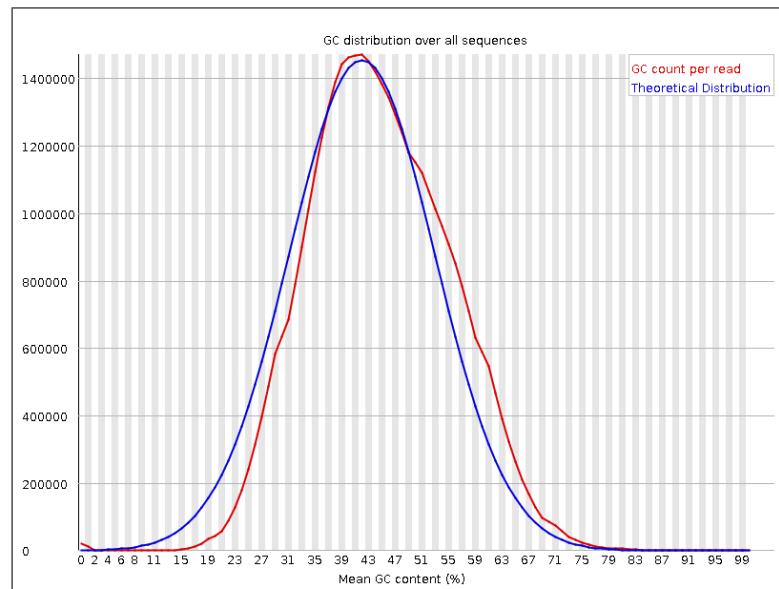
Nas figuras 19 e 20 podemos observar um sumário sobre a qualidade do alinhamento das leituras antes de utilizarmos o GATK. Tivemos aproximadamente 71.4 milhões

¹ [<https://github.com/arq5x/bedtools>](https://github.com/arq5x/bedtools)

Distribuição do conteúdo de GC nas leituras



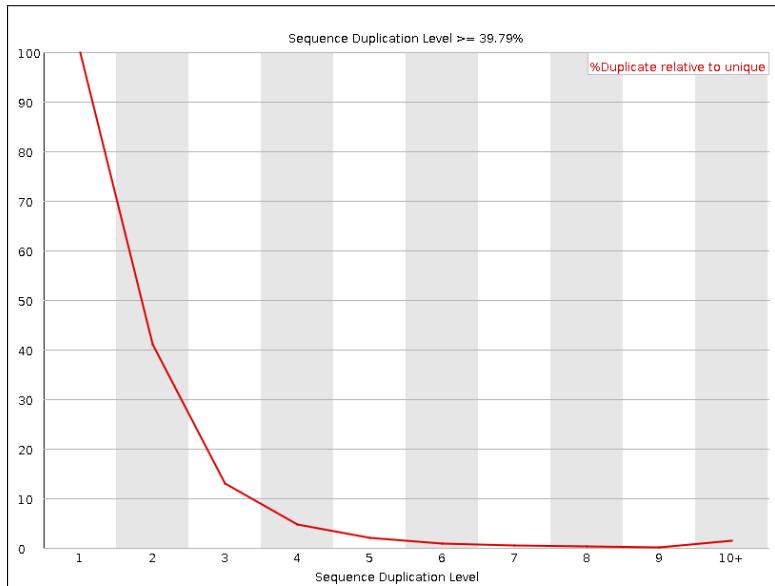
(a) Sequências Forward



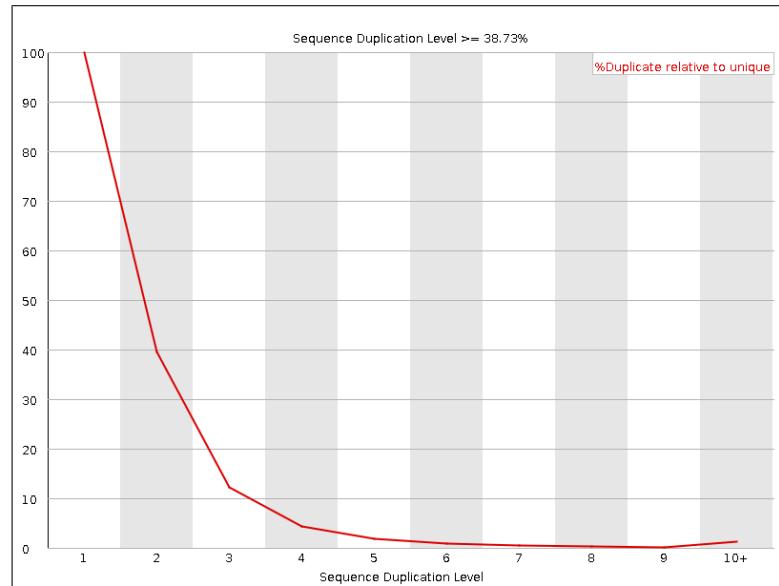
(b) Sequências Reverse

Figura 16 – Nesta figura podemos observar a distribuição do conteúdo GC ao longo das leituras.

Níveis de duplicação



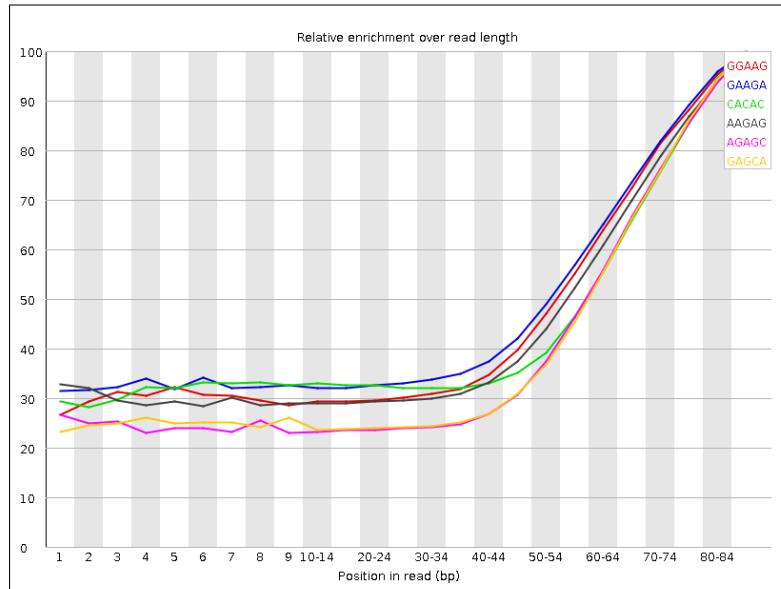
(a) Sequências Forward



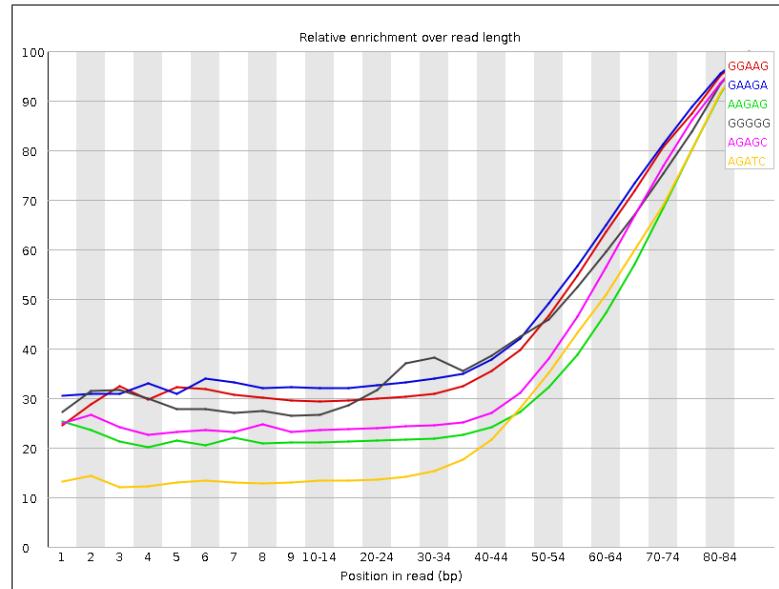
(b) Sequências Reverse

Figura 17 – Nesta figura podemos observar os níveis de duplicação. Uma quantidade alta do nível de duplicação pode indicar um problema com o sequenciamento.

Perfis de K-mer



(a) Sequências Forward



(b) Sequências Reverse

Figura 18 – Nesta figura podemos observar o enriquecimento relativo do número de kmers ao longo das leituras, um desbalanceamento grande de um kmer específico pode indicar um problema com o sequenciamento.

de leituras mapeadas e além disso 75% dessas leituras tiveram um escore de alinhamento MAPQ acima de 30 o que pode ser considerado um resultado muito bom para este tipo de análise.

Nas figuras 21 e 22 apresentamos um sumário sobre o alinhamento depois que o GATK foi utilizado. Embora o número de leituras tenha diminuído em relação a figura anterior, nota-se um aumento no valor da qualidade na região final das leituras presentes.

4.1.3.1 Remoção de leituras duplicadas

Primeiro foi utilizado o comando `MarkDuplicates` para remover leituras duplicadas em nosso arquivo de alinhamento, durante este processo 17% das leituras foram consideradas duplicadas e foram descartadas. Este passo é importante pois ajuda a reduzir os erros na hora de realizarmos o “*calling das variantes*”. Esta remoção não diminui a cobertura do sequenciamento mas ajuda a melhorar a qualidade das áreas que foram cobertas. Ao final restaram 59.570.255 milhões de leituras passaram para a próxima etapa.

Na tabela 7 apresentamos um resumo detalhado das leituras contidas no arquivo de alinhamento após este processo.

4.1.3.2 Realinhamento Local em regiões de Indels

4.1.3.2.1 Criação dos alvos para realinhamento

Este método realiza uma busca ao longo do alinhamento por regiões que possuam características específicas como SNPs presentes no final das leituras que sejam discordantes para as leituras *forward* e *reverse* ou que estejam em regiões com repetição de nucleotídeos, por exemplo (TTTTT). Então, ele marca estas regiões para que um realinhamento local seja realizado utilizando o método de Smith–Watermann com o objetivo de corrigir possíveis erros gerados durante o alinhamento dessas sequências e encontrar indels nesta região.

A seguir apresentamos o resultado da saída deste método:

- Tempo de execução 5719.67 segundos, 95.33 minutos, 1.59 horas.

Estatísticas do alinhamento antes de utilizarmos o GATK

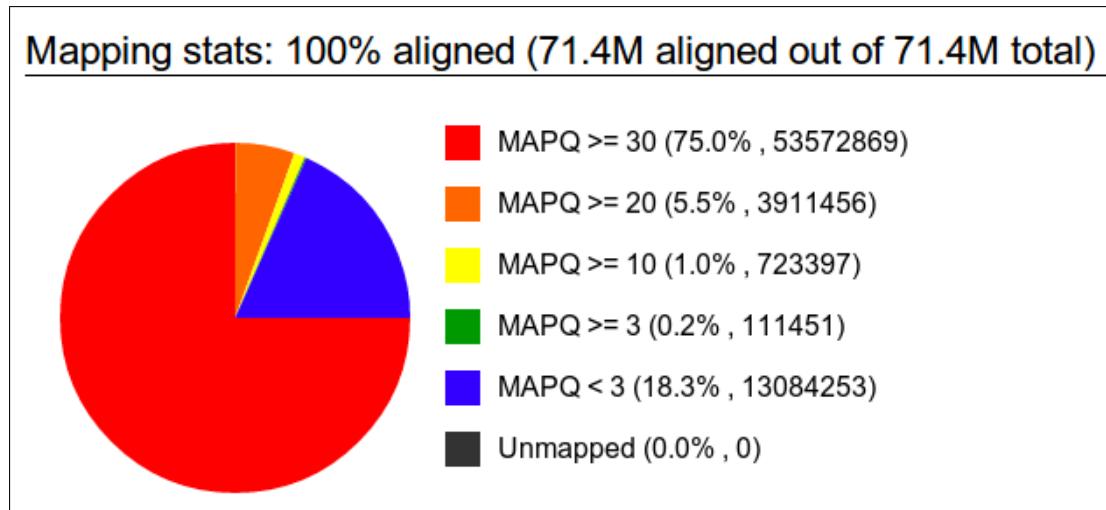


Figura 19 – Nesta figura podemos observar que o alinhamento obteve mais de 75% das leituras com escore de phred maior do que 30.

Qualidade média por base do alinhamento antes de utilizarmos o GATK

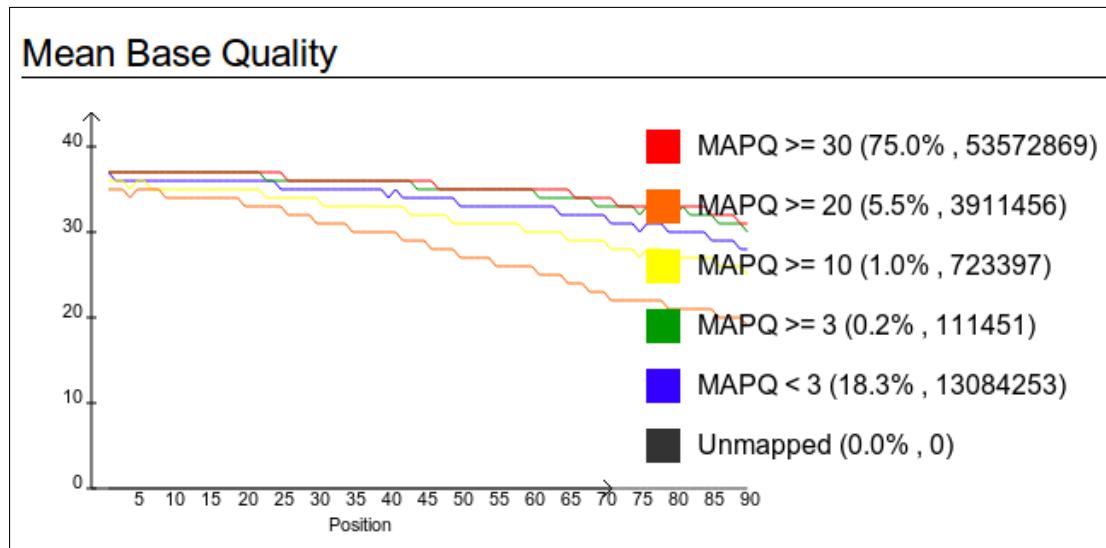


Figura 20 – Nesta figura podemos observar a distribuição da qualidade ao longo das bases para cada grupo de leituras do alinhamento.

Estatísticas do alinhamento depois de utilizarmos o GATK

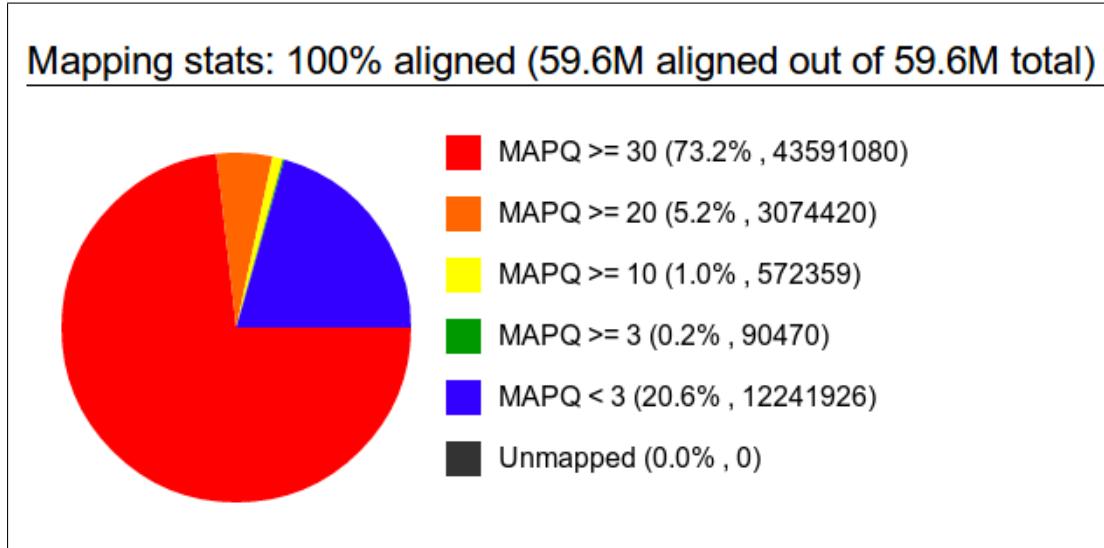


Figura 21 – Nesta figura podemos observar uma redução do número total de leituras devido a remoção de leituras com baixa qualidade dentro de cada um dos grupos.

Qualidade média por base do alinhamento depois de utilizarmos o GATK

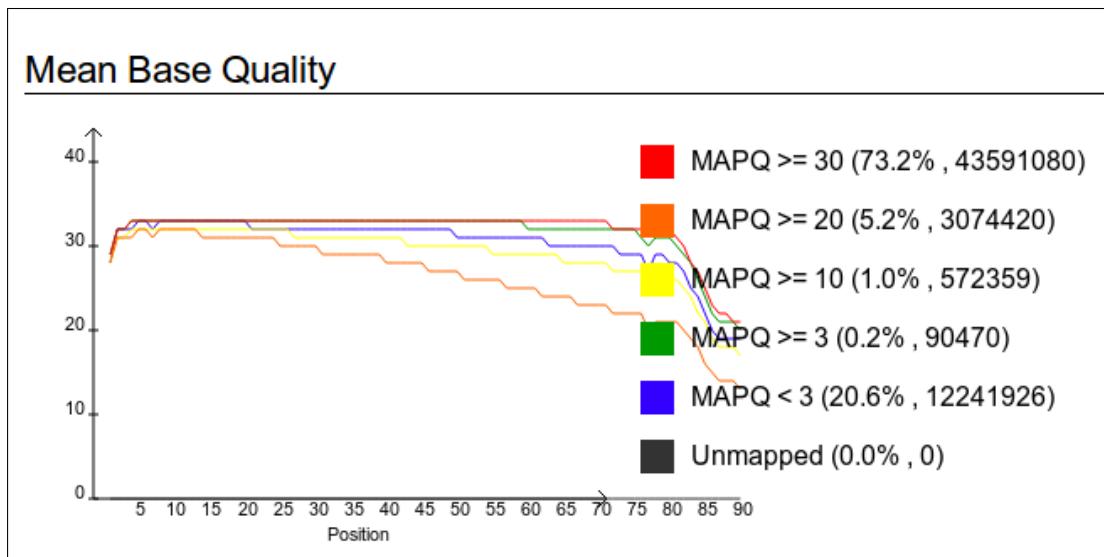


Figura 22 – Nesta figura podemos observar uma estratificação maior da qualidade das leituras após a filtragem das leituras com baixa qualidade.

Tabela 7 – Informações sobre o alinhamento após a remoção de leituras duplicadas

no total (leituras que passaram no controle de qualidade)	59.570.255
mapeadas (84.31% do total)	50.222.733
pareadas no sequenciamento	59.570.255
leitura1	29.756.446
leitura2	29.813.809
corretamente pareadas (80.78%)	48.122.446
com ela própria e com a companheira mapeada	49.778.115
únicas (0.75%)	444.618
com a leitura companheira mapeada em um cromossomo diferente	112.890
com a leitura companheira mapeada em um cromossomo diferente ($\text{mapQ} \geq 5$)	66.095

- 3.611.533 leituras foram filtradas durante o processo de um total de 51.194.954 (7,05 %).
- 3.611.499 leituras (7,05% do total) falharam o filtro de qualidade zero de mapeamento.
- 34 leituras (0,00% do total) falharam o filtro de leituras não mapeadas.

4.1.3.2.2 Aplicação da Recalibração

Este método realiza um realinhamento local nas leituras selecionadas pelo passo anterior para tentar melhorar a posição das leituras em relação a regiões com indels. Por exemplo, quando temos uma indel verdadeira que foi mapeada contra o genoma de referência, as leituras próximas a essa indel estarão mapeadas incorretamente se a região de início ou fim da leitura estiver próxima ao indel.

Após este processo o comando FixMate do programa Picard foi utilizado para remover as leituras órfãs, ou seja aquelas que o seu par tenha sido descartado durante este processo.

4.1.3.3 Recalibração do Escore de Qualidade da Base

Após o realinhamento de indels o GATK possui uma etapa de recalibração dos escores de qualidade. Este processo avalia a maneira como cada leitura varia em relação as outras leituras com o objetivo de deixar os valores mais uniformes e corrigir um possível viés de cada tecnologia.

A seguir apresentamos gráficos mostrando o que acontece com os dados antes e depois deste processo.

Na figura 23 apresentamos o antes e o depois para os valores de qualidade empíricos contra os valores reais estimados. Podemos notar que este método realiza uma normalização dos dados de maneira a corrigir um viés existente por causa das particularidades de cada tecnologia de sequenciamento.

Na figura 24 apresentamos a relação entre o número de bases e os valores de qualidade em phred escore. Neste gráfico podemos observar que a maior parte das bases

Escore de Qualidade Empírico vs Reportado

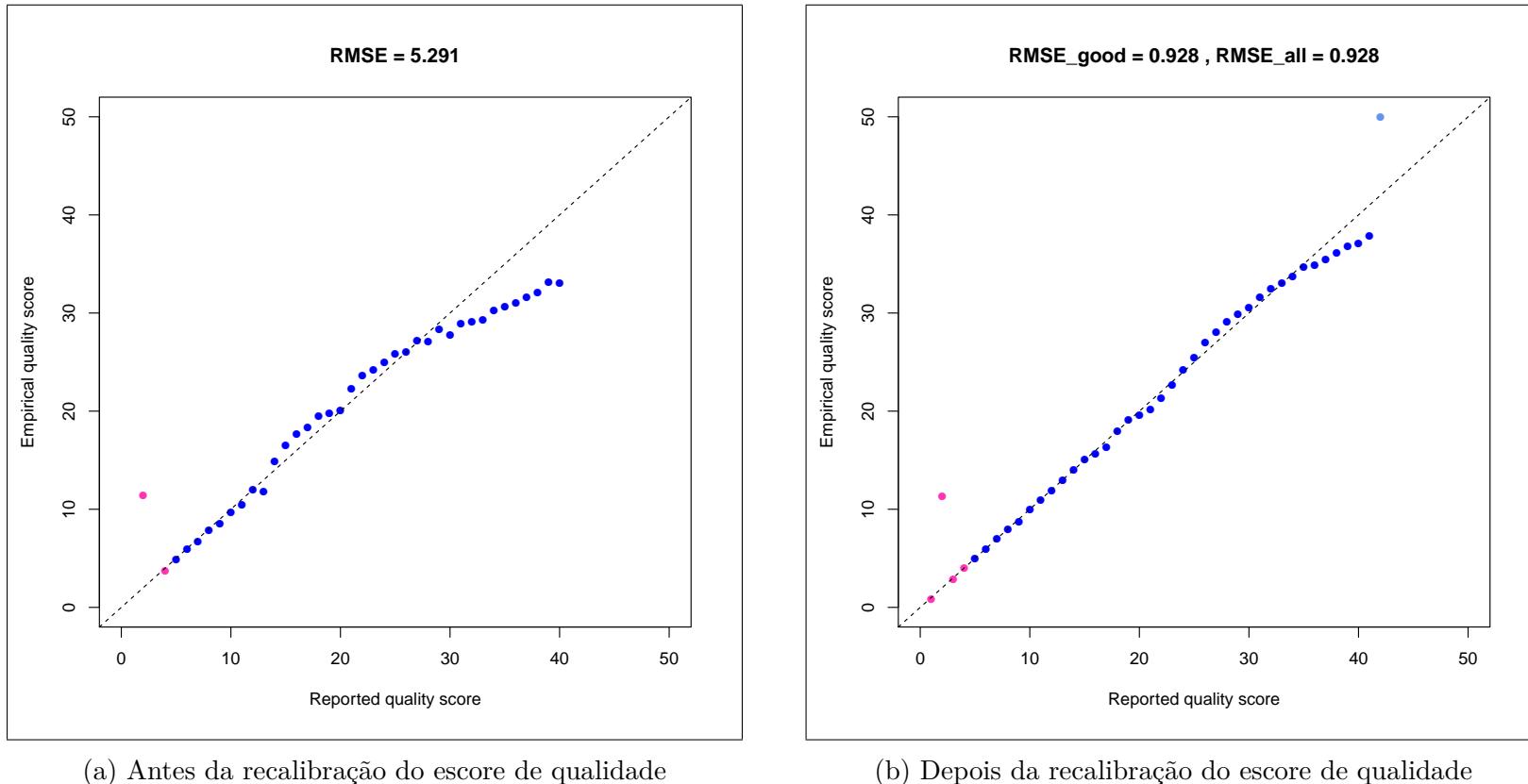


Figura 23 – Nesta figura podemos observar uma estratificação maior do escore de qualidade, este processo ajudar a corrigir um viés que existe no escore de qualidade de acordo com cada plataforma.

Número de Bases vs Escore de Qualidade

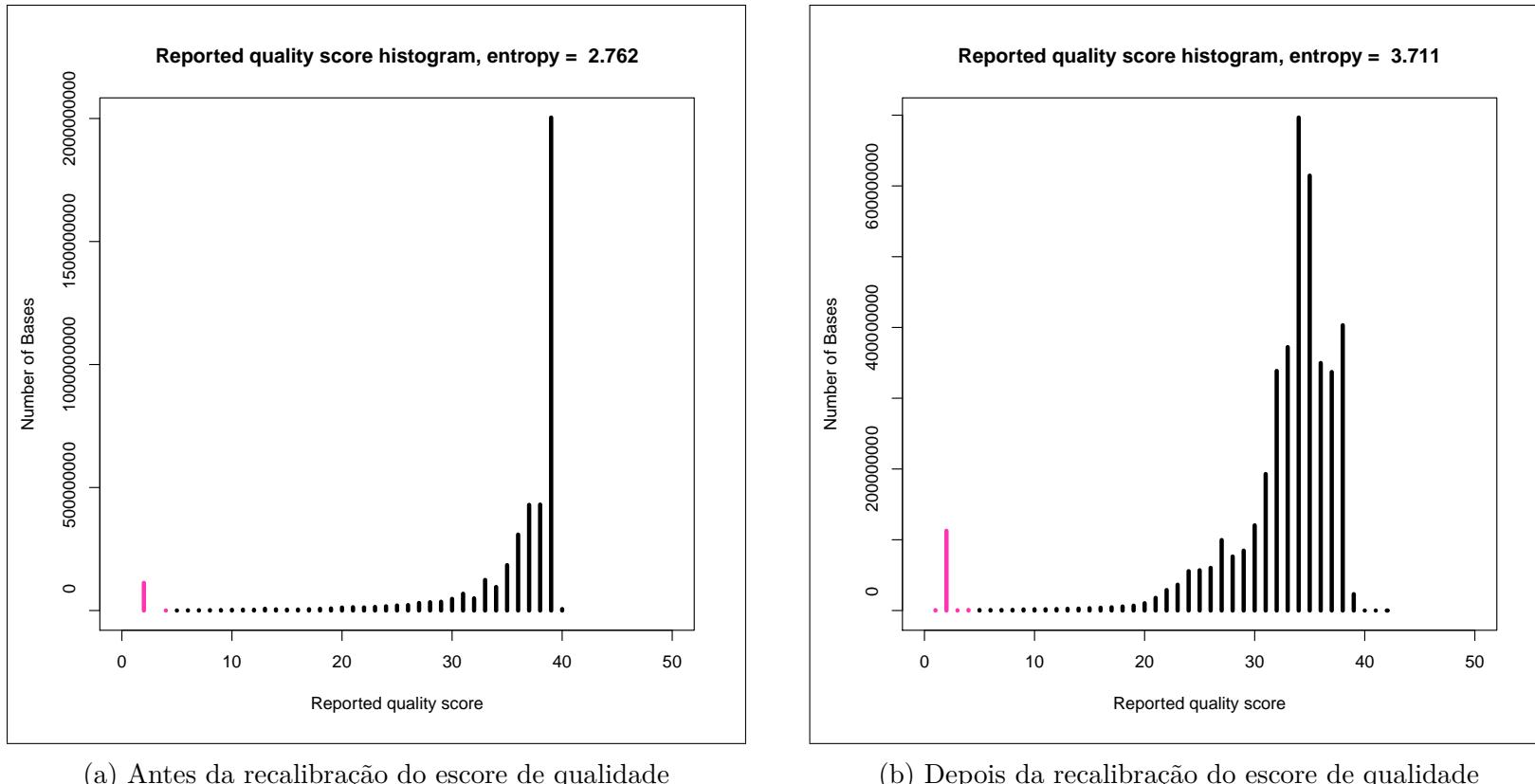
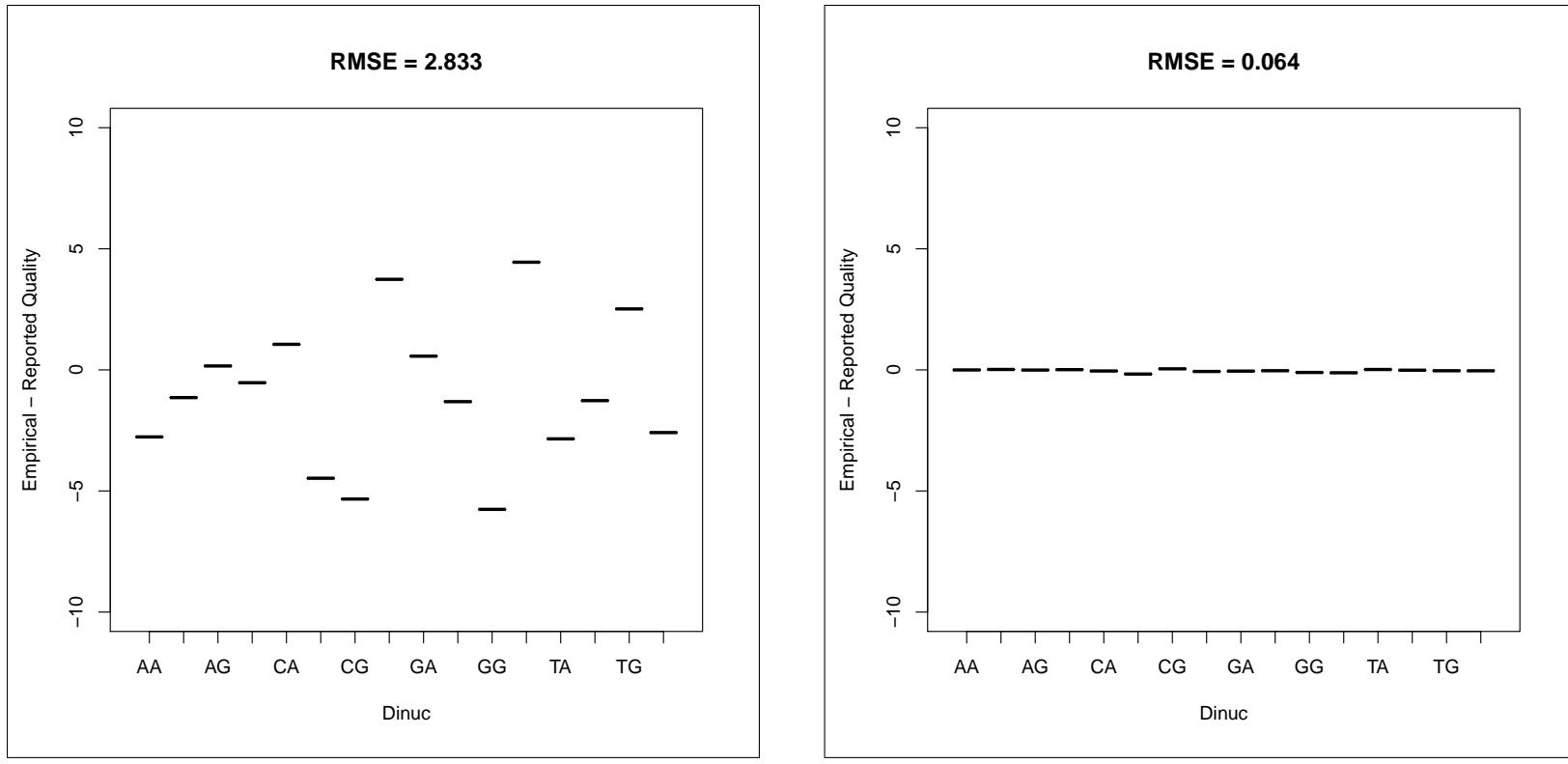


Figura 24 – Nessa figura podemos observar uma distribuição maior do valor médio do escore de qualidade.

Qualidade (Empírico - Reportado) vs Dinucleotídeos



(a) Antes da recalibração do escore de qualidade

(b) Depois da recalibração do escore de qualidade

Figura 25 – Nesta figura podemos observar uma correção da qualidade para cada dinucleotídeo.

estão com um phred escore entre 30 e 40 o que pode ser considerado um resultado muito bom.

Na figura 25 podemos notar um viés da relação entre a qualidade empírica e reportada para diferentes tipos de dinucleotídeos. Neste processo os valores são normalizados ao redor de 0, para ajudar a reduzir o número de falsos positivos no processo seguinte de calling de variantes.

4.1.3.4 Chamada de variantes

O processo de chamada “*calling*” de variantes foi realizado com o método UnifiedGenotyper do GATK considerado o padrão ouro atualmente para se fazer esse tipo de tarefa.

A tabela 8 apresenta um resumo dos resultados obtidos neste processo:

Podemos observar que tivemos um *calling* de 39.677 variantes em 47.195.691 bases visitadas no total, o que corresponde a 1.75% do genoma. De todas as bases cobertas 827.992 puderam ser genotipadas com um grau de confiança de 99.5%. A cobertura média desse exoma foi de 29.96X e o valor do escore de qualidade médio foi de 504.62.

4.1.3.5 Recalibração do Escore de Qualidade da Variante

Neste processo nós utilizamos um conjunto de dados com genótipos dos indivíduos do projeto HapMap 3 que foram validados utilizando arrays de SNPs e com isso criamos um modelo de treinamento sobre esses atributos. Este modelo é então aplicado no arquivo VCF para melhorar a qualidade das variantes obtidas e ajudar a reduzir o número de falsos positivos.

4.1.3.6 Processo de Filtragem de Indels

Este processo consiste na aplicação de alguns critérios de filtragem para eliminar indels de baixa qualidade, como por exemplo:

- QD < 2.0
- ReadPosRankSum < -20.0

Tabela 8 – Resultado do processo de calling de variantes realizado usando o método UnifiedGenotyper para o indivíduo RMS

Bases visitadas	47.195.691
Bases que podem ser chamadas	46.982.901
Bases chamadas com confiança	827.992
% de bases que podem ser chamadas de todas as regiões	99,549
% de bases que podem ser chamadas com confiança de todas as regiões	1,754
% de bases chamadas com confiança de todas as regiões	1,762
Chamadas realmente feitas (variantes realmente encontradas)	39.677

- InbreedingCoeff < -0.8
- FS > 200.0

Esses valores foram obtidos a partir do site do GATK em seu documento sobre “*Best Practices for Variant Detection*”

Site: <<https://www.broadinstitute.org/gatk/guide/best-practices>>

4.2 Anotação de Exomas

Após a obtenção do arquivo VCF final gerado pelo programa GATK, um script em python realiza a anotação dos dados utilizando os programas SNPEFF, Variant Annotator (GATK) e o Annovar integrando os resultados de cada um dos programas em um único arquivo VCF final. Após obtermos os resultados da anotação surgiu a ideia de construir uma ferramenta online que permitisse a filtragem das variantes por médicos e cientistas, de maneira que esta operação pudesse ser realizada através de uma interface web que permitisse a identificação de variantes que pudessem estar associadas com a doença.

4.3 Mendel,MD - Construção da Ferramenta

A seguir apresentamos o software Mendel,MD, que foi desenvolvido para investigação dos casos clínicos recebidos pelo Laboratório de Genômica Clínica da Faculdade de Medicina da UFMG. Esse programa foi criado para permitir o armazenamento, a anotação e a filtragem de variantes dos pacientes que foram estudados pelo nosso grupo, com a ideia de criar uma maneira fácil de atualizar os dados rapidamente, toda vez que novos conjuntos de dados, programas ou métodos fossem disponibilizados, de forma simples e modular, facilitando ao máximo a repetição das tarefas que fossem comuns. Este *pipeline* pode ser facilmente adaptado ou modificado para inclusão de novos métodos e novas ferramentas.

4.3.1 Banco de Dados

A modelagem do banco de dados foi feita através de um arquivo chamado *models.py* que possui classes em python que descrevem os campos que devem ser armazenados em cada uma das tabelas do banco de dados. Após a criação deste arquivo, o Django passa então a controlar os processos de criação e atualização desses campos. Além disso ele também fica responsável pela busca e remoção dos dados através uma técnica conhecida como Mapeamento de Objetos Relacionais (ORM) que facilita a criação de consultas em Python que são transformadas em SQL para se realizar consultas ao banco de dados. Isso é muito utilizado para filtrar as variantes de cada paciente de acordo com os parâmetros que forem escolhidos pelo médico ou pesquisador que estiver utilizando o Mendel,MD.

No anexo A.1 apresentamos o modelo que foi desenvolvido para armazenar as informações sobre cada indivíduo. Neste arquivo ficam armazenadas informações como por exemplo o nome de usuário que fez o upload do arquivo VCF, a data do upload, o nome completo do arquivo e algumas informações sobre o estado do arquivo dentro do sistema.

Após ser inserido no Mendel,MD, o arquivo VCF passa a ter três estados possíveis dentro do sistema: *new*, *annotated* e *populated*. O estado *new* indica que o arquivo acabou de ser enviado ao sistema e está na fila para ser anotado pelo nosso programa, o estado *annotated* significa que ele passou por todo o *pipeline* de anotação com sucesso e está pronto na fila para ser inserido no banco de dados e o estado *populated* indica que ele já foi inserido no banco de dados com sucesso e está pronto para ser analisado pelo usuário final.

No anexo B.1 apresentamos o modelo que foi desenvolvido para armazenar as informações sobre as variantes de cada indivíduo. Podemos observar que além dos campos já presentes no VCF, foram criados alguns campos para ajudar na filtragem de variantes como por exemplo a frequência da variante em relação a diferentes bancos de dados (ex. 1000genomesn, dbSNP e ESP6500) e alguns escores de patogenicidade como por exemplo SIFT e Polyphen-2 e CADD. Também podemos observar que alguns campos foram criados para armazenar as informações de duas ferramentas que foram integradas

no sistema SnpEff e VEP.

Para recuperarmos a partir do banco todas as variantes do primeiro indivíduo do nosso banco de dados usamos o seguinte código em Python (Django):

```
Variants.objects.all(individual_id=1)
```

Se quisermos obter todas as variantes desse indivíduo que são homozigóticas nós usamos o seguinte código:

```
Variants.objects.all(individual_id=1, variant_type="HOM")
```

Essa codificação dos campos em Python permite que eles sejam facilmente traduzidos para um código SQL compatível com o sistema gerenciador de banco de dados (SGBD) que estiver sendo utilizado pelo projeto, que no nosso caso é o banco PostgreSQL.

Os dados deste trabalho foram inicialmente armazenados em um banco MySQL e posteriormente migrados para um banco PostgreSQL. Isso aconteceu porque o número de registros armazenados na tabela de variantes ultrapassou 10 milhões e a consulta ao banco de dados começou a ficar muito lenta, por exemplo, quando muitos indivíduos fossem utilizados como controles durante o processo de análise e filtragem de variantes. Após a migração do banco nós obtivemos um aumento de desempenho considerável que ajudou a melhorar bastante a usabilidade do sistema.

Na tabela 9 nós apresentamos informações sobre o número de registros armazenados em cada uma das tabelas do banco de dados. Aqui é possível visualizar o número de indivíduos, genes, doenças e vias metabólicas que foram armazenados em cada uma das tabelas do banco. Esses dados são muito importantes para auxiliarem na filtragem de variantes de cada indivíduo.

Atualmente o nosso banco de dados possui 221 exomas e isso equivale a 25.304.952 variantes.

4.3.2 Dashboard

Para facilitar a visualização dos indivíduos no sistema foi desenvolvida uma interface chamada de *Dashboard* que exibe uma lista com todos os arquivos que foram enviados

Tabela 9 – Informação sobre o número de registros armazenados em cada uma tabelas do sistema.

Tabela	número de registros	Tabela	número de registros
account_emailaddress	8	djkombu_queue	3
account_emailconfirmation	8	filter_analysis_familyfilteranalysis	0
auth_group	0	filter_analysis_filteranalysis	0
auth_group_permissions	0	filter_analysis_filterconfig	0
auth_permission	150	genes_cgdcondition	3.607
auth_user	12	genes_cgentry	2.725
auth_user_groups	0	genes_cgentry_CONDITIONS	3.958
auth_user_user_permissions	0	genes_cgentry_INTERVENTION_CATEGORIES	3.635
cases_case	0	genes_cgentry_MANIFESTATION_CATEGORIES	7.319
cases_case_case_groups	0	genes_gene	37.215
cases_case_cases	0	genes_gene_diseases	5.725
cases_case_children	0	genes_genecategory	0
cases_case_control_groups	0	genes_genecategory_genes	0
cases_case_controls	0	genes_genegroup	0
cases_case_shared_with	0	genes_genelist	62
celery_taskmeta	712	genes_goterm	0
celery_tasksetmeta	0	genes_goterm_children	0
databases_varisnp	78.951	genes_goterm_parents	0
diseases_disease	6.845	genes_intervention	20
diseases_gene	4.715	genes_manifestation	19
diseases_gene_diseases	6.845	genes_membership	0
diseases_hgmdgene	0	individuals_controlgroup	0
diseases_hgmdgene_diseases	0	individuals_controlvariant	0
diseases_hgmdmutation	0	individuals_group	3
diseases_hgmdphenotype	0	individuals_group_members	76
django_admin_log	2	individuals_individual	221
django_content_type	50	individuals_individual_shared_with_groups	179
django_select2_keymap	0	individuals_individual_shared_with_users	0
django_session	511	individuals_usergroup	1
django_site	1	individuals_usergroup_members	2
djcelery_crontabschedule	0	pathway_analysis_pathway	289
djcelery_intervalschedule	0	socialaccount_socialaccount	0
djcelery_periodictask	0	socialaccount_socialapp	0
djcelery_periodictasks	0	socialaccount_socialapp_sites	0
djcelery_taskstate	0	socialaccount_socialtoken	0
djcelery_workerstate	0	variants_variant	25.304.952
djkombu_message	0		

para o sistema. Na figura 26 apresentamos essa interface e podemos observar que com ela é possível verificar diversas informações sobre cada indivíduo como o nome de cada arquivo, o número de variantes, a data de envio, o estado atual do arquivo no sistema entre outras informações. Também nesta página é possível realizar operações em massa como por exemplo, selecionar múltiplos indivíduos através de uma caixa de seleção ao lado de cada indivíduo pra poder enviar os arquivos para serem re-anotados ou re-inseridos no banco de dados sempre quando for necessário. Este tipo de interface facilita muito a manipulação e anotação dos VCFs sempre que houver novas informações para serem integradas na análise de exomas.

4.3.3 Upload de Genomas

A figura 27 apresenta a interface de submissão de indivíduos para o sistema. Essa interface foi desenvolvida utilizando a biblioteca de javascript chamada Jquery FileUpload e facilita o envio de arquivos VCFs para o sistema permitindo o upload simultâneo de indivíduos para o servidor usando para isso qualquer dispositivo (computador, tablet ou celular) que tenha acesso a internet.

4.3.4 Agendador de Tarefas

O Celery é um sistema agendador de tarefas assíncronas e distribuídas que permite a integração e execução de diferentes scripts e programas. Este programa foi utilizado para permitir a anotação automática dos dados de maneira totalmente assíncrona a partir do momento que o usuário realiza o upload dos dados no sistema. Nas figuras 28 e 29 apresentamos a interface do celery, na figura 29 é possível verificar diversos scripts sendo executados em paralelo para realizar a anotação de um exoma que foi inserido no sistema.

4.3.5 Conversão dos dados para CSV

Apesar do Mendel,MD ter sido criado para facilitar o processo de filtragem dos dados utilizando para isso uma interface web, nós também desenvolvemos um script em python para realizar a conversão dos dados de VCF para CSV após o término do processo

Dashboard - Interface para visualização dos indivíduos no sistema

257	mm150442ys	• Edit • Browse • Delete	sdpena	38618	July 1, 2015, 1:57 p.m.	July 2, 2015, 7:23 a.m.	0:19:47.130665	2:04:44.160600	None	populated	Reannotate Individual Repopulate Individual Populate to MongoDB
256	mm150524ao	• Edit • Browse • Delete	sdpena	101054	July 1, 2015, 1:49 p.m.	July 2, 2015, 9:29 a.m.	0:35:06.568579	4:10:29.182620	None	populated	Reannotate Individual Repopulate Individual Populate to MongoDB
255	mm150527wf	• Edit • Browse • Delete	sdpena	101754	July 1, 2015, 1:42 p.m.	July 2, 2015, 9:26 a.m.	0:32:11.318878	4:07:11.888756	None	populated	Reannotate Individual Repopulate Individual Populate to MongoDB
254	mm150531am	• Edit • Browse • Delete	sdpena	98474	July 1, 2015, 1:04 p.m.	July 2, 2015, 5:18 a.m.	0:21:28.254705	5:01:34.938282	None	populated	Reannotate Individual Repopulate Individual Populate to MongoDB
253	mm150539ps	• Edit • Browse • Delete	sdpena	101619	July 1, 2015, 12:54 p.m.	July 2, 2015, 5:18 a.m.	0:23:35.855933	8:05:02.637177	None	populated	Reannotate Individual Repopulate Individual Populate to MongoDB
252	mm150544si	• Edit • Browse • Delete	sdpena	97366	July 1, 2015, 12:43 p.m.	July 2, 2015, 5:18 a.m.	0:23:01.533086	8:05:03.016698	None	populated	Reannotate Individual Repopulate Individual Populate to MongoDB

Figura 26 – A partir desta interface é possível visualizar as variantes de cada indivíduo, reanotar os indivíduos e reinserí-los no banco de dados. Da esquerda para a direita temos as colunas: ID, nome da amostra, usuário, número de variantes, Data da criação, data da modificação, tempo de anotação, tempo de inserção no banco de dados e estado da amostra.

Interface para submissão dos indivíduos no sistema

Add VCFs

Here you can upload your VCF files to the system.
We support the following formats: VCF, VCF.GZ, VCF.ZIP and VCF.RAR

Every VCF File should take between 20 and 40 minutes to be annotated and inserted to the database.
You will receive an e-mail as soon this process finishes with the link to start analysing your individuals.

You can take this time to read our [Documentation](#) about how to filter your variants.

Shared with Groups: x LGC

+ Please, Select your VCF files...

Click here when you are done uploading your files!

Figura 27 – A interface de submissão dos indivíduos foi desenvolvida utilizando a biblioteca Jquery FileUpload para facilitar essa tarefa.

Interface do Celery

```
(mendelmd) raony@mendel:/var/www/html/mendelmd_master/mendelmd_source$ python manage.py celery worker
----- celery@mendel v3.1.13 (Cipater)
---- **** -----
-- * *** * -- Linux-3.16.0-43-generic-x86_64-with-Ubuntu-14.04-trusty
-- * - **** --
- ** ----- [config]
- ** ----- .> app:      default:0x7fbffe875350 (djcelery.loaders.DjangoLoader)
- ** ----- .> transport: django://localhost// 
- ** ----- .> results:   djcelery.backends.database:DatabaseBackend
- *** ... * ... .> concurrency: 30 (prefork)
- ***** -----
-- ***** ----- [queues]
----- .> celery           exchange=celery(direct) key=celery
```

Figura 28 – Nesta figura podemos visualizar a interface de comando do Celery. Nesta interface podemos verificar o output de cada programa utilizado durante a anotação de variantes.

Processo de anotação de variantes

```
realpath /var/www/html/mendelmd_master/annotator
vcffile /var/www/html/mendelmd_master/mendelmd_source/genomes/raony/261/sample.vcf
Running Command python /var/www/html/mendelmd_master/annotator/scripts/validator.py -i /var/www/html/mendelmd_master/mendelmd_source/genomes/raony/261/sample.vcf 2>log
/validator.log
Running Command python /var/www/html/mendelmd_master/annotator/scripts/sanity_check.py -i /var/www/html/mendelmd_master/mendelmd_source/genomes/raony/261/sample.vcf
Running Command python /var/www/html/mendelmd_master/annotator/scripts/snpeff.py -i sanity_check/checked.vcf 2>log/snpeff.log
Running Command python /var/www/html/mendelmd_master/annotator/scripts/vep.py -i sanity_check/checked.vcf
Running Command python /var/www/html/mendelmd_master/annotator/scripts/hi_index.py -i sanity_check/checked.vcf
Running Command python /var/www/html/mendelmd_master/annotator/scripts/hgmd.py -i sanity_check/checked.vcf
Running Command python /var/www/html/mendelmd_master/annotator/scripts/snpsift.py -i sanity_check/checked.vcf 2>log/snpsift.log
Running Command python /var/www/html/mendelmd_master/annotator/scripts/vcf_annotator_parallel.py -n 2 -i sanity_check/checked.vcf -r 1000genomes dbsnp clinvar esp6500 -a /var/www/html/mendelmd_master/annotator/data/1000genomes/ALL.wgs.phase3_shapeit2_mvncall_integrated_v5a.20130502.sites.vcf.gz /var/www/html/mendelmd_master/annotator/data/dbsnp/00-All.vcf.gz /var/www/html/mendelmd_master/annotator/data/dbsnp/clinvar.vcf.gz /var/www/html/mendelmd_master/annotator/data/esp6500/esp6500si.vcf.gz 2>log/pynnotator.log
Running Command python /var/www/html/mendelmd_master/annotator/scripts/cadd_vest_parallel.py -n 2 -i sanity_check/checked.vcf 2>log/cadd_vest.log

Warning: The index file is older than the data file: /var/www/html/mendelmd_master/annotator/data/hgmd/hgmd.sorted.bed.gz.tbi
Finished HGMD, it took 0:00:00.450626
Finished Hi Index, it took 0:00:00.486530
Finished CADD VEST, it took 0:00:02.812356
Finished vcf_annotator, it took 0:00:03.894890
UNIVERSAL->import is deprecated and will be removed in a future perl at /var/www/html/mendelmd_master/annotator/lib/vep/ensembl-tools-release-77/scripts/variant_effect_predictor/Bio/Tree/TreeFunctionsI.pm line 94.
Finished snpsift, it took 0:00:04.688008
Finished VEP, it took 0:00:05.848358
Finished.snpEff, it took 0:01:13.810551
Merging all VCF Files...
Running Command python /var/www/html/mendelmd_master/annotator/scripts/merge.py -i sanity_check/checked.vcf
Finished Merging VCF, it took 0:00:14.325045
Annotation Completed!.
Finished Annotation, it took 0:01:28.674569
adding: ann_sample/annotation.final.vcf (deflated 78%)
```

Figura 29 – Nesta figura podemos observar diversos scripts sendo executados em paralelo utilizando o Celery para realizar a anotação de exomas através do uso de diferentes programas como SnpEff e VEP.

de anotação. Além disso foi desenvolvido uma programa usando wxPython que permite a conversão entre arquivos do tipo VCF para CSV de maneira local utilizando para isso uma interface gráfica com janelas e botões. Isso permite que o usuário converta os dados de VCF para CSV para que eles possam ser filtrados manualmente pelo Médico ou Pesquisador usando um programa de planilhas, como por exemplo o Excel ou o Libre Office Calc.

A figura 30 apresenta a interface gráfica desenvolvida para realizar essa conversão entre os formatos VCF e CSV. Essa transformação entre os formatos também realiza a de-normalização dos dados presentes nas colunas INFO do VCF de maneira que todos os dados dessa coluna fiquem separados em colunas diferentes no arquivo CSV final.

Além disso, também foi adicionada uma opção para que o usuário pudesse alterar a ordem das colunas no arquivo de saída.

4.3.6 Anotação de Variantes

Para realizar a integração de diferentes ferramentas e fontes de informação, foi desenvolvido um *framework* que realiza toda a anotação das variantes e que integra a maior parte dos dados e ferramentas existentes relacionados a este tipo de análise. A figura 31 é a principal figura deste trabalho, onde é apresentado o *framework* que foi desenvolvido durante este trabalho. Pode-se observar nesta figura todos os processos que ocorrem dentro do pipeline de anotação desenvolvido para o Mendel,MD.

Após a inserção dos indivíduos no sistema, nós primeiramente realizamos uma validação dos arquivos VCFs através de um método chamado “*vcf-validator*” que faz parte do programa *VCFTools*. Este método realiza diversos testes no arquivo VCF de entrada e ao final gera um arquivo com todos os problemas que foram encontrados. Após essa validação inicial o arquivo passa então por um método que desenvolvemos chamado de “*sanity-check*” que prepara o arquivo VCF para ser anotado por diferentes ferramentas.

Uma das primeiras ferramentas integradas para a anotação dos dados foi o SNPEFF que entre outras coisas fornece informações importantes sobre cada mutação como por exemplo a classificação do seu impacto de acordo com as seguintes classes: MODIFIER, LOW, MODERATE e HIGH. Essas classes são extremamente úteis na hora de

Programa desenvolvido para conversão entre arquivos VCF e CSV

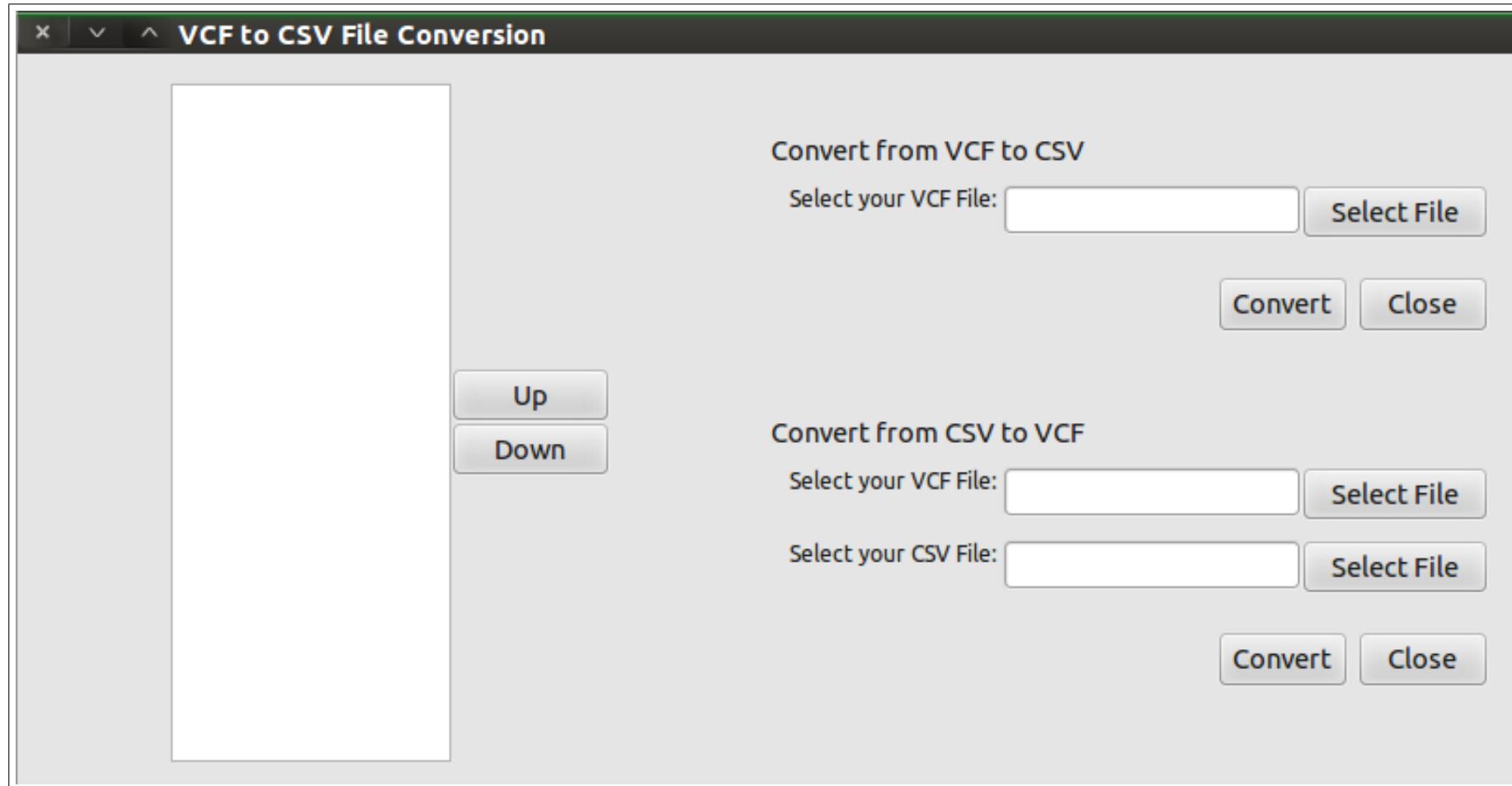


Figura 30 – Interface do programa desenvolvido para conversão entre os formatos VCF e CSV.

realizarmos a filtragem de variantes, o mais recomendado aqui seria primeiro fazer a busca em variantes que são MODERATE e HIGH pois aí estarão presentes as mutações que são mais graves e possivelmente podem causar uma alteração da estrutura de uma proteína.

Outro programa integrado pela nossa anotação foi o Variant Effect Predictor (VEP). Este programa realiza a anotação das variantes em relação aos tipos de mutações encontradas, aos aminoácidos que estão alterados, e caso ela seja uma mutação não-sinônima, anota a posição em relação ao cDNA. Além disso, o VEP também fornece escores de patogenicidade como por exemplo o SIFT e o Polyphen2 que são os escores mais utilizados atualmente quando buscamos por variantes patogênicas.

O banco de dados dbNFSP trouxe a capacidade de agregar centenas de informações diferentes para nossa análise como por exemplo anotação em relação a diferentes bancos de dados, escores de patogenicidade e de conservação de mutações. Para integrar essa ferramenta nós desenvolvemos um script em python chamado de “*pynnotator*” que por sua vez utiliza bibliotecas como *pysam* e *parallel python* para realizar a anotação dos dados de uma maneira rápida eficiente, inclusive fazendo o uso de múltiplos cores do processador ao mesmo tempo. Esse tipo de implementação não foi trivial mas ajudou bastante a diminuir o tempo necessário para se realizar a anotação de cada VCF contra uma grande quantidade de dados. O tempo médio de anotação para cada exoma enviado para o sistema é de apenas dez minutos.

4.3.7 Controle de Qualidade sobre os dados

Para se realizar o controle de qualidade sobre os dados foi desenvolvida uma interface para calcular e mostrar métricas de qualidade calculadas a partir dos VCFs de cada indivíduo. Na figura 32 podemos observar algumas o número de variantes homozigotas e heterozigotas (0/1 e 1/1). Além disso nesta seção é possível verificar outras métricas como o número de variantes para cada indivíduo, a cobertura média de cada exoma, a qualidade média de suas variantes, o número de variantes por cromossomo, o número de variantes por classe funcional entre outras opções que são apresentadas.

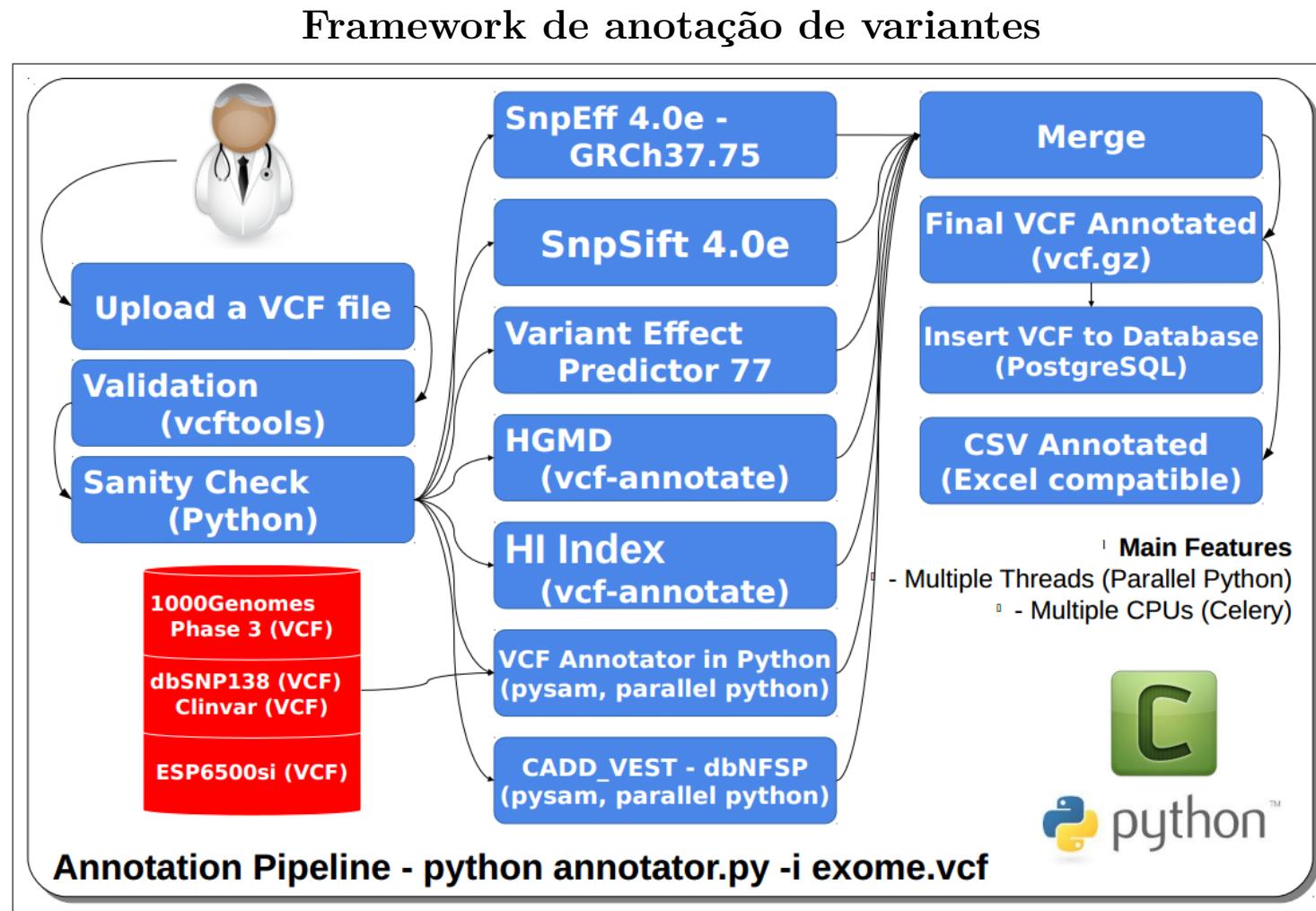


Figura 31 – Este foi o framework de anotação de variantes desenvolvido durante este trabalho, ele realiza a integração de diversas ferramentas e bancos de dados diferentes através de scripts desenvolvidos em python para automatizar todo o processo.

Interface para visualizar métricas sobre os dados inseridos

HOME	SUMMARY	SNP EFFECT	FUNCTIONAL CLASS	IMPACT	FILTER	QUALITY	READ DEPTH	CLINICAL ASSOCIATED		
VARIANTS PER CHROMOSOME										
Type	Total Variants									
Total SNVs	35817	1 1/4	12770 1/1	1 0/3	22828 0/1	151 1/2	8 1/3	23 0/2	29 2/2	6 2/3
Total Gene-associated SNVs	35408	1 1/4	12618 1/1	1 0/3	22579 0/1	145 1/2	8 1/3	23 0/2	27 2/2	6 2/3

Figura 32 – Nesta interface podes visualizar o número total de variantes, os tipos de variante encontradas no indivíduo e diversas outras métricas que são importantes para auxiliarem na filtragem de variantes como o número médio de qualidade e de cobertura para cada indivíduo.

4.3.8 Genes

Na figura 33 apresentamos a interface desenvolvida para realizar a busca de genes no sistema. Com essa interface é possível buscar genes por “gene symbol”, por exemplo “*SUCLA2*” ou então por uma parte do nome do gene “succinate” e então selecionar os genes obtidos nos resultados para serem utilizados no método de filtragem de variantes.

Também foi desenvolvida um opção para armazenar listas de genes personalizadas como por exemplo, com genes que já estivessem associados com doenças Mendelianas dominantes e recessivas. A figura 34 apresenta as listas de genes que foram inseridas no Mendel,MD.

4.3.9 Doenças

Para obter dados sobre doenças Mendelianas nós utilizamos o site OMIM que possui atualmente 6845 doenças e 4715 genes. Esses dados foram inseridos no Mendel,MD para que fosse possível buscar por genes associados a doenças Mendelianas e para que eles pudessem ser inseridos no processo de filtragem de variantes.

Na figura 35 apresentamos a interface que permite a busca por doenças Mendelianas. Aqui é possível buscar por doenças e também selecionar os resultados para serem visualizados no método de filtragem de variantes na busca por variantes candidatas que estejam presentes nos indivíduos afetados pela doença. Nesta página o usuário pode digitar uma doença e selecionar todos os genes associados com ela para buscar variantes em seus indivíduos.

4.3.10 Filtragem de Variantes

Após a anotação dos dados pelo sistema, o usuário pode filtrar as variantes dos indivíduos utilizando para essa tarefa um formulário web que permite a eliminação de variantes utilizando diferentes critérios de filtragem para tentar identificar a variante que possa ser responsável por causar a doença do paciente.

Este método foi implementado para permitir que essa tarefa pudesse ser repetida muitas vezes, facilitando a compreensão do que acontece durante cada etapa do processo

Interface para Busca de Genes

Genes

succinate

Page 1 of 1.

Use selected genes for Filter Analysis

#	Symbol	Name	Options
<input type="checkbox"/>	SUCLA2P3	succinate-CoA ligase, ADP-forming, beta subunit pseudogene 3	View Gene Variants
<input type="checkbox"/>	SUCLA2-AS1	SUCLA2 antisense RNA 1	View Gene Variants
<input type="checkbox"/>	SUCLA2P1	succinate-CoA ligase, ADP-forming, beta subunit pseudogene 1	View Gene Variants
<input type="checkbox"/>	SUCLA2P2	succinate-CoA ligase, ADP-forming, beta subunit pseudogene 2	View Gene Variants
<input type="checkbox"/>	SUCLA2	succinate-CoA ligase, ADP-forming, beta subunit	View Gene Variants

Figura 33 – Essa interface foi desenvolvida para permitir a busca de genes específicos, após a obtenção do resultado é possível investigar variantes apenas na lista de genes desejada.

Interface com grupos de genes adicionados

Gene Lists	
Create GeneList	
Name	Options
Allergy/Immunology/Infectious_manifestation_dominant	View Delete
Allergy/Immunology/Infectious_manifestation_recessive	View Delete
Allergy/Immunology/Infectious_manifestation_xlinked	View Delete
Audiologic/Otolaryngologic_manifestation_dominant	View Delete
Audiologic/Otolaryngologic_manifestation_recessive	View Delete
Audiologic/Otolaryngologic_manifestation_xlinked	View Delete
Biochemical_manifestation_dominant	View Delete
Biochemical_manifestation_recessive	View Delete
Biochemical_manifestation_xlinked	View Delete
Cardiovascular_manifestation_dominant	View Delete

Figura 34 – Nesta interface podemos verificar os grupos de genes que foram criados pelos usuários de acordo com alguns critérios desejados.

Interface para Busca de Doenças

Diseases

Situs

5 Diseases

#	Name	Genes
<input type="checkbox"/>	Retinitis pigmentosa with or without situs inversus, 615434 (3)	ARL2BP, BART
<input type="checkbox"/>	Ciliary dyskinesia, primary, 1, with or without situs inversus, 244400 (3)	DNAI1, CILD1, ICS, PCD
<input type="checkbox"/>	Ciliary dyskinesia, primary, 3, with or without situs inversus, 608644 (3)	DNAH5, HL1, PCD, CILD3
<input type="checkbox"/>	Ciliary dyskinesia, primary, 7, with or without situs inversus, 611884 (3)	DNAH11, DNAHC11, CILD7, DNAHBL
<input type="checkbox"/>	Ciliary dyskinesia, primary, 9, with or without situs inversus, 612444 (3)	DNAI2, CILD9

Figura 35 – Nesta interface podemos visualizar o resultado da busca por doenças com a palavra situs. A partir desta interface é possível visualizar variantes apenas em genes associados a um ou mais tipos específicos de uma determinada doença escolhida pelo usuário.

e permitindo a combinação de diferentes opções de filtragem para chegar a uma lista pequena de candidatos.

As figuras a seguir ilustram o processo de filtragem implementado no Mendel,MD e que foram divididos em 3 etapas.

A figura 36 mostra a primeira etapa onde o usuário pode selecionar os indivíduos em que gostaria de visualizar as variantes existentes e também os indivíduos que gostaria que fossem utilizados como controles no processo de exclusão de variantes. Além disso o usuário pode utilizar uma lista de genes e SNPs que gostaria de incluir ou excluir nos indivíduos selecionados.

Na figura 37 apresentamos a segunda etapa onde o usuário possui diversas opções para definir sobre o tipo de variante que gostaria de visualizar nos resultados. A primeira opção seria em relação ao tipo de variante homozigótica (Ex. 1/1, 2/2 e 3/3) ou heterozigótica (Ex. 0/1, 0/2 e 0/3), esses números correspondem ao genótipos que foram codificados no arquivo VCF para cada indivíduo. Existe uma opção que foi desenvolvida para selecionar apenas variantes em um único cromossomo ou então em uma região específica de um cromossomo como por exemplo: chr:17, pos:80789468-80789469. Isso pode ajudar na investigação de regiões homozigóticas onde possam existir variantes candidatas.

Nesta aba o usuário também pode selecionar o tipo da mutação, o cromossomo, a posição, a coluna de filtro do VCF (Ex. PASS, LowQual), o efeito da variante, a classe funcional, o impacto, o dbSNP Build de quando a variante foi inserida no dbSNP, a cobertura, a qualidade, o número de variantes por gene, exibir apenas variantes presentes em genes comuns entre os indivíduos selecionados, mostrar apenas variantes que não estejam presentes no dbSNP, mostrar apenas variantes anotadas como “*clinically associated*” pelo clinvar e também excluir variantes presentes em regiões com segmento de duplicação

Conforme apresentado na figura 38, a terceira etapa permite a filtragem utilizando valores de máximo e mínimo da frequência possível das variantes em bancos de dados como 1000Genomes, dbSNP e ESP6500. Por último foram incluídos filtros para se eliminar variantes utilizando os escores de SIFT e Polyphen-2.

Além disso, também é possível utilizar outras opções como por exemplo, apenas genes relacionados a doenças específicas do OMIM. Ao começar a digitar, usamos uma

Filtragem de Variantes - 1^a Etapa

+ Filter Options

MAIN VARIANTS DATABASES

SELECT VARIANTS FROM

INDIVIDUALS:

SNP LIST:

GROUPS:

SAVED GENE LIST:

GENE LIST:

EXCLUDE VARIANTS FROM

EXCLUDE INDIVIDUALS:

EXCLUDE SNP LIST:

EXCLUDE GROUPS:

EXCLUDE SAVED GENE LIST:

EXCLUDE GENE LIST:

SELECT INHERITANCE:

RECESSIVE HOMOZYGOUS RECESSIVE COMPOUND HETEROZYGOUS DOMINANT HETEROZYGOUS X-LINKED RECESSIVE HEMIZYGOUS X-LINKED DOMINANT HETEROZYGOUS

SELECT YOUR DISEASES:

OMIM:

CLINICAL GENOMICS DATABASE:

HGMD:

OPEN RESULT IN A NEW WINDOW

Figura 36 – Essa é a primeira tela da filtragem de variantes onde o usuário precisa selecionar as opções referentes aos indivíduos, genes e snps que deseja analisar.

Filtragem de Variantes - 2^a Etapa

MAIN VARIANTS DATABASES

MUTATION TYPE:

VARIANT EFFECT: CODON_CHANGE_PLUS_CODON_DELETION, CODON_CHANGE_PLUS_CODON_INSERTION, CODON_DELETION

DBSNP BUILD: >= 130

EXCLUDE VARIANTS AT VARISNP

CHR: . GATKStandard, GATKStandard;LowQual, GATKStandard;LowQual;repeat

POS:

FUNCTIONAL CLASS: NONE, SILENT, MISSENSE, NONSENSE

IMPACT: HIGH, MODERATE, MODIFIER, LOW

READ DEPTH: >= 10

VARIANTS PER GENE: <=

QUAL: <=

SHOW ONLY VARIANTS PRESENT IN COMMON GENES BETWEEN ALL THE INDIVIDUALS SELECTED
 SHOW ONLY VARIANTS AT EXACTLY SAME POSITION BETWEEN ALL THE INDIVIDUALS SELECTED
 EXCLUDE ALL VARIANTS PRESENT IN LATEST DBSNP BUILD
 SHOW ONLY VARIANTS PRESENT AT HGMD

Figura 37 – Nesta interface é possível selecionar algumas opções das variantes presentes nos arquivos VCF.

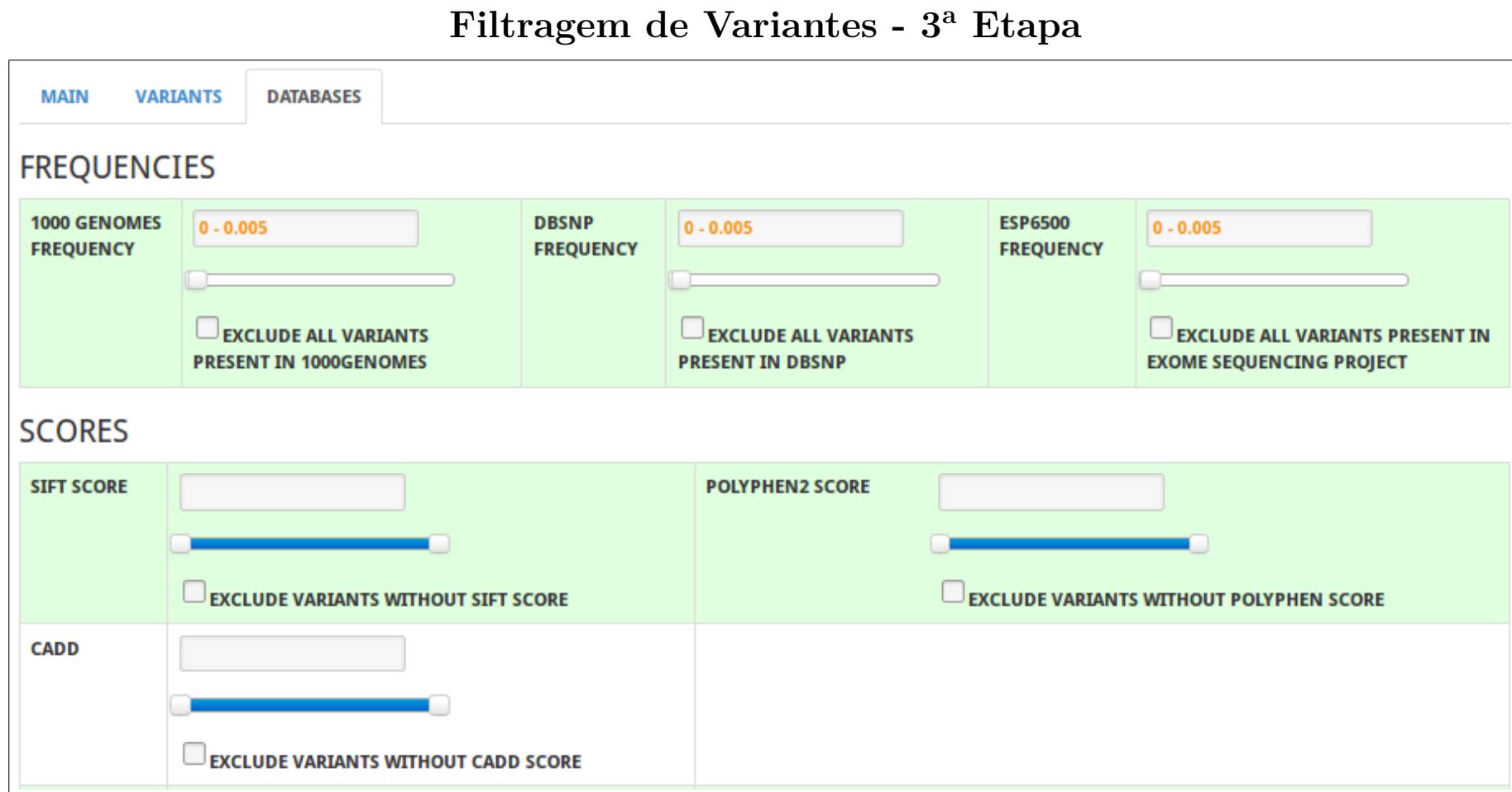


Figura 38 – Nesta interface é possível selecionar valores em relação aos Bancos de Dados e Escores de Priorização que foram anotados para cada variante.

função *autocomplete* que foi desenvolvida usando uma biblioteca de Javascript chamada de Select2 que retorna uma lista de doenças para que o usuário possa adicionar em suas pesquisas. Isso facilita muito a investigação de doenças que possuam um fenótipo parecido mas que sejam causadas por genes diferentes. Rapidamente podemos adicionar diversos nomes de doenças na pesquisa que o sistema irá encontrar apenas os genes relacionados a estas doenças.

4.3.10.1 1-Click

Nesta interface foi desenvolvida uma configuração padrão de filtros que são recomendados para o início do processo de filtragem de variantes. Esta interface foi desenvolvida para facilitar e automatizar a identificação de variantes causadoras de doenças Mendelianas.

O método 1-Click foi inspirado em uma opção da ferramenta de análise filogenética Phylogeny.fr e permite que o usuário selecione apenas os indivíduos da análise e o tipo de herança genética mais provável o que reduz drasticamente o número de variantes candidatas para tentar encontrar a real causadora da doença do indivíduo. Este método faz com que os filtros sejam configurados automaticamente para o usuário. Esses valores foram definidos de maneira empírica e são apenas uma sugestão sobre como realizar a configuração dos filtros.

As configurações pré-definidas estão listadas a seguir:

- SnpEff Impact: HIGH ou MODERATE
- Profundidade de Leitura maior ou igual a 10
- Mostrar apenas variantes em genes comuns aos indivíduos selecionados
- Excluir Variantes presentes no banco VariSNP
- Frequência da variante no 1000Genomes menor que 0.005
- Frequência da variante no dbSNP137 menor que 0.005
- Frequência da variante no Exome Variant Server menor que 0.005

4.3.10.2 Análise de Filtros para Famílias

Este método permite a análise de famílias (Ex. Trios, Quartetos e etc) que podem ser utilizadas no processo de filtragem para encontrar variantes que sejam heterozigóticas nos pais dos indivíduos e que sejam homozigóticas nos filhos afetados. Isso também permite a identificação de variantes candidatas que tenham um padrão de herança chamado de heterozigoto composto, ou seja, quando o filho recebe um alelo do gene com defeito de cada um dos pais. Além disso esta opção também permite a visualização de variantes chamadas *de novo*, ou seja, aquelas que estão presentes nos filhos mas que obrigatoriamente não estejam presentes em nenhum dos pais selecionados.

Para isso nós desenvolvemos uma interface onde é possível definir quem são os pais dos pacientes durante a análise. Então este método usa essa informação obtida para eliminar as variantes que estejam presentes nos pais e que não obedeçam aos critérios de herança estabelecidos. Na figura 39 podemos observar o formulário para este tipo de análise, e na figura 40 podemos verificar que nos resultados para cada genótipo encontrado nos filhos existe o genótipo de cada um dos pais. Quando selecionamos por exemplo a opção 'heterozigoto composto' o que acontece por trás é que o sistema mostra nos resultados apenas genes candidatos que possuem pelo menos uma variante de cada um dos pais. Esse tipo de análise ajuda muito a reduzir o número de genes candidatos.

4.3.10.3 Visualização de variantes

Ao encontrar uma variante de interesse é possível verificar todas as informações disponíveis sobre aquela variante no sistema clicando no botão "View". Para isso nós desenvolvemos uma interface que mostra todos os campos do banco de dados para aquela variante específica. Esta interface possui centenas de anotações para cada variante.

4.3.10.4 Exportação de resultados

Após o usuário realizar a filtragem dos dados do paciente é possível exportar as variantes restantes em formato csv clicando sobre o botão "export to csv" para que elas possam ser investigadas manualmente por um clínico utilizando programas como por

Interface do Family Analysis

Main	Variants	Databases	Diseases	Saved Configs	Saved Analysis	FAQ	
Father	Mother	Inheritance					
x exome_5_ls var annotated	x exome_6_dc var annotated	Autosomal Recessive					
Select Variants From				Exclude Variants From			
Individuals:	Groups:	Exclude individuals:	Exclude groups:				
x exome_3_eds var annotated	Select your Group of Cas	Select your Controls	Select your Group of CoI				
x exome_4_els var annotated	Saved Gene Lists:	Exclude SNP list:	Saved Gene Lists:				
Snp list:	Select your GeneLists		Select your GeneLists to				
	Gene list:		Exclude gene list:				

Figura 39 – Family Analysis - Neste formulário é possível selecionar quem são os pais dos indivíduos que estão sendo analisados para usar essa informação na hora da filtragem de dados

Resultado do Family Analysis

SUCLA2																									
Omim - GeneCards - NCBI																									
	Options	Individual	Chr	Rsid	Pos	Qual	Ref	Alt	Filter	Gen	Father	Mother	Read Depth	Effect	Impact	Func Class	1kgenomes	dbSNP	ESP6500	sift	PP2	VEST	CADD	OMIM	HGMD
<input type="checkbox"/>	<button>View</button>	exome_4_els var annotated	13	rs140963290 dbSNP	48528384	2634.54	T	C	PASS	1/1	HET	HET	67							0.00	1.00	0.99	4.48		
<input type="checkbox"/>	<button>View</button>	exome_3_eds var annotated	13	rs140963290 dbSNP	48528384	3294.36	T	C	PASS	1/1	HET	HET	82							0.00	1.00	0.99	4.48		

Figura 40 – Nos resultados do Family Analysis é possível visualizar o genótipo de cada um dos pais para cada indivíduo nos resultados da análise

exemplo o Excel.

4.3.11 Comparação de Exomas

Para possibilitar a comparação de exomas de diferentes indivíduos e até mesmo de exomas do mesmo indivíduo gerados a partir de tecnologias diferentes foi desenvolvido um método de comparação de VCFs. O algoritmo implementado nesta comparação possui duas etapas: primeiro procura apenas por posições que sejam comuns aos dois indivíduos sendo comparados, depois verifica se o genótipo nessas posições é igual ou diferentes entre os dois arquivos.

Na figura 41 podemos observar a comparação do genótipo de dois irmãos (Exome_2_MB e Exome_3_EDS) através deste método. Nesta caso nós encontramos 48.110 variantes em posições em comum entre os dois irmãos sendo que 84.2% dessas variantes tinham o mesmo genótipo nos dois indivíduos selecionados.

4.4 Casos Clínicos

Nesta seção iremos discutir dois casos clínicos que foram recebidos para estudo pelo Laboratório de Genômica Clínica da Faculdade de Medicina da UFMG.

4.4.1 Caso Clínico LGC 1

O paciente RMS foi diagnosticado com duas doenças raras e pouco conhecidas *situs inversus totalis* e panhipopituitarismo.

Situs Inversus Totalis é uma doença congênita onde todos os órgãos internos do indivíduo encontram-se invertidos, como exemplo, o coração encontra-se localizado no lado direito do peito ao invés do lado esquerdo. Este doença geralmente é acompanhada de problemas no trato respiratório, infertilidade, infecções no aparelho auditivo, e redução ou ausência de olfato. Estes problemas são causados por defeitos nos cílios e flagelos das células, sendo que um cílio é composto por um grupo de microtúbulos que são organizados em pares chamados de braços de dineína, recobertos por uma membrana chamada

Interface para comparação de indivíduos

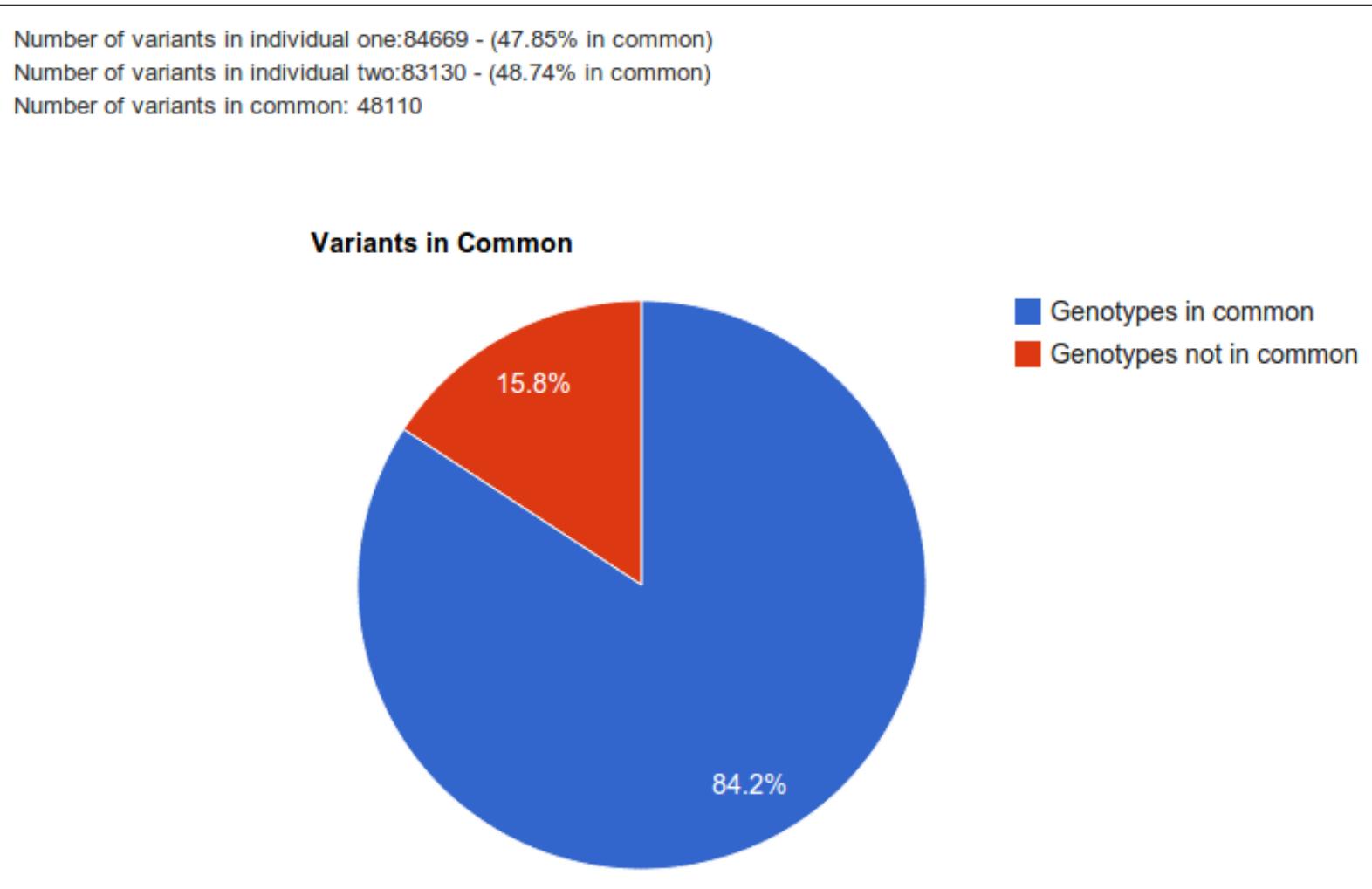


Figura 41 – Essa interface foi desenvolvida para permitir a comparação de dois indivíduos ou VCFs.

axonema. A ausência ou anormalidade dos braços de dineína conectando os nove pares de microtúbulos pode ser revelada através de uma microscopia eletrônica da célula. Apesar desta ser geralmente uma doença autossômica recessiva, ela também pode ser causada por mutações no cromossomo X (GEBBIA et al., 1997).

Esta doença foi desenhada pela primeira vez por Leonardo da Vinci entre os períodos de 1452–1519, e foi reconhecida por Marco Aurélio Severino em 1643, porém, só foi descrita de uma maneira mais científica em 1793 por Matthew Baillie em seu livro "The Morbid Anatomy of Some of the Most Important Parts of the Human Body". Acredita-se que a incidência desta doença seja de 1 em cada 10.000 nascimentos. (BHATT; BHADKAMKAR, 1959; HALÁSZ et al., 2008)

Atualmente existem diversos genes associados com esta doença como por exemplo *DNAI1*, *DNAH5*, *DNAH11*, *ZIC3* e *CCDC11*, *IVS* (NEESEN et al., 2001; OLBRICH et al., 2002; BARTOLONI et al., 2002; GEBBIA et al., 1997; PERLES et al., 2012).

Panhipopituitarismo é uma doença caracterizada pela ausência total ou parcial da glândula pituitária, o que provoca uma deficiência na produção de todos hormônios relacionados a este órgão. Este doença pode ser causada tanto por defeitos genéticos congênitos (presentes desde o nascimento) quanto por acidentes que danifiquem o hipotálamo ou a hipófise do indivíduo.

Entre os genes associados a esta doença podemos citar: *HESX1*, *SOX2*, *SOX3*, *GLI2*, *LHX3*, *LHX4*, *PROP1*, *POU1F1* (VIAROLI, 2012).

Um caso de panhipopituitarismo com situs inversus totalis foi descrito pela literatura em um paciente Húngaro de 56 anos(HALÁSZ et al., 2008). Neste artigo o autor realizou o sequenciamento dos genes *PIT1*, *PROP1*, *PITX2* a procura de mutações que pudessem estar relacionadas com a doença e no final eles concluíram não ter encontrado nenhuma mutação que pudesse ser responsável pelas doenças. Após a publicação deste artigo foi realizado o sequenciamento do exoma deste paciente e os dados foram compartilhados com nosso laboratório para análise junto com o exoma do nosso paciente RMS. A análise em conjunto desses dados não levou a nenhuma conclusão sobre uma mutação, ou gene, que pudesse ser compartilhado pelos dois indivíduos.

Além deste caso descrito pela literatura, outra paciente feminina de Boston, tam-

bém foi diagnosticada com situs inversus totalis e panhipopituitarismo. Nós também recebemos os exomas dela e de sua família (pai, mãe e irmão) totalizando um quarteto de exomas que também foram analisados por nossa ferramenta a procura de mutações que pudessem ser compartilhadas entre os indivíduos afetados.

O tempo total de execução do pipeline completo para o indivíduo RMS foi de 12 horas.

Na tabela 10 apresentamos o processo de filtragem de variantes sob um modelo recessivo que utilizado para o indivíduo RMS.

Ao final da análise o gene restante *GLRA4*, é um receptor de glicina alpha 4 e que não encontra-se associado a nenhuma das duas síndromes em estudo. Portanto, neste caso o mais recomendado seria aumentar o número de genes candidatos. A partir do momento que o número de genes obtidos estiver suficientemente pequeno, o usuário já pode investigar os genes um a um, para verificar se algum deles pode estar associado com a doença em estudo.

Podemos observar que nenhum dos 13 genes candidatos (*ZNF674*, *C21orf62*, *GPRIN2*, *U52112.12*, *OR52I1*, *GLRA4*, *VIL1*, *DSPP*, *HEPH*, *TMEM199*, *RP1L1*, *SARM1*, *PRR21*) pareceu estar associado com a doença, que nenhum deles foi descrito anteriormente como sendo o causador de nenhuma das duas doenças que foram diagnosticadas no indivíduo.

Apesar de todos os esforços não foi possível encontrar as causas destas doenças no paciente. Até mesmo a hipótese de que as variantes responsáveis possam estar em genes diferentes foi levantada devido a grande variedade de genes que poderiam ser responsáveis por causar ambas as doenças como por exemplo os genes *OR52I1*, *GLRA4*, *VIL1* e *DSPP*.

Este caso clínico foi importante para aprendermos a trabalhar com os dados de exomas e para automatizarmos o processo de análise para os futuros casos clínicos que seriam recebidos pelo nosso laboratório.

4.4.2 Caso Clínico LGC 2

Este caso clínico foi composto por 4 indivíduos de uma mesma família, sendo dois irmãos (Exome_3_EDS e Exome_4_ELS) que haviam sido diagnosticados com síndrome

Tabela 10 – Processo de Filtragem utilizado para o paciente RMS

Método de filtragem utilizado	Número de Variantes	Número de Genes
Valores iniciais	39.677	12.341
Apenas variantes com o filtro PASS no VCF dos indivíduos.	36.362	11.840
Remoção de variantes encontradas em 292 exomas usados como controle.	11.737	5.112
Apenas variantes homozigóticas diferentes da referência Ex. 1/1, 2/1, 1/2.	3.916	2.266
Apenas variantes não sinônimas	753	487
Apenas variantes com um profundidade de leitura maior do que 10 (para uma cobertura média de 30X).	644	428
Apenas variantes em genes que possuem ≤ 2 variantes por gene.	434	386
Remoção de variantes em regiões de segmento de duplicação.	368	328
Apenas variantes com frequência menor do que 0.5% nos indivíduos do 1000Genomes	14	13
Apenas variantes com frequência menor do que 0.5% no dbSNP137	14	13
Apenas variantes com frequência menor do que 0.5% nos indivíduos do projeto ESP6500	8	8
Apenas variantes com escore de SIFT entre 0 e 0.05 (Damaging)	2	2
Apenas variantes com escore de Polyphen-2 entre 0.85 e 1.0 (Damaging)	1	1

de Leigh (OMIM:256000) e os seus pais (Exome_5_LS e Exome_6_DC) que não eram afetados pela doença.

A síndrome de Leigh possui 16 genes conhecidos no OMIM que podem conter variantes responsáveis por causar a doença, portanto este foi um caso ideal onde o sequenciamento e análise de exomas facilitou bastante o diagnóstico do paciente.

A figura 42 mostra os passos que foram utilizados para identificação da variante candidata no gene *SUCLA2*.

Na figura 43 nós apresentamos a visualização das variantes desta família em todos os cromossomos do genoma humano.

Nesta análise foram identificados 2 genes candidatos: *SUCLA2* e *OR5H2*. Uma pesquisa bibliográfica revelou um artigo publicado em 2007 ([CARROZZO et al., 2007](#)) com mutações no gene *SUCLA2* associadas com *Leigh-Like Syndrome*(OMIM:612073).

Esse foi o primeiro caso clínico em que conseguimos identificar uma nova mutação, em um gene *SUCLA2* (GENEID:8803) que já havia sido descrito pela literatura como associado a uma doença Mendeliana. Esse caso mostrou que mesmo que fossem sequenciados todos os genes conhecidos associados a síndrome de Leigh, ainda assim a verdadeira mutação causadora da doença do paciente não seria encontrada.

Este caso também mostrou que o principal objetivo da filtragem de variantes não seria a identificação de um único gene candidato no final do processo mas o ideal seria retornar uma lista de genes e variantes que pudessem ser analisadas separadamente pelo médico ou pesquisador, que então seria capaz de identificar o gene e mutação responsável pela doença a partir de uma lista de possíveis candidatos.

A mutação encontrada no gene *SUCLA2* foi confirmada por PCR e estudos funcionais mostraram que ela seria a real causadora da doença do paciente.

4.4.3 Outros casos clínicos

No dia 17 de maio de 2012 foram recebidos 12 novos exomas de 5 casos clínicos diferentes. A identificação dos exomas recebidos e uma breve descrição de seus casos clínicos são apresentados na tabela a seguir.

Processo de filtragem de variantes do caso LGC 2

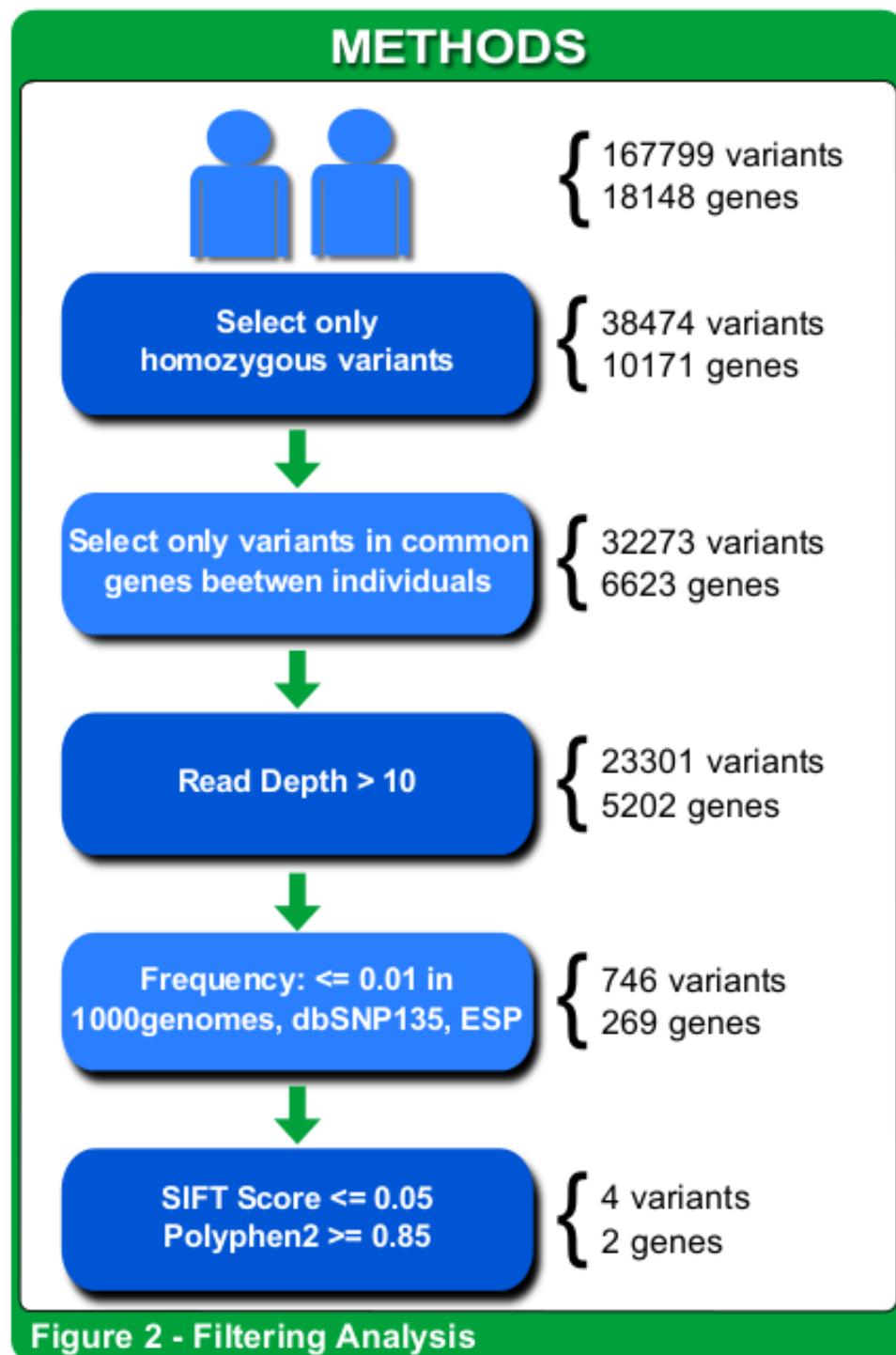


Figura 42 – Nesta figura podemos observar o que aconteceu com o número de variantes e genes em cada etapa da filtragem para este caso clínico.

Visualização das variantes do caso LGC 2

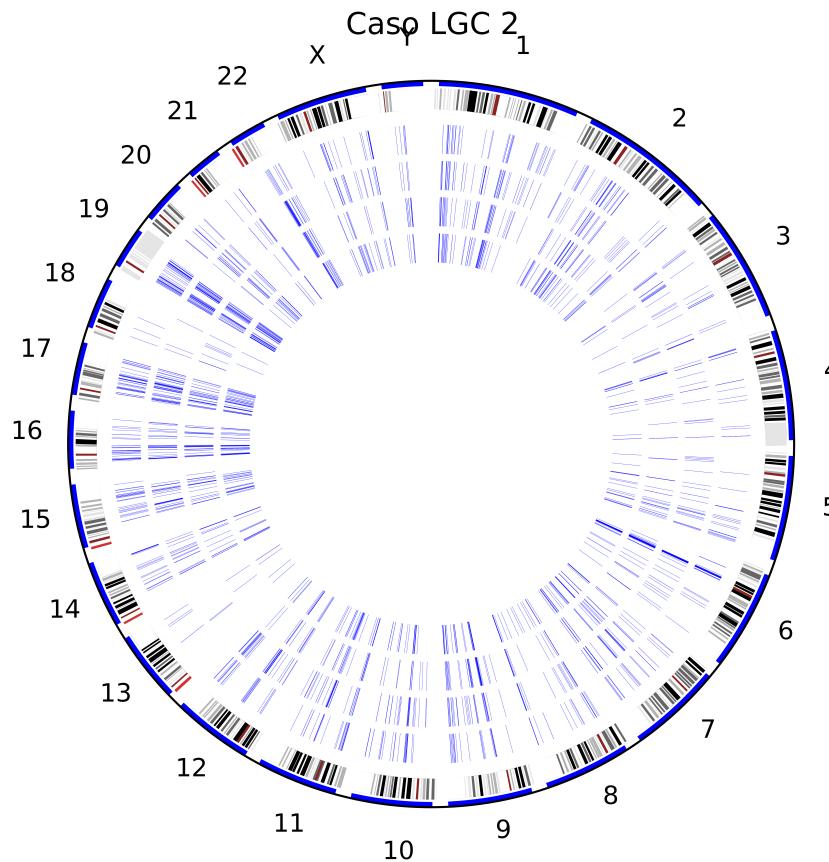


Figura 43 – Esta figura foi criada como uma representação as variantes para o caso LGC 2 - Os indivíduos Exome_3_EDS, Exome_4_ELS, Exome_5_LS e Exome_6_DC estão representados no sentido de dentro para fora da imagem.

Tabela 11 – Descrição dos 12 exomas recebidos

Identificação	Descrição
Exome_1_HJ	Paciente com uma calcificação no cérebro
Exome_2_MB	Paciente (não relacionado com Exome_1_HJ) também com uma calcificação no cérebro
Exome_3_EDS	Paciente com diagnóstico de Síndrome de Leigh (caso 2)
Exome_4_ELS	Irmão do indivíduo Exome_3_EDS também afetado pela mesma doença (caso 2)
Exome_5_LS	Mãe normal dos indivíduos Exome_3_EDS e Exome_4_ELS (caso 2)
Exome_6_DC	Pai normal dos indivíduos Exome_3_EDS e Exome_4_ELS (caso 2)
Exome_7_EUF	Paciente com o diagnóstico de Esclerose Tuberous
Exome_8_MF	Mãe do paciente Exome_7_EUF diagnosticada com Síndrome de Marfan
Exome_9_ELF	Mãe normal da paciente Exome_8_MF
Exome_10_EF	Pai normal da paciente Exome_8_MF
Exome_11_AP	Pai normal do indivíduo RMS
Exome_12_IN	Mãe normal do indivíduo RMS

Esses 12 exomas foram sequenciados utilizando o sequenciador SOLID modelo 5500xl e o kit para o enriquecimento das regiões exônicas que utilizado foi o “SureSelect V3” da Nimblegen. Cada um desses exomas possuem uma cobertura média de 35X e o número médio de variantes por exoma foi de 80 mil. Este número de variantes é considerado um pouco alto em relação ao número variantes de exomas sequenciados por outras tecnologias. Isso é um problema bastante conhecido dos sequenciadores da plataforma SOLID.

Além dos exomas dos casos descritos neste trabalho muitos outros foram recebidos pelo nosso laboratório e foram analisados por outros pesquisadores do nosso laboratório utilizando para isso o Mendel,MD. Em especial ([LINHARES et al., 2014a](#)) e ([LINHARES et al., 2014b](#)) são exemplos de dois casos clínicos do nosso laboratório que foram publicados. O primeiro caso, foi de dois irmãos afetados por uma doença chamada de leiomiomatose que é caracterizada pela proliferação de tumores do músculo liso para outras regiões do corpo como pélvis e abdômen. Foram encontradas variantes em dois genes *NDRG4* e *RLTPL*. No segundo caso tivemos dois irmãos com miofibromatose infantil onde foram encontradas variantes nos genes *PDGFRB* e *PTPRG*, sendo que as variantes do gene *PDGFRB* foram herdadas da mãe e as variantes do gene *PTPRG* foram herdadas do pai. Foi sugerido que a mãe dos pacientes apesar de possuir a mesma mutação não seria afetada pela doença o que caracteriza um modelo de penetrância incompleta para este caso clínico.

Através do uso do Mendel,MD foi possível levantar muitas hipóteses sobre as possíveis variantes que poderiam estar associadas a síndromes que ainda não haviam sido descritas pela literatura. Em colaboração com o Laboratório GENE - Núcleo de Genética Médica, foram sequenciados e analisados 57 casos que ainda não possuíam um diagnóstico definitivo ou onde havia uma suspeita de que os paciente pudesse ter doenças mendelianas.

Em 29 dos 57 casos clínicos analisados (51%) foi possível chegar a um diagnóstico definitivo após utilizar o Mendel,MD para analisar o exoma dos pacientes.

Além dos nossos próprios casos, a pesquisadora Dra. Judith Conroy do Children's University Hospital em Dublin utilizou o Mendel,MD para analisar casos de 42 crianças

com encefalopatia epilética infantil precoce. Ela conseguiu encontrar variantes patogênicas em 26% dos pacientes o que estaria próximo dos valores de sucesso desta metodologia de acordo com a literatura (YANG et al., 2013).

4.5 Mendel,MD - Validação

Apesar do Mendel,MD ter auxiliado no diagnóstico de muitos casos clínicos do nosso próprio laboratório, ainda seria necessário realizar uma validação utilizando exomas de casos clínicos que já fossem descritos pela literatura, onde o gene e a mutação responsáveis por causar a doença já estivessem identificados e validados experimentalmente.

Para isso nós fizemos uma pesquisa no Pubmed procurando por artigos científicos que utilizaram o sequenciamento de exomas nos dois últimos anos para estudar doenças Mendelianas e com isso criamos uma lista de 100 artigos que foram selecionados. Depois da criação desta lista, nós enviamos e-mails para os autores dos estudos pedindo que os genótipos dos pacientes fossem compartilhados com o objetivo de fazer uma validação do Mendel,MD utilizando dados da literatura.

O objetivo principal desta validação foi verificar se os usuários do Mendel,MD seriam capazes de identificar o gene, a variante e a doença do paciente em cada caso clínico, utilizando os dados do exoma do paciente.

Na tabela 12 apresentamos uma lista com os exomas recebidos e que foram utilizados neste processo de validação do Mendel,MD. Ao todo nós recebemos 19 exomas de 11 casos clínicos diferentes. Esses dados foram padronizados para o formato VCF e foram inseridos no Mendel,MD através da nossa interface de upload.

A seguir nós descrevemos brevemente como cada caso clínico foi analisado. Em primeiro lugar nós definimos os valores mínimos de profundidade de leitura de acordo com a cobertura média obtida para cada um dos exomas.

Para realizar na validação dos casos primeiramente nós pedimos para um médico criar uma lista com os sintomas para cada caso clínico.

Tabela 12 – Casos Recebidos para Validação

Casos Clínicos	Indivíduos	Modelo de Herança da doença
1	DG00658	Autossômica Recessiva
2	P6NV	Autossômica Recessiva
3	P2	Autossômica Recessiva
4	NI1000380 e NI1001037	Autossômica Dominante
5	924 3037 e 3121	Autossômica Dominante
6	NI40816, NI54126, NI48062, NI43664	Autossômica Dominante
7	DG1365	Autossômica Recessiva
8	P5	Autossômica Recessiva (Heterozigoto composto)
9	P3	Autossômica Recessiva (Heterozigoto composto)
10	S0001	Autossômica Dominante
11	child, father, mother	Autossômica Recessiva

4.5.1 Caso 1

Para o caso 1 nós recebemos o indivíduo DG00658 com uma doença autossômica recessiva. Este indivíduo tinha inicialmente 86.445 variantes e 63.25X de cobertura. Nós utilizamos o método 1-Click e como resultado nós obtivemos uma lista com mutações em 29 genes candidatos. Desses 29 genes, apenas 10 genes estavam presentes no OMIM como já sendo associados com doenças Mendelianas.

Nós comparamos essa lista de genes associados a doenças com a lista de sintomas do paciente e selecionamos o *POC1A* associado com a doença “Short Stature, Onychodysplasia, Facial Dysmorphism, and Hypotrichosis” como sendo um bom candidato. Esse gene tem uma mutação de G>A na posição 3-52183866 (R81*) com rsID rs397514487 que foi classificada pelo SnpEff com efeito STOP_GAINED, impacto HIGH e classe funcional NONSENSE. É importante notar que esta mutação não possui escores de patogenicidade como SIFT e Polyphen-2 porque ela não causa uma mudança no aminoácido mas ela possui um CADD escore de 7.57, o que é um valor extremamente alto.

4.5.2 Caso 2

No caso 2 nós recebemos o individuo P6NV que tinha uma doença autossômica recessiva. Após realizar o 1-Click foram obtidos 10 genes candidatos, sendo que 4 já estavam presentes no OMIM. Comparando a lista de genes do individuo com a lista de sintomas nos selecionados o gene *MRPL44* como sendo o melhor candidato para este caso clínico. Este gene possui uma mutação homozigótica no cromossomo 2 posição 224824538 (L156R) que é uma variante não-sinônima, missense com um SIFT de 0 e um Polyphen-2 de 0.95.

4.5.3 Caso 3

Nos caso três o indivíduo P2 foi diagnosticado com uma doença autossômica recessiva, após realizarm o 1-Click, foram obtidos 22 genes candidatos, sendo que 9 já estavam presentes no OMIM. Após investigar essa lista de genes nós selecionamos o gene *AARS* como sendo o melhor candidato de acordo com a lista de sintomas definidos para este caso

clínico. Este gene possui uma mutação homozigótica na posição 6:44272249 (R592W) que é não-sinônima, missense, com um SIFT de 0 e um Polyphen-2 de 0.81.

4.5.4 Caso 4

Para o caso 4 nós tínhamos dois indivíduos (nl1001037 e nl1000380) desta vez utilizando um modelo de herança dominante no 1-Click, nós encontramos 63 genes candidatos, sendo que 11 genes já estavam presentes no OMIM e por fim selecionamos o gene *WDR35* como sendo o melhor candidato para este caso. Este gene tinha 3 mutações heterozigotas, sendo que uma foi encontrada no indivíduo nl1001037 na posição 2:20133230 (A875T) que era não-sinônima, missense, com um SIFT de 0 e um Polyphen-2 de 1.0. O outro indivíduo nl1000380 tinha duas mutações candidatas uma na posição 2:20145548 (E626G) sendo não sinônima e a outra na posição 2:20189045 em uma região de splicing.

4.5.5 Caso 5

Para o caso 5 com os indivíduos 924, 3037 e 3121 com uma doença autossômica dominante, nós utilizamos o 1-Click e encontramos 25 genes presentes no OMIM, a partir desta lista nós selecionamos o gene *MAX* que continha 3 mutações no cromossomo 14 nas posições 65569057, 65544630, 65544703 em cada um dos indivíduos. Essas mutações são consideradas START_LOST, SPLICE_SITE_DONOR e STOP_GAINED respectivamente.

4.5.6 Caso 6

Para o caso 6 nós tivemos quatro indivíduos nl54126, nl48062, nl43664 e nl40816. Nós utilizamos o modelo de herança dominante para filtragem de variantes que retornou uma lista com 19 genes candidatos, desta lista apenas 4 genes estavam presentes no OMIM e o gene *SETPB1* foi selecionado como sendo o melhor candidato baseado na lista de sintomas do paciente por possuir uma mutação em cada um dos indivíduos nas seguintes posições 18:42531914 (G870D), 18:42531907 (D868N), 18:42531907 (D868N) e 18:42531917 (I871T) todas consideradas MODERATE e MISSENSE.

4.5.7 Caso 7

No caso 7 nós tivemos apenas um único indivíduo DG1365. Utilizando o modelo de herança recessivo nós encontramos 23 genes candidatos, sendo apenas 4 genes já presentes no OMIM. Como nenhum dos candidatos se mostrou bom o suficiente, nós reduzimos o valor mínimo de profundidade de leitura para este caso e com isso identificamos o gene *DOCK6* que parecia estar relacionado com a lista de sintomas deste caso clínico. Neste gene foi encontrada uma mutação na posição 19:11353955 LT454 que foi classificada pelo SnpEff como HIGH e frameshift. Este caso foi muito importante para mostrar que os valores padrões de filtragem nem sempre serão a melhor escolha.

4.5.8 Casos 8 e 9

Para os casos 8 e 9, nós recebemos dois indivíduos P3 e P5 mas nenhuma informação sobre a lista de sintomas dos pacientes, nós apenas sabíamos o modelo de herança que era heterozigoto composto. Um problema inicial na conversão dos dados para VCF fez com que a variante deste caso fosse excluída do arquivo inserido no Mendel,MD. Após as devidas correções no arquivo convertido nós conseguimos identificar as variantes corretas para estes dois casos clínicos.

Para o caso 8 nós obtivemos 46 genes, sendo que apenas 4 estavam associados com doenças no OMIM. Após considerar a lista de sintomas do paciente, o gene *TK2* foi selecionado como sendo um bom candidato para validação. Este gene tinha duas mutações, a primeira na posição 16:66551110 (R225W) que era não-sinônima, com um SIFT escore de 0, Polyphen-2 de 0.97 e um escore de CADD de 2.53, e a segunda na posição 16:66551095 (T230A) com um SIFT escore de 0.15, Polyphen-2 de 0.06 e com um CADD escore de 1.94.

Para o caso 9 nós utilizando o modelo de herança heterozigoto composto nós encontramos 28 genes sendo apenas 3 já associados com doenças Mendelianas. Nós selecionamos o gene *FARS2* com duas mutações nas posições 6:5545494 (I329T) com um SIFT escore de 0.00, Polyphen-2 escore de 1.0 e com um CADD escore de 5.20, a segunda mutação foi encontrada na posição 6:5613508 (D391V) com um escore de SIFT de 0, um Polyphen-2

de 1 e um cadd escore de 2.25.

4.5.9 Caso 10

Para o caso 10, nós recebemos o paciente S0001 com um doença autossômica dominante, após utilizarmos o 1-Click nós obtivemos uma lista com 261 genes, sendo 51 genes já presentes no OMIM. Após comparar a lista de sintomas para este caso nós selecionamos o gene *GJC2* que possui uma mutação na posição 1:228345602 (S48L) que é não-sinônima, MISSENSE com um SIFT escore de 0.14, um Polyphen-2 escore of 0.95 e um CADD escore de 4.01.

4.5.10 Caso 11

Para o caso 11, recebemos um trio de exomas (*father, mother, children*) onde o filho do casal tinha uma doença autossômica recessiva. Após utilizarmos o 1-Click nós obtivemos 8 genes, sendo que apenas um gene já estava presente no OMIM. Após reduzirmos a quantidade mínima de profundidade de leitura para 5, nós identificamos 31 genes sendo que 6 já estavam presentes no OMIM. Dessa lista nós selecionamos o gene *KIF1A* como sendo o melhor candidato. Este gene tinha uma mutação na posição 2:241723190 com um SIFT de 0, um Polyphen-2 de 0.60 e um CADD escore de 3.56. Este caso também serve como um exemplo de que devemos sempre verificar a cobertura média dos exomas antes de definirmos cada uma das opções de filtragem de variantes. A criança neste caso tinha uma cobertura de 23.52X o que seria uma boa indicação para reduzir o valor mínimo de profundidade de leitura.

4.6 Validação da usabilidade da ferramenta

Para comprovar a usabilidade do sistema, o Professor Sérgio Pena realizou um teste com suas alunas do curso de Bases Moleculares ministrado no Intituto de Ciências Biológicas (ICB) no segundo semestre de 2014. O objetivo desse teste foi verificar se outras

pessoas também seriam capazes de analisar os casos clínicos e identificar o gene, a variante e a doença Mendeliana para cada caso usando para isso o Mendel,MD.

Cada aluna do curso recebeu um caso clínico para analisar e todas foram capazes encontrar as mutações corretas para cada caso clínico utilizando o Mendel,MD. Após o término dessa disciplina, foi elaborado um questionário que foi enviado para cada aluna com três perguntas para avaliarem o Mendel,MD de acordo com alguns critérios. A seguir apresentamos as perguntas que foram enviadas as alunas.

- 1) Que nota geral você dá à sua experiência com o software Mendel,MD?
 - A - 0-20
 - B - 30-40
 - C - 40-60
 - D - 70-80
 - E - 90-100
- 2) Em termos de facilidade de uso em comparação com softwares que vocês estão acostumadas a usar, como você avaliaria o Mendel,MD?
 - A - Muito fácil
 - B - Fácil
 - C - Médio
 - D - Difícil
 - E - Muito difícil
- 3) Quais são as suas sugestões para melhoria do Mendel,MD em facilidade de uso e eficiência?

Ao todo três alunas do curso responderam ao questionário que foi enviado por e-mail pelo Professor Sérgio Pena e as respostas obtidas para cada pergunta estão descritas a seguir.

Na pergunta número um, as três alunas classificaram o Mendel,MD com uma nota geral: E (90-100). Isso significa que o Mendel,MD obteve uma boa qualificação em relação a experiência que os usuários tiveram com o sistema em geral.

Em relação a pergunta número dois sobre a facilidade de utilização do software nós obtivemos as seguintes respostas: A (Muito Fácil), B (Fácil) e B (Fácil). Isso mostra que obtivemos também um boa qualificação em relação a facilidade de uso do Mendel,MD.

Todas as sugestões que foram enviadas pelas alunas na pergunta descritiva número três foram levadas em consideração e ajudaram a melhorar a facilidade de uso do sistema. Essa validação que foi realizada ajudou a fornecer ainda mais evidência de que o programa poderia ser realmente utilizado por outros pesquisadores para realizar a identificação de mutações candidatas para estudar diferentes casos clínicos.

5 Discussão

Os resultados apresentados no capítulo anterior fornecem evidências de que em alguns casos foi realmente possível realizar o diagnóstico clínico de pacientes com doenças Mendelianas através do uso do sequenciamento e análise de exomas. Ou seja, em alguns casos, mesmo com apenas um único indivíduo, foi possível identificar a mutação e o gene causador da doença investigada, sem que para isso fosse necessário utilizar mais de um indivíduo afetado para realizarmos a análise.

O desenvolvimento do Mendel,MD mostrou que é possível oferecer uma solução rápida e eficiente para a análise de exomas e ao mesmo tempo oferecer uma interface bastante simples e amigável aos seus usuários, de modo a permitir o acesso aos médicos e pesquisadores a essa enorme quantidade de dados que são gerados pelos sequenciadores de nova geração e também para incorporar a esses dados novas informações sobre a anotação dessas variantes que estejam presentes em diferentes tipos de bancos de dados.

Apesar da aplicação deste método ser relativamente nova na prática clínica, isso tem trazido enormes benefícios para auxiliar no diagnóstico clínico, e na solução de alguns casos que ainda não haviam sido resolvidos através de outros exames convencionais. É importante ressaltar que mesmo com a popularização do uso do sequenciamento de exomas, sempre será necessário confirmar a existência da mutação através de outros métodos de sequenciamento tradicionais, como por exemplo, utilizando o método de Sanger para fazer a validação e a confirmação da existência das mutações identificadas. Sem isso não seria possível chegar a um diagnóstico definitivo sobre o caso clínico estudado. Nos últimos dois anos, com o aumento da cobertura média dos exomas que recebemos para serem analisados (em média 100X), nós obtivemos uma melhora considerável o que facilitou ainda mais a análise e permitiu inclusive a identificação de algumas indels que possuíam uma boa cobertura no exoma.

A análise de exomas pode ser utilizada tanto para direcionar os estudos sobre uma doença específica, como por exemplo, para a identificação de um novo gene candidato que ainda não estiver descrito pela literatura científica como sendo associado com o fenótipo

em estudo, ou então, para diagnosticar um paciente buscando por mutações que já estejam descritas pela literatura como sendo patogênicas e que já estejam presentes em bancos de dados públicos como o OMIM, HGMD ou Clinvar. Essa é uma parte essencial da e muito importante da ferramenta que foi desenvolvida.

5.1 Sobre o armazenamento dos dados

Em relação ao espaço para armazenamento dos dados genômicos o que muitas pessoas não percebem é que o tamanho dos arquivos para se armazenar as informações sobre cada indivíduo é considerado bastante pequeno. Sabemos que um genoma humano possui em média 4 milhões de variantes em relação ao genoma de referência que possui 4 bilhões de pares de base. Um exoma possui algo entre 20 e 40 mil variantes. Para armazenarmos a informação sobre um genoma humano em um arquivo VCF nós precisamos de apenas 40mb de espaço em disco. Como exemplo podemos citar os arquivos VCFs disponibilizados pelos pesquisadores James Watson e J Craig Venter que possuem 37 megabytes e 39 megabytes de tamanho respectivamente. Um arquivo VCF gerado pelo nosso laboratório contendo o exoma de um único individuo possui em média 10mb de tamanho, isso porque este arquivo contém algumas informação extras sobre as variantes além do genótipo do indivíduo para cada posição encontrada.

É importante notar que enquanto um arquivo FASTQ de um exoma humano com 30x de cobertura tem em média 5GB de tamanho, o arquivo VCF final desse mesmo indivíduo com todas variantes identificadas possui apenas 10MB de tamanho. Um arquivo BAM de um exoma com essa cobertura possui em média 10Gb de espaço em disco.

Para o nosso banco de dados, que possui mais de 200 indivíduos armazenados, são necessários 100GB de espaço em disco e apenas 29GB para armazenar todos os arquivos VCFs compactados desses indivíduos, esse tamanho também inclui todas as anotações de centenas de outras fontes de informação que foram incorporadas a esses arquivos VCFs para permitir a análise desses dados. Nosso banco de dados contém as mesmas informações presentes nos arquivos VCFs porém de uma forma mais descompactada, não normalizada e indexada para acelerar a busca e recuperação das informações sobre cada

indivíduo utilizando nossa interface web de filtragem de variantes.

Em uma palestra em abril de 2015 com o título “*Genomics, Big Data, and Medicine Seminar Series*” o pesquisador George Church afirmou que seriam necessários apenas 9 Petabytes de espaço em disco para armazenar todas as informações genômicas de toda a raça humana. Esse número apesar de inicialmente parecer grande pode ser considerado pequeno quando comparado com a enorme quantidade de informações que empresas como o Google ou alguns institutos de pesquisa como o NIH estão acostumadas a trabalhar.

Link: <<https://www.youtube.com/watch?v=iVG4EaMrXfI>>

5.2 Sobre a anotação de variantes

A grande vantagem do Mendel,MD em relação aos outros softwares disponíveis para análise genômica é que ele é capaz de integrar de maneira rápida e eficiente todas as informações necessárias para realizarmos a identificação de variantes durante a anotação dos dados. Para realizarmos essa anotação nós utilizamos no total 20GB de arquivos contendo todas as informações de referência que são adicionadas ao arquivos VCF de entrada pela nossa análise.

O *framework* desenvolvido por este trabalho para anotação dos arquivos VCFs pode ser considerado um dos pontos mais fortes deste trabalho. Para que esta tarefa fosse executada de maneira eficiente e em tempo hábil foi necessário a utilização de técnicas modernas de programação durante o seu desenvolvimento que permitissem o uso de múltiplos núcleos, a paralelização da execução dos processos, o uso de filas de espera, a utilização de um *cluster* de computadores e a sincronização dos dados entre diferentes máquinas.

Uma das grandes vantagens da nossa análise que a diferencia das outras existentes foi a implementação do uso de múltiplos núcleos de processamento para realizarmos a anotação dos dados. Isso reduziu bastante o tempo necessário para realizarmos a anotação e permitiu que ela fosse repetida diversas vezes sempre que novos dados estivessem disponíveis para download.

Um dos problemas com a anotação de variantes é que tanto os dados quanto os programas utilizados para anotação estão sendo desenvolvidos e atualizados de uma ma-

neira constante. Isso significa que a cada nova versão disponível de arquivos VCFs (Ex. 1000Genomes, OMIM, dbSNP ou dbNFSP) é necessário repetir a anotação dos exomas para melhorar a análise dos dados. A cada nova versão de arquivo VCF disponível para download ou de um programa utilizado o ideal seria repetir a anotação para todos os indivíduos do banco de dados. Isso pode ajudar a resolver alguns casos clínicos que ainda não foram solucionados e permite eliminar cada vez mais variantes falso-positivas, reduzindo ainda mais o número de variantes candidatas do paciente que precisam ser investigadas manualmente.

5.3 Sobre a filtragem de variantes

Em relação a filtragem de variantes nossa estratégia principal nesta etapa da análise foi a de dividir essa tarefa em diversas rotinas pequenas de uma maneira que isso facilitasse a integração de diferentes opções e métodos durante a filtragem dos dados.

Um dos primeiros filtros implementados foi em relação ao tipo da variante pesquisada, se ela seria homozigótica (Ex. 1/1, 2/2, 3/3) ou heterozigótica (0/1, 0/2, 0/3, 1/2) de acordo com a suspeita sobre o caso clínico que estivesse sendo investigado. Essa opção também foi utilizada para diferenciar entre os modelos de herança dominantes e recessivos para cada doença Mendeliana específica. Outro filtro bastante útil foi a criação de um campo de texto onde o usuário pudesse entrar com listas de SNPs (Ex. rs1234, rs334, rs556) ou de genes (Ex. *PITX2*, *SUCLA2*, *BRCA1*) para poder visualizar as variantes presentes apenas nesses locais de interesse. Isso permitiu o uso de listas de genes já conhecidos para a identificação variantes relacionadas a doenças Mendelianas e trouxe a possibilidade de excluir SNPs ou genes e eliminar aqueles que com certeza não estariam associados com a doença.

Uma das coisas que foi observada na filtragem de variantes foi o alto número de variantes nos resultados em genes associados a receptores olfativos e genes pertencentes a família das mucinas. Esses genes podem ser facilmente excluídos da lista de resultados pelo usuário.

A próxima opção implementada foi a possibilidade de excluir variantes que esti-

vessem presentes nos indivíduos “normais” que seriam utilizados como controles durante a filtragem de variantes. Isso foi bastante utilizado para os casos clínicos onde o exomas dos pais dos pacientes afetados também foram sequenciados junto com o paciente. Esse método se mostrou bastante efetivo para eliminar as variantes falso-positivas que poderiam estar associadas com um viés da tecnologia do sequencialmente e que com certeza não estariam associadas com a doença investigada.

Essa análise permite a combinação de diferentes opções de filtros para ajudar a encontrar a melhor variante candidata possível. Todas as opções dessa análise foram adicionadas de maneira sequencial (gradativa) através do desenvolvimento de um formulário web (Ex. sliders, checkbox, input text, select).

Um dos atributos mais importantes da filtragem de variantes foi a inserção da biblioteca de Javascript chamada de Select2. O Django possui um plugin chamado “Django-Select2” que permite a criação de campos no formulário que funcionem com a opção de auto-completar dos dados (*auto-complete*)). Isso permitiu a busca por indivíduos, grupos de indivíduos, genes, grupos de genes, e doenças Mendelianas durante a filtragem de variantes, de forma que o usuário pudesse digitar, por exemplo, ‘exome_3_eds’ e através do uso de uma requisição por ajax utilizando uma o formato json como resposta, o sistema faz uma consulta no banco de dados e retorna uma lista com todos os registros encontrados, como ‘exome_3_eds var annotated’ no resultado que então pode ser facilmente incluído na lista de indivíduos afetados ou na lista de controles. Isso facilita muito o preenchimento dos dados no formulário e permite a modificação das opções de maneira muito rápida.

5.4 Variantes falso-positivas e falso-negativas

Um dos principais problemas atuais com o sequenciamento e a análise de exomas é o alto número de variantes falsos-positivas que estão presentes nos resultados finais de cada paciente mas foram causadas por um viés existente na tecnologia utilizada para realizar o sequenciamento dos dados. Um dos grande desafios atuais é encontrar uma maneira efetiva de eliminar essas variantes sem jogar fora variantes que sejam realmente verdadeiras e talvez tenham um baixo escore de qualidade. Existem algumas técnicas que

podem ser utilizadas para ajudar a reduzir este número, como por exemplo, a definição de um valor de limiar mínimo para os valores de qualidade e de profundidade de leitura das variantes. Para utilizar esses valores de filtragem é importante saber os valores médios de cada exoma que estiver sendo analisado para isso criamos uma interface que faz todos os cálculos dessas métricas para o usuário.

As variantes falso-negativas seriam aquelas que existem ao longo do exoma do paciente, mas que não foram detectadas através daquele tipo de sequenciamento realizado. Em alguns casos a variante causadora da doença Mendeliana pode não ter sido coberta por aquele kit de captura que foi utilizado. O tamanho da região coberta e o preço dos kits são duas coisas que variam bastante de acordo com cada empresa que produz os reagentes.

Não podemos esquecer ainda existem que muitas variantes que estão associadas a doenças complexas e podem não estar presentes na região exônica. Por causa disso algumas variantes não poderão ser detectadas através deste método. Antes de realizarmos o sequenciamento de um paciente, o ideal seria sempre fazer um exame chamado “array de SNPs” para eliminar a possibilidade de que variações grandes como por exemplo as CNVs sejam as reais causadoras da doença do paciente.

5.5 Exportação dos dados

Outra característica, extremamente importante do Mendel,MD é a possibilidade de se exportar todos os dados referentes às variantes e todas as anotações que foram incorporadas aos arquivos VCF de entrada uma vez que todos os dados forem processados pelo nosso *framework* de anotação. O arquivo final possui centenas de informações incorporadas em cada posição do genoma a partir de outras fontes de dados. Uma das vantagens de gerar um arquivo no formato VCF no final é que ele também poderá ser utilizado normalmente por outros que sejam capazes de realizar esse tipo de análise, mesmo após serem anotados pela nossa ferramenta.

Após o usuário realizar a filtragem dos dados utilizando o Mendel,MD ele pode simplesmente exportar os resultados da filtragem em formato CSV com todos os campos

presentes no nosso banco de dados em relação a cada variante para que esses dados possam então ser visualizados por exemplo utilizando programas como o Microsoft Excel ou o LibreOffice Calc. Ao exportar esses dados dados, nós incluímos todas as informações presentes no banco de dados sobre cada variante e não apenas aquelas que são mostradas na visualização dos resultados da filtragem de variantes. Isso ajuda a investigar outras informações sobre essas variantes escolhidas e ajuda a trazer mais evidências para a variante que for escolhida e que pode ser a real causadora da doença Mendeliana estudada.

5.6 Visualização dos Dados

Depois de selecionar uma lista de variantes candidatas utilizando o Mendel,MD existem dois programas que podem ser utilizados para visualizarmos as variantes estudantes o IGV desenvolvido pelo Broad Institute e o Genome Browser desenvolvido pelo a empresa GoldenHelix. Ambos podem ser utilizados para visualizarmos a região que possui cada variantes utilizando os arquivos no formato BAM e VCF.

Caso a variante esteja em uma região sem muitos problemas, com alta cobertura e uma boa distribuição de leituras para as sequências *forward* e *reverse* isso ajuda a trazer mais evidência para a análise de que a variante possa de fato existir no paciente.

Uma das bibliotecas utilizados por esse trabalho para fazer a visualização dos dados foi o matplotlib em Python. Utilizando conhecimentos de matemática como por exemplo, o sistema de coordenadas polares, foi possível converter os dados dos pacientes em relação as posições referentes aos cromossomos para uma distância em relação a um ponto fixo e um ângulo em relação a esse ponto. Usando essa metodologia foi possível criar gráficos circulares para visualizar os dados de vários indivíduos de uma maneira equivalente a uma ferramenta chamada de CircosPlot que é bastante utilizada em Bioinformática. A grande diferença entre a nossa abordagem e a do CircosPlot é que nosso programa aceita a entrada de indivíduos no formato VCF e permite uma maior customização da visualização dos dados conforme apresentado na figura 43.

5.7 Questões Éticas

O sequenciamento de exomas também trouxe algumas questões éticas que precisam ser discutidas por este trabalho. Como este método realiza o sequenciamento de todos os genes humanos conhecidos (cerca de 20 mil), muitas vezes isso pode levar ao que chamamos de descobertas “acidentais” de variantes, ou seja, aquelas que estejam relacionadas a outros tipos de doenças que são sejam o foco da pesquisa atual que está sendo realizada. Em 2013 a *American College of Medical Genetics and Genomics* (ACMG) ([GREEN et al., 2013](#)) publicou uma lista com alguns genes que supostamente deveriam ser obrigatoriamente investigados pelos médicos em busca de mutações causais que pudessem ajudar no tratamento de algumas doenças. Essas mutações são chamadas de “*clinically actionable*” e deveriam ser reportadas toda vez em que fosse realizado o sequenciamento do exoma de um paciente. Entre os genes dessa lista podemos citar os genes *BRCA1*, *BRCA2*, *TSC2*, *TP53* entre outros. Apesar da publicação deste documento, muitos médicos ainda preferem não reportar as variantes encontradas nessa lista de genes.

Um estudo recente publicado em 2015 ([MIDDLETON et al., 2015](#)) com 6944 pessoas de 75 países diferentes mostrou que 98% dos entrevistados possuem interesse em serem informados sobre doenças graves que possam ser evitadas ou então tratadas. Atualmente existe uma certa pressão para que essas informações descobertas em estudos por exemplo de *clinical-trials* sejam entregues de volta para o participante mesmo quando a variante não esteja relacionada com o estudo que estiver sendo realizado. Esse estudo também mostrou que existe uma grande diferença de opinião entre os profissionais que conduzem as pesquisas e os participantes da pesquisa que foi realizada.

Ainda é preciso lembrar que nem os pacientes nem os profissionais de saúde estão preparados para trabalhar com esse novo tipo de informação e ainda serão necessários alguns anos para que essa técnica se torne mais precisa e confiável. Um dos grandes problemas enfrentados atualmente por esta técnica é a definição de que uma mutação nova que for encontrada seja realmente patogênica e para isso são necessários muitos estudos funcionais que comprovem realmente essa associação entre a variante, o gene e a doença Mendeliana.

5.7.1 Em relação ao número de pacientes

Um dos problemas com a aplicação deste método para a investigação de pacientes afetados por doenças Mendelianas é a necessidade de um número mínimo de indivíduos afetados. Muitas vezes devido a baixa incidência da doença na população não existem outros pacientes disponíveis para serem sequenciados junto com o paciente e finalmente serem utilizados na filtragem de variantes em busca de mutações em genes que sejam comuns aos indivíduos afetados. Um dos maiores desafios do sequenciamento de exomas é de encontrar um novo gene candidato e a mutação causadora da doença utilizando para isso apenas os dados do exoma de um único indivíduo afetado pela doença. Apesar disso, já existem muitos casos descritos pela literatura onde essa identificação foi possível utilizando apenas um único indivíduo. Lembrando que a mutação pode estar em outra posição de um gene já conhecido como causador da doença.

Mas para que a análise de exomas tenha alto poder estatístico, o ideal seria utilizar sempre o maior número possível de indivíduos que forem afetados pela mesma doença. Isso ajuda a reduzir muito o número de genes e variantes candidatas e facilita bastante a identificação do gene causador da doença Mendeliana. Não podemos esquecer que ao analisarmos vários indivíduos com a mesma doença é possível que sejam genes diferentes encontrados em cada um dos indivíduos. Por causa disso o ideal é sempre começar a sua análise com um único indivíduo e adicionar novos indivíduos afetados de maneira gradual. Também pode-se utilizar famílias com mais de um indivíduo afetado para facilitar a análise.

5.7.2 Validação Experimental

Apesar da análise de exomas ter se tornado um poderosa ferramenta para investigação clínica de doenças Mendelianas ainda é preciso utilizar diversos outros experimentos para poder comprovar a associação entre a mutação encontrada e a doença que estiver sendo estudada. A análise de exomas pode ser utilizada inicialmente para selecionar bons genes candidatos mas isso não exclui a necessidade de que as mutações precisem ser validadas através de outras técnicas experimentais.

5.7.3 Sobre o futuro da análise de exomas e genomas

Com o barateamento dos custos para se realizar o sequenciamento de exomas, espera-se que essa técnica continue a avançar e trazer soluções cada vez mais rápidas para o diagnóstico clínico de pacientes. O aumento do número de variantes com frequência conhecida em diferentes populações em bancos de dados públicos, promete facilitar a identificação de variantes patogênicas de uma maneira cada vez mais rápida e eficiente. Enquanto não houver softwares que sejam “amigáveis” o suficiente para serem utilizados por médicos e pesquisadores, essa técnica não será plenamente adotada na prática clínica.

5.8 Custo do Sequenciamento e da Interpretação de Exomas

Apesar do preço do sequenciamento de exomas estar atualmente em 2015 na faixa de U\$ 445.00 dólares para uma cobertura mínima de 30X, utilizando para isso um sequenciador de DNA modelo Illumina HiSeq, ou então U\$800 dólares para uma cobertura de 100X, esse preço ainda não inclui o custo para se realizar a análise desses dados.

Na verdade não existe um preço definido para se realizar a análise dos dados mas acredita-se que ela custe muito mais do que o valor do sequenciamento dos dados do paciente.

Fonte: <<https://www.scienceexchange.com/services/whole-exome-seq>>

5.9 Serviços Online e Softwares Comerciais

Já existem algumas empresas nos Estados Unidos como por exemplo a DNAnexus (<<https://www.dnanexus.com/>>) que permitem realizar toda a análise dos dados genômicos de maneira totalmente online utilizando apenas um navegador web. Com esse serviço também é possível realizar o *upload* dos dados em formato FASTQ e gerar arquivos BAMs e VCFs utilizando diferentes programas e pipelines de análise de dados diferentes que forem escolhidos pelo usuário.

Um dos softwares mais utilizados para fazer a visualização e a confirmação de variantes pelo nosso laboratório é o Alamut (<<http://www.interactive-biosoftware.com/>>)

[alamut-visual/>](#)). Esse programa permite a visualização de arquivos BAMs e VCFs para investigar a região onde a variante foi detectada e uma de suas grandes vantagens é a possibilidade de inserir mutações ao longo do genoma e verificar o seu impacto utilizando diferentes escores de patogenicidade. Isso pode ser utilizado para confirmar os escores calculados por outros programas como por exemplo ANNOVAR, VEP ou dbNFSP.

Também existem alguns outros softwares comerciais como por exemplo Enlis (<https://www.enlis.com/>) e VarSeq (<http://goldenhelix.com/VarSeq/>) que permitem a re- realização desse tipo de análise de maneira totalmente offline com programas desenvolvidos para Desktop que podem serem executados em qualquer computador com Windows, Linux ou MAC desde que ela tenha os requisitos mínimos para sua execução.

A desvantagem de utilizar esses programas para Desktop para analisarmos os dados genômicos é que muitos programas ainda precisam de uma máquina com um alto desempenho para realizarem a anotação e análise dos dados e isso precisa ser executados no próprio computador do usuário de uma maneira totalmente nativa.

Uma das principais vantagens de se utilizar um ferramenta web como o Mendel,MD é que elas são geralmente mais rápidas, por possuírem um bom processador e bastante memória disponível, o que facilita bastante o processamento dos dados e aumenta a velocidade de consultas ao banco de dados.

6 Conclusão e Perspectivas

O Mendel,MD foi um exemplo prático de como é possível construir um software de que seja responsável pela aquisição, organização e análise de dados genômicos de diferentes casos clínicos que foram investigados pelo nosso laboratório. A disponibilização do código fonte de forma *open-source* e pública na internet promete ajudar outros laboratórios pelo mundo a analisar seus próprios casos clínicos e facilitar bastante a tarefa de identificar mutações candidatas para serem validadas experimentalmente.

O aumento da quantidade de dados biológicos disponíveis atualmente para serem analisados trouxe um grande desafio para a ciência atual. Vivemos em uma época onde os sequenciadores produzem muito mais dados do que os cientistas são capazes de analisar. Por causa disso é preciso que haja o desenvolvimento de novos algoritmos e tecnologias cada vez mais poderosas que sejam capazes de integrar diferentes tipos de experimentos (Ex. DNA, RNA e Proteína) para tentar extrair algum tipo de informação útil desses dados que possam ajudar diretamente no tratamento de pacientes. Isso promete causar uma revolução na Medicina nos próximos anos.

O uso do sequenciamento de exomas trouxe uma nova metodologia para a prática clínica que promete ajudar a solucionar casos clínicos que até hoje não foram solucionados por outros métodos convencionais. Além disso, o chamado *screening* de crianças recém-nascidas, promete no futuro ajudar até mesmo no tratamento de algumas doenças genéticas e em alguns raros casos ajudar na definição do tipo de medicamento que pode ser utilizado para tratar cada paciente.

Atualmente, alguns pesquisadores sugerem que o exoma seja utilizado de uma maneira chamada de “*exome-first approach*”. Isso significa utilizar o sequenciamento de exomas antes mesmo de pedir outros exames clínicos mais caros, com o objetivo de tentar identificar possíveis SNPs, indels ou CNVs que possam estar associadas de alguma forma com a doença do paciente. Quando isso acontecer, os médicos terão acesso não somente ao exoma do paciente mas também a todos os outros exomas que estarão presentes nos bancos de dados.

Com o aumento do número de indivíduos completamente sequenciados nos bancos de dados públicos, espera-se que o número de novas variantes encontradas em um único indivíduo diminua cada vez mais e com isso a tarefa de identificar variantes relevantes tende a ficar cada vez mais simples, especialmente para as variantes que sejam patogênicas, ou seja, aquelas variantes que já estejam associadas com doenças mendelianas pela literatura no OMIM ou no HGMD. Isso promete trazer grandes benefícios para o diagnóstico dos pacientes ao longo dos próximos anos.

A visualização dessa enorme quantidade de dados, também é uma área em constante evolução, e que promete facilitar cada vez mais a identificação de variantes clinicamente relevantes a partir dos dados genômicos de cada paciente. No futuro, ao consultarmos um médico será possível realizarmos perguntas mais complexas sobre a nossa saúde e com certeza a informação genômica poderá ser utilizada por diferentes especialistas para direcionar o tratamento de cada paciente de uma maneira um pouco mais individualizada.

Para concluir, o aumento do número de casos clínicos solucionados através do uso de sequenciamento de exomas promete continuar crescendo. Isso irá ajudar ainda mais a impulsionar a adoção deste método na prática clínica auxiliando o médico cada vez mais na hora de realizar o diagnóstico clínico de seus pacientes. Para que isso aconteça ainda será que preciso que esta técnica torne-se mais precisa e confiável e que novos programas sejam desenvolvidos para facilitar o acesso a este tipo de informação.

Referências

- ABECASIS, G. R. et al. An integrated map of genetic variation from 1,092 human genomes. *Nature*, v. 491, n. 7422, p. 56–65, nov 2012. ISSN 1476-4687. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3498066/>>. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3498066/>>. Citado na página 12.
- ADAMS, H. G. Medical Informatics: Computer Applications in Health Care. *JAMA: The Journal of the American Medical Association*, v. 265, n. 4, p. 522, jan 1991. ISSN 0098-7484. Disponível em: <<http://jama.jamanetwork.com/article.aspx?doi=10.1001/jama.1991.03460040100041>>. Citado na página 1.
- ADZHUBEI, I.; JORDAN, D. M.; SUNYAEV, S. R. Predicting Functional Effect of Human Missense Mutations Using PolyPhen-2. In: *Current Protocols in Human Genetics*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2013. Chapter 7, n. January, p. 7.20.1–7.20.41. ISBN 0471142905. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/23315928>>. Disponível em: <<http://doi.wiley.com/10.1002/0471142905.hg0720s76>>. Citado na página 37.
- BAMSHAD, M. J. et al. Exome sequencing as a tool for Mendelian disease gene discovery. *Nature reviews. Genetics*, Nature Publishing Group, v. 12, n. 11, p. 745–55, nov 2011. ISSN 1471-0064. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3219469/>>. Citado na página 5.
- BARTOLONI, L. et al. Mutations in the DNAH11 (axonemal heavy chain dynein type 11) gene cause one form of situs inversus totalis and most likely primary ciliary dyskinesia. *Proceedings of the National Academy of Sciences of the United States of America*, v. 99, n. 16, p. 10282–6, aug 2002. ISSN 0027-8424. Disponível em: <www.ncbi.nlm.nih.gov/pmc/articles/PMC1214246/>. Citado na página 90.
- BERNSTAM, E. V.; SMITH, J. W.; JOHNSON, T. R. What is biomedical informatics? *Journal of biomedical informatics*, Elsevier Inc., v. 43, n. 1, p. 104–10, feb 2010. ISSN 1532-0480. Disponível em: <www.ncbi.nlm.nih.gov/pmc/articles/PMC2820037/>. Citado na página 2.
- BHATT, V. P.; BHADKAMKAR, A. R. *Situs inversus totalis*. 1959. 107–110 p. Citado na página 90.
- CARROZZO, R. et al. SUCLA2 mutations are associated with mild methylmalonic aciduria, Leigh-like encephalomyopathy, dystonia and deafness. *Brain*, v. 130, n. Pt 3, p. 862–874, mar 2007. ISSN 00068950. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1730108/>>. Citado na página 93.
- CHOI, M. et al. Genetic diagnosis by whole exome capture and massively parallel DNA sequencing. *Proceedings of the National Academy of Sciences of the United States of America*, v. 106, n. 45, p. 19096–101, nov 2009. ISSN 1091-6490. Disponível em: <www.ncbi.nlm.nih.gov/pmc/articles/PMC2748615/>. Citado na página 3.

- CHONG, J. X. et al. The Genetic Basis of Mendelian Phenotypes: Discoveries, Challenges, and Opportunities. *The American Journal of Human Genetics*, v. 97, n. 2, p. 199–215, aug 2015. ISSN 00029297. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0002929715002451>>. Citado na página 3.
- 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*, v. 38, n. 6, p. 1767–1771, apr 2009. ISSN 03051048. Disponível em: <<http://www.ncbi.nlm.nih.gov/article/fcgi?artid=2847217&tool=pmcentrez&rendertype=ab>>. Citado 2 vezes nas páginas 21 e 24.
- COLLINS, F. S.; VARMUS, H. A New Initiative on Precision Medicine. *The New England journal of medicine*, v. 372, n. 9, p. 793–5, jan 2015. ISSN 1533-4406. Disponível em: <<http://www.nejm.org/doi/abs/10.1056/NEJMp1415160>>. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/25635347>>. Citado na página 13.
- DANECEK, P. et al. The variant call format and VCFtools. *Bioinformatics (Oxford, England)*, v. 27, n. 15, p. 2156–8, aug 2011. ISSN 1367-4811. Disponível em: <<http://www.ncbi.nlm.nih.gov/article/fcgi?artid=3137218&tool=pmcentrez&rendertype=ab>>. Citado na página 25.
- DEPRISTO, M. a. et al. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature genetics*, v. 43, n. 5, p. 491–8, may 2011. ISSN 1546-1718. Disponível em: <<http://www.ncbi.nlm.nih.gov/article/fcgi?artid=3083463&tool=pmcentrez&rendertype=abstract>>. Citado na página 29.
- DURBIN, R. M. et al. A map of human genome variation from population-scale sequencing. *Nature*, v. 467, n. 7319, p. 1061–73, oct 2010. ISSN 0028-0836. Disponível em: <<http://dx.doi.org/10.1038/nature09534>>. Citado na página 12.
- FERNALD, G. H. et al. Bioinformatics challenges for personalized medicine. *Bioinformatics (Oxford, England)*, v. 27, n. 13, p. 1741–8, jul 2011. ISSN 1367-4811. Disponível em: <www.ncbi.nlm.nih.gov/pubmed/21596790>. Citado na página 15.
- FIELD, D. et al. Open software for biologists: from famine to feast. *Nature biotechnology*, v. 24, n. 7, p. 801–3, jul 2006. ISSN 1087-0156. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/16841067>>. Citado na página 19.
- FU, W. et al. Analysis of 6,515 exomes reveals the recent origin of most human protein-coding variants. *Nature*, v. 493, n. 7431, p. 216–20, jan 2013. ISSN 1476-4687. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/23201682>>. Citado na página 13.
- GEBBIA, M. et al. X-linked situs abnormalities result from mutations in ZIC3. *Nature genetics*, 1997. Citado na página 90.
- GRAY, K. a. et al. Genenames.org: the HGNC resources in 2013. *Nucleic acids research*, v. 41, n. Database issue, p. D545–52, jan 2013. ISSN 1362-4962. Disponível em: <www.ncbi.nlm.nih.gov/pubmed/23161694>. Citado na página 33.

- GREEN, R. C. et al. ACMG recommendations for reporting of incidental findings in clinical exome and genome sequencing. *Genetics in medicine : official journal of the American College of Medical Genetics*, v. 15, n. 7, p. 565–74, jul 2013. ISSN 1530-0366. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3692767/>>{tool=pmcentrez}{rendertype=ab}. Citado na página 113.
- GYMREK, M. et al. lobSTR: A short tandem repeat profiler for personal genomes. *Genome Research*, v. 22, n. 6, p. 1154–1162, 2012. ISSN 10889051. Citado na página 40.
- HALÁSZ, Z. et al. Laterality disturbance and hypopituitarism. A case report of co-existing situs inversus totalis and combined pituitary hormone deficiency. *Journal of endocrinological investigation*, v. 31, n. 1, p. 74–8, jan 2008. ISSN 1720-8386. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1829690/>>. Citado na página 90.
- HARAKSINGH, R. R.; SNYDER, M. P. Impacts of variation in the human genome on gene regulation. *Journal of Molecular Biology*, The Authors, v. 425, n. 21, p. 3970–3977, 2013. ISSN 00222836. Disponível em: <[http://dx.doi.org/10.1016/j.jmb.2013.07.015](https://doi.org/10.1016/j.jmb.2013.07.015)>. Citado 2 vezes nas páginas 9 e 12.
- HOGEWEG, P. The roots of bioinformatics in theoretical biology. *PLoS computational biology*, v. 7, n. 3, p. e1002021, mar 2011. ISSN 1553-7358. Disponível em: <www.ncbi.nlm.nih.gov/pmc/articles/PMC311479/>. Citado na página 2.
- HOMER, N.; MERRIMAN, B.; NELSON, S. F. BFAST: an alignment tool for large scale genome resequencing. *PloS one*, v. 4, n. 11, p. e7767, jan 2009. ISSN 1932-6203. Disponível em: <www.ncbi.nlm.nih.gov/pmc/articles/PMC2770639/>{tool=pmcentrez}{rendertype=ab}. Citado na página 28.
- HU, H. et al. X-exome sequencing of 405 unresolved families identifies seven novel intellectual disability genes. *Molecular Psychiatry*, n. August 2014, p. 1–16, 2015. ISSN 1359-4184. Disponível em: <<http://www.nature.com/doifinder/10.1038/mp.2014.193>>. Citado na página 6.
- JAMESON, J. L.; LONGO, D. L. Precision Medicine — Personalized, Problematic, and Promising. *New England Journal of Medicine*, v. 372, n. 23, p. 2229–2234, jun 2015. ISSN 0028-4793. Disponível em: <<http://www.nejm.org/doi/abs/10.1056/NEJMsb1503104>>. Citado na página 13.
- KIRCHER, M. et al. A general framework for estimating the relative pathogenicity of human genetic variants. *Nature genetics*, Nature Publishing Group, v. 46, n. 3, p. 310–5, mar 2014. ISSN 1546-1718. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4487276/>>{tool=pmcentrez}{rendertype=ab}. Citado na página 37.
- KRUMM, N. et al. Copy number variation detection and genotyping from exome sequence data. *Genome research*, v. 22, n. 8, p. 1525–32, aug 2012. ISSN 1549-5469. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3409265/>>{tool=pmcentrez}{rendertype=ab}. Citado na página 40.
- KULIKOWSKI, C. a. et al. AMIA Board white paper: definition of biomedical informatics and specification of core competencies for graduate education in the

- discipline. *Journal of the American Medical Informatics Association*, v. 19, n. 6, p. 931–938, 2012. ISSN 1067-5027. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC334470/>>tool=pmcentrez&rendertype=abstract>. Citado na página 1.
- LANDER, E. S. et al. Initial sequencing and analysis of the human genome. *Nature*, v. 409, n. 6822, p. 860–921, feb 2001. ISSN 0028-0836. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC11237011/>>. Citado na página 12.
- LANGRETH, R.; WALDHOLZ, M. New Era of Personalized Medicine : Targeting Drugs for Each Unique Genetic Profile. *The Oncologist*, v. 4, p. 426–427, 1999. ISSN 1549-490X. Citado na página 13.
- LI, H.; DURBIN, R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)*, v. 25, n. 14, p. 1754–60, jul 2009. ISSN 1367-4811. Disponível em: <<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2705234/>>tool=pmcentrez&rendertype=abstract>. Citado 2 vezes nas páginas 23 e 26.
- LI, J. et al. CONTRA: copy number analysis for targeted resequencing. *Bioinformatics (Oxford, England)*, v. 28, n. 10, p. 1307–13, may 2012. ISSN 1367-4811. Disponível em: <www.ncbi.nlm.nih.gov/pmc/articles/PMC3348560/>tool=pmcentrez&rendertype=ab>. Citado na página 40.
- LI, R. et al. SOAP2: An improved ultrafast tool for short read alignment. *Bioinformatics*, v. 25, n. 15, p. 1966–1967, 2009. ISSN 13674803. Citado na página 28.
- LINHARES, N. et al. Short Communication Modulation of expressivity in PDGFRB-related infantile myofibromatosis: a role for PTPRG? *Genetics and Molecular Research*, v. 13, n. 3, p. 6287–6292, 2014. ISSN 16765680. Disponível em: <<http://www.funpecrp.com.br/gmr/year2014/vol13-3/pdf/gmr5016.pdf>>. Citado na página 97.
- LINHARES, N. D. et al. Exome sequencing identifies a novel homozygous variant in NDRG4 in a family with infantile myofibromatosis. *European Journal of Medical Genetics*, Elsevier Masson SAS, v. 57, n. 11-12, p. 643–648, 2014. ISSN 17697212. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1769721214001712>>. Citado na página 97.
- LIU, X.; JIAN, X.; BOERWINKLE, E. dbNSFP: A lightweight database of human nonsynonymous SNPs and their functional predictions. *Human Mutation*, v. 32, n. 8, p. 894–899, 2011. ISSN 10597794. Citado na página 38.
- MAOJO, V.; KULIKOWSKI, C. A. Bioinformatics and Medical Informatics: Collaborations on the Road to Genomic Medicine? *Journal of the American Medical Informatics Association*, v. 10, n. 6, p. 515–522, nov 2003. ISSN 1067-5027. Disponível em: <<http://jamia.oxfordjournals.org/lookup/doi/10.1197/jamia.M1305>>. Citado na página 2.
- MARTIN-SANCHEZ, F. et al. Synergy between medical informatics and bioinformatics: facilitating genomic medicine for future health care. *Journal of Biomedical*

- Informatics*, v. 37, n. 1, p. 30–42, feb 2004. ISSN 15320464. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S1532046403000856>>. Citado na página 2.
- MIDDLETON, A. et al. Attitudes of nearly 7000 health professionals, genomic researchers and publics toward the return of incidental results from sequencing research. *European Journal of Human Genetics*, Nature Publishing Group, n. February, p. 1–9, 2015. ISSN 1018-4813. Disponível em: <<http://www.nature.com/doifinder/10.1038/ejhg.2015.58>>. Citado na página 113.
- NEESEN, J. et al. Disruption of an inner arm dynein heavy chain gene results in asthenozoospermia and reduced ciliary beat frequency. *Human molecular genetics*, v. 10, n. 11, p. 1117–28, may 2001. ISSN 0964-6906. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/11371505>>. Citado na página 90.
- NG, P. C.; HENIKOFF, S. SIFT: Predicting amino acid changes that affect protein function. *Nucleic Acids Research*, v. 31, n. 13, p. 3812–3814, jul 2003. ISSN 1362-4962. Disponível em: <<http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/gkg509http://linkinghub.elsevier.com/retrieve/pii/S1077314208001331http://www.ncbi.nlm.nih.gov/pubmed/12824425http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC168916>>. Citado na página 37.
- NG, P. C. et al. Genetic variation in an individual human exome. *PLoS genetics*, v. 4, n. 8, p. e1000160, jan 2008. ISSN 1553-7404. Disponível em: <<http://www.ncbi.nlm.nih.gov/articlerender.fcgi?artid=2493042{&}tool=pmcentrez{&}rendertype=ab>>. Citado na página 5.
- NG, S. B. et al. Exome sequencing identifies the cause of a mendelian disorder. *Nature genetics*, Nature Publishing Group, v. 42, n. 1, p. 30–5, jan 2010. ISSN 1546-1718. Disponível em: <<http://dx.doi.org/10.1038/ng.499http://www.ncbi.nlm.nih.gov/articlerender.fcgi?artid=2847889{&}tool=pmcentrez{&}rendertype=abstract>>. Citado na página 6.
- NG, S. B. et al. Targeted capture and massively parallel sequencing of 12 human exomes. *Nature*, Nature Publishing Group, v. 461, n. 7261, p. 272–6, sep 2009. ISSN 1476-4687. Disponível em: <www.ncbi.nlm.nih.gov/pubmed/19684571http://dx.doi.org/10.1038/nature08250>. Citado 2 vezes nas páginas 5 e 7.
- OLBRICH, H. et al. Mutations in DNAH5 cause primary ciliary dyskinesia and randomization of left-right asymmetry. *Nature genetics*, v. 30, n. 2, p. 143–144, feb 2002. ISSN 10614036. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/11788826>>. Citado na página 90.
- PERLES, Z. et al. A human laterality disorder associated with recessive CCDC11 mutation. *Journal of medical genetics*, v. 49, n. 6, p. 386–90, jun 2012. ISSN 1468-6244. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/22577226>>. Citado na página 90.
- ROSSUM, G. V. *Python Software Foundation. Python Language Reference, version 2.7*. 1995. Citado na página 20.

- STENSON, P. D. et al. The Human Gene Mutation Database: 2008 update. *Genome medicine*, v. 1, n. 1, p. 13, jan 2009. ISSN 1756-994X. Disponível em: <<http://www.ncbi.nlm.nih.gov/articlerender.fcgi?artid=2651586&tool=pmcentrez&rendertype=abstractwww.ncbi.nlm.nih.gov/pubmed/19>>. Citado na página 33.
- TAYLOR, M. S.; PONTING, C. P.; COPLEY, R. R. Occurrence and Consequences of Coding Sequence Insertions and Deletions in Mammalian Genomes Occurrence and Consequences of Coding Sequence Insertions and Deletions in Mammalian Genomes. p. 555–566, 2004. ISSN 1088-9051. Citado 2 vezes nas páginas 9 e 12.
- VENTER, J. C. et al. The sequence of the human genome. *Science (New York, N.Y.)*, v. 291, n. 5507, p. 1304–1351, feb 2001. ISSN 0036-8075. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/11181995>>. Citado na página 12.
- VIAROLI, F. Panhypopituitarism, a Cause of Early Sudden Infant Death Syndrome? *Journal of Clinical Case Reports*, v. 02, n. 13, p. 2–4, 2012. ISSN 21657920. Disponível em: <<http://www.omicsgroup.org/journals/2165-7920/2165-7920-2-193.digital/2165-7920-2-193.html>>. Citado na página 90.
- WANG, K.; LI, M.; HAKONARSON, H. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic acids research*, v. 38, n. 16, p. e164, sep 2010. ISSN 1362-4962. Disponível em: <<http://www.ncbi.nlm.nih.gov/articlerender.fcgi?artid=2938201&tool=pmcentrez&rendertype=ab>>. Citado na página 38.
- YANG, Y. et al. Clinical whole-exome sequencing for the diagnosis of mendelian disorders. *The New England journal of medicine*, v. 369, n. 16, p. 1502–11, 2013. ISSN 1533-4406. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/24088041>>. Citado na página 98.

ANEXO A – Modelo utilizado para armazenar os dados de cada indivíduo.

Podemos observar que diferentes tipos de campos de informação foram utilizados como por exemplo ForeignKey, CharField, TextField e FileField

Código A.1 – Modelo utilizado para armazenar os dados de cada indivíduo.

```
from django.db import models
from django.contrib.auth.models import User
from django.conf import settings
from django.utils.text import slugify
from datetime import datetime

class UserGroup(models.Model):
    name = models.CharField(max_length=600)
    members = models.ManyToManyField(User, editable=True,
→     related_name="members", blank=True, null=True)
    def __unicode__(self):
        return self.name

class Individual(models.Model):
    def get_upload_path(self, filename):
        string = "%s/genomes/%s/%s/%s" % (settings.BASE_DIR,
→     slugify(self.user.username), self.id, filename).replace(' ', '_')
        return string
    user = models.ForeignKey(User, editable=False)
```

```
shared_with_users = models.ManyToManyField(User, editable=True,
→ related_name="shared_with_users", blank=True, null=True)

shared_with_groups = models.ManyToManyField(UserGroup, editable=True,
→ related_name="shared_with_groups", blank=True, null=True)

name = models.CharField(max_length=600)

is_featured = models.BooleanField(default=True)

is_public = models.BooleanField(default=False)

vcf_file = models.FileField(upload_to=get_upload_path, blank=True,
→ help_text="File Format: VCF",max_length=600)

vcf_header = models.TextField(null=True, blank=True)

status = models.CharField(max_length=100, blank=True, editable=False)

n_variants = models.IntegerField(null=True, blank=True, editable=False)

n_lines = models.IntegerField(null=True, blank=True, editable=False)

creation_date = models.DateTimeField(auto_now_add=True,null=True,
→ blank=True)

modified_date = models.DateTimeField(null=True, blank=True)

annotation_time = models.CharField(max_length=200, null=True,
→ blank=True)

insertion_time = models.CharField(max_length=200, null=True, blank=True)

insertion_time_mongo = models.CharField(max_length=200, null=True,
→ blank=True)

def __unicode__(self):
    return self.name

def save(self, *args, **kwargs):
    if not self.creation_date:
        self.creation_date = datetime.now()
    self.modified_date = datetime.now()
```

```
    return super(Individual, self).save(*args, **kwargs)

class Group(models.Model):
    name = models.CharField(max_length=128)
    members = models.ManyToManyField(Individual)

    def __unicode__(self):
        return self.name

class ControlGroup(models.Model):
    name = models.CharField(max_length=600)
    vcf_file = models.FileField(upload_to=get_upload_path, blank=True,
→   help_text="File Format: VCF", max_length=600)

    def __unicode__(self):
        return self.name

    def save(self, *args, **kwargs):
        super(ControlGroup, self).save(*args, **kwargs)
        PopulateControls.delay(self.id)

    def get_upload_path(self, filename):
        string = "upload/controls/%s/%s" % (self.id, filename).replace(' ', '_')
        return string

class ControlVariant(models.Model):
    controlgroup = models.ForeignKey(ControlGroup)
    index = models.TextField(db_index=True)#ex. 1-2387623-G-T
```

ANEXO B – Modelo utilizado para armazenar os dados sobre as variantes de cada indivíduo.

Código B.1 – Modelo utilizado para armazenar os dados sobre as variantes de cada indivíduo.

```

from django.db import models
from individuals.models import *

class Variant(models.Model):
    individual = models.ForeignKey(Individual)

    index = models.TextField(db_index=True)#ex. 1-2387623-G-T
    pos_index = models.TextField(db_index=True)#ex. 1-326754756

    #First save all 9 VCF columns
    chr = models.CharField(db_index=True, max_length=100,
    ↵ verbose_name="Chr")
    pos = models.IntegerField(db_index=True)
    variant_id = models.CharField(db_index=True, max_length=100,
    ↵ verbose_name="ID")
    ref = models.TextField(null=True, blank=True)
    alt = models.TextField(null=True, blank=True)
    qual = models.FloatField(db_index=True)
    filter = models.CharField(db_index=True, max_length=100)
    info = models.TextField(null=True, blank=True)

```

```
format = models.TextField(null=True, blank=True)

genotype_col = models.TextField(null=True, blank=True)
genotype = models.CharField(db_index=True, max_length=100)

#metrics from genotype_info DP field
read_depth = models.IntegerField(db_index=True)

gene = models.TextField(db_index=True, null=True, blank=True)
mutation_type = models.CharField(db_index=True, null=True,
→ max_length=100)
vartype = models.CharField(db_index=True, null=True, max_length=100)

#Annotation From 1000genomes
genomes1k_maf = models.FloatField(db_index=True, null=True, blank=True,
→ verbose_name="1000 Genomes Frequency")
dbsnp_maf = models.FloatField(db_index=True, null=True, blank=True,
→ verbose_name="dbSNP Frequency")
esp_maf = models.FloatField(db_index=True, null=True, blank=True,
→ verbose_name="ESP6500 Frequency")

#dbsnp
# dbsnp_pm = models.TextField(db_index=True, null=True, blank=True)
# dbsnp_clnsig = models.TextField(db_index=True, null=True, blank=True)
dbsnp_build = models.IntegerField(db_index=True, null=True)

#VEP
sift = models.FloatField(db_index=True, null=True, blank=True)
sift_pred = models.TextField(db_index=True, null=True, blank=True)

polyphen2 = models.FloatField(db_index=True, null=True, blank=True)
```

```
polyphen2_pred = models.TextField(db_index=True, null=True, blank=True)

condel = models.FloatField(db_index=True, null=True, blank=True)
condel_pred = models.TextField(db_index=True, null=True, blank=True)

vest = models.FloatField(db_index=True, null=True, blank=True)
cadd = models.FloatField(db_index=True, null=True, blank=True)

cadd = models.FloatField(db_index=True, null=True, blank=True)
rf_score = models.FloatField(db_index=True, null=True, blank=True)
ada_score = models.FloatField(db_index=True, null=True, blank=True)

#hi_index
# hi_index_str = models.TextField(null=True, blank=True)
# hi_index = models.FloatField(db_index=True, null=True, blank=True)
# hi_index_perc = models.FloatField(db_index=True, null=True,
↪ blank=True)

#OMIM
is_at_omim = models.BooleanField(default=False)

#HGMD
is_at_hgmd = models.BooleanField(default=False)
hgmd_entries = models.TextField(null=True, blank=True)

#snpeff annotation
snpeff_effect = models.TextField(db_index=True, null=True, blank=True)
snpeff_impact = models.TextField(db_index=True, null=True, blank=True)
snpeff_func_class = models.TextField(db_index=True, null=True,
↪ blank=True)
```

```
snpeff_codon_change = models.TextField(db_index=True, null=True,
↪ blank=True)

snpeff_aa_change = models.TextField(db_index=True, null=True,
↪ blank=True)

# snpeff_aa_len = models.TextField(db_index=True, null=True, blank=True)

snpeff_gene_name = models.TextField(db_index=True, null=True,
↪ blank=True)

snpeff_biotype = models.TextField(db_index=True, null=True, blank=True)

snpeff_gene_coding = models.TextField(db_index=True, null=True,
↪ blank=True)

snpeff_transcript_id = models.TextField(db_index=True, null=True,
↪ blank=True)

snpeff_exon_rank = models.TextField(db_index=True, null=True,
↪ blank=True)

# snpeff_genotype_number = models.TextField(db_index=True, null=True,
↪ blank=True)

#vep annotation

vep_allele = models.TextField(db_index=True, null=True, blank=True)

vep_gene = models.TextField(db_index=True, null=True, blank=True)

vep_feature = models.TextField(db_index=True, null=True, blank=True)

vep_feature_type = models.TextField(db_index=True, null=True,
↪ blank=True)

vep_consequence = models.TextField(db_index=True, null=True, blank=True)

vep_cdna_position = models.TextField(db_index=True, null=True,
↪ blank=True)

vep_cds_position = models.TextField(db_index=True, null=True,
↪ blank=True)

vep_protein_position = models.TextField(db_index=True, null=True,
↪ blank=True)

vep_amino_acids = models.TextField(db_index=True, null=True, blank=True)
```

```
vep_codons = models.TextField(db_index=True, null=True, blank=True)
vep_existing_variation = models.TextField(db_index=True, null=True,
↪ blank=True)
vep_distance = models.TextField(db_index=True, null=True, blank=True)
vep_strand = models.TextField(db_index=True, null=True, blank=True)
vep_symbol = models.TextField(db_index=True, null=True, blank=True)
vep_symbol_source = models.TextField(db_index=True, null=True,
↪ blank=True)
vep_sift = models.TextField(db_index=True, null=True, blank=True)
vep_polyphen = models.TextField(db_index=True, null=True, blank=True)
vep_condel = models.TextField(db_index=True, null=True, blank=True)

def get_fields(self):
    return [(field.name, field.verbose_name.title().replace('_', ' '
↪ ''))
        for field in Variant._meta.fields]
```

ANEXO C – BWA - Script desenvolvido para realizar o alinhamento dos arquivos em formato FASTQ.

Código C.1 – BWA - Script desenvolvido para realizar o alinhamento dos arquivos em formato FASTQ.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from optparse import OptionParser
import os

#python bwa.py -f reads1.fastq -q reads2.fastq

parser = OptionParser()
usage = "usage: %prog [options] -f reads1.fastq -q reads2.fastq"
parser = OptionParser(usage=usage)

parser.add_option("-f", dest="reads1",
                  help="reads 1 in FASTQ format", metavar="FASTQ")
parser.add_option("-q", dest="reads2",
                  help="reads 2 in FASTQ format", metavar="FASTQ")

(options, args) = parser.parse_args()

reads1=options.reads1
```

```
reads2=options.reads2

bwa_dir="/lgc/programs/bwa/"
st_dir="/lgc/programs/samtools"
reference="/lgc/datasets/gatk_data/b37/human_g1k_v37.fasta"

class Bwa():
    def __init__(self, reads1, reads2):
        """
        Execute the alignment of the two fastq files from input.

        """
        print "Start Analysis..."
        self.alignment()
        self.sai_to_sam()
        self.sam_to_bam()
        #self.clean_files()

    def alignment(self):
        print "Aligning reads..."
        command = "%s/bwa aln -t 24 -I %s %s > reads1.sai" % (bwa_dir,
        ↵ reference, reads1)
        os.system(command)
        command = "%s/bwa aln -t 24 -I %s %s > reads2.sai" % (bwa_dir,
        ↵ reference, reads2)
        os.system(command)

    def sai_to_sam(self):
        print "Convert Sai to SAM"
```

```
        command = """%s/bwa sampe %s -r
↪   "@RG\tID:Exome\tLB:Exome\tSM:Exome\tPL:ILLUMINA" reads1.sai reads2.sai
↪   %s %s > exome.sam"""\ % (bwa_dir, reference, reads1, reads2)

        os.system(command)

def sam_to_bam(self):
    print "Convert SAM to BAM"
    #import BAM
    command = "%s/samtools view -bS exome.sam > exome.bam" % (st_dir)#
    ↪   reference
    os.system(command)
    # #Sort BAM
    command = "%s/samtools sort exome.bam exome.sorted" % (st_dir)
    os.system(command)
    # #Index BAM
    command = "%s/samtools index exome.sorted.bam exome.sorted.bam.bai" %
    ↪   (st_dir)
    os.system(command)
    #Calmd
    #command = "%s/samtools calmd -Abr exome.sorted.bam %s >
    ↪   exome.baq.bam" % (st_dir, reference)
    #os.system(command)

def clean_files(self):
    command = "rm exome.sam exome.bam reads1.sai reads2.sai" % (st_dir)
    os.system(command)
    command = "mv exome.sorted.bam exome.bam" % (st_dir)
    os.system(command)
    command = "mv exome.sorted.bam.bai exome.bam.bai" % (st_dir)
    os.system(command)
```

```
if __name__ == '__main__':
    Bwa(reads1, reads2)
```

ANEXO D – GATK- Script desenvolvido para executar o programa GATK.

Código D.1 – GATK- Script desenvolvido para executar o programa GATK.

```

#!/usr/bin/env python

# -*- coding: utf-8 -*-

from optparse import OptionParser
import os
from time import time
import datetime

__author__ = "Raony Guimaraes"
__copyright__ = "Copyright 2015, The Exome Pipeline"
__credits__ = ["Raony Guimaraes"]
__license__ = "GPL"
__version__ = "1.2"
__maintainer__ = "Raony Guimaraes"
__email__ = "raonyguimaraes@gmail.com"
__status__ = "Production"

#run example
#python gatk.py -i alignment/exome.sorted.bam

parser = OptionParser()
usage = "usage: %prog [options] -f reads1.fastq -q reads2.fastq"

```

```
parser = OptionParser(usage=usage)

parser.add_option("-i", dest="input_file",
                  help="BAM File Sorted in BAM format", metavar="BAM")
parser.add_option("-t", dest="target_array",
                  help="Target Array", metavar="BEDFILE")

(options, args) = parser.parse_args()

input_file=options.input_file
target_array=options.target_array

#PROGRAMS

gatk_dir="/lgc/programs/GenomeAnalysisTK-1.5-21-g979a84a"
gatk_dir="/lgc/programs/GenomeAnalysisTK-2.3-0-g9593e74"

pic_dir="/lgc/programs/picard-tools-1.66/picard-tools-1.66"
st_dir="/lgc/programs/samtools"

sting_dir = "/lgc/programs/Sting/dist/"
sting_res = "/lgc/programs/Sting/public/R/"

#FOLDERS

log_dir = "logs"

reference="/lgc/datasets/gatk_data/b37/human_g1k_v37.fasta"
#reference="/lgc/datasets/hg19/Homo_sapiens_assembly19.fasta"
```

```
dbsnp="/lgc/datasets/dbsnp/dbsnp-135.vcf"
dbsnp="/lgc/datasets/dbsnp/137/00-All.vcf"

omni="/lgc/datasets/gatk_data/b37_resources/1000G_omni2.5.b37.sites.vcf"
hapmap="/lgc/datasets/gatk_data/b37_resources/hapmap_3.3.b37.sites.vcf"
indels="/lgc/datasets/gatk_data/b37_resources/1000G_phase1.indels.b37.vcf"
exome_dataset="/lgc/datasets/exome_datasets/292_illumina_samples_from_bi_and_washu.vcf

exon_10bp = "/lgc/datasets/ucsc/refseq_plus10.withoutchr.bed"

class Gatk():
    def __init__(self, input_file):
        print "Starting GATK..."

        #You can change the order of the tasks :D
        starttime = datetime.datetime.now()
        print "Start Time: %s" % (starttime)

        input_file = self.MarkDuplicates(input_file, True)
        input_file = self.LocalRealignmentAroundIndels(input_file, True)
        input_file = self.BaseQualityScoreRecalibration(input_file, True)
        #CALCULATE Depth of Coverage
        self.DepthofCoverage(input_file, True)
        input_file = self.UnifierGenotyper(input_file, True)
        self.VariantQualityScoreRecalibration(input_file)
        #die()
        #self.VariantEval(input_file)
        #self.Dindel(input_file)

        timetaken = datetime.datetime.now() - starttime
```

```
print "Time Taken: %s" % (timetaken)

#don't forget to clean files after processing!!!!!

def MarkDuplicates(self, input_file, flag=True):

    print "Mark Duplicates..."

    filename = self.getfilename(input_file)
    output_file = "%s.dedup.bam" % (filename)

    command = """
        java -Xmx40g -Djava.io.tmpdir=/projects/tmp -jar %s/MarkDuplicates.jar
        ↵   \
        INPUT=%s \
        REMOVE_DUPLICATES=true \
        VALIDATION_STRINGENCY=LENIENT \
        AS=true \
        METRICS_FILE=%s.dedup.metrics \
        OUTPUT=%s """ % (pic_dir, input_file, filename, output_file)
    if flag:
        os.system(command)

    #Reindex BAM File
    command = "%s/samtools index %s %s.bai" % (st_dir, output_file,
    ↵   output_file)
    if flag:
        os.system(command)
```

```
os.system(command)

return (output_file)

def LocalRealignmentAroundIndels(self, input_file, flag=True):

    print "Starting LocalRealignmentAroundIndels..."

    filename = self.getfilename(input_file)
    output_file = "%s.real.fixed.bam" % (filename)

    print "Starting RealignerTargetCreator..."
    command = """
        java -Xmx40g -Djava.io.tmpdir=/projects/tmp -jar
        %s/GenomeAnalysisTK.jar -T RealignerTargetCreator \
        -l INFO \
        -R %s \
        -I %s \
        --known %s \
        --known %s \
        -log %s/RealignerTargetCreator.log \
        -o %s.intervals """ % (gatk_dir, reference, input_file, dbsnp, indels,
        log_dir, filename)

        #-B:intervals,BED $EXON_CAPTURE_FILE \
        #-L $EXON_CAPTURE_FILE \
        if flag:
            os.system(command)
```



```
return(output_file)

def BaseQualityScoreRecalibration(self, input_file, flag=True):
    print "Starting Base Quality Score Recalibration..."

    filename = self.getfilename(input_file)
    output_file = "%s.recal.bam" % (filename)

#3.1 CountCovariates Before

    command = """
        java -Xmx40g -Djava.io.tmpdir=/projects/tmp -jar
        %s/GenomeAnalysisTK.jar -T CountCovariates \
        -l INFO \
        -I %s \
        -R %s \
        -knownSites %s \
        -cov ReadGroupCovariate -cov QualityScoreCovariate -cov CycleCovariate
        -cov DinucCovariate \
        -recalFile %s.before.csv \
        -log %s/CountCovariates_before.log \
        -nt 4
    """ % (gatk_dir, input_file, reference, dbsnp, filename, log_dir)

    if flag:
        os.system(command)

#3.2 TableRecalibration (try -baq RECALCULATE)
```

```
command = """  
  
java -Xmx40g -Djava.io.tmpdir=/projects/tmp -jar  
↪ %s/GenomeAnalysisTK.jar -T TableRecalibration \  
-l INFO \  
-I %s \  
-R %s \  
-recalFile %s.before.csv \  
-log %s/TableRecalibration.log \  
-o %s """ % (gatk_dir, input_file, reference, filename, log_dir,  
↪ output_file)  
  
if flag:  
    os.system(command)  
  
  
#Reindex Bam FILE  
  
command = "%s/samtools index %s %s.bai" % (st_dir, output_file,  
↪ output_file)  
  
if flag:  
    os.system(command)  
  
  
# 3.2.1 CountCovariates After  
  
command = """  
  
java -Xmx12g -jar %s/GenomeAnalysisTK.jar -T CountCovariates \  
-l INFO \  
-I %s \  
-R %s \  
-knownSites %s \  
-cov ReadGroupCovariate -cov QualityScoreCovariate -cov CycleCovariate  
↪ -cov DinucCovariate \  
-recalFile %s.after.csv \  
-log %s/CountCovariates_after.log \  
""" % (gatk_dir, input_file, reference, filename, log_dir, output_file)
```

```
-nt 4 """ % (gatk_dir, output_file, reference, dbsnp, filename,
↪ log_dir)

if flag:
    os.system(command)

#3.3.2 Analyze Covariates - Generate Graphs Before and After Table
↪ Recalibration

#Create output folders
if flag:
    os.system("mkdir %s.recal.stats.before" % (filename))
    if flag:
        os.system("mkdir %s.recal.stats.after" % (filename))

# #Generate graphs before
command = """
    java -Xmx40g -Djava.io.tmpdir=/projects/tmp -jar
↪ %s/AnalyzeCovariates.jar \
    -l INFO \
    --recal_file %s.before.csv \
    -outputDir %s.recal.stats.before/ \
    -Rscript /usr/bin/Rscript \
    -ignoreQ 5 \
    -resources %s """ % (sting_dir, filename, filename, sting_res)
if flag:
    os.system(command)

#Generate graphs after
```

```
command = """  
  
java -Xmx40g -Djava.io.tmpdir=/projects/tmp -jar  
↪ %s/AnalyzeCovariates.jar \  
    -l INFO \  
    --recal_file %s.after.csv \  
    -outputDir %s.recal.stats.after/ \  
    -Rscript /usr/bin/Rscript \  
    -ignoreQ 5 \  
    -resources %s """ % (sting_dir, filename, filename, sting_res)  
  
if flag:  
  
    os.system(command)  
  
  
return(output_file)  
  
  
def DepthofCoverage(self, input_file, flag=True):  
  
    print "Starting DepthOfCoverage..."  
  
    filename = self.getfilename(input_file)  
  
  
    command = """  
  
java -Xmx40g -Djava.io.tmpdir=/projects/tmp -jar  
↪ %s/GenomeAnalysisTK.jar -T DepthOfCoverage \  
    -I %s \  
    -R %s \  
    -o %s_depthofcoverage \  
    -L %s \  
    -log %s/DepthofCoverage.log \  
    """ % (gatk_dir, input_file, reference, filename, target_array,  
↪ log_dir)  
  
if flag:  
  
    os.system(command)
```

```
#[-geneList refSeq.sorted.txt] \
#-ct 4 -ct 6 -ct 10 \
#[ -L my_capture_genes.interval_list]

def UnifierGenotyper(self, input_file, flag=True):
    print "Starting UnifierGenotyper..."

    filename = self.getfilename(input_file)
    output_file = "%s.raw.vcf" % (filename)
    # # #Standard Raw VCF
    command = """
        java -Xmx40g -Djava.io.tmpdir=/projects/tmp -jar
    ↳ %s/GenomeAnalysisTK.jar -T UnifiedGenotyper \
        -l INFO \
        -I %s \
        -R %s \
        --dbsnp %s \
        --genotype_likelihoods_model BOTH \
        -stand_call_conf 50.0 \
        -stand_emit_conf 10.0 \
        -dcov 50 \
        -A AlleleBalance \
        -A DepthOfCoverage \
        -A FisherStrand \
        -o %s \
        -log %s/UnifiedGenotyper.log \
        -L %s \
        -nt 4
    """ % (gatk_dir, input_file, reference, dbsnp, output_file, log_dir,
    ↳ target_array)
```

```
if flag:
    os.system(command)
    #-B:targetIntervals,BED $EXON_CAPTURE_FILE \
    #-B:intervals,BED $EXON_CAPTURE_FILE \
    #-L targets.interval_list
    return (output_file)

def VariantQualityScoreRecalibration(self, input_file, flag=True):
    print "Starting Variant Quality Score Recalibration..."
    filename = self.getfilename(input_file)
    #output_file = "%s.snps.vcf" % (filename)

    print "SelectVariants - Select SNPs from the unified genotype raw
    ↳ VCF"
    command = """
        java -Xmx4g -Djava.io.tmpdir=/projects/tmp -jar
    ↳ %s/GenomeAnalysisTK.jar -T SelectVariants \
        -R %s \
        --variant %s \
        -selectType SNP \
        -log %s>SelectVariants.snps.log \
        -o %s.snps.vcf """ % (gatk_dir, reference, input_file, log_dir,
    ↳ filename)
    if flag:
        os.system(command)

    #create folder
    command = "mkdir VariantRecalibrator"
    if flag:
        os.system(command)
```

```
print "Start VariantRecalibrator"

command = """

java -Xmx4g -Djava.io.tmpdir=/projects/tmp -jar
↪ %s/GenomeAnalysisTK.jar -T VariantRecalibrator \
-R %s \
-input %s.snps.vcf \
-resource:hapmap,known=false,training=true,truth=true,prior=15.0 %s \
-resource:omni,known=false,training=true,truth=false,prior=12.0 %s \
-resource:dbsnp,known=true,training=false,truth=false,prior=8.0 %s \
-resource:exome_dataset,known=true,training=true,truth=true,prior=8.0

↪ %s \
-an QD -an HaplotypeScore -an MQRankSum -an ReadPosRankSum -an FS -an
↪ MQ \
--maxGaussians 4 \
-recalFile VariantRecalibrator/%s.recal \
-tranchesFile VariantRecalibrator/%s.tranches \
-rscriptFile VariantRecalibrator/%s.plots.R \
-log %s/VariantRecalibrator.log \
-nt 4 """ % (gatk_dir, reference, filename, hapmap, omni, dbsnp,
↪ exome_dataset, filename, filename, log_dir)

if flag:
    os.system(command)

##percentBadVariants 0.05 \
#-mode SNP \
#Apply Recalibrator

command = """
```

```
java -Xmx4g -Djava.io.tmpdir=/projects/tmp -jar
↪ %s/GenomeAnalysisTK.jar -T ApplyRecalibration \
-R %s \
-input %s.snps.vcf \
-tranchesFile VariantRecalibrator/%s.tranches \
-recalFile VariantRecalibrator/%s.recal \
--ts_filter_level 99.0 \
-log %s/ApplyRecalibration.log \
-o %s.snps.recal.vcf """ % (gatk_dir, reference, filename, filename,
↪ filename, log_dir, filename)

if flag:
    os.system(command)

#Filtering SNPS Variant Filtration
command = """
java -Xmx4g -Djava.io.tmpdir=/projects/tmp -jar
↪ %s/GenomeAnalysisTK.jar -T VariantFiltration \
-R %s \
--variant %s.snps.recal.vcf \
--clusterWindowSize 10 \
--filterExpression "MQ0 >= 4 && ((MQ0 / (1.0 * DP)) > 0.1)" \
--filterName "HARD_TO_VALIDATE" \
--filterExpression "DP < 5" \
--filterName "LowCoverage" \
--filterExpression "QUAL < 30.0" \
--filterName "VeryLowQual" \
--filterExpression "QUAL > 30.0 && QUAL < 50.0" \
--filterName "LowQual" \
--filterExpression "QD < 1.5" \
--filterName "LowQD" \
--filterExpression "SB > -10.0" \
```

```
--filterName "StrandBias" \
-o %s.snps.recal.filtered.vcf """ % (gatk_dir, reference, filename,
↪ filename)

if flag:
    os.system(command)

print "SelectVariants.Select Indels from the Unified Genotyper raw
↪ VCF"

    command = """
        java -Xmx4g -Djava.io.tmpdir=/projects/tmp -jar
↪ %s/GenomeAnalysisTK.jar -T SelectVariants \
        -R %s \
        --variant %s \
        -selectType INDEL \
        -log %s>SelectVariants.indels.log \
        -o %s.indels.vcf """ % (gatk_dir, reference, input_file, log_dir,
↪ filename)

    if flag:
        os.system(command)

#Variant Filtration

    command = """
        java -Xmx4g -Djava.io.tmpdir=/projects/tmp -jar
↪ %s/GenomeAnalysisTK.jar -T VariantFiltration \
        -R %s \
        --variant %s.indels.vcf \
        --filterExpression "QD < 2.0 || ReadPosRankSum < -20.0 || FS > 200.0"
↪ \
        --filterName GATK_standard \
        --missingValuesInExpressionsShouldEvaluateAsFailing \
        -log %s/VariantFiltration.indels.log \
    """
```

```
-o %s.indels.filtered.vcf """ % (gatk_dir, reference, filename,
↪ log_dir, filename)

if flag:
    os.system(command)

#CombineVariants: Combine Recalibrated SNPs and Filtered Indels
command = """
java -Xmx4g -Djava.io.tmpdir=/projects/tmp -jar
↪ %s/GenomeAnalysisTK.jar -T CombineVariants \
-R %s \
--variant %s.snps.recal.vcf \
--variant %s.indels.filtered.vcf \
-genotypeMergeOptions UNIQUIFY \
-log %s/CombineVariants.log \
-o %s.snps.indels.vcf """ % (gatk_dir, reference, filename,
↪ log_dir, filename)

if flag:
    os.system(command)

command = """grep "^#" %s.snps.indels.vcf > exome.passed.vcf"""\ %
↪ (filename)
if flag:
    os.system(command)

command = """grep "PASS" %s.snps.indels.vcf >> exome.passed.vcf"""\ %
↪ (filename)
if flag:
    os.system(command)

def VariantEval(self, input_file, flag=True):
```

```
filename = self.getfilename(input_file)

#VariantEval

command = """
java -Xmx4g -Djava.io.tmpdir=/projects/tmp -jar
↪ %s/GenomeAnalysisTK.jar -T VariantEval \
-R %s \
-B:dbsnp,VCF %s \
-o %s.eval.gatkreport \
-B:eval,VCF $INPUT \
-l INFO
-nt 4 """ % (gatk_dir, reference, dbsnp, filename, input_file)

if flag:
    os.system(command)

def getfilename(self, filepath):

    filename = ".".join(filepath.split("/")[-1].split(".")[: -1])
    return filename

if __name__ == '__main__':
    Gatk(input_file)
```

ANEXO E – Unified Genotyper - GATK -

Script desenvolvido para realizar o Calling de Variantes

Código E.1 – Unified Genotyper - GATK - Script desenvolvido para realizar o Calling de Variantes.

```

#!/usr/bin/env python
# -*- coding: utf-8 -*-

from optparse import OptionParser
import os
from time import time
import datetime
import re
#Example
#python /projects/12exomes/data/unifiergenotyper.py -i
    ↳ /projects/12exomes/data/Exome_3_EDS.realigned-recalibrated.bam
    ↳ /projects/12exomes/data/Exome_4_ELS.realigned-recalibrated.bam
    ↳ /projects/12exomes/data/Exome_5_LS.realigned-recalibrated.bam
    ↳ /projects/12exomes/data/Exome_6_DC.realigned-recalibrated.bam -o
    ↳ quartet_only_SNPs_.vcf -t
    ↳ /lgc/datasets/exome_targets/SureSelect_All_Exon_V2_hg19.20110105.bed

__author__ = "Raony Guimaraes"
__copyright__ = "Copyright 2015, The Exome Pipeline"

```

```

__credits__ = ["Raony Guimaraes"]
__license__ = "GPL"
__version__ = "1.0.1"
__maintainer__ = "Raony Guimaraes"
__email__ = "raonyguimaraes@gmail.com"
__status__ = "Production"

#run example
#python unifiergenotyper.py -i
    ↳ /projects/12exomes/data/Exome_3_EDS.realigned-recalibrated.bam
    ↳ /projects/12exomes/data/Exome_4_ELS.realigned-recalibrated.bam
    ↳ /projects/12exomes/data/Exome_5_LS.realigned-recalibrated.bam
    ↳ /projects/12exomes/data/Exome_6_DC.realigned-recalibrated.bam

parser = OptionParser()
usage = "usage: %prog [options] -f reads1.fastq -q reads2.fastq"
parser = OptionParser(usage=usage)

parser.add_option("-i", dest="input_file",
                  help="BAM File Sorted in BAM format", metavar="BAM") #,
    ↳ nargs=12
parser.add_option("-t", dest="target_array",
                  help="Target Array", metavar="BEDFILE")
parser.add_option("-o", dest="output",
                  help="Output Filename", metavar="VCFFILE")

(options, args) = parser.parse_args()

input_file=options.input_file

filename = ".".join(input_file.split("/")[-1].split(".")[:-1])

```

```
target_array=options.target_array

#reference="/lgc/datasets/gatk_data/b37/human_g1k_v37.fasta"
reference="/lgc/datasets/gatk_data/b37/human_g1k_v37chr.fasta"
#dbsnp="/lgc/datasets/dbsnp/dbsnp-135chr.vcf"

#reference="/lgc/datasets/gatk_data/b37/human_g1k_v37chr.fasta"
#reference="/lgc/datasets/hg19/all/hg19_cancer.fasta"
#reference="/lgc/datasets/gatk_data/hg19/ucsc.hg19.fasta"
#dbsnp="/lgc/datasets/dbsnp/137/00-All.vcf"

dbsnp="/lgc/datasets/dbsnp/141/All.vcf"

gatk_dir="/lgc/programs/GenomeAnalysisTK-3.1-1"

command = """
java -Xmx40g -Djava.io.tmpdir=/projects/tmp -jar
→ %s/GenomeAnalysisTK.jar -T UnifiedGenotyper \
-R %s \
-I %s \
-l INFO \
--dbsnp %s \
-A AlleleBalance \
-stand_call_conf 50.0 \
-stand_emit_conf 10.0 \
-dcov 200 \
-o %s \
-log %s-UnifiedGenotyper.log \
-L %s \
-nt 16 \
```

```
""" % (gatk_dir, reference, input_file, dbsnp, options.output,
→ options.output, re.escape(target_array))
```

```
print command
```

```
os.system(command)
```