Annotating the genomic variants in Next Generation Sequencing data through computational approach

By

Mehwish Noureen

NUST201463256MRCMS64014F

Research Center for Modeling and Simulation
National University of Sciences & Technology
Islamabad, Pakistan

2016

Annotating the genomic variants in Next Generation Sequencing data through computational approach

By

Mehwish Noureen

NUST201463256MRCMS64014F

A thesis submitted in partial fulfillment of the requirement for the degree of
Masters in Computational Sciences and Engineering

With
Majors in Bioinformatics

Research Center for Modeling and Simulation
National University of Sciences & Technology
Islamabad, Pakistan 2016

FORM TH-4

National University of Sciences & Technology

MASTER THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by: **Mehwish Noureen (NUST201463256MRCMS64014F)** Titled: **Annotating the genomic variants in Next Generation Sequencing data through computational approach** be accepted in partial fulfillment of the requirements for the award of **MS CS&E** degree with (________ Grade).

Examination Committee Members

1. Name: **Dr. Rehan Zafar Paracha** Signature: ________________
2. Name: **Dr. Mehak Rafiq** Signature: ________________
3. Name: **Dr. Rabia Amir** Signature: ________________

Supervisor's name: **Dr. Shumaila Sayyab** Signature: _______________

Date: __________________

________________________ ________________________

Head of Department Date

COUNTERSIGNED

Date: __________________ ____________________

Dean/Principal

Declaration

I hereby declare that the work presented in the following thesis is my own effort, except where otherwise acknowledged, and that the thesis is my own composition. No part of the thesis has been previously presented for any other degree.

Mehwish Noureen

NUST201463256MRCMS64014F

DEDICATED TO

MY BELOVED PARENTS

Acknowledgements

First and Foremost praise is to **ALLAH Almighty**, the greatest of all, on whom ultimately we depend for sustenance and guidance and my utmost respect to His last **Prophet (P.B.U.H.)**. I would like to thank Almighty Allah for giving me opportunity, determination and strength to do my research.

I would like to thank and express my deep and sincere gratitude to my supervisor **Dr. Shumaila Sayyab** for her continuous support, guidance and encouragement. I am also thankful to members of my thesis Guidance and Examination Committee **Dr. Mehak Rafiq**, **Dr. Rehan Zafar Paracha** and **Dr. Rabia Amir** for their motivation and suggestions.

I take this opportunity to express gratitude to all of the RCMS faculty members for their help and support.

I would like to thank my friends Maria Altaf Satti, Qurat ul ain, Tabeer Fatima and Fauzia Ehsan for their support and help during these two years.

I wish to express my thanks and gratitude to my parents, the ones who can never ever be thanked enough, for the overwhelming love and care they bestow upon me, and who have supported me financially as well as morally and without whose proper guidance it would have been impossible for me to complete my higher education.

I would like to thank my sisters and brothers for their support, guidance and unconditional love.

Mehwish Noureen

Abstract

Next Generation Sequencing (NGS) has made possible the parallel analysis of sequencing data. Different NGS platforms are generating huge amount of data. This data undergoes different analysis pipeline depending upon the platform from which it is produced. Bioinformatics has made this analysis easier by the development of different tools and pipelines. Several pipelines are available for performing different types of analysis which includes variant calling, phylogenetic analysis and many others. In variant calling, once the variants have been called, it is important to identify their biological significance and their location in the genome. This can be helpful to identify their roles in different disease. Several tools are available for this purpose. Each tool has its own features, with certain limitations. Some tools require the annotation sets from the user, while others require programming skills to use it. So, an automated pipeline is required to overcome these limitations. This thesis provides the detail about an automated pipeline implemented in R programming language named as AutoAnnotate. AutoAnnotate takes only the VCF file as an input and generates the annotation results for the user, along with different charts and tables representing annotation information in different ways.

Table of Contents

List of Figures

List of Tables

Acronyms

DNA	Deoxyribonucleic acid
NGS	Next Generation Sequencing
SNP	Single Nucleotide Polymorphism
SFF	Standard flowgram format
PCR	Polymerase chain reaction
INDEL	Insertion and Deletion
RNA	Ribonucleic acid
mRNA	Messenger RNA
cDNA	Complementary DNA
BLAST	Basic local alignment search tool
MAQ	Mapping and Assembly with Quality
dbSNP	Single Nucleotide Polymorphism Database
OMIM	Online Mendelian Inheritance in Man
HGMD	Human Gene Mutation Database
VCF	Variant Calling Format
SAM	Sequence Alignment Map
BAM	Binary Alignment Map
GATK	Genome Analysis Toolkit
BWA	Burrows-Wheeler Aligner
UCSC	University of California, Santa Cruz
SeqAnt	Sequence Annotator
CNVs	Copy Number Variations
SVA	Sequence Variant Analyzer

VASST	Variant Annotation, Analysis and Search Tool
GVF	Genome Variation Format
SNVs	Single Nucleotide Variants
VAT	Variant Analysis Tool
ENA	European Nucleotide Archive
PROVEAN	Protein Variation Effect Analyzer
NCBI	National Center for Biotechnology Information
GTF	Gene Transfer Format
GUI	Graphical User Interface
GO	Gene Ontology
HGVBase	Human Genome Variation Database
BED	Browser Extensible Data
UTR	Untranslated Region
BWA-SW	Burrows-Wheelers Aligner, Smith-Waterman alignment
HOPE	Have (y)OUR Protein Explained

CHAPTER 1

INTRODUCTION

CHAPTER 1

1. Introduction

The history of the DNA sequencing technology is quite rich and broad [1, 2]. In order to determine the DNA sequence chemical methods were introduced by different groups in late 1970’s. These methods involved the mechanism of cleaving the DNA chemically [3] or by adding the dideoxy-nucleotides when the DNA is being synthesized [4]. Later in 1990’s semi-automated implementation of Sanger’s method was used to sequence the DNA [5-7]. Figure 1.1 depicts the Sanger sequencing [8]. Clinicians and researchers were able to sequence the DNA on daily basis by the discovery of fluorescently labeled dideoxy-nucleotides [9] and automated capillary electrophoresis [10]. The base accuracy obtained by this technique is >99.9% [11] and read lengths up to 1000 base pairs can be obtained [8]. Human genome sequencing [12, 13] revealed that Sanger platform was not capable of diploid genome analysis. Some of the improvements were made to resolve these issues [14]. However, these shortcomings are intrinsic in nature [11]. The Sanger’s automated method is referred to as the ‘first generation’ and the latest techniques as the next generation sequencing (NGS) [15].

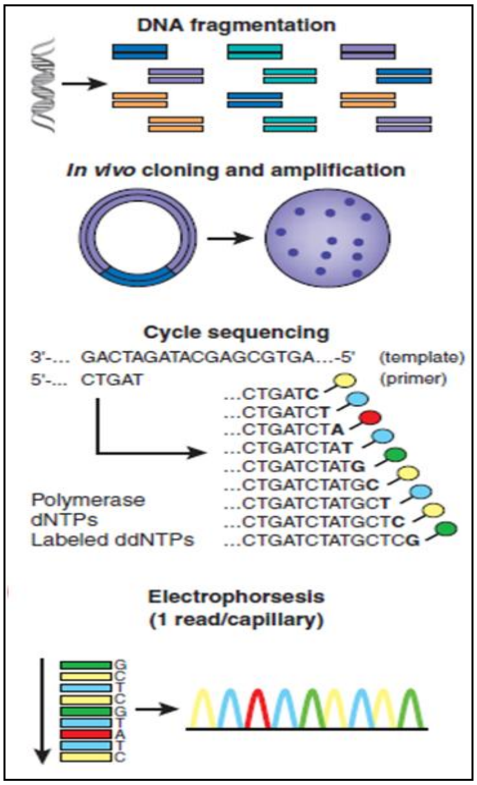


Figure 1.1 Sanger Sequencing [8]

1.1 Next Generation Sequencing

NGS has revolutionized the scientist thinking towards the different areas of research [15]. NGS is different from other methods as it has reduced the cost and has made possible the parallel analysis [16]. Multiple platforms of NGS exist in the market. Each platform has some advantage over the other with respect to the applications in which they are used [15]. Different NGS technologies which are available commercially today include 454 pyrosequencing (Roche), Illumina genome analyzer (Solexa), the SOLiD and Helicos [17].

1.2 NGS Platforms

1.2.1 Roche 454

The first next generation sequencing system which was made commercially available was Roche 454 [16]. The technology used by this system is known as pyrosequencing [18]. Figure 1.2 shows the pyrosequencing method [8].

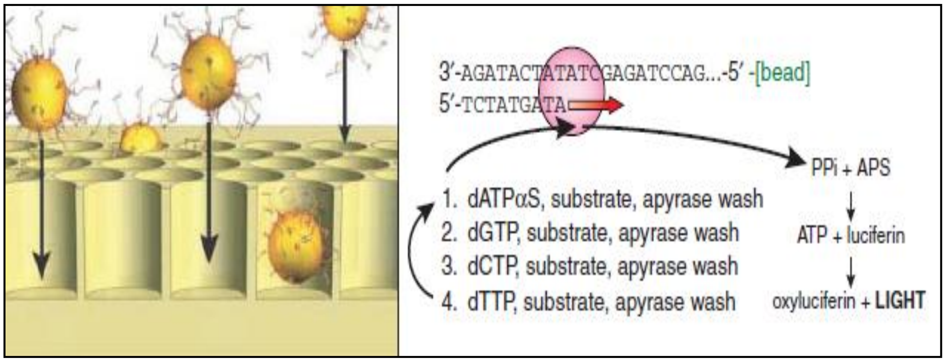

3′-AGATACTATATCGAGATCCAG...-5′ -[bead]
5′-TCTATGATA
PPi + APS
1. dATPαS, substrate, apyrase wash
2. dGTP, substrate, apyrase wash
3. dCTP, substrate, apyrase wash
4. dTTP, substrate, apyrase wash
ATP + luciferin
oxyluciferin + LIGHT

Figure 1.2 Pyrosequencing Method [8]

The length of the reads obtained from Roche 454 was 100 to 150 base pairs. It could generate the 20Mb in a run [19, 20]. Roche 454 when compared with the other NGS platforms has certain advantages which include the read length and sequencing speed. The disadvantages of Roche 454 include the expensive reagents [16] and the high error rate related to homopolymers (that is consecutive occurrence of similar bases for example GGGG or TTTTT). Because of homopolmers, major errors of Roche 454 are insertion or deletions (INDELs) of single base pairs [1].

Software

GS FLX Titanium software is used for the processing of the data obtained from Roche 454. The main part of this software is GS RunProcessor. It is used for the normalization, signal conversion and for generation of sequencing data. Different files produced by GS RunProcessor include the files in Standard flowgram format (SFF). This file contains the information about the bases being called and the quality scores of the reads. SFF file can be converted into the other file formats like fastq format for the further downstream analysis [16].

1.2.2 Illumina Genome Analyzer/HiSeq System

One of the NGS platforms is the Illumina genome analyzer also known as Solexa [21, 22]. The Solexa technology uses the bridge PCR for amplification of the sequencing features. Figure 1.3 shows the amplification by bridge PCR [23, 24].

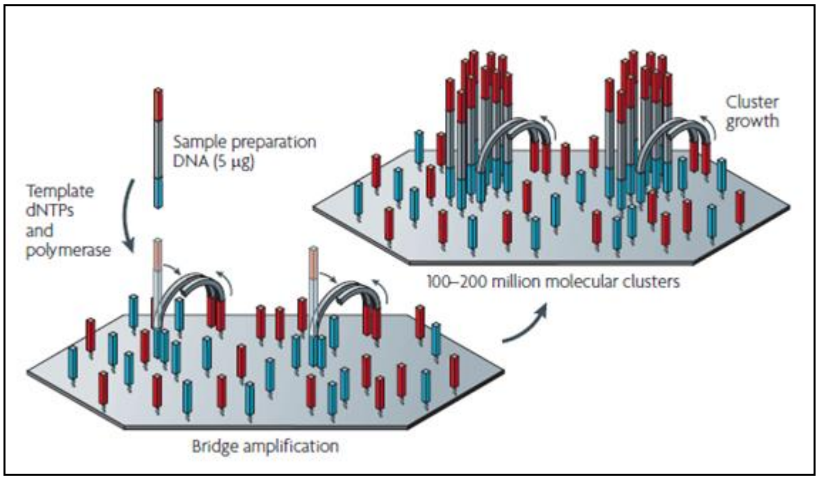


Figure 1.3 Bridge PCR [23, 24]

The length of the reads obtained by Solexa is upto 36 base pairs. The longer reads can be obtained but has the increased error rate. The error generated by Solexa includes the substitutions [8]. The insertions or deletions produced by Solexa are less as compared to the Roche 454 [1].

Software

The softwares that are used include HiSeq and Real time analyzer. The files obtained after sequencing contain the bases that are being called and the quality scores. These files can be converted into other formats. Software named CASAVA is used for matching the large number of reads with the genome [16].

1.2.3 SOLiD

In 2005, SOLiD was developed by the George Church's laboratory. The technology uses the hybridization and ligation approaches [25]. The length of the reads obtained from SOLiD is about 35 base pairs. Filtering helps SOLiD to achieve up to 99.85% accuracy. SOLiD has a lot of applications, which include transcriptome profiling, whole genome resequencing, target region sequencing and epigenome [16]. One of the short comings of SOLiD includes the short length of reads [26]. The setup used by SOLiD requires the computational skills, cluster system and is expensive [16].

Software

The data obtained from SOLiD is analyzed by the BioScope. MaxMapper algorithm is used for mapping [16].

1.2.4 Helicos

In 2003, Stephen Quake and his colleagues developed Helicos [27]. Among the available NGS platforms it has the uniqueness in its ability of using the DNA templates which are not amplified to produce the information about the sequencing [11]. The length of the reads obtained from Helicos is about 30 base pairs on average and in seven day run >20 Gb sequencing data is produced [28, 29]. The accuracy of Helicos is > 99.99%. But the major error in Helicos is the deletion of nucleotides [11].

The comparison of the NGS platforms is shown in table 1.1

Table 1.1 NGS Platform Comparison

Platforms	Technique	Read Length	Accuracy	Advantage	Disadvantage
Roche 454	Pyrosequencing	100-150 bp	99.9%	Fast, Read length	Indels, Expensive reagents
Illumina	Bridge PCR amplification	36 bp	98%	Highthroughput	Short reads
SOLiD	Ligation	35 bp	99.85%	Accuracy	Short reads
Helicos	Without amplification	30 bp	99.9%	Accuracy	Error rate with homopolymers
Ion Torrent	Proton Detection	200 bp	99.6%	Short run times	New
PacBio	Real-time Sequencing	3000 bp	99.999%	No PCR, Longest Read length	High error rate

Among the available NGS platforms Illumina has the advantage of cheap reagents whereas the accuracy of SOLiD is high [30] and Roche 454 has the reads with the longer lengths [16]. NGS platforms are not limited to sequencing the DNA. They can also be used to sequence the mRNA [31]. The technologies used for RNA sequencing include digital gene expression [32]. RNA is sequenced by first converting it into cDNA which then can be sequenced like DNA [31]. RNA sequencing (RNA-Seq) is cheaper than the genomic sequencing and is widely used. By doing RNA-Seq answer to the different questions like gene expression level quantification, alternative splicing detection, fusion of genes [33-36] or editing of RNA [37-39] can be obtained. NGS analysis can be done at the multiple levels which include genomic, transcriptomic, epigenetic and proteomic level [40]. In short NGS methods [15] are widely used for sequencing *de novo* [41], quantification of gene expression levels [42-44] and studying the population genetics [45-47]. Figure 1.4 shows the different NGS sequencing methodologies [48].

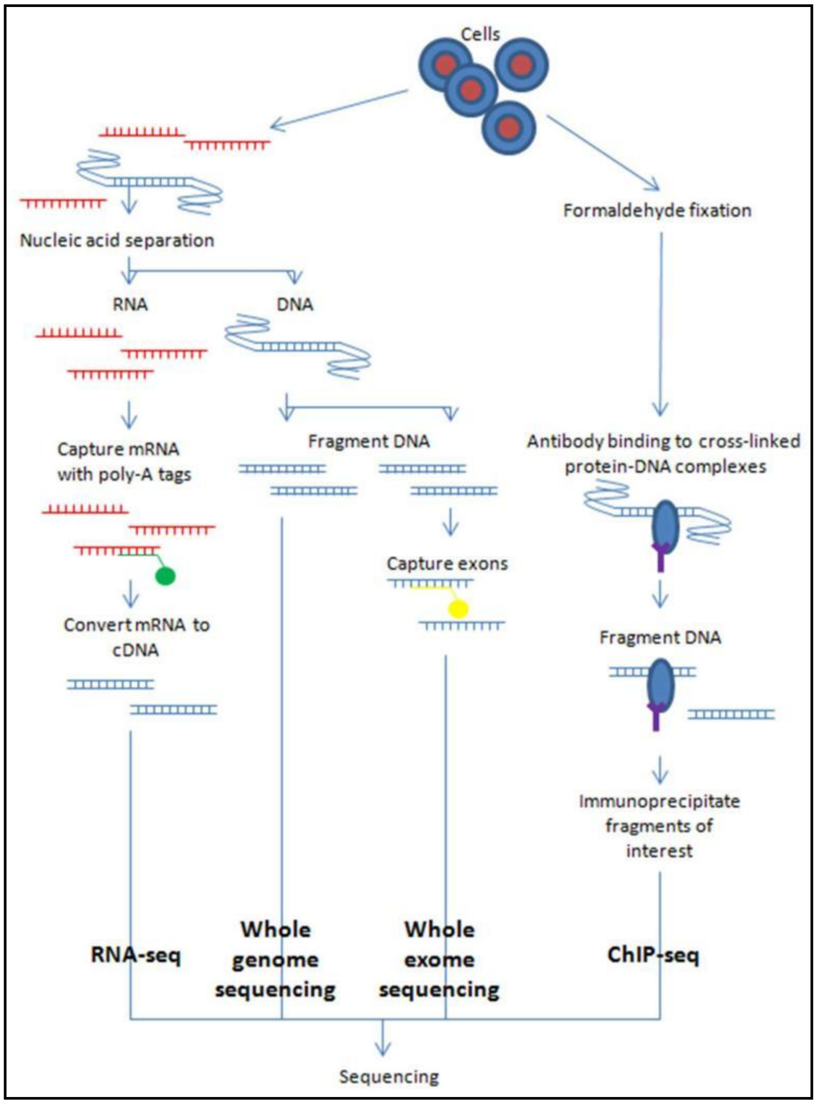


Figure 1.4 NGS sequencing methodologies [48]

1.3 NGS Data Analysis

The huge amount of data is being produced by the Next generation sequencing (NGS) platforms [49]. Bioinformatics involves investigation of the NGS data by first converting the signals into data which is then transformed into useful information and finally the information into the facts. These analyses can be categorized as primary, secondary and tertiary [50]. Figure 1.5 shows the analytical NGS pipeline.

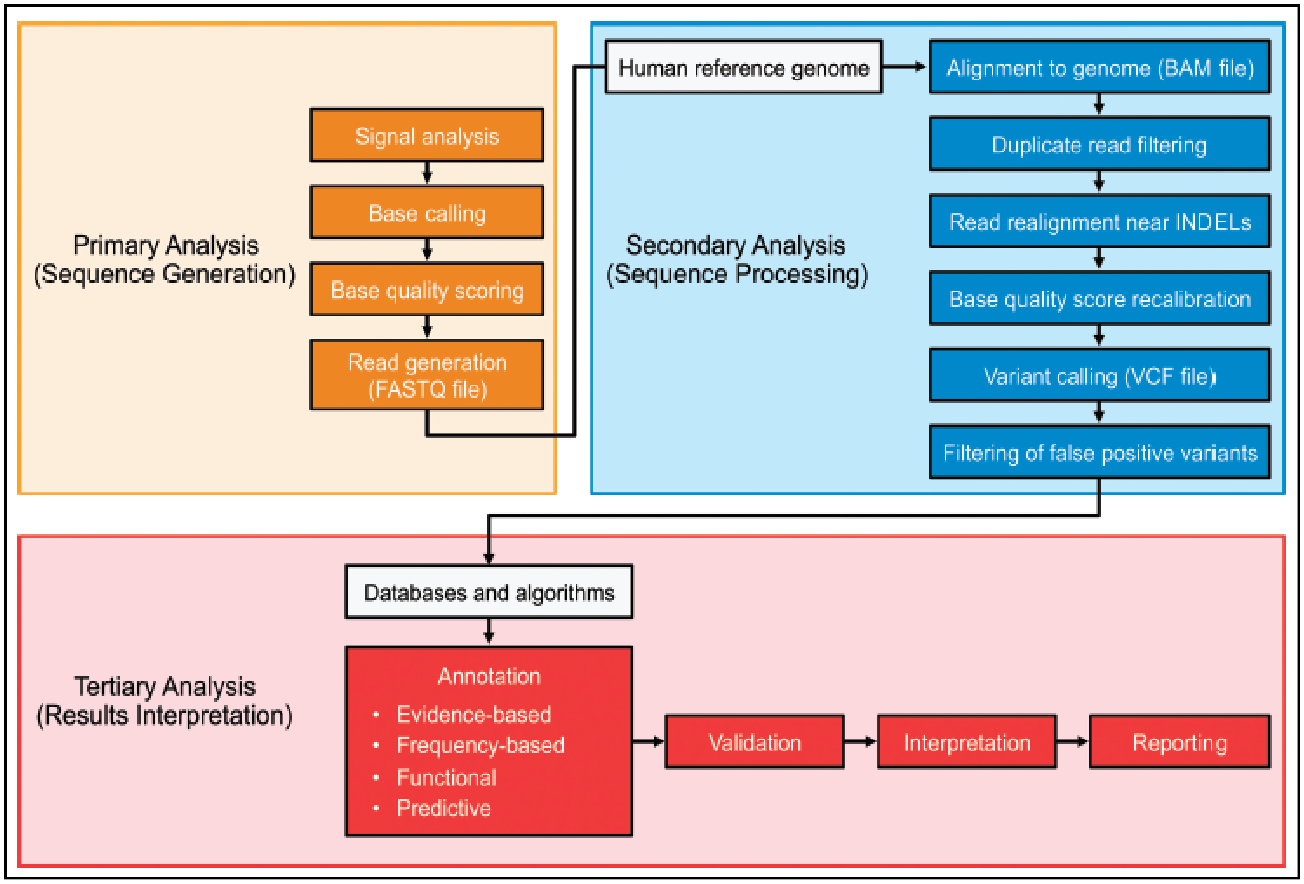


Figure 1.5 Analytical NGS pipeline [50]

In short, primary stage of the analysis involves conversion of the raw sequencing signals into the nucleotide bases and short reads. Reads are stores in a fastq files. FASTQ file contains the reads along with their quality scores. Secondary stage involves aligning the reads obtained from the primary stage to the reference genome or performing the *de novo* assembly. Binary Alignment Map (BAM) files contain the aligned reads. Alignment leads to the detection of the variants stored in Variant Call Format (VCF) files. Finally the tertiary stage involves annotating, validating and interpreting the data [51].

1.3.1 Tools for NGS Data Analysis

Different bioinformatics tools are available for analyzing the NGS data. The tools can be categorized into four general groups, which includes calling bases, reads alignment to reference, *de novo* assembly, and interpreting, annotating and detect variants [52]. The alignment tools like basic local alignment search tool (BLAST) [53] and BLAST-like alignment tool [54] are used for alignment of long reads. Mapping and Assembly with Quality (MAQ) [55], Bowtie [56] and TopHat [57] are also used for the alignment purposes. Cufflinks [44] and Scripture [43] are used for transcriptome assembly and finding the genes that are differentially expressed. Lots of tools are available for analyzing the data at the different stages. Use of the appropriate tools varies with the varying NGS platform [52]. Depending upon the application the features of NGS software vary [58]. Table 1.2 provides the information about the different types of tools available for the NGS data analysis.

Table 1.2 Tools for NGS Data Analysis

TOOL	FEATURE	URL
Bowtie	Alignment	http://bowtie-bio.sourceforge.net
BWA	Alignment	http://bio-bwa.sourceforge.net/
Bowtie 2	Alignment	http://bowtie-bio.sourceforge.net/bowtie2
MAQ	Alignment	http://maq.sourceforge.net
ELAND	Alignment	www.illumina.com
SOAPdenovo	Denovo assembly	http://soap.genomics.org.cn/
Varscan2	Variant Caller	http://varscan.sourceforge.net/
JoinSNVMix2	Variant Caller	https://code.google.com/p/joint-snv-mix/
Mutect	Variant Caller	https://github.com/broadinstitute/mutect
Varscan2	Variant Caller	http://varscan.sourceforge.net/
GATK	Variant Caller	https://www.broadinstitute.org/gatk/download
CNVnator	Copy Number Variation caller	http://sv.gersteinlab.org/cnvnator/
Breakdancer	Structural variant caller	http://breakdancer.sourceforge.net/
Slider	SNP detection	www.bcgsc.ca/platform/bioinfo
SeattleSeq	Annotation	http://snp.gs.washington.edu/SeattleSeqAnnotation/
Oncotator	Annotation	http://www.broadinstitute.org/cancer/cga/oncotator
Annovar	Annotation	http://www.openbioinformatics.org/annovar/

1.4 NGS Applications

NGS methods are broadly used in different areas of research that includes sequencing of the bacterial and viral genomes *de novo* [28, 59]. The first genome to be sequenced using NGS technology was the bacterial genome. It was sequenced using Roche 454 platform [60]. Different 854 eukaryotic and 3920 bacterial genomes have been sequenced completely according to Genome Online database as of June 2012 [61]. Along with the genomic sequencing of the different organisms, sequencing of humans at the whole genome level is becoming usual as well

[62]. The rapid increase of the sequencing of the genomes has made the knowledge in the field of genetics and genomic alterations to grow exponentially [63]. The progress and advancement in sequencing technologies has allowed the scientist and researchers to understand the mechanism of disease form DNA to the RNA level [11]. It has also allowed the medical researchers and clinicians to know about the disease traits that are inherited and find out the susceptibility loci [64]. NGS methodologies are widely used for finding out the variations at the genetic level by whole genome resequencing or target region sequencing [65].

1.5 Variations

Identification of the causative variants of complex and rare diseases, at the genetic level by whole genome and whole exome sequencing has revealed the importance of these methods [62]. Molecular changes in disorders caused by single gene have been detected by whole-exome sequencing [66, 67]. Small insertions and deletions called INDELS and single nucleotide polymorphism (SNP) can be identified by sequencing [68]. Identification of INDELS and SNP is an important task when dealing with the resequenced genomic data [69]. About 12,000 variants present in coding region can be identified by whole-exome sequencing on average [70].

1.5 .1 Single Nucleotide Polymorphism (SNP)

SNP arises when a single nucleotide base gets substituted by another base [71]. In humans the most common form of the genetic variation is SNPs. Average human genome consists of 1/1000 of SNPs [72]. Figure 1.6 explains the phenomena of SNP [73].

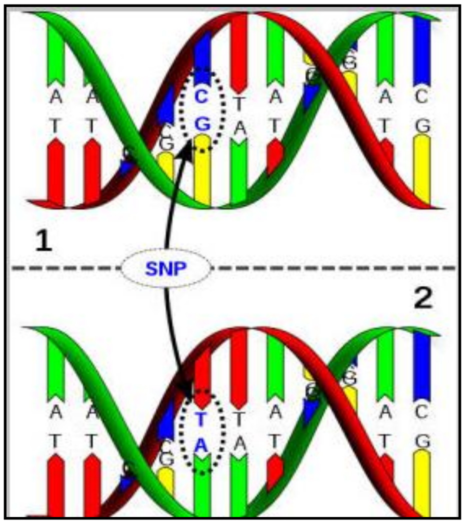


Figure 1.6 SNP phenomena [73]

SNPs can affect the normal function of DNA leading to change at RNA and protein level. They can be grouped on the basis of their location in the genome. Table 1.3 shows the classes of SNPs [74].

Table 1.3 SNPs classes

TYPE	LOCATION
Regulatory SNPs	Regulatory regions of genes
Synonymous SNPs	Exons that do not change the codon to substitute an amino Acid
Non-synonymous SNPs	Exons that incur an amino acid substitution
Intronic SNPs	Fall within introns

Synonymous and Non Synonymous SNPs

Synonymous SNPs is a class of SNPs in which change of a single nucleotide base does not change the type of amino acid that is being coded. Whereas Non Synonymous SNPs is a class of SNPs in which change of a single nucleotide base results in change of the amino acid that is being coded.

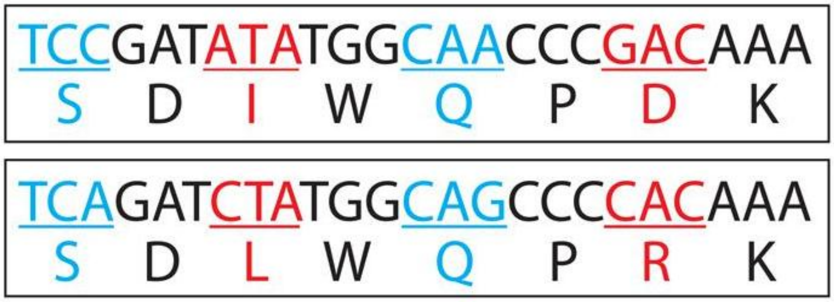


Figure 1.7 Synonymous and Non Synonymous SNPs

In the above figure the codons highlighted in blue are showing the Synonymous SNPs. TCC and TCA code for the same amino acid which is Serine. Similarly CAA and CAG code for the same amino acid Glutamine. The codons highlighted in red are showing the Non Synonymous SNPs. ATA codes for Isoleucine whereas the substitution of the A to C in CTA changes the amino acid to Leucine. Similarly GAC code for Aspartate whereas the substitution of G to C in CAC changes the amino acid to Arginine.

Databases

More than 800 databases containing the genetic variations are available but only some of them are used widely [75]. Single Nucleotide Polymorphism Database (dbSNP) [76] is one of the most important SNP database and has about 5,000,000 human SNPs [74]. Polymorphisms associated with the diseases are present in the databases like Swiss-Prot [77], Online Mendelian Inheritance in Man (OMIM) [78], Human Genome Variation Database (HGVBase) [79] and the Human Gene Mutation Database (HGMD) [80].

1.5.2 Insertions and Deletions

INDELs range in size from 1 to 50 base pairs. INDELs can be categorized as frameshift and Non frameshift [81].

Frameshift and Non Frameshift INDELS

If the length of an INDEL is not the multiple of three then this type of INDELs are known as Frameshift INDELS. However, if the INDEL length is a multiple of three then it is categorized as Non Frameshift INDEL [81].

1.6 Identification of SNPs and INDELs

The most difficult part in the analysis of the NGS data is the calling or identification of the variants because it is affected by the design of the study [68]. Once the reads are aligned to the reference genome in the alignment step, the next step in is the identification of the variants. Variants present in the sample can be identified by simply comparing the sample with the reference genome as the sample reads are already mapped to the reference genome. The identified variants are stored in the VCF file. VCF is a standard format in which the identified variants including INDELS, SNPs and large structural variants are stored [82].

1.6.1 Tools for calling variants

Some of the most widely used caller for SNPs and INDELs are discussed below:

1.6.1.1 Samtools

A variety of tools for alignment [83, 84] have been developed for accurate mapping of the reads with the reference genomes. Alignments produced by these tools have different formats thus making the downstream analysis complex. The Sequence Alignment/Map (SAM) format is developed to have a common alignment format. It supports the paired and single end reads and can combine the different types of reads. Samtools is a package for alignment manipulation in SAM or BAM format. It can be used for different alignment format conversion and SNP and INDEL calling. Samtools are implemented in C and java [85]. Bayesian statistical model is used by Samtools to calculate the posterior probabilities for the possible three genotypes. It uses the binomial distribution for calculating the likelihood [86].

1.6.1.2 Genome Analysis Tool Kit

Genome Analysis Tool Kit (GATK) is a tool kit developed by the Broad Institute which provides a variety of tools. Its main focus is finding out the variants and genotyping. Unified Genotyper is a module of GATK for calling variants [87]. GATK uses the statistical models for calling variants. It also uses the Bayesian algorithm for estimating the probabilities which are then filtered [86]. BAM file format is supported by GATK [88].

1.6.1.3 Mapping and Assembly with Quality

Li, Heng *et al* developed a statistical model for calling SNP and genotype, Mapping and Assembly with Quality (MAQ) aligns the reads to the genome used as a reference and identifies SNPs, short INDELS and variants from the alignment results. MAQ allocates a quality score named as phred-scaled quality score to each alignment in order to judge how reliable the alignments are. Single alignment is reported by MAQ. While calling SNP, MAQ uses the Bayesian statistical model to produce the consensus sequence. By comparing the consensus sequence with the reference, SNPs can be identified and later filtered on the basis of the rules defined [83].

1.6.1.4 VarScan

VarScan uses the heuristic methods, as well as Fisher exact test to identify the variants [86]. It contains a pipeline for detecting the variants in NGS data alignments. VarScan uses the alignment file to sort and score the alignments, removes the reads having low identity or aligning at multiple positions in the reference genome. For each read one of the best alignments is monitored for changes in sequence. Multiple reads having the variants are combined into the novel SNPs and insertions or deletions [89].

1.7 Annotation

The process of inferring the structural and functional information [90] about the genes generally by identifying the similarities with the already annotated sequences is known as annotation. Different NGS platforms are producing a huge amount of data about the genomic variants. In order to identify the significant variants, annotation of these variants is required. It is important to know the changes caused by these variants to understand how they play a role in various biological mechanisms.

1.8 Conclusion

There are different NGS platforms which are available for sequencing the data. Different steps are involved in analyzing these NGS data sets. Several tools are available for performing the analysis at each step. Each tool has its own specifications and functionalities.

1.9 Problem Statement

Most of the annotators are not user friendly and some of them work for the limited number of organisms. Beside this they require input files preparation, so there is a need to automate the whole pipeline for annotating the genomic variants.

1.10 Objectives of Study

Following are the objectives of the study:

- Optimize the NGS data analysis pipeline.
- Identify the genomic variants (Single Nucleotide Polymorphisms and INDELs) in real data.
- Annotate the genomic variants with available variant annotator.
- Develop an optimized and automated pipeline for annotating the genomic variants.

CHAPTER 2

LITERATURE REVIEW

CHAPTER 2

2. Literature Review

NGS has revolutionized the scientist's way of thinking because of its high throughput ability [15]. It is different from other sequencing methods in a way that it has made the parallel analysis possible by reducing the cost [16]. Different NGS platforms are present each having its own advantages with respect to the area of application in which they are used [15]. The huge amount of the data is being produced by these platforms [49]. Analysis of this data can be classified into primary, secondary and tertiary stages [50]. The primary stage involves the conversion of the sequencing signals into the raw reads. Secondary level involves the read alignment up to the variant detection steps. Tertiary stage involves the annotation, interpretation and validation of the results [51]. Different tools are available to perform the analysis at the different stages. Among all the steps the variant annotation is very important because by annotating the variants one gets insight into the facts about how these variants are affecting the different biological mechanism. For this purpose a number of tools are present. Some of these tools are discussed below.

2.1 ANNOVAR

Annovar is one of the tools used for the annotation of the genomic variants that are identified by high throughput sequencing technologies. It is command line software and can be used on the systems as standalone application where Perl modules are already installed. It is freely available and is open source. It takes the text files as an input, in which every line represents the information about one genomic variant. It needs to download the annotation data sets of genes from UCSC Genome Browser in order to perform the annotation of genomic variants with reference to their effects on gene functionality. Three types of annotations are provided by Annovar as its principal functionality. It provides the gene based annotations as one of its functionalities. It determines the effect of genomic variants (SNV, INDELs or CNVs) on the protein. Other than gene based annotations it provides other functionalities, such as to annotate the variants with respect to the regions in which these variants are located. In location based annotation the variants are annotated on the basis of the genomic features for example transcription factor binding sites, conserved regions. Filter based annotation is also provided by

Annovar which compares the variants to the already reports mutations those that are present in 1000 Genomes Project and similar projects. It downloads the files from the respective websites [91].

2.1.1 Limitations of Annovar

Beside these functionalities Annovar has certain limitations. It is platform dependent. It needs a Linux based environment to work. For Annovar to work on windows user need to set up the environment. User has to install software (e.g cgywin) which provides Linux based environment in windows. Beside this perl modules must be installed for Annovar to work. Different genomes for example humans, mouse and other can be specified by the users while using Annovar if they are present in the UCSC genome browser. User needs to prepare the input files while working with Annovar. It needs to download annotation data sets for genes in order to perform the annotation with respect to the effect of variants on genes. Also it takes the text files as an input. It provides the scores only for human's non synonymous mutations as SIFT and Polyphen (softwares for studying the affect of amino acid change on protein function) work for humans only. As it is a command line so it is difficult for the people having less programming background to work with it.

2.2 VariantAnnotation: an R/ bioconductor package

VariantAnnotation is a package in R programming language. It provides the user a facility to explore and annotate the genomic variants identified. Its functionalities include reading, writing and filtration of the VCF files [92]. This package contains different functions which can be called to perform the different tasks. User can get the information about the regions in which the variants are located. It also provides the information about the type of amino acid that is being coded and how the amino acid changes affect the function of the protein. readVCF function of the package takes the VCF file as an input and save the contents of the VCF file in a R object. locateVariants function of this package help the user to obtain the information about the genomic region in which a variant is located. predictCoding function gives the information about the affect of amino acid on the protein function.

2.2.1 Limitations of VariantAnnotation Package

Following are some of the limitations of the VariantAnnotation package. It works only for the organisms for which the annotation library is present in R. It has different functions which takes different types of input. User having less programming background finds a difficulty while passing the parameters to a function.

2.3 SIFT 4G

SIFT 4G is an improved and faster version of SIFT. SIFT only provides information about the affect of amino acid change on the function of proteins, whereas SIFT 4G also annotates the variants. SIFT 4G works by finding out the homologous sequences and gets the top 5,000 hits; (align them with the Smith Waterman algorithm). In the next step it picks up the sequences having a suitable diversity. In the final step it generates the SIFT prediction score. It takes the vcf file as an input. The database required should be downloaded before performing annotation. It generates excel and vcf files as an output [93]. The work flow of SIFT 4G is shown in Figure 2.1.

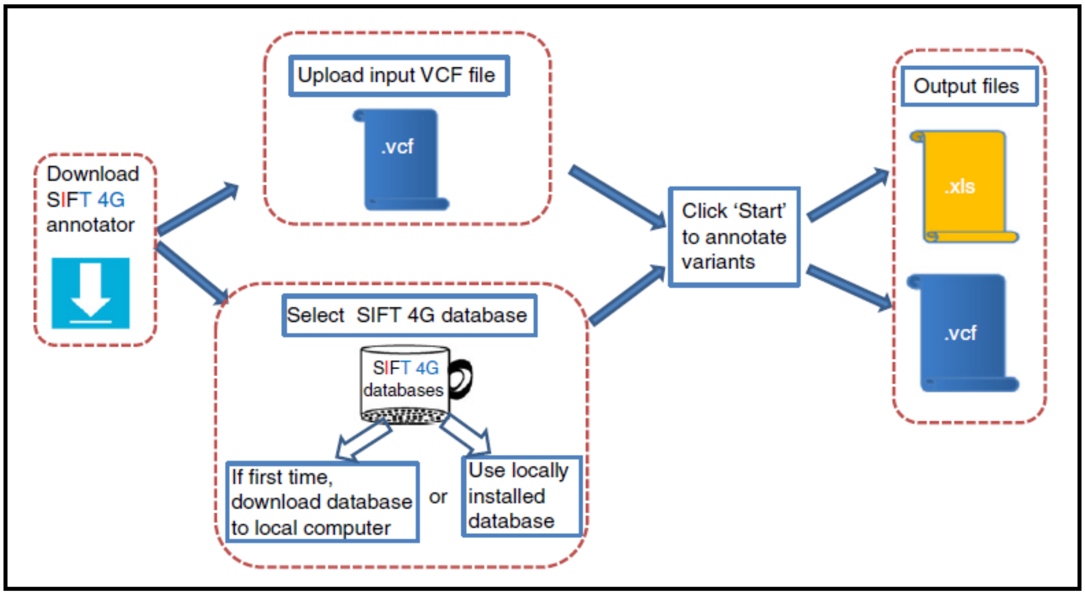

Download SIFT 4G annotator
Upload input VCF file
.vcf
Select SIFT 4G database
SIFT 4G databases
If first time, download database to local computer
or
Use locally installed database
Click 'Start' to annotate variants
Output files
.xls
.vcf

Figure 2.1 SIFT4G workflow [93]

2.3.1 Limitations of SIFT4G

Following are the limitations of SIFT 4G. It works for certain number of organisms. Before performing annotations database should be downloaded. It takes the VCF file as an input which must have at least eight columns. If the number of columns are less than eight it will not work. So user has to prepare the VCF file by adding null columns in it in order to obtain the results.

2.4 AnnTools

AnnTools is an annotation tool programmed in Python language. It can annotate the different types of human genomic variants. It is a fast and freely available tool. It provides several scripts for installation and annotation purposes. It requires Python 2.6 version or the latest version to run. It can take VCF and variant pileup files as an input. It also provides the functionality for custom annotations. To perform the custom annotations user needs to download the UCSC tables and by using the helper tools provides by Anntools import these tables in database [94].

2.4.1 Limitations of AnnTools

Some of the limitations of AnnTools are as follows. User needs to upgrade the version of python if the version is not greater than 2. It works only for annotation of human genomic variants, thus does not work for the other species. It does not provide the information about the affect of variant on the protein function. As it provides the scripts for the installation and annotation purposes one must have a programming background to work with it.

2.5 Sequence Annotator

Sequence Annotator (SeqAnt) is a web based tool for the annotation of the genomic variants. It is written in Perl programming language which contains the scripts. Users who don’t want to use the web version can also download the Perl scripts and use them for annotation purposes on the local machine. It requires different input files which include a fasta file containing the reference sequence, multifasta file and also the genomic position file in Browser Extensible Data (BED) format. Beside these it also requires the variant list file containing the information about the variants. It also requires a file containing the information about the samples along with the variant file. The annotation results can be downloaded in a text file that is tab delimited and also in a BED format for viewing it in the UCSC genome browser. Figure 2.2 shows the work flow of SeqAnt [95].

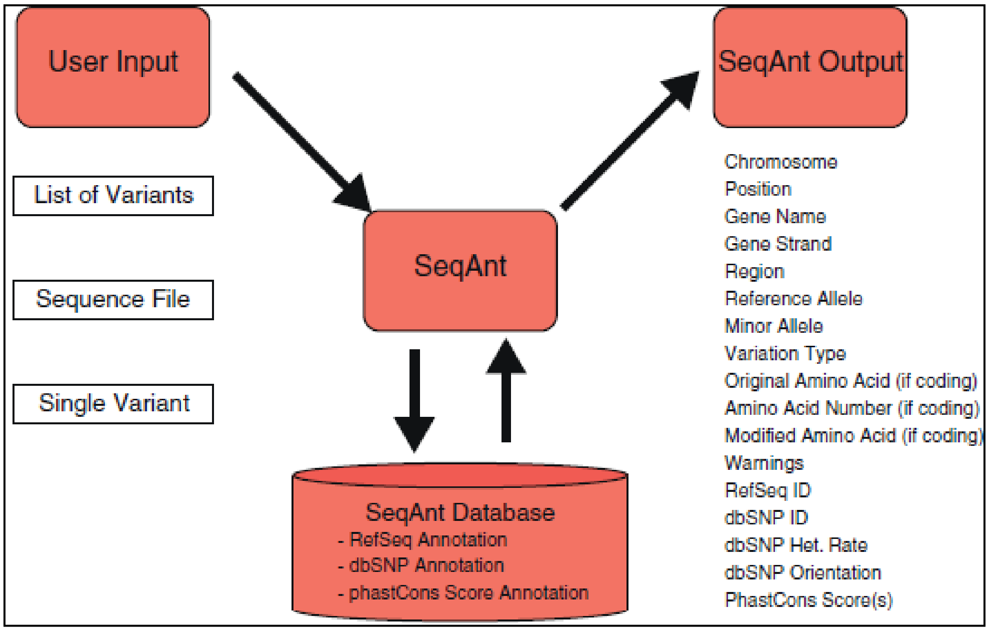


Figure 2.2 SeqAnt workflow [95]

2.5.1 Limitations of Sequence Annotator

SeqAnt has following limitations. It asks the user to provide all the files necessary to perform the annotations. It works for the limited number of organisms for example human, mouse, fly and worm.

2.6 SeattleSeq

SeattleSeq is a web based application for annotation of the SNPs and INDELs found in humans. To perform the batch analysis of different variants it provides the facility to upload the files directly as compared to the other web based applications [68]. It provides the information about the type of amino acid changed and also provides the PolyPhen (online tool which provides prediction whether the amino acid change will be deleterious or tolerant) prediction about the variants. It takes the VCF file as an input. INDELs annotation is limited in it [96].

2.6.1 Limitations of SeattleSeq

Following are the limitations of SeattleSeq. It provides the annotations only for human genomic variants. It provides limited annotations for INDELs or CNVs [97].

2.7 Sequence Variant Analyzer

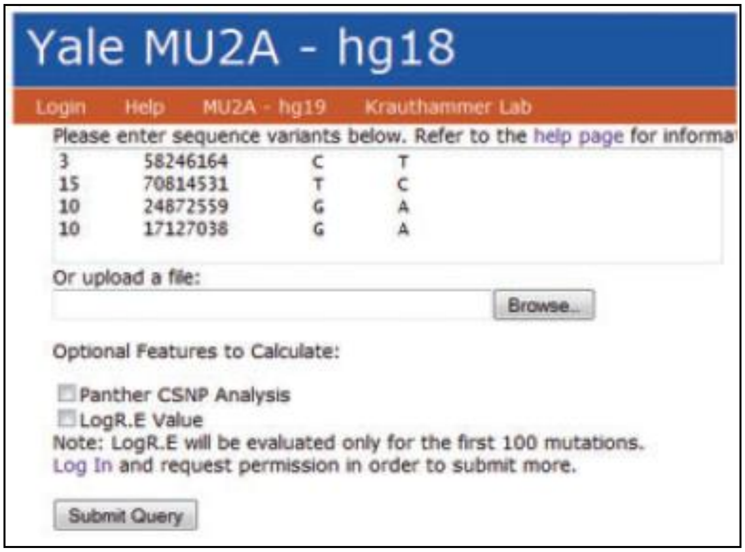
Sequence Variant Analyzer (SVA) is developed in Java programming language. It is used as a standalone application for the annotation of the genomic variants. It also provides a browser in which these variants can be visualized. For annotations of genomic variants, SVA uses a number of biological and genomic databases that are publically available. It also provides users a facility to update the gene annotations data sets. This can be done by downloading the annotation files directly [98].

2.7.1 Limitations of Sequence Variant Analyzer

Some of the limitations of SVA are as follows. It provides the annotations only for human genomic variants. Although it is standalone application but the program setup and project configuration file writing becomes difficult without expertise in Information Technology (IT) [68]. Powerful workstation is required for it which includes greater than one tera byte hard disk, greater than 48 GB RAM and processor required must be quad core [68].

2.8 MU2a

MU2a is a web based Java application [97]. It is an open source application. People with less programming background find it easier to use because of its web interface. Its web interface has a text area in which user can paste the information (like chromosome number, reference allele and alternative allele) about the variants. It also provides the user the option to upload a file from their computer. It provides the annotation for the genomic variant (SNPs) and along with how these variants effect the protein function. It generates the output in excel and tab delimited files. Figure 2.3 shows the web interface of MU2a [99].



Please enter sequence variants below. Refer to the help page for informa

```
3    58246164   C   T
15   70814531   T   C
10   24872559   G   A
10   17127038   G   A
```

Or upload a file:

Optional Features to Calculate:

□ Panther CSNP Analysis
□ LogR.E Value

Note: LogR.E will be evaluated only for the first 100 mutations.
Log In and request permission in order to submit more.

Figure 2.3 MU2a web interface [99]

2.8.1 Limitations of MU2a

One of the limitations of MU2a is that it does not provide the annotation for INDELs but only works for SNPs [97].

2.9 Variant Annotation Analysis and Search tool

The Variant Annotation Analysis and Search tool (VAAST) is command line software [97]. To download VAAST one must have an academic license. It uses the probabilistic methods to identify the variants that are deleterious and affects the genes. It takes two files as an input called target file and background file. The target file contains the information about the variants identified in cases whereas background file contains the variants found in controls. The input file formats allowed include Genome Variation Format (GVF) and VCF. The General Feature Format (GFF3) file is also taken which contains the information about the genomic feature. It also uses some of the condenser files. It generates the text file as an output which can be viewed in the browser provided by VAAST [100]. It identifies the disease causal variants by using the information about the scores of the variants [96]. It can also identify the rare variants. The workflow of VAAST is shown in Figure 2.4 [100].

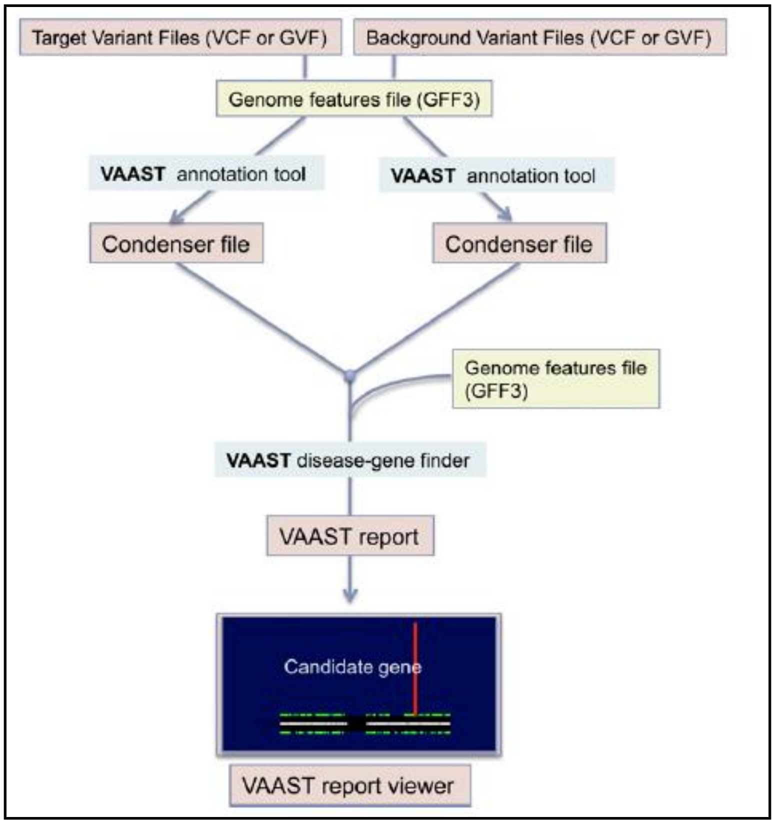

Target Variant Files (VCF or GVF)
Background Variant Files (VCF or GVF)
Genome features file (GFF3)
VAAST annotation tool
VAAST annotation tool
Condenser file
Condenser file
Genome features file (GFF3)
VAAST disease-gene finder
VAAST report
Candidate gene
VAAST report viewer

Figure 2.4 VASST workflow [100]

2.9.1 Limitations of Variant Annotation Analysis and Search tool

Following are some of the VAAST limitations. Its main functionality is to find out the genes that are associated with the diseases. It provides limited information about the annotation [97]. It works only for the case-control studies.

2.10 NGS-SNP

NGS-SNP is a freely available tool implemented in Perl. It has several Perl scripts which are used for the annotation of Single Nucleotide Variants (SNVs). Among the several scripts the main script of NGS-SNP is annotate_SNPs.pl. It takes the list of SNPs and as an output it generates the annotations along with the list of SNPs. It takes the output of Mapping and Assembly with Quality (MAQ) and SAMtools as an input. One can use it by running the Perl scripts or by using it on the virtual machine. It classifies the SNPs as non synonymous, synonymous or three prime Untranslated Region (3'UTR) [101].

2.10.1 Limitations of NGS-SNP

Some of the NGS-SNP limitations are as follows. It provides the annotation only for SNPs. As it has several scripts one must have a programming aptitude to work with it. It may take several days to complete the annotation process [68].

2.11 Variant Analysis Tool

Variant Analysis Tool (VAT) performs the functional annotations of the genes. VAT program is written in C language. It has the different modules. It can be used through command line or as a web application. It takes the variant and annotation sets as an input and generates the annotated variants in VCF file. It also provides the user with a facility to visualize the variants. Figure 2.5 shows the workflow of VAT [102].

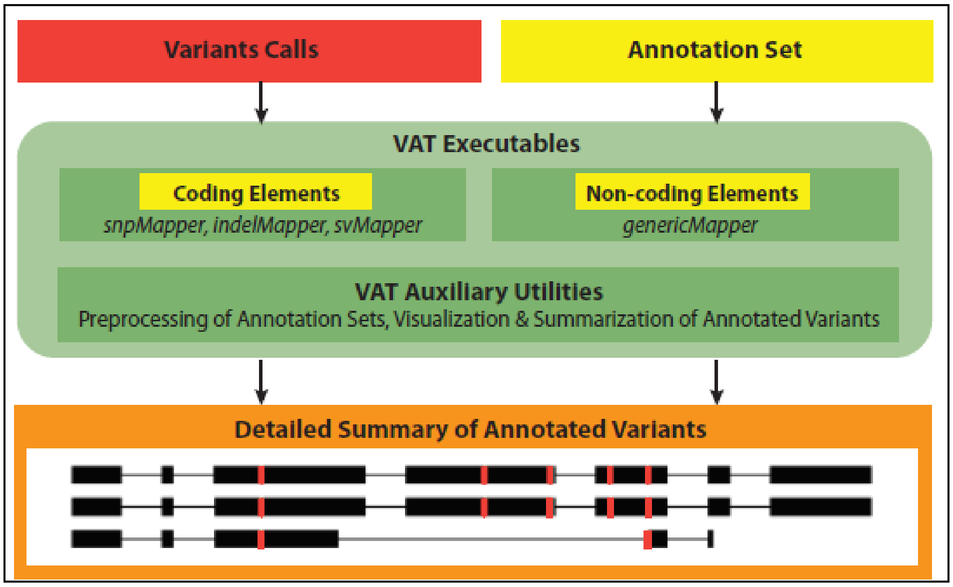

Variants Calls
Annotation Set
VAT Executables
Coding Elements
snpMapper, indelMapper, svMapper
Non-coding Elements
genericMapper
VAT Auxiliary Utilities
Preprocessing of Annotation Sets, Visualization & Summarization of Annotated Variants
Detailed Summary of Annotated Variants

Figure 2.5 VAT workflow [102]

2.11.1 Limitations of Variant Analysis Tool

VAT has the following limitations. One of the major limitations of VAT is that it requires certain other modules or packages to work [68]. It asks the user to provide the annotation sets along with the variants as an input.

Comparison of Tools

Table 2.1 Comparison of tools

Tool	Input	Language	Platform	Limitations
Annovar	vcf, pileup	Perl	Linux, Mac, Windows	Input files preparation, Command line, Platform dependant
Variant Annotation	vcf	R	Linux, Windows	Limited number of organisms, Command line
SIFT4G	vcf	Java	Linux, Mac, Windows	Limited number of organisms, Download database, takes vcf file with 8 columns
Anntools	vcf, variant pileup	Python	Linux, Mac	Works only for human variants, Programming background, does not provide the effect of variant on protein function
SeqAnt	Fasta & BED file	Perl	Web based	Works for limited number of organisms, User needs to provide all the input files
SeattleSeq	vcf	Java	Web based	Works only for human variants, limited annotations for indels
SVA	vcf	Java	Linux	Works only for human variants, powerful workstation is required
MU2a	vcf	Java	Web based	Does not provide annotation for INDELs
VASST	GVF, vcf	C	Linux, Mac	Case-control studies, finds genes related to disease, limited information about annotation, User need to install the dependencies.
NGS-SNP	vcf, pileup	Perl	Linux, Mac	Does not provide annotation for INDELs, Several days required to complete the process
VAT	vcf, annotation sets	C	Web based	Requires other modules and packages to work, annotation sets provided by user

CHAPTER 3

METHODS

CHAPTER 3

3.1 Methodology

The aim of this study was to develop an automated pipeline for annotation of the genomic variants (SNPs and INDELs). This was done by first optimizing the NGS data analysis pipeline. Later on this optimized pipeline was used to identify the variants. These variants were later annotated with SIFT4G. The methodology that was used is as follows.

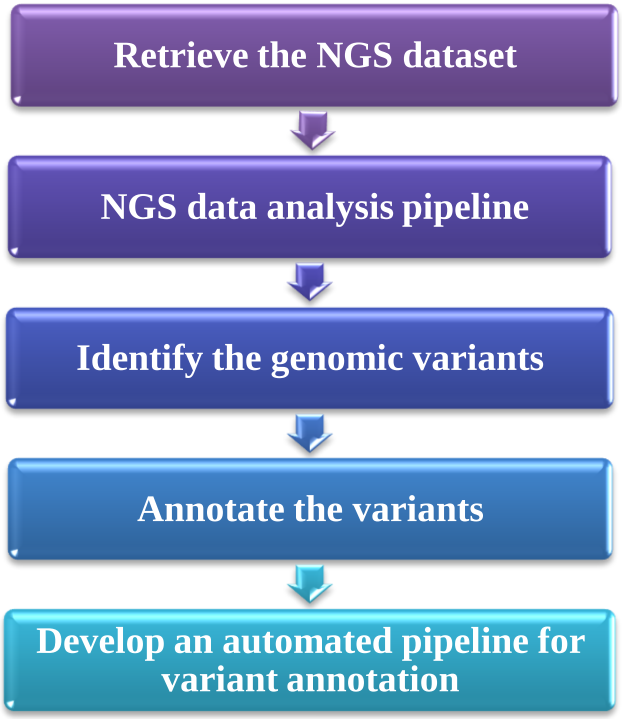

Retrieve the NGS dataset
NGS data analysis pipeline
Identify the genomic variants
Annotate the variants
Develop an automated pipeline for variant annotation

Figure 3.1 Methodology

3.1 NGS Dataset

The training dataset for this study was taken from the article by Ali *et.al* [103]. The raw sequence data of this [103] study is available at European Nucleotide Archive (ENA) having the accession number **PRJEB7798**. The organism under study was Mycobacterium Tuberculosis. Whole Genome Sequencing was done using paired end Illumina HiSeq 2000 [103]. The raw reads sequence data of 5 samples was downloaded from the ENA. The brief information about the sample data is described in the table 3.1.

Table 3.1 Sample Information

Sample No.	Sample Accession	Run Accession	Read Count	Base Count
1	SAMEA3139819	ERR688008	20,728,746	4,187,206,692
2	SAMEA3139820	ERR688009	7,997,503	1,615,495,606
3	SAMEA3139821	ERR688010	7,920,984	1,600,038,768
4	SAMEA3139827	ERR688016	10,040,027	2,028,085,454
5	SAMEA3139828	ERR688017	9,812,375	1,982,099,750

3.2 NGS Data Analysis Pipeline

The pipeline used for NGS data analysis is shown in Figure 3.2.

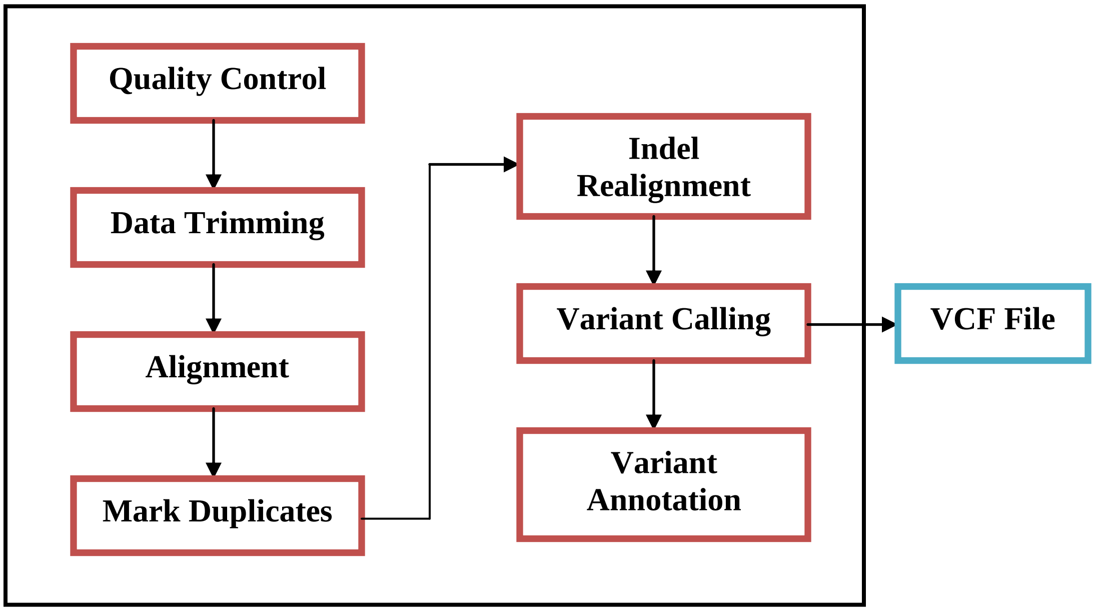


Figure 3.2 NGS Data Analysis Pipeline

NGS datasets were analyzed using the above pipeline. Most of the NGS platforms provide the data of the sequencing reads in a format called FASTQ format. The first step in the analysis pipeline is to check the quality of the raw sequencing reads. After this, the data trimming step is optional and is done when the quality of the raw sequencing reads is poor. The next step in the pipeline is alignment, involves the alignment of the raw reads to the reference genome [104]. PCR and other duplicates are being added in the NGS data while amplification or sequencing. This often causes the incorrect variant calls therefore it is essential to filter out these duplicates. For this purpose the Mark Duplicates step in the processing pipeline identifies the duplicated sequence alignments. Once these duplicates are identified, they are ignored in the preceding steps of the pipeline. False positive calls are being made due to the INDELs that are present at the end of the reads which causes the misalignments to occur. INDEL realignment step in the pipeline reduces the number of the misaligned bases. After this a file containing the refined alignment is generated which later on is used for calling the variants. Variants that are called are stored in a VCF file [105]. The final step of the processing pipeline is to annotate the called variants to give them a biological meaning and describe how these variants affect the different biological mechanisms.

3.3 Optimized NGS Data Analysis Pipeline

The NGS data analysis pipeline that was optimized is as shown in Figure 3.3.

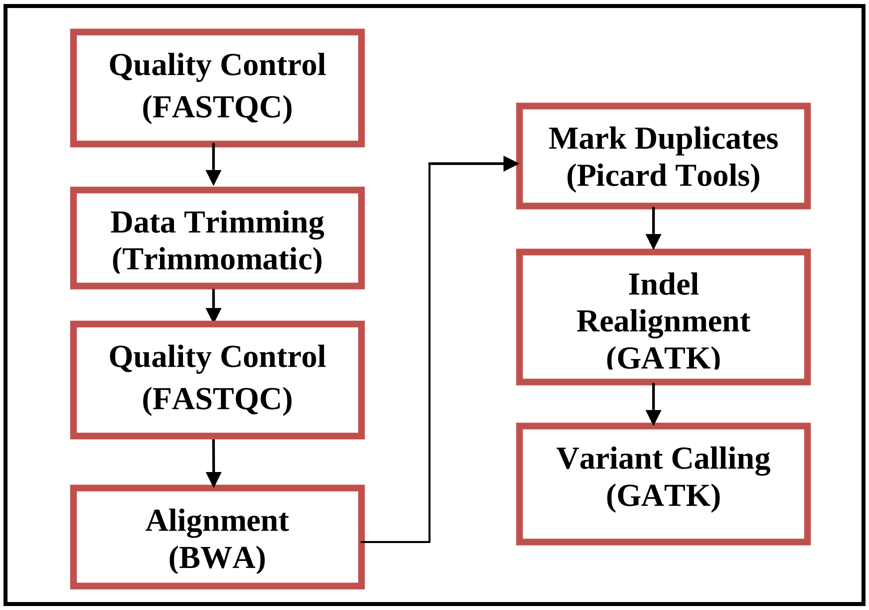


Figure 3.3 Optimized NGS Data Analysis Pipeline

As there are large number of tools available to perform each steps of the NGS data analysis pipeline, so it is important to select the tool according to one's dataset. For example Burrows Wheelers Aligner (BWA) has three algorithms which work best under different circumstances. Here BWA-mem was used for alignment as it works faster as compared to others.

3.4 Identification of Genomic Variants

Genomic variants were identified using the optimized NGS data analysis pipeline shown in Figure 3.3. The various steps of the analysis pipeline performed are discussed in detail below.

3.4.1 Input Files

The input files required for the identification of the genomic variants include the fastq files along with the reference genome. The fastq files of Mycobacterium Tuberculosis were downloaded from the ENA under the accession number **PRJEB7798.** The reference genome **H37RV** was downloaded from NCBI in a FASTA format under the Genbank accession number **AL123456.3.**

3.4.2 Step 1: Quality Control

The raw sequencing reads once obtained from the NGS platforms are checked for the quality. The quality control step checks the quality of the reads and identifies the reads whose quality is below a certain threshold. All the fastq files downloaded from the ENA were checked for the quality.

3.4.2.1 Tool used for Quality Control

The tool used to perform the quality control was FASTQC. It can take the file in different formats for example FASTQ, SAM, BAM. FASTQC identifies the areas that have a problem that arose during the sequencing or while preparing the library. It generates the results in the form of graphs and tables. And it reports the problematic areas in the html report [106]. Block diagram in Figure 3.4 shows the workflow of the FASTQC.

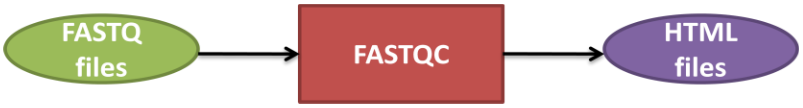


Figure 3.4 FASTQC workflow

3.4.3 Step 2: Data Trimming

The quality of the reads obtained was not good so the next step in the processing pipeline was data trimming. This step trims the reads of the poor quality. All the fastq files had the poor quality reads so they were trimmed in this step.

3.4.3.1 Tool used for Data Trimming

The tool used to perform the trimming was Trimmomatic-0.35. It takes only the FASTQ files as an input. It is a command line tool used to trim the Illumina data. It has two running modes one is single end and the other is paired end. As the data set used was paired end so paired end mode of Trimmomatic was used. Paired end mode takes the two files one forward and the other reverse as an input. The output generates the four files two forward reads files one containing the forward paired and other containing the forward unpaired reads and similarly the two reverse reads files one containing the reverse paired and other containing the reverse unpaired reads. Following parameters were used for trimming: Leading 3, Trailing 3, Sliding Window 4:15 and Min length 36. Block diagram in Figure 3.5 shows the workflow of Trimmomatic.

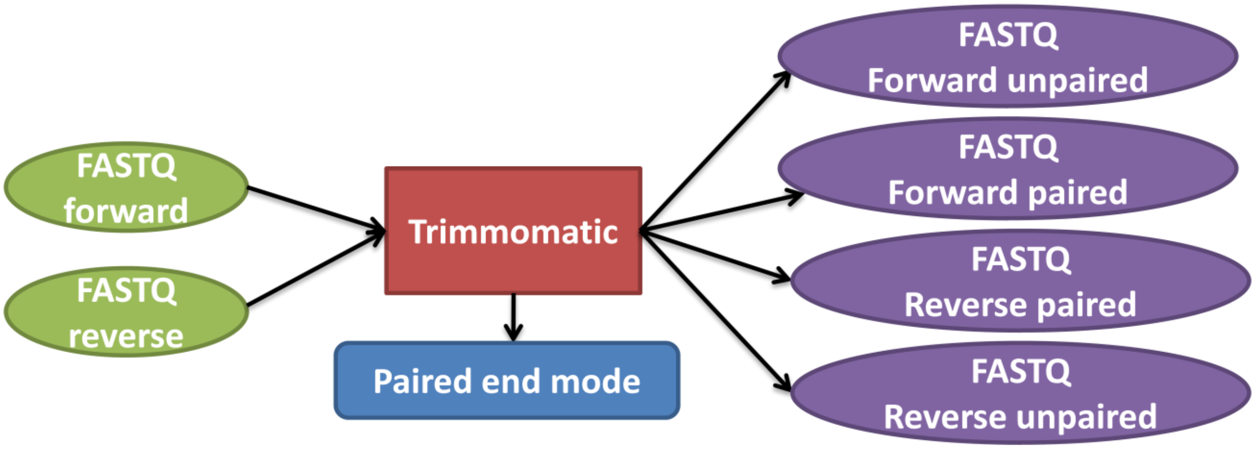


Figure 3.5 Trimmomatic workflow

The two output files forward paired and reverse paired contains the trimmed reads that can be used for further analysis, as they do contain the poor quality reads. It can be checked by performing quality control.

3.4.4 Step 3: Quality Control after Trimming

As all the fastq files had reads with poor quality. The reads lying in the poor quality region cwere trimmed. The files obtained after trimming were then again checked for the quality. This was done to confirm that the data contains no poor quality reads.

3.4.4.1 Tool

For quality control FASTQC was used. Its working is described in step 3.4.1.

3.4.5 Step 4: Alignment

Once the reads are checked for the quality, the next step in the processing pipeline is to align the reads to the reference genome. Before the alignment is done the sequence dictionary file was created using Picard tools. After this the reads were aligned and the read group information was added. All the fastq files were aligned to the reference genome.

3.4.5.1 Tool used for Alignment

The tool that was used to align the reads to the reference genome was Burrows-Wheelers Aligner (BWA). The tool maps the reads against a reference genome. It consists of three different algorithms. The three algorithms are Burrows-Wheelers Aligner, Smith-Waterman alignment (BWA-SW), BWA-backtrack and BWA-mem. BWA-mem is the most accurate and fast algorithm. It takes the fastq files and the reference genome as an input. As the data used was paired end so it had two fastq files one containing the forward end reads and the other containing the reverse end reads. It generates the BAM files containing the alignment information about the reads as an output. Block diagram in Figure 3.6 shows the workflow of BWA.

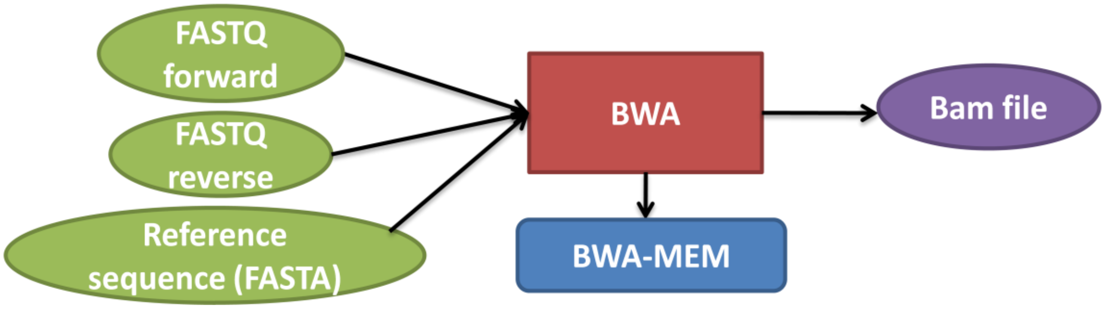


Figure 3.6 BWA workflow

3.4.6 Step 5: Mark Duplicates

After the alignment is done the next step which was performed was the duplicate marking. Duplicates are added to the NGS data during amplification or sequencing. It is important to filter out these duplicates as it leads to the incorrect variant calls. For this purpose the Mark Duplicates step in the processing pipeline identifies the duplicated sequence alignments. Once these duplicates are identified they will be ignored in the following steps of the pipeline. Duplicates in all BAM files were marked.

3.4.6.1 Tool used for Duplicate marking

The tool used for marking the duplicates was Picard tool. Picard tools are java based command line tools. In Picard tools the Mark Duplicates module was used. It takes the BAM files as an input and generates the BAM with the marked duplicates as an output. Block diagram in Figure 3.7 shows the workflow of Mark Duplicates module of the Picard tools.

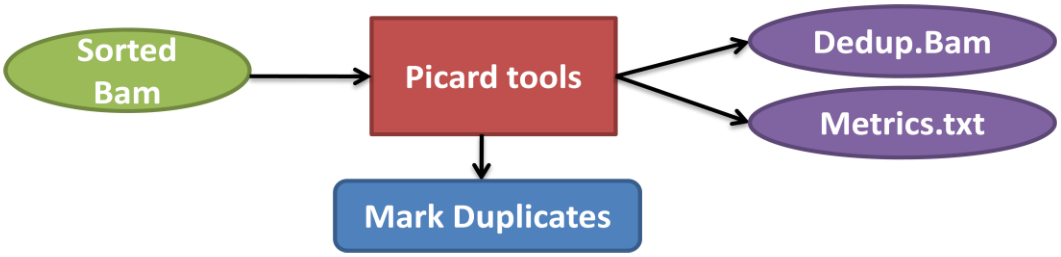


Figure 3.7 Picard tools (Mark Duplicates) workflow

3.4.7 Step 6: INDEL Realignment

False positive calls are being made due to the INDELs that are present at the end of the sequences which causes the misalignments to occur. INDEL realignment step in the pipeline reduces the number of the misaligned bases.

3.4.7.1 Tool used for INDEL Realignment

The tool used for realignment of the INDELs was GATK. The INDEL realigner tool was used for this purpose. The BAM files in which the duplicates were marked were given as an input to the tool and it generated the BAM files in which the INDELs were realigned. Block diagram in Figure 3.8 shows the workflow of INDEL realigner module of the GATK tools.

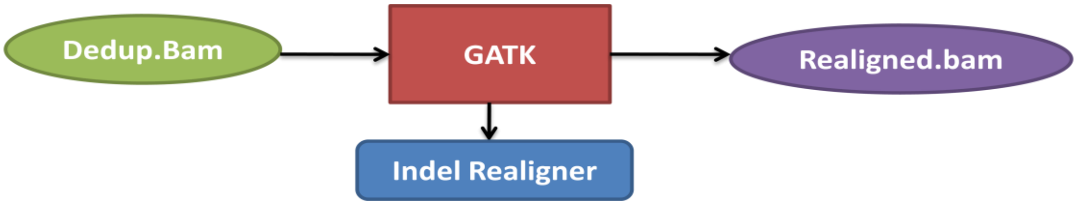


Figure 3.8 GATK (INDEL Realigner) workflow

3.4.8 Step 7: Variant Calling

After the realignment of INDELs the regions that are different in the samples when compared to the reference are identified. The identified variants are reported and are stored in a file.

3.4.8.1 Tool used for calling variants

Variants were called using the Haplotype Caller module of the GATK tools. It can call the SNPs and INDELs. It takes the BAM files as input and generates the VCF file as an output. The VCF file contains the information about the variants that are being called. Block diagram in Figure 3.9 shows the workflow of Haplotype Caller module of the GATK tools.

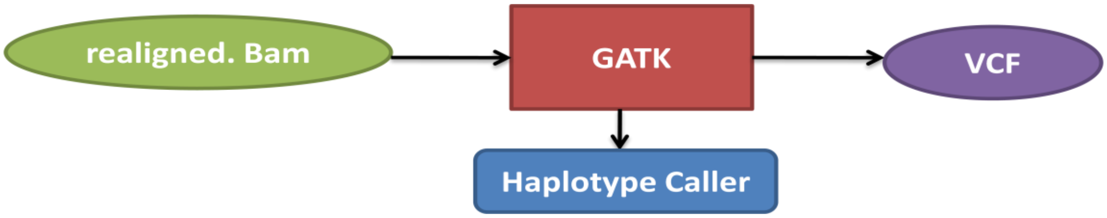


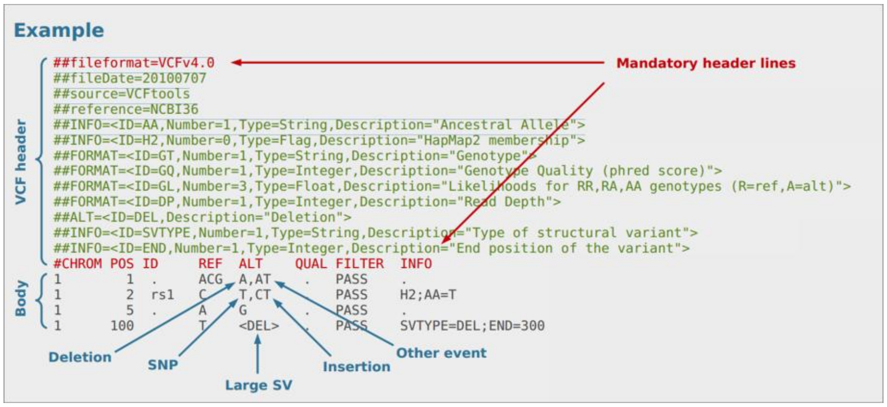
Figure 3.9 GATK (Haplotype Caller) workflow

3.5 Variant Annotation

Once the alignment and variant calling has been done a large number of changes that are present in the genome are reported in comparison to the reference genome [96]. The next step, after the identification of variants is to obtain the information about how these changes or variations affect the biological process under study. This information can be obtained by annotating the identified variants.

3.5.1 VCF Format

Variant Call Format (VCF) is a text format file. The variants that are identified are reported in a VCF file. So it is important to know what a VCF file is and what information it contains about the variants. It contains the header and the information about each variant in genome. There are eight columns that are compulsory in a VCF file. The column names are as follows: 1) #CHROM, 2) POS, 3) ID, 4) REF, 5) ALT, 6) QUAL, 7) FILTER and 8) INFO. Figure 3.10 shows how a VCF file looks like.


Example

```
##fileformat=VCFv4.0
##fileDate=20100707
##source=VCFtools
##reference=NCBI36
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality (phred score)">
##FORMAT=<ID=GL,Number=3,Type=Float,Description="Likelihoods for RR,RA,AA genotypes (R=ref,A=alt)">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##ALT=<ID=DEL,Description="Deletion">
##INFO=<ID=SVTYPE,Number=1,Type=String,Description="Type of structural variant">
##INFO=<ID=END,Number=1,Type=Integer,Description="End position of the variant">
#CHROM POS ID     REF  ALT    QUAL FILTER  INFO
1        1 .      ACG  A,AT   .    PASS    .
1        2 rs1    C    T,CT        PASS    H2;AA=T
1        5 .      A    G      .    PASS    .
1      100 .      T    <DEL>  .    PASS    SVTYPE=DEL;END=300
```

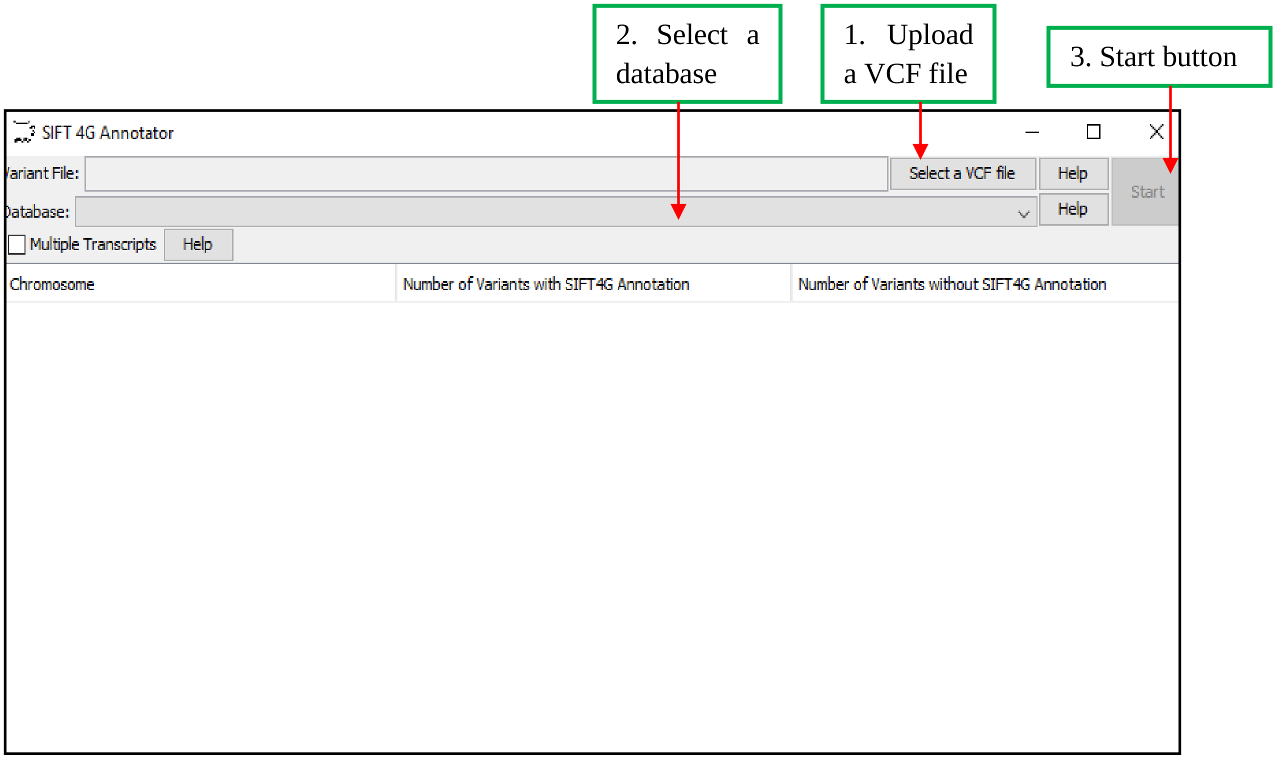
Mandatory header lines
VCF header
Body
Deletion
SNP
Large SV
Insertion
Other event

Figure 3.10 VCF file example

The #CHROM column contains the chromosome number on which the variant is located. The POS column tells the position of the variant in the genome. The ID column contains the ID of the variant if it has been previously identified. The REF column contains the alleles present in the reference genome whereas the ALT column contains the allele present in the sample under study. QUAL column describes the quality of the variant. FILTER column tells whether the variant has passed the applied filters. INFO column contains other information about the variant.

3.5.2 Tool used for performing annotation

The variants obtained from the variant calling step were annotated using SIFT 4G tool. The VCF file given as an input must have eight columns. It generates the results in a VCF file and Excel file also [93]. Figure 3.11 shows the Graphical User Interface (GUI) of SIFT 4G annotator.



2. Select a database

1. Upload a VCF file

3. Start button

Figure 3.11 SIFT 4G Annotator GUI [93]

The VCF files for all the samples were annotated using the SIFT 4G annotator. The Mycobacterium tuberculosis H37RV database was downloaded to perform the annotations. The output for all the files was saved in Excel files. Java Runtime Environment was downloaded to run the SIFT 4G annotator.

3.6 Automated Pipeline Development

The automated pipeline is named as AutoAnnotate. To automate the pipeline for annotation of the SNPs and INDELs, R programming language was used. R is freely available and is used for statistical computing and graphics. It can be used for manipulation of data. It is platform independent and can be run on Windows, Linux and MacOX. The pipeline takes only the VCF file as an input and performs the different computations required and generates the output files. It generates several files as an output. Two excel files are generated, one containing the annotated SNPs and the other containing the annotated INDELs. It also generates .pdf files containing the charts and tables. Different types of charts are generated which include bar, pie, petal pie and stacked bar charts. The charts and tables display the number of the SNPs and INDELs annotated. The workflow of the pipeline that was automated for annotation of the SNPs and INDELs is shown in Figure 3.12.

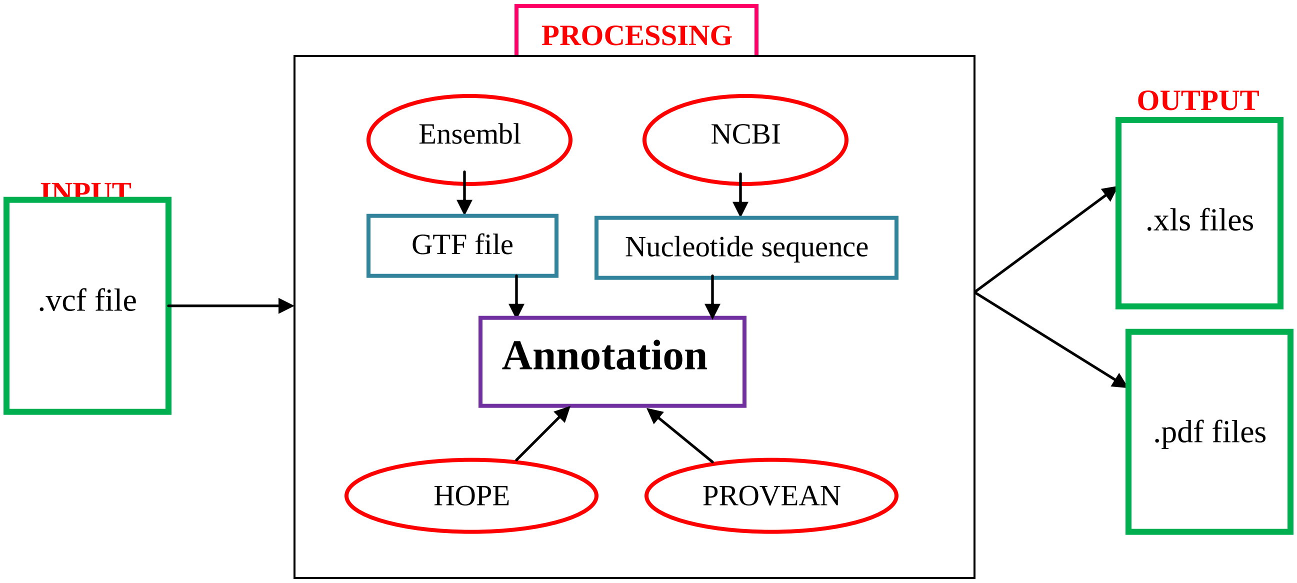


Figure 3.12 AutoAnnotate workflow

The processing part displayed in Figure 3.12 is described more in detail in Figure 3.13.

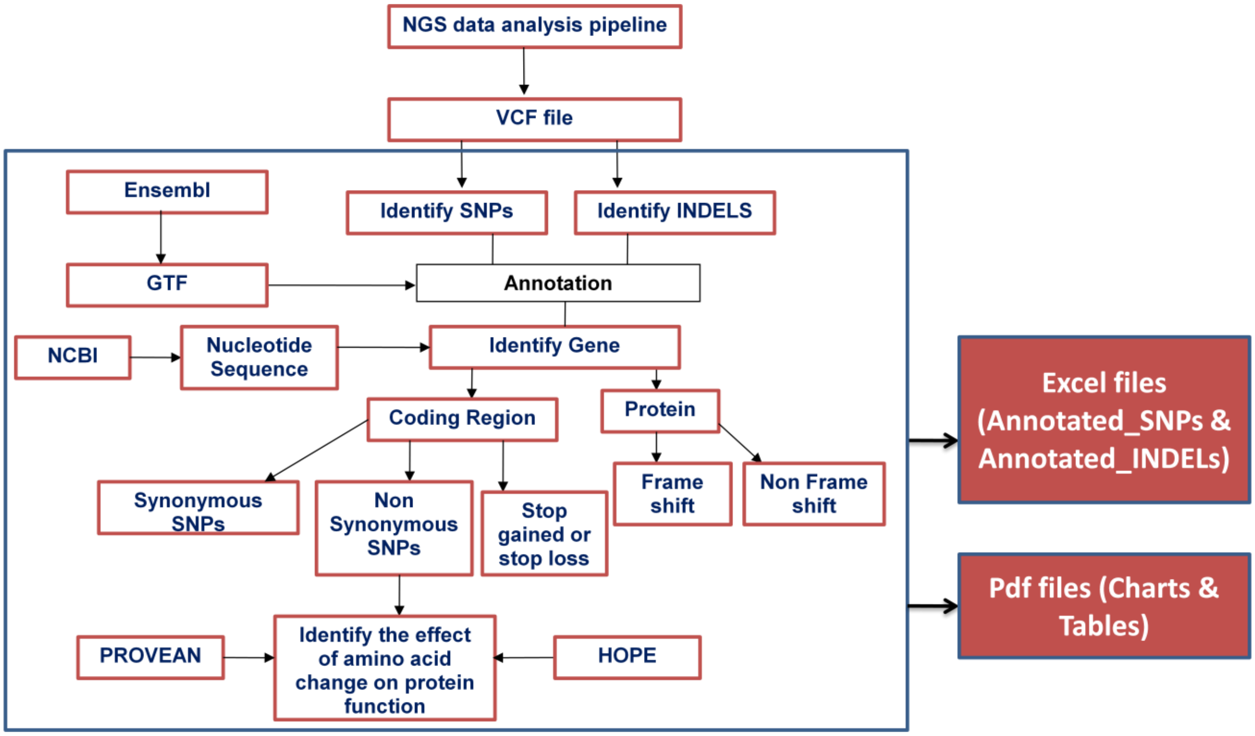
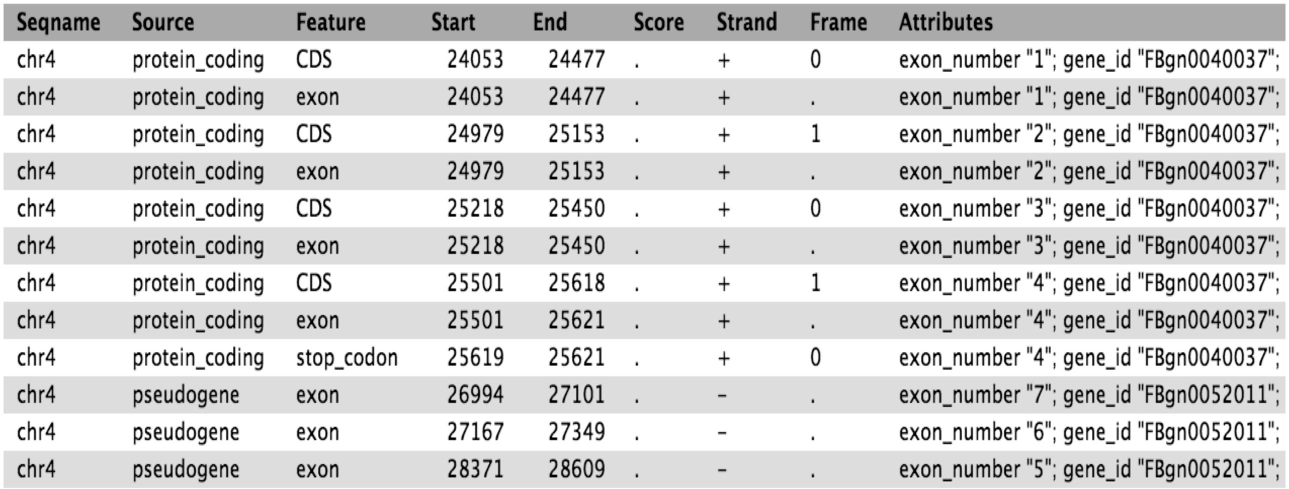


Figure 3.13 Pipeline Automated

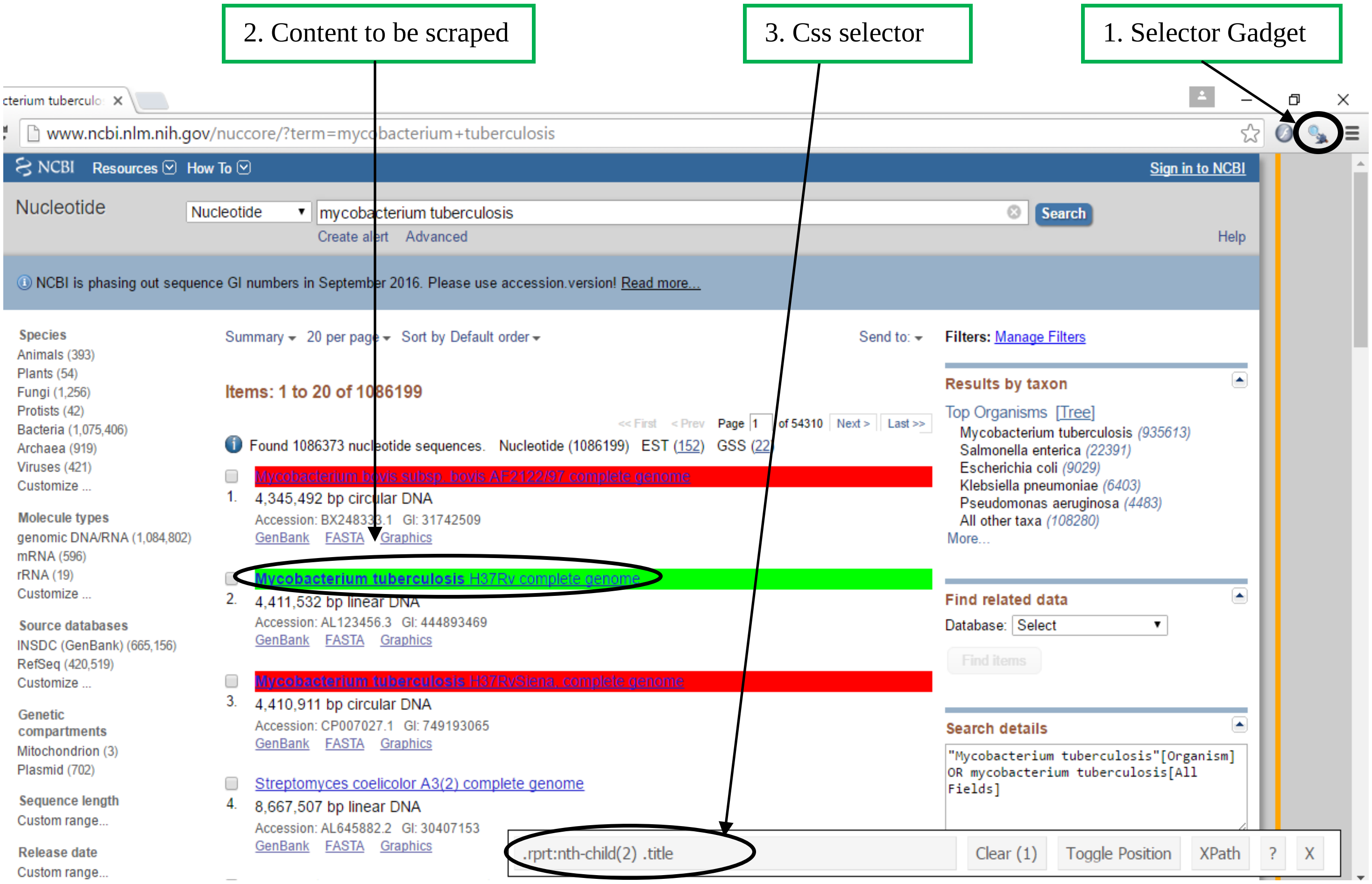
The description of the automated pipeline is as follows. The output of the NGS data analysis pipeline which is a VCF file is used as an input for the pipeline. Once the VCF file is read the SNPs and INDELS are identified. The next step involves the annotation of these variants to find out where these variants are located in the genome. For this one must have a Gene Transfer Format (GTF file). The GTF file is downloaded from Ensembl. Ensembl is one of the genome browsers and it provides the information about genomes of different organisms. GTF file contains the information about the structure of the genes. It contains nine columns and is tab delimited. Example of a GTF file is displayed in Figure 3.14. The nine columns containing the data are shown in Figure 3.14.



Seqname	Source	Feature	Start	End	Score	Strand	Frame	Attributes
chr4	protein_coding	CDS	24053	24477	.	+	0	exon_number "1"; gene_id "FBgn0040037";
chr4	protein_coding	exon	24053	24477	.	+	.	exon_number "1"; gene_id "FBgn0040037";
chr4	protein_coding	CDS	24979	25153	.	+	1	exon_number "2"; gene_id "FBgn0040037";
chr4	protein_coding	exon	24979	25153	.	+	.	exon_number "2"; gene_id "FBgn0040037";
chr4	protein_coding	CDS	25218	25450	.	+	0	exon_number "3"; gene_id "FBgn0040037";
chr4	protein_coding	exon	25218	25450	.	+	.	exon_number "3"; gene_id "FBgn0040037";
chr4	protein_coding	CDS	25501	25618	.	+	1	exon_number "4"; gene_id "FBgn0040037";
chr4	protein_coding	exon	25501	25621	.	+	.	exon_number "4"; gene_id "FBgn0040037";
chr4	protein_coding	stop_codon	25619	25621	.	+	0	exon_number "4"; gene_id "FBgn0040037";
chr4	pseudogene	exon	26994	27101	.	-	.	exon_number "7"; gene_id "FBgn0052011";
chr4	pseudogene	exon	27167	27349	.	-	.	exon_number "6"; gene_id "FBgn0052011";
chr4	pseudogene	exon	28371	28609	.	-	.	exon_number "5"; gene_id "FBgn0052011";

Figure 3.14 GTF file example

Using the information from a GTF and VCF file the SNPs and INDELs are annotated. The #CHROM and POS columns of VCF file are picked and are mapped with the Seqname (column1), Start (column 4) and End (column 5) columns of the GTF. In this way regions for all the variants in which they are located are identified. After this the SNPs and INDELs in the coding region are identified. Later the genes on which these variants are located are identified. The complete nucleotide sequence of the organism's genome is downloaded from Center for Biotechnology Information (NCBI). To retrieve the information from NCBI the concept of web scraping is used. Web scraping is a concept of retrieving the information from the web programmatically. To scrap the data from the web, the css selector (pattern of the element) of the content that is being scraped should be known. To know about the css selector of a web page Selectorgadget was used. It is a java based tool which allows one to know about the css selector of the content that one need to extract from a web page. To use this tool one must install it and add it to the bookmark bar of the browser. Figure 3.15 shows how selectorgadget helps to find out the css selector of the content to be extracted from a web page.



www.ncbi.nlm.nih.gov/nuccore/?term=mycobacterium+tuberculosis

Nucleotide: mycobacterium tuberculosis

NCBI is phasing out sequence GI numbers in September 2016. Please use accession.version! Read more...

Species: Animals (393), Plants (54), Fungi (1,256), Protists (42), Bacteria (1,075,406), Archaea (919), Viruses (421), Customize ...

Molecule types: genomic DNA/RNA (1,084,802), mRNA (596), rRNA (19), Customize ...

Source databases: INSDC (GenBank) (665,156), RefSeq (420,519), Customize ...

Genetic compartments: Mitochondrion (3), Plasmid (702)

Sequence length: Custom range...

Release date: Custom range...

Items: 1 to 20 of 1086199

<< First < Prev Page 1 of 54310 Next > Last >>

Found 1086373 nucleotide sequences. Nucleotide (1086199) EST (152) GSS (22)

1. Mycobacterium bovis subsp. bovis AF2122/97 complete genome
4,345,492 bp circular DNA
Accession: BX248333.1 GI: 31742509
GenBank FASTA Graphics

2. Mycobacterium tuberculosis H37Rv complete genome
4,411,532 bp linear DNA
Accession: AL123456.3 GI: 444893469
GenBank FASTA Graphics

3. Mycobacterium tuberculosis H37RvSiena, complete genome
4,410,911 bp circular DNA
Accession: CP007027.1 GI: 749193065
GenBank FASTA Graphics

4. Streptomyces coelicolor A3(2) complete genome
8,667,507 bp linear DNA
Accession: AL645882.2 GI: 30407153
GenBank FASTA Graphics

Results by taxon

Top Organisms [Tree]
Mycobacterium tuberculosis (935613)
Salmonella enterica (22391)
Escherichia coli (9029)
Klebsiella pneumoniae (6403)
Pseudomonas aeruginosa (4483)
All other taxa (108280)
More...

Find related data
Database: Select
Find items

Search details

```
"Mycobacterium tuberculosis"[Organism]
OR mycobacterium tuberculosis[All
Fields]
```

.rprt:nth-child(2) .title

Figure 3.15 SelectorGadget working

To get the css selector one must follow the steps shown in the above figure. First one needs to click on the selectorgadget icon in the bookmark menu of the browser. Then click on the content to be extracted. Once the content is selected the css selector will appear in the box at the bottom of the page. In this way, css selector can be used to extract the content from the web page. Similar concept was used to retrieve the sequence from NCBI. Once the sequence is retrieved the protein sequence of the gene is obtained to check the codon in the reference genome. The codon being coded after the SNP occurred at a particular position in the genome. This information is used for the classification of SNPs as synonymous, non synonymous and nonsense. Later the effect of non synonymous SNPs on the protein is identified using the Protein Variation Effect Analyzer (PROVEAN) and Have (y)Our Protein Explained (HOPE) tools. The jobs on both the tools are submitted programmatically using the concept of web scraping. PROVEAN is an online tool which can predict the effect of the amino acid substitutions on the protein [107]. The tools give the PROVEAN score and prediction whether the amino acid substitution is deleterious or neutral. The jobs to the PROVEAN protein server are submitted one by one and the results are extracted as soon as the server reports them. HOPE server is also used for predicting the effect of non synonymous SNPs on the protein. It provides the information how the point mutation affects the structure of the protein. It generates the output in the form of an HTML report. The job to HOPE server is submitted programmatically and the report is downloaded. INDELs are classified as frame shift and non frame shift INDELs. The output of the pipeline is in the form of excel and pdf files. The two excel files are generated. The file Annotated_SNPs.xls contains the annotated SNPs and Annotated_INDELs.xls contains the annotated INDELs. The pdf file contains the charts and tables. The different types of charts are generated example bar, pie, petal pie and stacked bar charts. The charts and tables provide the number of the variants (SNPs and INDELs) present in a VCF file. In this way the whole pipeline is automated in R. User only needs to provide the VCF file the rest of the work is done by the pipeline and at the end user gets the results of annotation.

AutoAnnotate uses the already available packages of R. rvest package is used for web scraping, Biostrings package for fetching the nucleotide sequence, refGenome for GTF file processing, ggplot2 for generating the plots and gridExtra package for generating the tables.

CHAPTER 4

RESULTS & DISCUSSION

CHAPTER 4

4. Results & Discussion

Results of this study are discussed in detail in this chapter. The NGS dataset mentioned in Chapter 3 was downloaded from ENA and using the optimized NGS data analysis pipeline the genomic variants (SNPs and INDELs) were identified in this dataset.

4.1 Quality Control (QC) results before trimming

NGS data analysis pipeline first step is to check the quality of the reads. The QC results for all the samples are shown in Figure 4.1 (a-c).

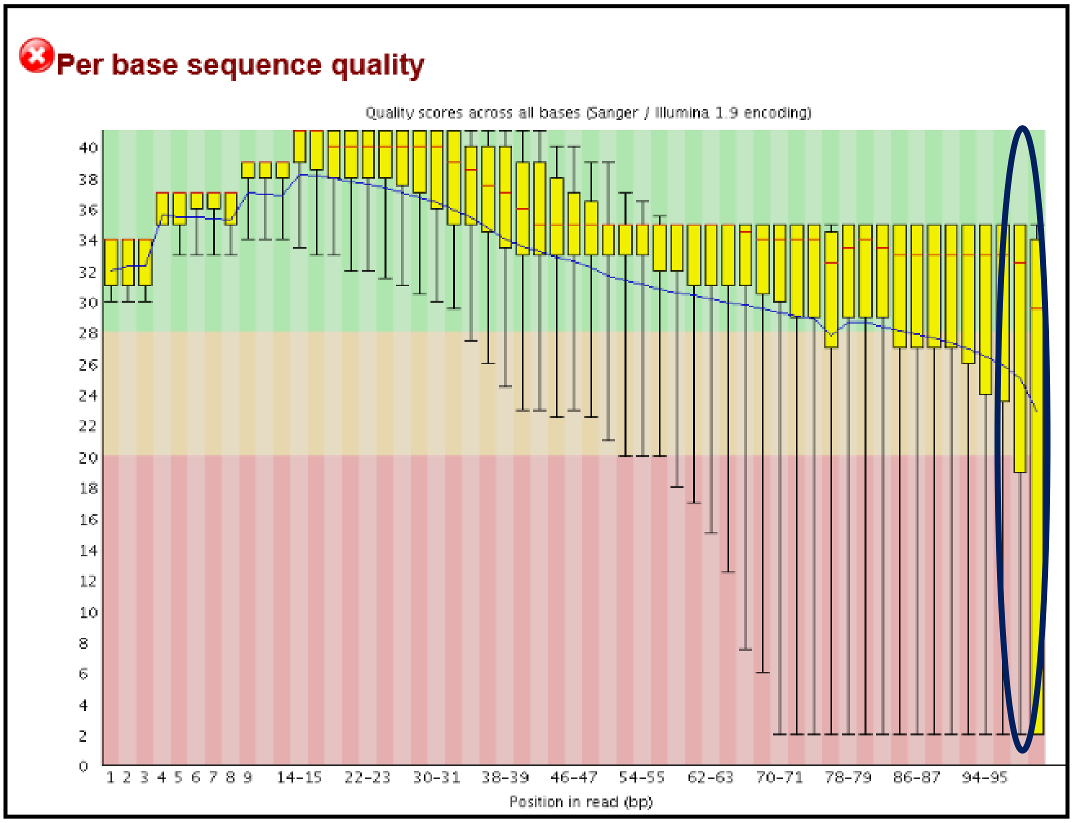


Figure 4.1 (a) FASTQC results of SAMEA3139819

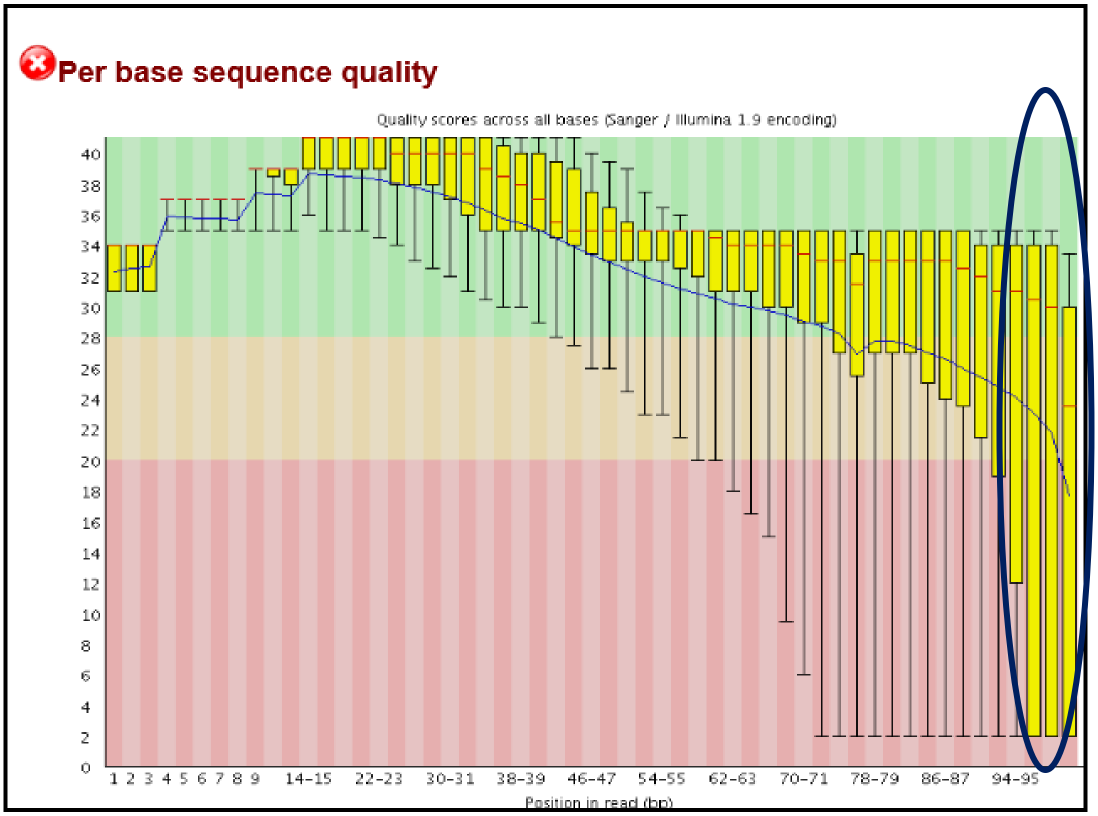


Figure 4.1 (b) FASTQC result of SAMEA3139820

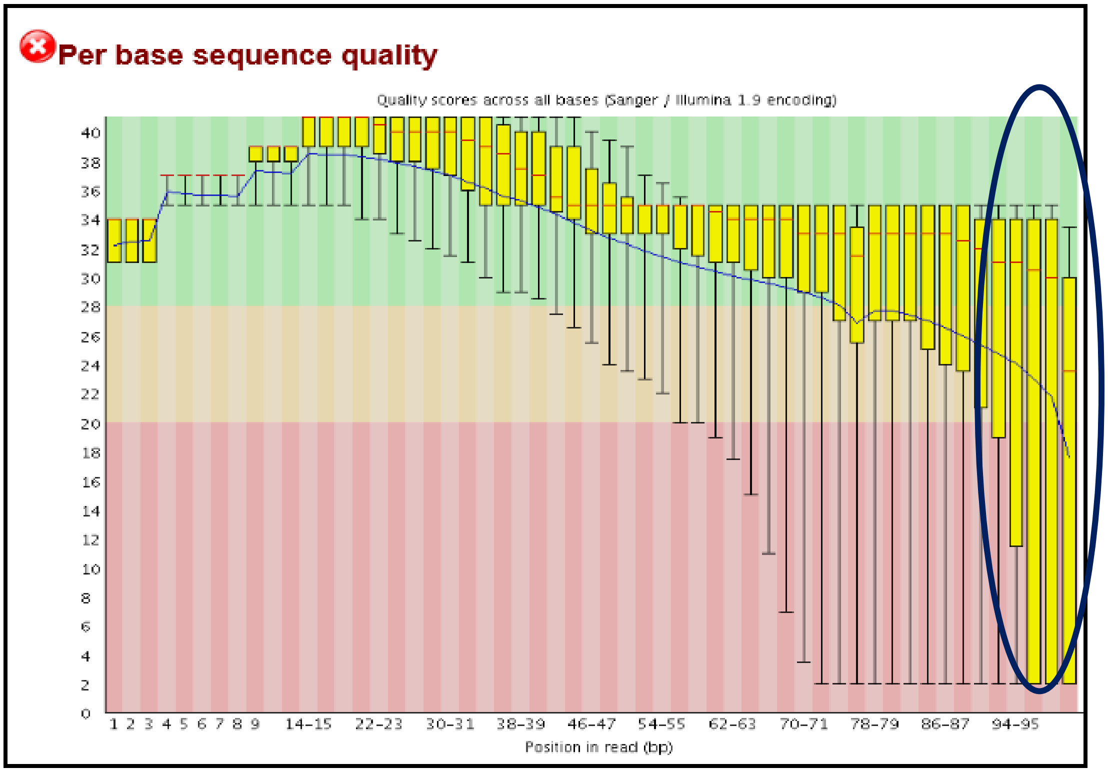


Figure 4.1 (c) FASTQC result of SAMEA3139821

The results in Figure 4.1 (a-c) show quality of the reads in three of the samples. Other two samples have similar results as shown in figure 4.1 (a-c). All the samples contain the poor quality reads. The plot shows the quality scores on the y-axis. In the plot the y-axis is divided into the three regions. The third region contains the good quality region (green), moderate quality region (orange) and the poor quality region (red). The poor quality reads are highlighted in the Figure 4.1 (a-c).

4.2 Quality Control (QC) results after trimming

As the quality of some of reads was not good so these reads were trimmed and the quality of these reads were checked after trimming. Figure 4.2 (a-c) show the samples QC results after the trimming of the poor quality reads was done.

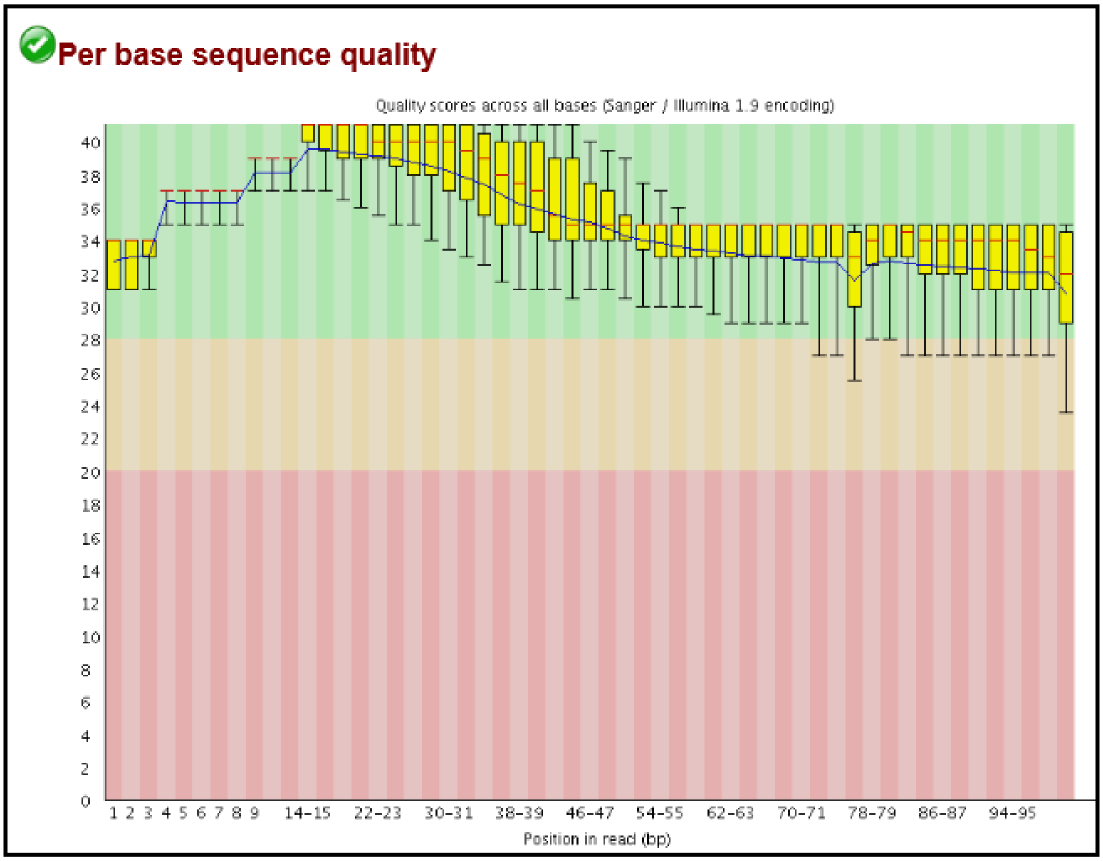


Figure 4.2 (a) FASTQC result of SAMEA3139819

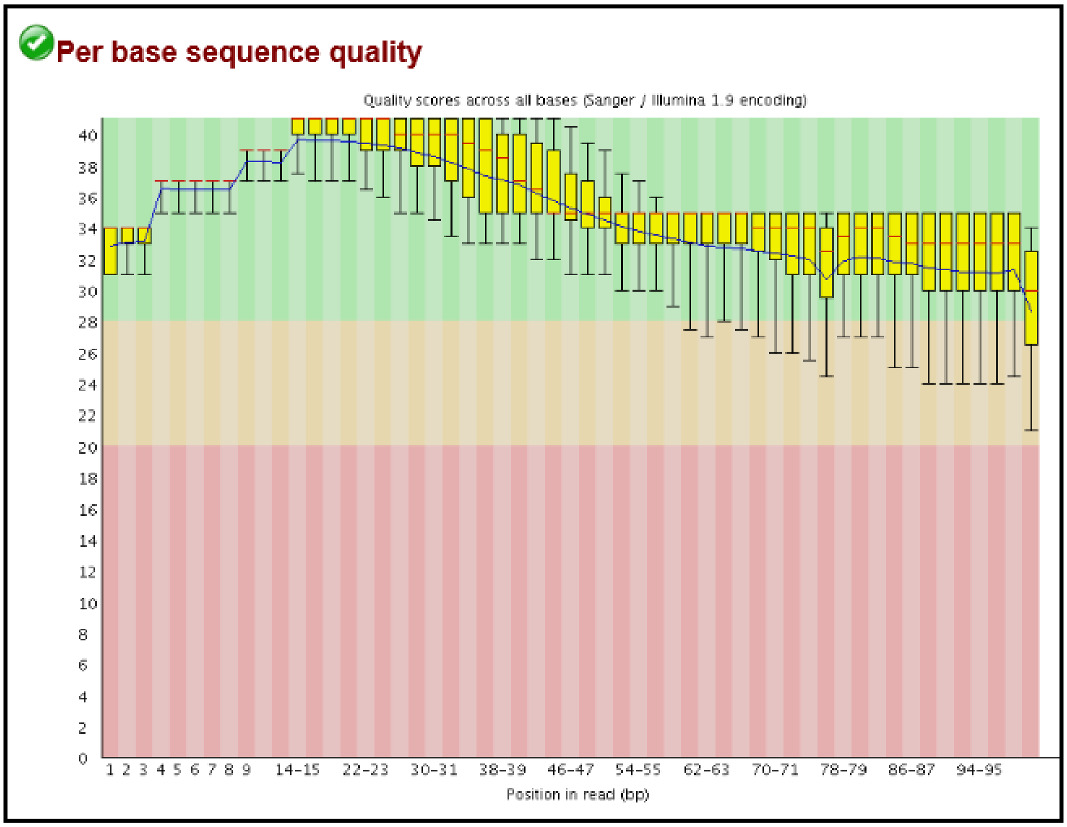


Figure 4.2 (b) FASTQC result of SAMEA3139820

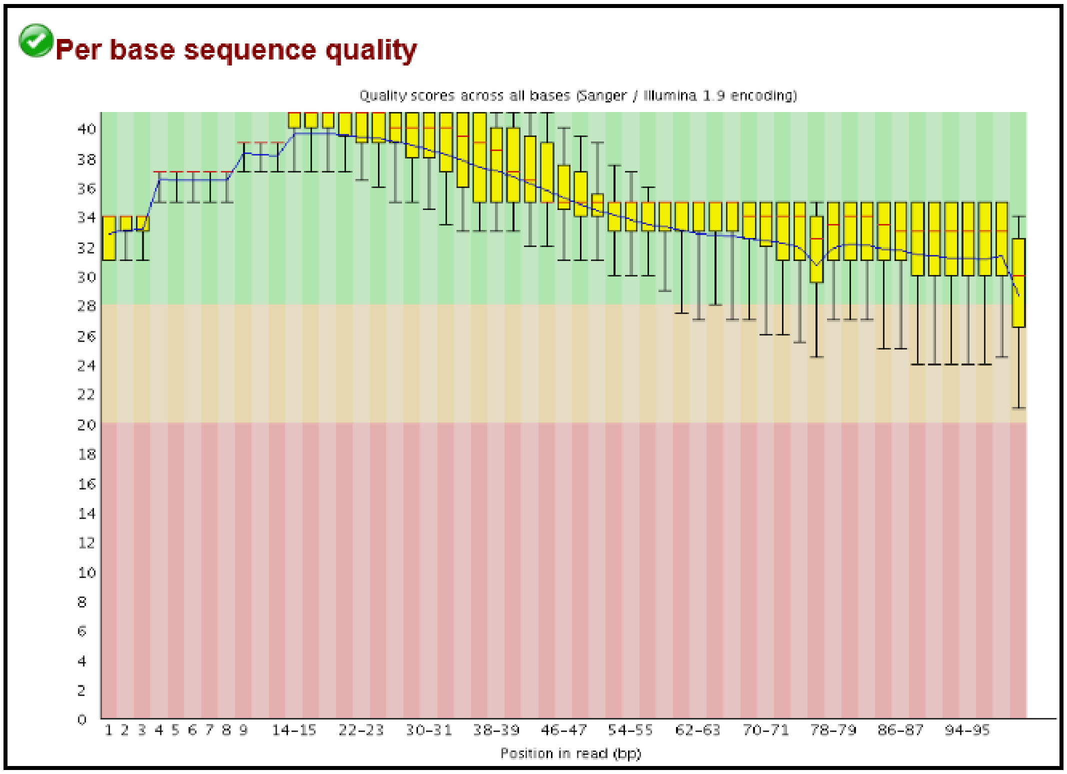


Figure 4.2 (c) FASTQC result of SAMEA3139821

QC after trimming was done in order to check whether the quality of reads has improved or not. Figure 4.2 (a-c) shows that all the reads after trimming lie in the good quality region (green). Now these reads can be used for the downstream analysis.

4.3 Alignment Summary

Once the alignment was done using BWA-mem, later the alignment summary for all the samples was obtained using *CollectAlignmnetSummaryMetrics* module of the Picard tools. Alignment summary information for paired end data is shown in the Table 4.1.

Table 4.1 Alignment Summary Results

CATEGORY	FIRST_OF_PAIR	SECOND_OF_PAIR	PAIR
TOTAL_READS	18352977	18352977	36705954
PF_READS	18352977	18352977	36705954
PCT_PF_READS	1	1	1
PF_NOISE_READS	0	0	0
PF_READS_ALIGNED	18258267	18251712	36509979
PCT_PF_READS_ALIGNED	0.99484	0.994482	0.994661
PF_ALIGNED_BASES	1748915856	1739412091	3.49E+09
PF_HQ_ALIGNED_READS	17989442	17981491	35970933
PF_HQ_ALIGNED_BASES	1724298806	1714846439	3.44E+09
PF_HQ_ALIGNED_Q20_BASES	1699926627	1690183700	3.39E+09
PF_HQ_MEDIAN_MISMATCHES	0	0	0
PF_MISMATCH_RATE	0.001373	0.001352	0.001363
PF_HQ_ERROR_RATE	0.001357	0.001336	0.001347
PF_INDEL_RATE	0.000054	0.000058	0.000056
MEAN_READ_LENGTH	96.1031	95.886713	95.99491
READS_ALIGNED_IN_PAIRS	18241006	18241006	36482012
PCT_READS_ALIGNED_IN_PAIRS	0.999055	0.999413	0.999234
BAD_CYCLES	0	0	0
STRAND_BALANCE	0.499482	0.500641	0.500062
PCT_CHIMERAS	0.0099	0.0099	0.0099
PCT_ADAPTER	0.000403	0.000402	0.000402

Table 4.1 contains the details about the alignment summary of sample 1. Similar metrics were generated for other samples as well. There are three categories in this table. The First of Pair category contains information about the read that is first in the pair whereas the Second of Pair category contains information about the read that is second in the pair and the Pair category contains the collective information about the reads. Parameters are described in the Table 4.2.

Table 4.2 Alignment Parameters Summary Description

Parameters	Description
TOTAL_READS	Number of reads in the different pairs as well as in pair.
PF_READS	Number of reads that have passed the Illumina filter.
PCT_PF_READS	Percentage of the reads that passed the Illumina filter.
PF_NOISE_READS	Number of reads that are found as noise.
PF_READS_ALIGNED	Number of reads aligned with the reference genome.
PCT_PF_READS_ALIGNED	Percentage of reads (PF) aligned with the reference genome.
PF_ALIGNED_BASES	Number of the bases aligned with the reference genome.
PF_HQ_ALIGNED_READS	Number of reads aligned with reference genome with a quality (Q) equal (Q20) or greater than 20.
PF_HQ_ALIGNED_BASES	Number of bases aligned with reference genome having high quality.
PF_HQ_ALIGNED_Q20_BASES	Number of bases aligned with reference genome with a quality (Q) equal (Q20) or greater than 20.
PF_HQ_MEDIAN_MISMATCHES	It represents the median of the mismatches with high quality.
PF_MISMATCH_RATE	Rate of the mismatched bases present in reads that are aligned with the reference at good quality.
PF_HQ_ERROR_RATE	Percentage of the mismatched bases compared to the reference genome.
PF_INDEL_RATE	Number of INDELS, occurring every 100 aligned bases.
MEAN_READ_LENGTH	Represent reads length mean.
READS_ALIGNED_IN_PAIRS	Number of reads aligned along with their pair to reference genome.
PCT_READS_ALIGNED_IN_PAIRS	Percentage of reads aligned along with their pair to reference genome.
BAD_CYCLES	Number of the cycles where 80 % of bases are not called.
STRAND_BALANCE	Ratio of number of reads that are aligned to genome's positive strand to the number of reads that are aligned with a reference genome.
PCT_CHIMERAS	Percentage of reads that align outside the insert size.
PCT_ADAPTER	Percentage of reads that are not aligned and are similar to adapter sequence.

The collective alignment summary for all the samples is described in Table 4.3.

Table 4.3 Collective Alignment Summary

Category	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
TOTAL_READS	36705954	14664584	14449752	17948606	36705954
PF_READS	36705954	14664584	14449752	17948606	36705954
PCT_PF_READS	1	1	1	1	1
PF_NOISE_READS	0	0	0	0	0
PF_READS_ALIGNED	36509979	14560431	14354898	17862251	36509979
PCT_PF_READS_ALIGNED	0.994661	0.992898	0.993436	0.995189	0.994661
PF_ALIGNED_BASES	3488327947	1381360517	1367651676	1693573843	3488327947
PF_HQ_ALIGNED_READS	35970933	14367579	14156734	17603045	35970933
PF_HQ_ALIGNED_BASES	3439145245	1364009113	1349649544	1670154989	3439145245
PF_HQ_ALIGNED_Q20_BASES	3390110327	1344668649	1330222101	1646001910	3390110327
PF_HQ_MEDIAN_MISMATCHES	0	0	0	0	0
PF_MISMATCH_RATE	0.001363	0.001198	0.001126	0.001177	0.001363
PF_HQ_ERROR_RATE	0.001347	0.001183	0.001111	0.001161	0.001347
PF_INDEL_RATE	0.000056	0.000053	0.000045	0.000038	0.000056
MEAN_READ_LENGTH	95.994906	95.662043	95.627128	95.171371	95.994906
READS_ALIGNED_IN_PAIRS	36482012	14548468	14343558	17855094	36482012
PCT_READS_ALIGNED_IN_PAIRS	0.999234	0.999178	0.99921	0.999599	0.999234
BAD_CYCLES	0	0	0	0	0
STRAND_BALANCE	0.500062	0.500042	0.500008	0.500033	0.500062
PCT_CHIMERAS	0.0099	0.013237	0.008944	0.003286	0.0099
PCT_ADAPTER	0.000402	0.000149	0.000086	0.000047	0.000402

4.4 Duplication Metrics

Duplicates were marked after the alignment was done. Duplication metrics were obtained using *MarkDuplicates* module of Picard tools. Duplication Metrics were obtained for all the samples. Table 4.4 shows the collective information of duplication metrics for all the samples.

Table 4.4 Duplication Metrics

Category	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
UNPAIRED_READS_EXAMINED	27967	11963	11340	7157	30660
READ_PAIRS_EXAMINED	18241006	7274234	7171779	8927547	17745536
UNMAPPED_READS	195975	104153	94854	86355	200626
UNPAIRED_READ_DUPLICATES	26183	10085	9187	6407	29284
READ_PAIR_DUPLICATES	6302818	1268742	807927	761606	814722
READ_PAIR_OPTICAL_DUPLICATES	0	0	0	0	0
PERCENT_DUPLICATION	0.345983	0.174965	0.113205	0.085634	0.046696

The parameters in category column are described in Table 4.5.

Table 4.5 Duplication Metrics Parameter Description

Parameters	Description
UNPAIRED_READS_EXAMINED	Number of mapped reads identified that don’t have a mapped pair.
READ_PAIRS_EXAMINED	Number of mapped reads along with their pair mapped also.
UNMAPPED_READS	Number of reads that are not mapped.
UNPAIRED_READ_DUPLICATES	Number of duplicates marked.
READ_PAIR_DUPLICATES	Number of duplicated read pairs.
READ_PAIR_OPTICAL_DUPLICATES	Number of duplicated read pairs generated because of optical duplication.
PERCENT_DUPLICATION	Percentage of aligned sequence identified as duplicate.

4.5 GC Bias Plot

To find out the Guanine-Cytosine content (GC) in the samples GC bias plots were generated using *CollectGcBiasMetrics* module of the Picard tools. Figure 4.3 shows the GC bias plot.

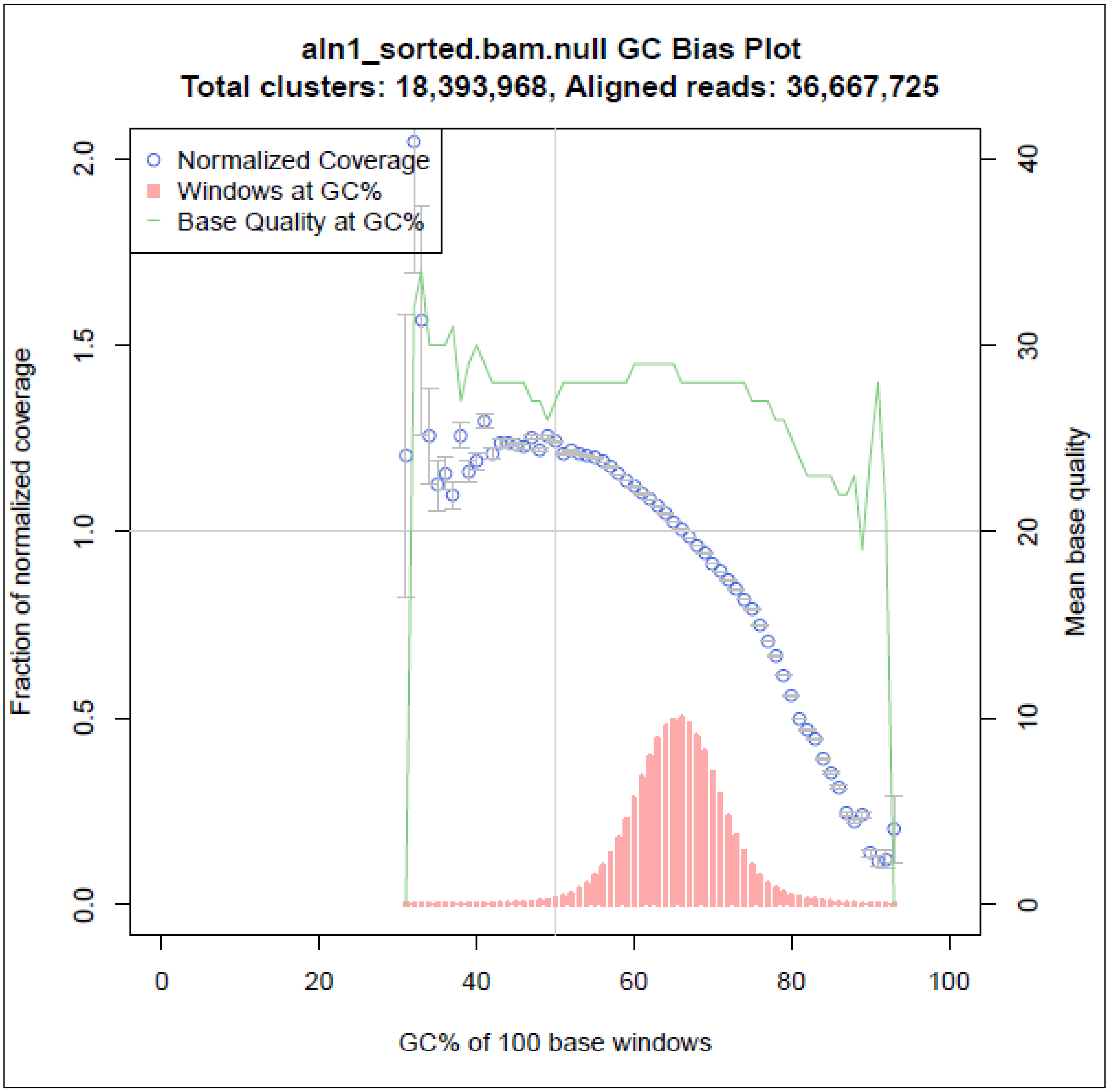


Figure 4.3 Sample 1 GC Bias Plot

GC bias plot has 100 base window GC % content on the x-axis. In the GC bias plot blue circles show the normalized coverage, red lines indicate the reference sequence % GC and green line indicates base quality. GC bias plot was generated for all samples and the plots obtained have the results quite similar to the one shown in Figure 4.3.

4.6 Insert Size Metrics

Insert size metrics was calculated for all samples using *CollectInsertSizeMetrics* module of the Picard tools. Along with the metrics the module also generated the histogram. Table 4.6 shows the collective information of insert size metrics for all the samples.

Table 4.6 Insert Size Metrics

Field	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
MEDIAN_INSERT_SIZE	230	242	246	161	252
MEDIAN_ABSOLUTE_DEVIATION	13	16	14	23	18
MIN_INSERT_SIZE	2	2	3	2	2
MAX_INSERT_SIZE	4393630	4381941	4392694	4367435	4397674
MEAN_INSERT_SIZE	224.51874	228.0313	236.0546	168.1635	238.2449
STANDARD_DEVIATION	33.555345	44.61613	37.97004	42.83281	43.61087
READ_PAIRS	11817877	5926781	6309926	8136587	16780537

The parameters in field column are described in Table 4.7.

Table 4.7 Insert Size Metrics Parameters Description

Parameters	Description
MEDIAN_INSERT_SIZE	Paired end reads median insert size, where both reads are mapped to the similar chromosome.
MEDIAN_ABSOLUTE_DEVIATION	Distribution of the median absolute deviation.
MIN_INSERT_SIZE	Minimum insert size.
MAX_INSERT_SIZE	Maximum insert size obtained as result of alignment.
MEAN_INSERT_SIZE	Distribution mean insert size.
STANDARD_DEVIATION	Distribution standard deviation.
READ_PAIRS	Number of paired reads observed in the distribution.

In Figure 4.4 (a) x-axis shows the insert size whereas y-axis shows the read count. It is evident from the Table 4.6 that the median insert sizes for the sample 4 is 161. Histogram for all the samples was obtained.

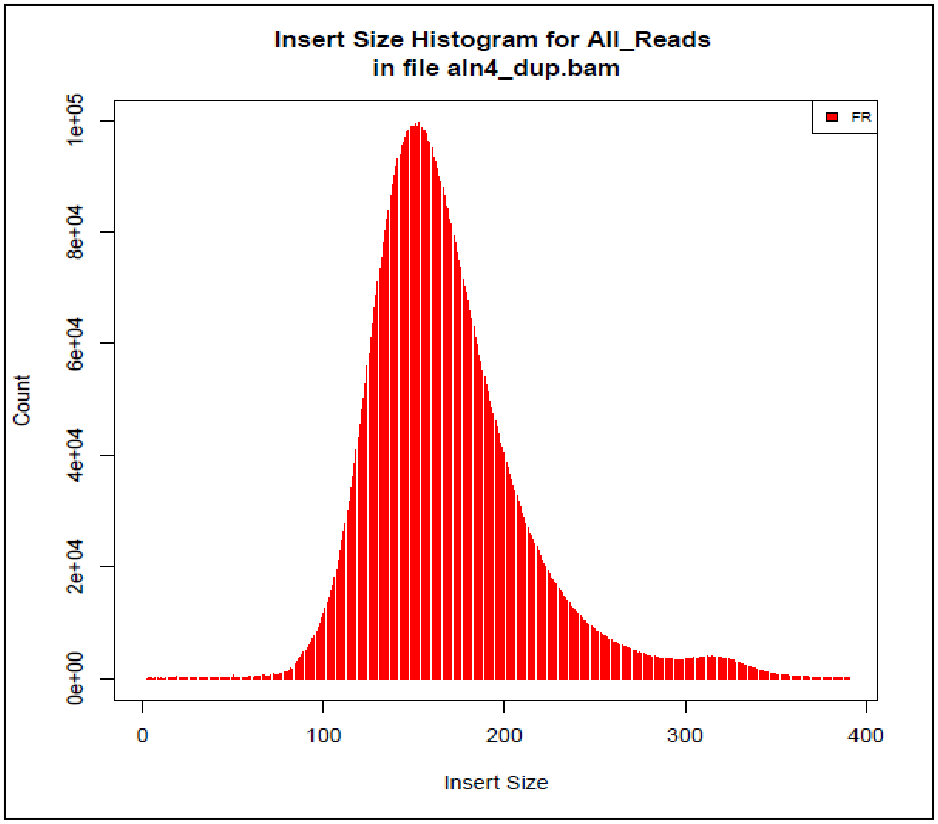


Figure 4.4 (a) Insert Size Histogram of Sample 4

Figure 4.4 (b-e) shows the histogram for the other four samples.

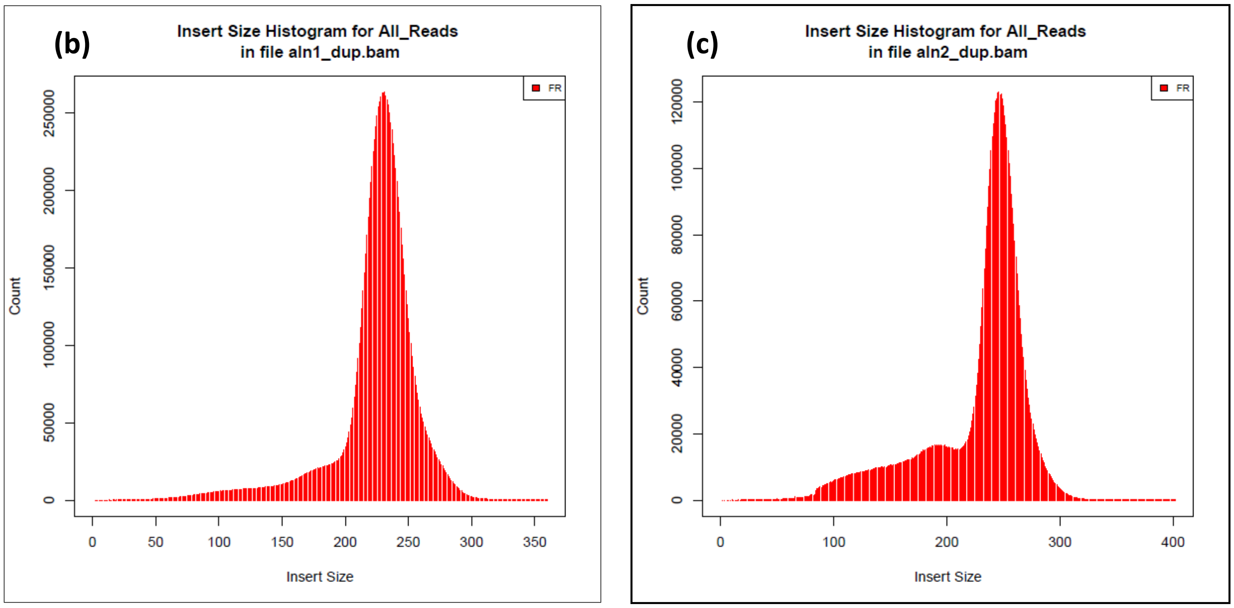


Figure 4.4 (b-c) Insert Size Histogram of Sample 1 & Sample 2

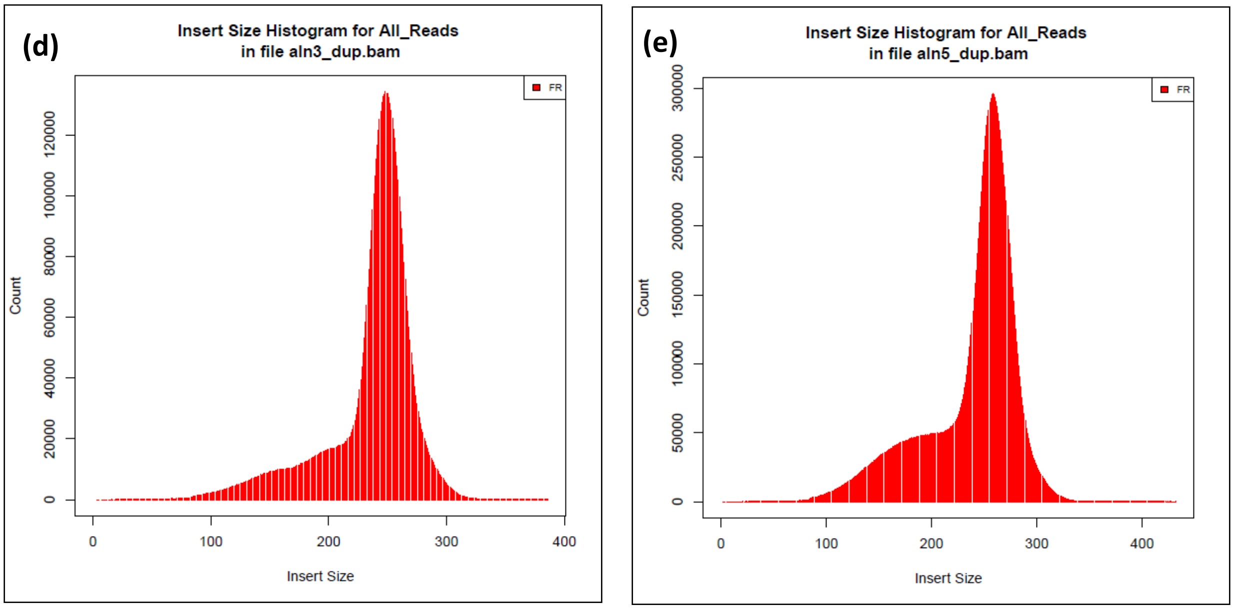


Figure 4.4 (d-e) Insert Size Histogram of Sample 3 & Sample 5

4.7 Variant Calling

Variants in the dataset were called using *HaplotypeCaller* module of GATK. The number of called variants in each sample are shown in Table 4.8.

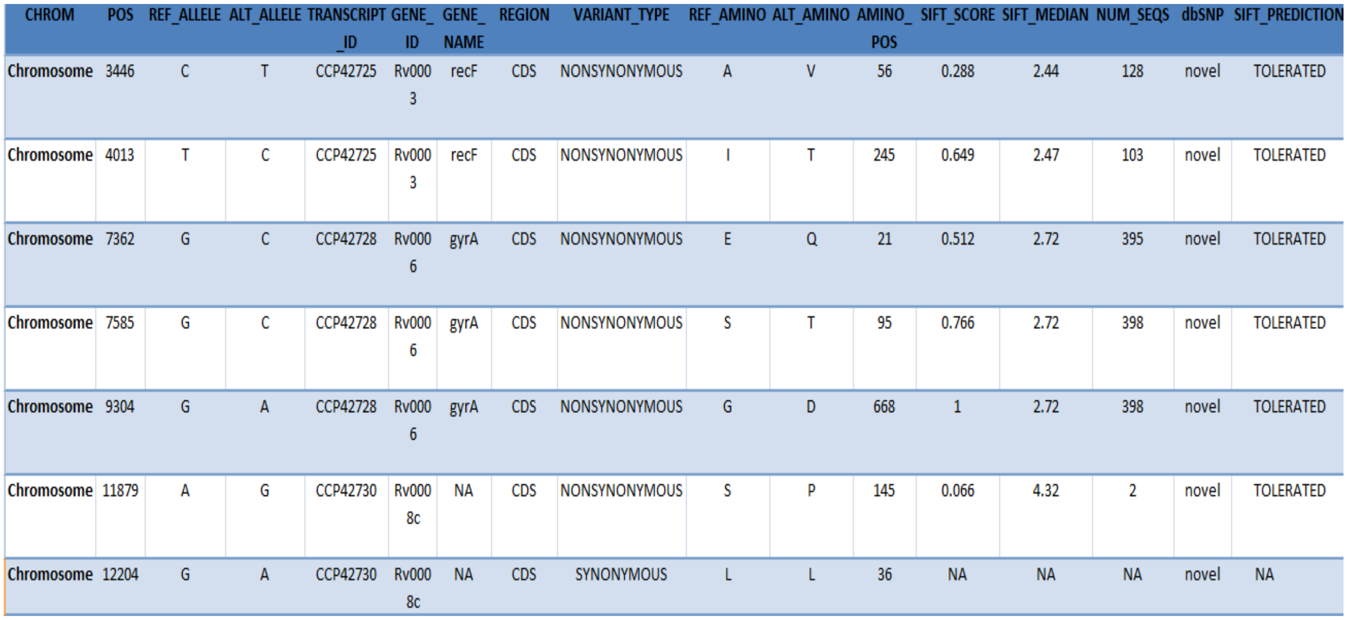
Table 4.8 Number of Variants

Sample	Number of variants
1	1638
2	1697
3	1735
4	1721
5	1879

4.8 Annotation

Annotation was done using SIFT4G software. Files having the variants for all the samples were annotated. Results were obtained in the excel file. Samples had the different number of variants. Table 4.8 shows the number of variants in each sample.

Figure 4.5 shows the annotation results for sample 1.



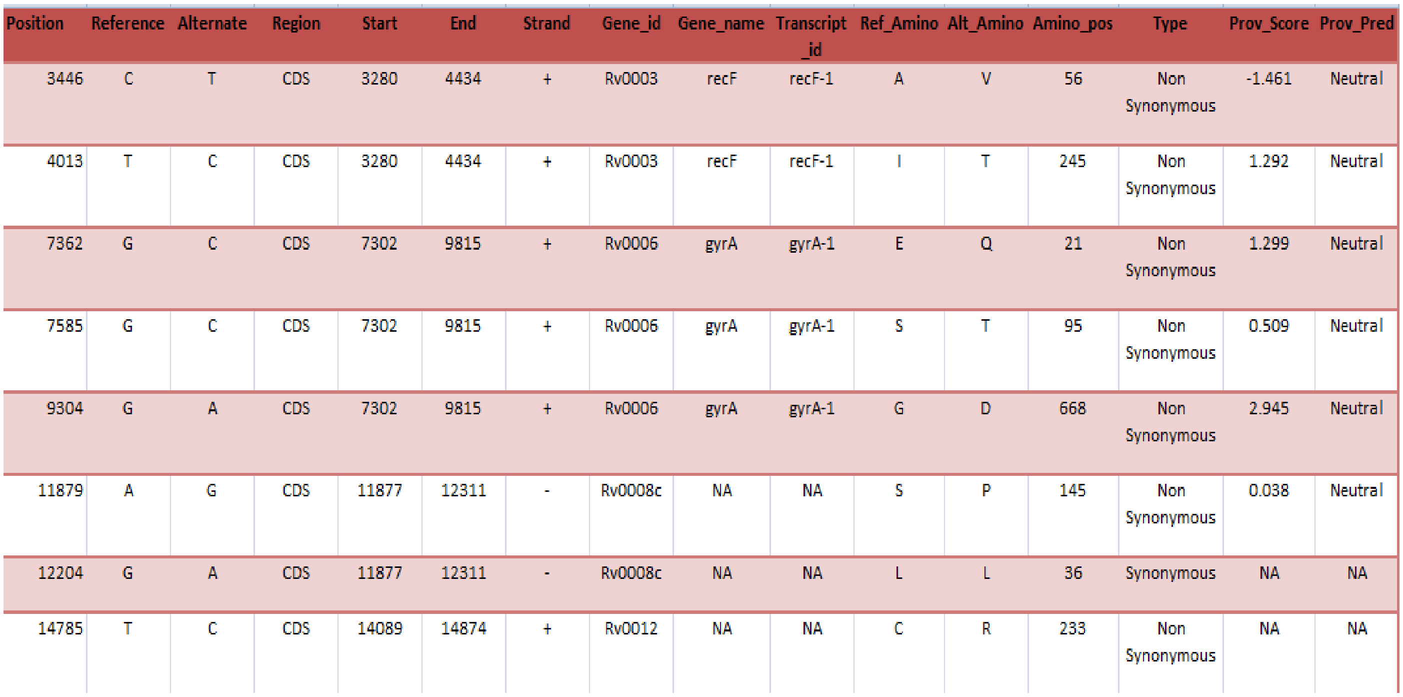
CHROM	POS	REF_ALLELE	ALT_ALLELE	TRANSCRIPT_ID	GENE_ID	GENE_NAME	REGION	VARIANT_TYPE	REF_AMINO	ALT_AMINO	AMINO_POS	SIFT_SCORE	SIFT_MEDIAN	NUM_SEQS	dbSNP	SIFT_PREDICTION
Chromosome	3446	C	T	CCP42725	Rv0003	recF	CDS	NONSYNONYMOUS	A	V	56	0.288	2.44	128	novel	TOLERATED
Chromosome	4013	T	C	CCP42725	Rv0003	recF	CDS	NONSYNONYMOUS	I	T	245	0.649	2.47	103	novel	TOLERATED
Chromosome	7362	G	C	CCP42728	Rv0006	gyrA	CDS	NONSYNONYMOUS	E	Q	21	0.512	2.72	395	novel	TOLERATED
Chromosome	7585	G	C	CCP42728	Rv0006	gyrA	CDS	NONSYNONYMOUS	S	T	95	0.766	2.72	398	novel	TOLERATED
Chromosome	9304	G	A	CCP42728	Rv0006	gyrA	CDS	NONSYNONYMOUS	G	D	668	1	2.72	398	novel	TOLERATED
Chromosome	11879	A	G	CCP42730	Rv0008c	NA	CDS	NONSYNONYMOUS	S	P	145	0.066	4.32	2	novel	TOLERATED
Chromosome	12204	G	A	CCP42730	Rv0008c	NA	CDS	SYNONYMOUS	L	L	36	NA	NA	NA	novel	NA

Figure 4.5 Sample 1 Annotation Results

Excel files obtained for all the samples have annotated SNPs along with INDELs. Output excel file contains different columns containing the information about position of variant in the genome, reference allele, alternative allele, transcript id, gene id, gene name, region in which the variant is located, type of the variant, reference amino acid, alternative amino acid, SIFT score, SIFT median, Single Nucleotide Polymorphism database (dbSNP) and SIFT prediction about the variant effect on the protein functionality.

4.9 AutoAnnotator

AutoAnnotator is an automated pipeline developed in R. It takes the vcf file as an input. It generates the annotated SNPs in an excel file named as Annotation_SNPs.xls and annotated INDELs in an excel file named as Annotated_INDELs.xls. The annotation results for SNPs in sample 1 are shown in Figure 4.6 (a).

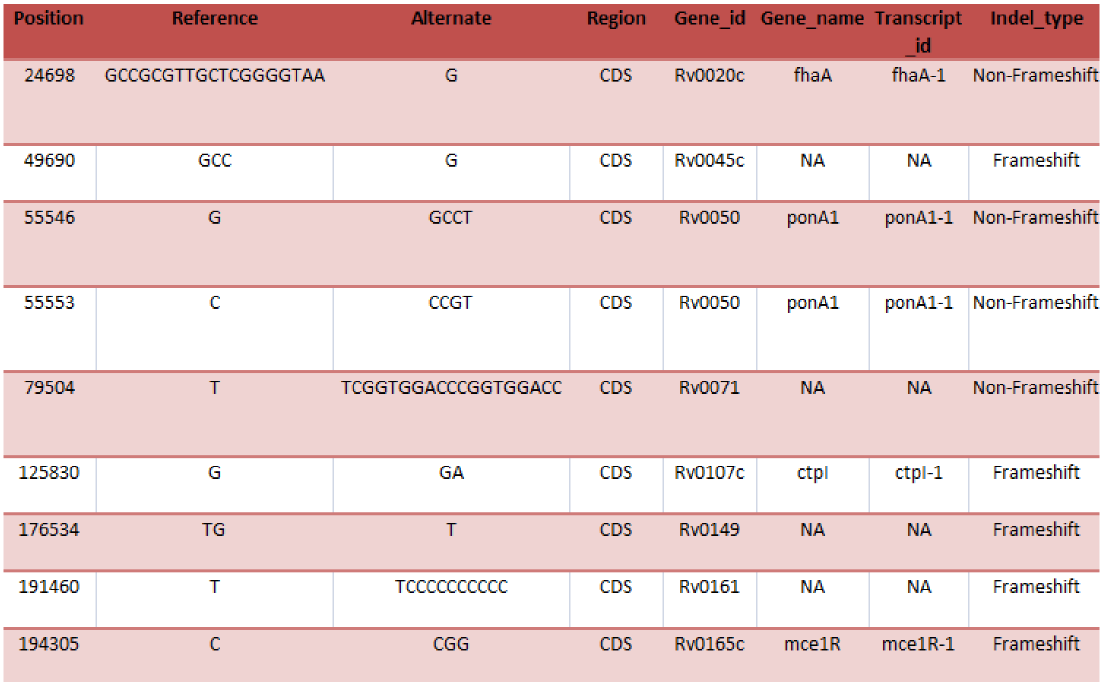


Position	Reference	Alternate	Region	Start	End	Strand	Gene_id	Gene_name	Transcript_id	Ref_Amino	Alt_Amino	Amino_pos	Type	Prov_Score	Prov_Pred
3446	C	T	CDS	3280	4434	+	Rv0003	recF	recF-1	A	V	56	Non Synonymous	-1.461	Neutral
4013	T	C	CDS	3280	4434	+	Rv0003	recF	recF-1	I	T	245	Non Synonymous	1.292	Neutral
7362	G	C	CDS	7302	9815	+	Rv0006	gyrA	gyrA-1	E	Q	21	Non Synonymous	1.299	Neutral
7585	G	C	CDS	7302	9815	+	Rv0006	gyrA	gyrA-1	S	T	95	Non Synonymous	0.509	Neutral
9304	G	A	CDS	7302	9815	+	Rv0006	gyrA	gyrA-1	G	D	668	Non Synonymous	2.945	Neutral
11879	A	G	CDS	11877	12311	-	Rv0008c	NA	NA	S	P	145	Non Synonymous	0.038	Neutral
12204	G	A	CDS	11877	12311	-	Rv0008c	NA	NA	L	L	36	Synonymous	NA	NA
14785	T	C	CDS	14089	14874	+	Rv0012	NA	NA	C	R	233	Non Synonymous	NA	NA

Figure 4.6 (a) Annotated SNPs

The file containing the annotated SNPs has 16 columns. The columns contain information about the Position of SNP, Reference allele, Alternative allele, Region in which the SNP is located, start and position of the region in which SNP is located, Strand on which the gene is present whether it is present on positive or the negative stand, Gene ID, Gene name, Transcript name, Amino acid present in reference, amino acid in alternate, Type of the SNP whether it is synonymous or non synonymous, PROVEAN score and PROVEAN prediction about the affect of amino acid change on protein function whether it is tolerated which means it does not have a significant effect on the protein function or it is deleterious which means it has significant effect on the protein function.

The annotation results for INDELs in sample 1 are shown in Figure 4.6 (b).



Position	Reference	Alternate	Region	Gene_id	Gene_name	Transcript_id	Indel_type
24698	GCCGCGTTGCTCGGGGTAA	G	CDS	Rv0020c	fhaA	fhaA-1	Non-Frameshift
49690	GCC	G	CDS	Rv0045c	NA	NA	Frameshift
55546	G	GCCT	CDS	Rv0050	ponA1	ponA1-1	Non-Frameshift
55553	C	CCGT	CDS	Rv0050	ponA1	ponA1-1	Non-Frameshift
79504	T	TCGGTGGACCCGGTGGACC	CDS	Rv0071	NA	NA	Non-Frameshift
125830	G	GA	CDS	Rv0107c	ctpI	ctpI-1	Frameshift
176534	TG	T	CDS	Rv0149	NA	NA	Frameshift
191460	T	TCCCCCCCCCC	CDS	Rv0161	NA	NA	Frameshift
194305	C	CGG	CDS	Rv0165c	mce1R	mce1R-1	Frameshift

Figure 4.6 (b) Annotated INDELs

File containing annotated INDELs contains eight columns. It includes position at which INDEL is located, reference allele, alternative allele, region, Gene ID, Gene name, Transcript and type of indel whether it is frame shift or non frame shift.

Different types of plots generated are shown in Figure 4.7 (a, b, c, d, e, f, g).

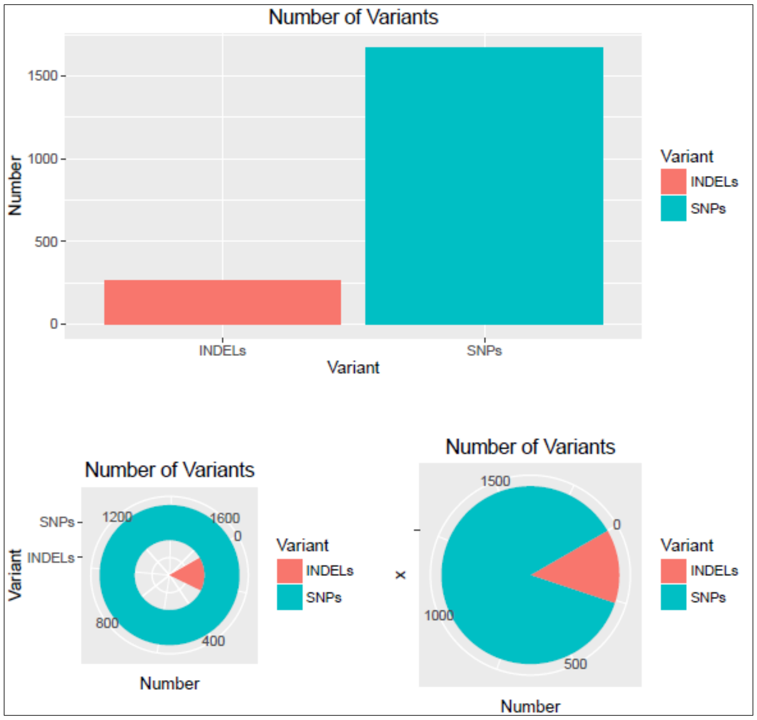


Figure 4.7 (a) Plots: Number of variants

Figure 4.7 (a) displays three types of plots. First the Bar chart shows the total number of INDELs and SNPs found in the data. Y-axis of the plot shows the number of variants found. The petal pie chart represents the number of variants. Pie chart also displays the number of variants in the form of pie.

Figure 4.7 (b) shows the Bar chart of different types of variants found in the data.

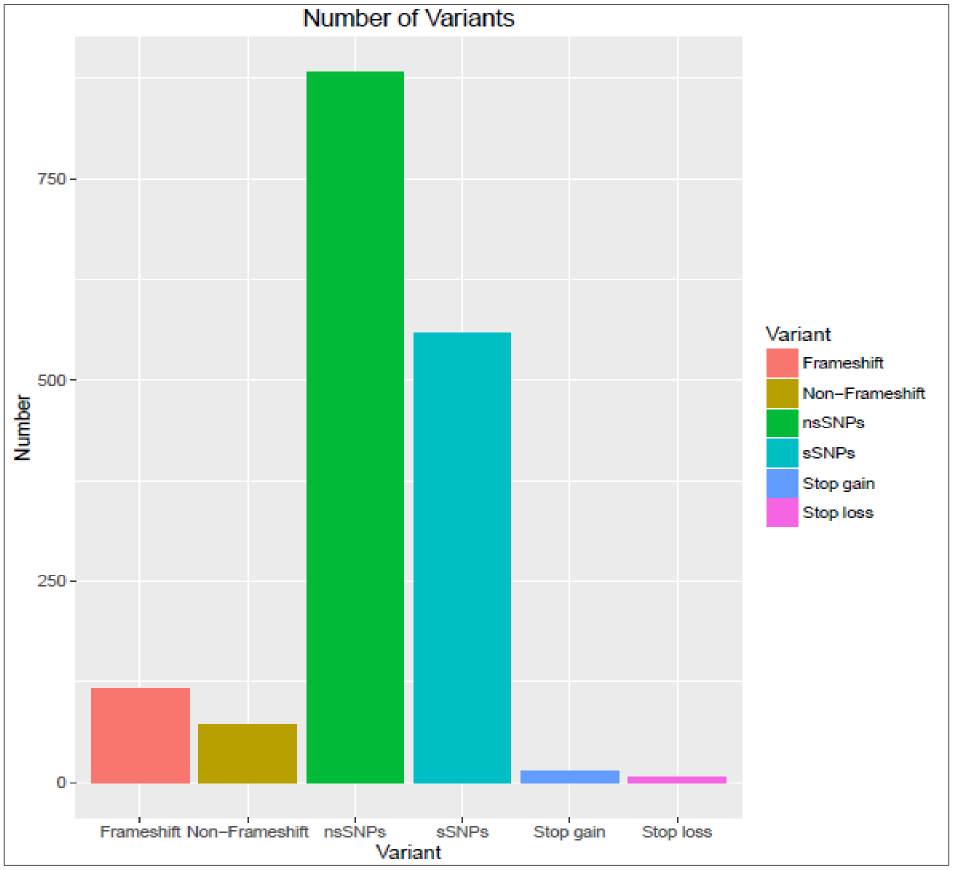


Figure 4.7 (b) Bar Chart: Types of variants

Bar chart in Figure 4.7 (b) displays the different types of variants on the x-axis which includes frame shift INDELs, non frame shift INDELs, non-synonymous SNPs (nsSNPs), substitution and synonymous SNPs (sSNPs). Y-axis of the plot shows the number of each variant found in the data. From the above figure it is evident that non synonymous SNPs (green bar) are more than 750 in number and synonymous SNPs (purple bar) are more than 500 in number.

Figure 4.7 (c) shows the pie chart of different types of variants found in the data.

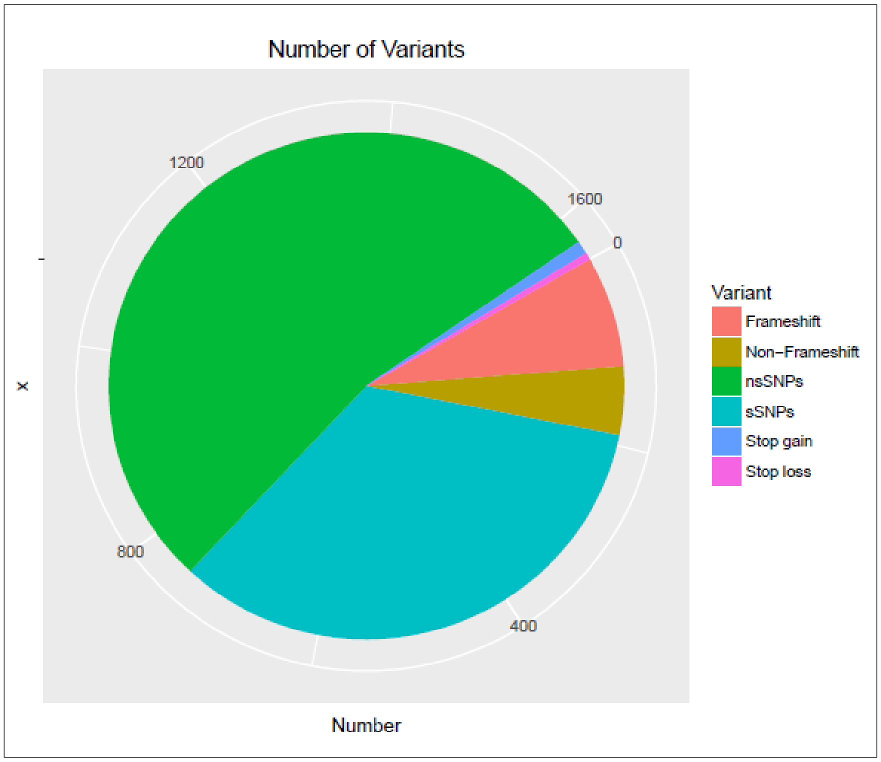


Figure 4.7 (c) Pie Chart: Types of variants

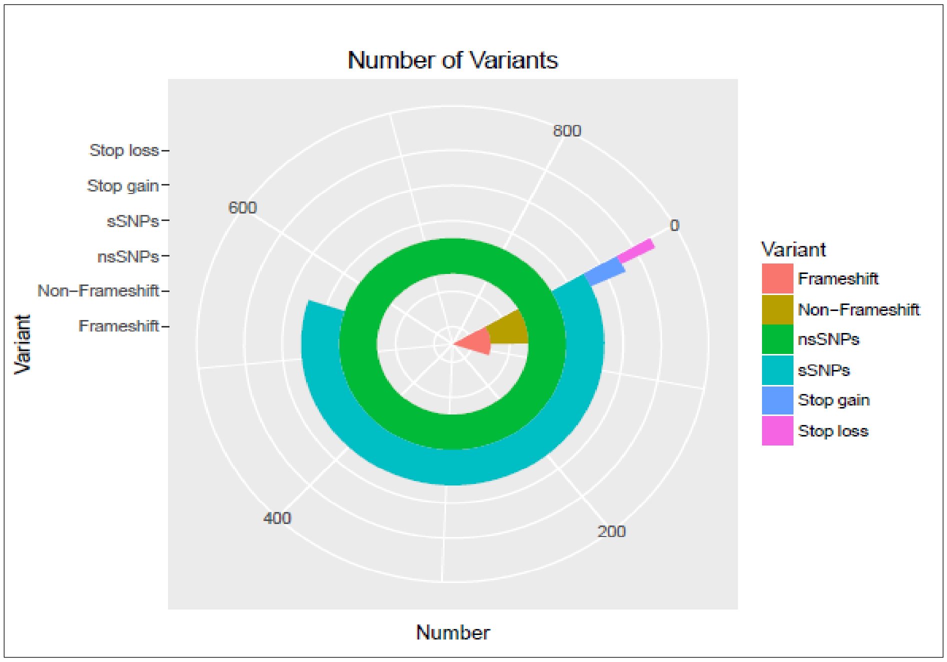


Figure 4.7 (d) Petal Pie: Types of variants

Figure 4.7 (c, d) displays the similar information as of Figure 34 (b) but in a different way. Figure 4.7 (c) shows the different types of variants present in the data in the form of pie. It is clear from the figure that synonymous and nonsynonymous SNPs cover the larger area as compared to the other types of variants. Figure 4.7 (d) displays the number of variants in the form the petals. It shows that stop gain and stop loss occurrence is lower as compared to the other variants.

Figure 4.7 (e) shows total number of insertion and deletion in the data in the form of bar chart.

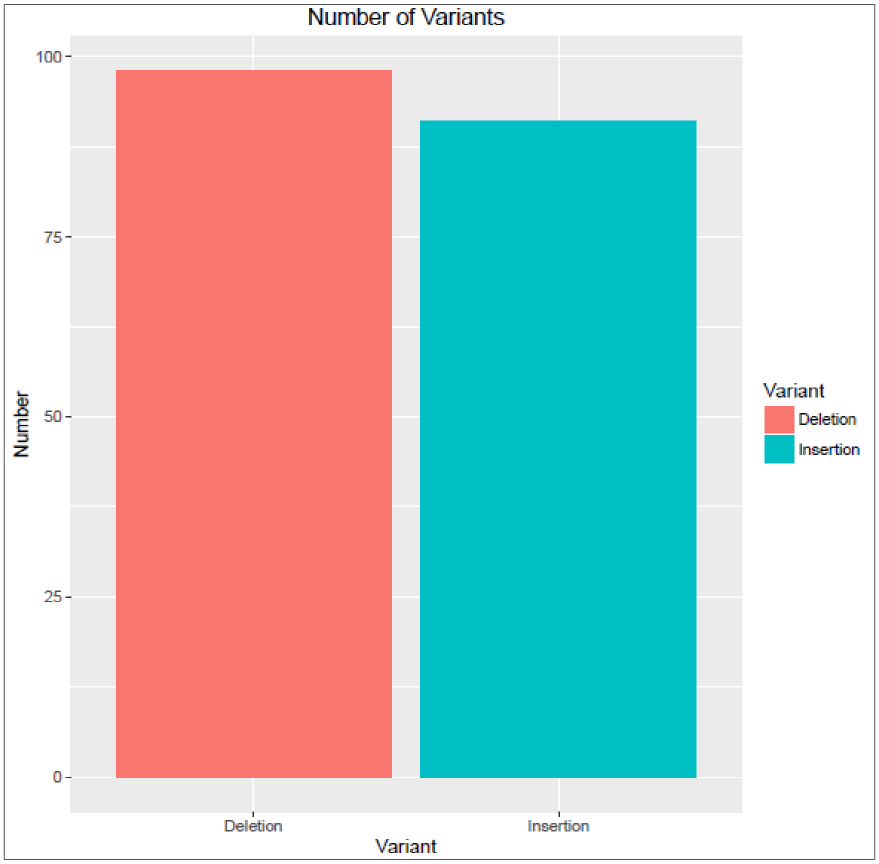


Figure 4.7 (e) Bar Chart: INDELs

Figure 4.7 (e) displays the variant type, which is insertion and deletion on the x-axis and the number of times it is occurring on the y-axis. Blue bar indicates insertions whereas red bar indicates deletions and it is evident from the above figure that deletions are more in number than insertions. Figure 4.7 (f , g) show the stacked bar chart displaying further classification of INDELs as the frame shift insertion or deletions and non-frame shift insertion or deletions.

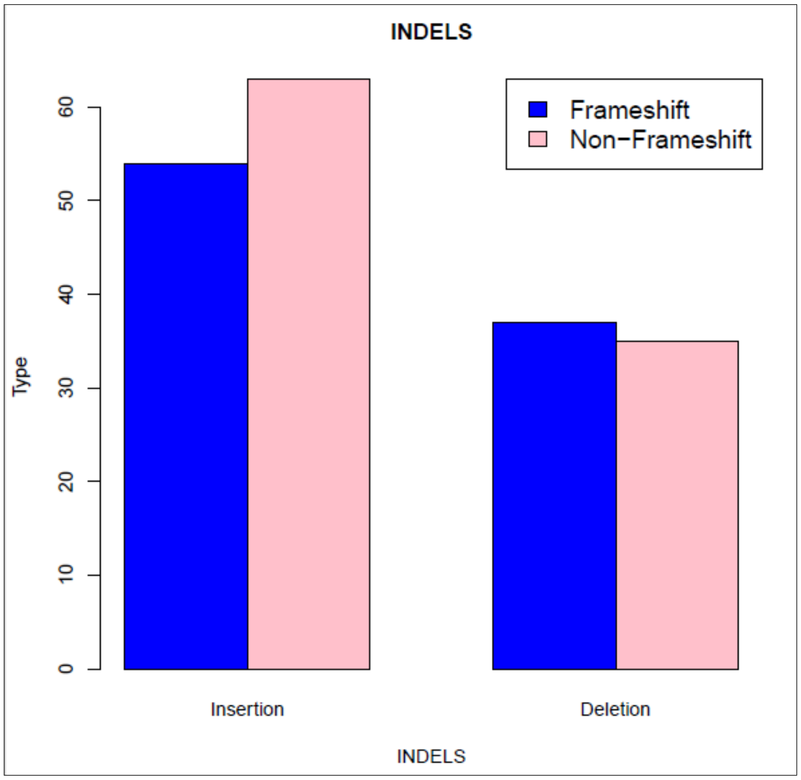


Figure 4.7 (f) Stacked Bar Chart: INDELs Type

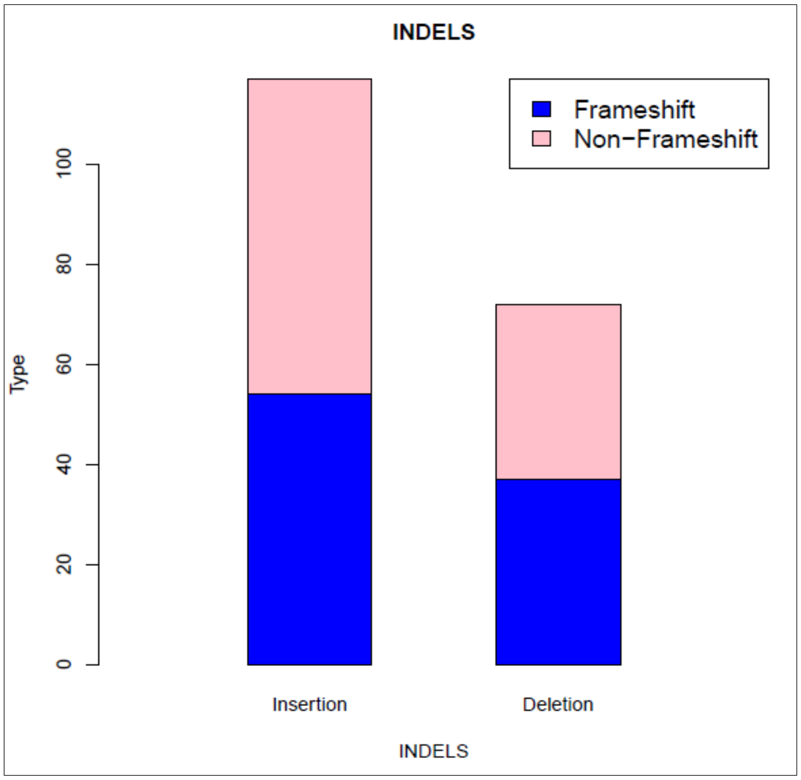
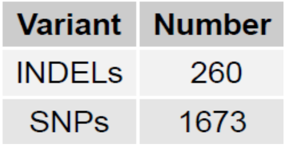


Figure 4.7 (g) Stacked Bar Chart: INDELs Type

Figure 4.7 (f, g) shows that the non-frame shift insertion and deletions in pink color and frame shift insertion and deletions in blue color. It is clear from the above figures that non-frame shift insertions are more in number that frame shift insertions and frame shift deletions are more in number than non-frame shift deletions in the dataset used.

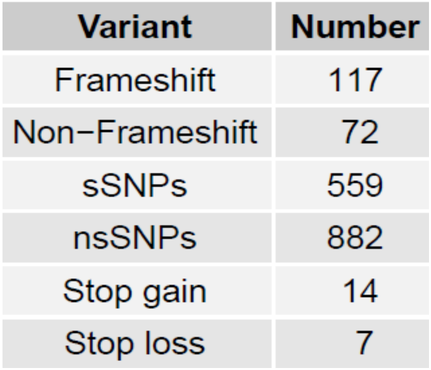
Tables containing the number of variants are generated. Table 4.9, 4.10 and 4.11 show the results.

Table 4.9 Total number of variants



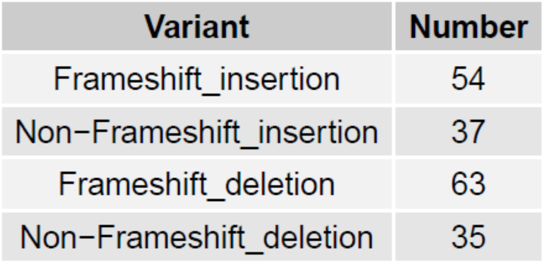
Variant	Number
INDELs	260
SNPs	1673

Table 4.10 Number of Types of Variants



Variant	Number
Frameshift	117
Non−Frameshift	72
sSNPs	559
nsSNPs	882
Stop gain	14
Stop loss	7

Table 4.11 Number of Types of INDELs



Variant	Number
Frameshift_insertion	54
Non−Frameshift_insertion	37
Frameshift_deletion	63
Non−Frameshift_deletion	35

The tool also shows the affect of variant on protein function. This is done by programmatically retrieving the detail report generated by HOPE server about the affect of the amino acid change on the protein structure and function. One of the results retrieved are discussed here. The report contains the information about the reference amino acid and the mutant. It generates the structure of wild type and mutant amino acids shown in Figure 4.8.

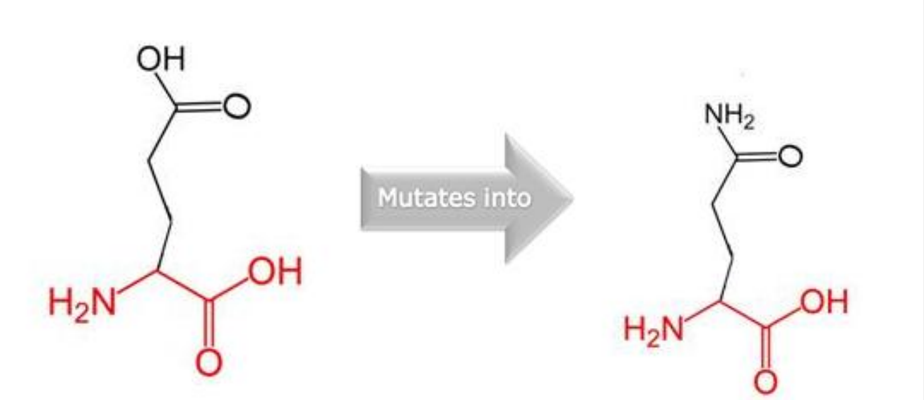


Figure 4.8 Reference and Alternative amino acid structures

Reference amino acid which is glutamic acid mutates into glutamine at position 21. Glutamic acid is negatively charged and glutamine has no charge which means it is neutral.

Conservation

It also provides the information whether the amino acid residue is conserved at this position or not. Here glutamic acid is not conserved at this position. So, on the basis of this observation it concludes that the mutation is not damaging for protein.

Domains

It also provides information about the domain to which the residue belongs. Here glutamic acid is part of an interpro domain named DNA Gyrase, Subunit A. In order to describe the function of the domain it is annotated with the following Gene Ontology (GO) terms: DNA Binding (GO:0003677), ATP Binding (GO:0005524) and DNA topoisomerase Type Ii (ATP hydrolyzing) activity (GO:0003918).

It indicates that domain has following functions:

Table 4.12 Domains Functions

Function	GO term
Nucleic Acid Binding	GO:0003676
Ion Binding	GO:0043167
Nucleotide Binding	GO:0000116
Binding	GO:0005488
Nucleoside Binding	GO:0001822
Hydrolase Activity	GO:0016787
Isomerase Activity	GO:0016853

Mutated residue is the part of domain which is involved in binding of other molecules so the mutated residue may affect the binding.

Amino Acid Properties

Reference residue is negatively charged whereas the mutant residue is neutral. It may cause the loss of interactions between the molecules.

Images of the protein generated are as shown in Figure 4.9, 4.10 (a, b, c).

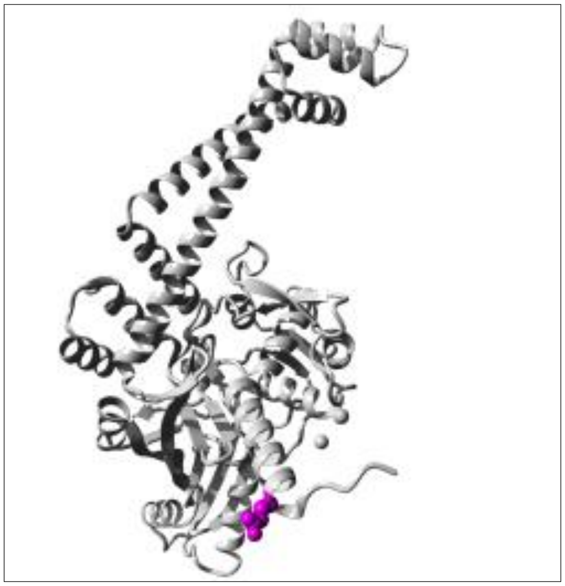


Figure 4.9 Protein Structure

In Figure 4.9 protein is shown in grey color, the mutant side chain is shown as balls in Magenta color.

The close view of the mutation is shown in Figure 4.10 (a, b, c)

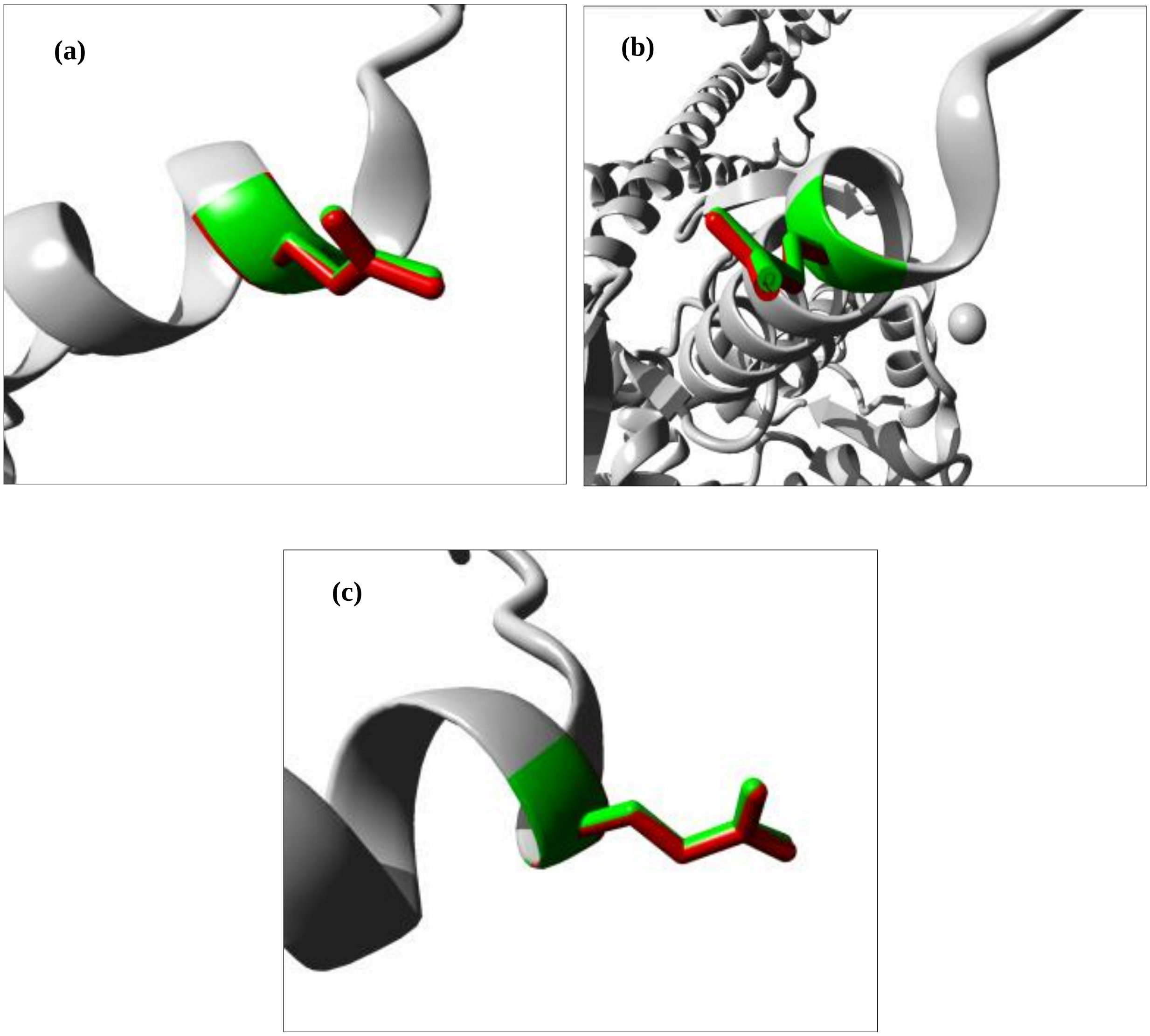


Figure 4.10 (a, b, c) Mutated structure

Figure 4.10 (a, b, c) the close view of mutation is displayed through different angles. Protein is shown in grey color whereas the wild type and mutant residues are shown in red and green color respectively.

4.10 Comparison of Annotation Tools

Table 4.13 Comparison of Annotation Tools

Tool	Obtain annotation sets from user	Platform independent	Unlimited number of organisms	Effect of variants on protein function	SNPs and INDELs Annotation	Visualization
Annovar	✓	✓	✓	✓	✓	**X**
Variant Annotation	**X**	✓	**X**	✓	✓	**X**
SIFT4G	✓	✓	**X**	✓	✓	**X**
Anntools	**X**	**X**	**X**	**X**	✓	**X**
SeqAnt	✓	✓	**X**	✓	✓	**X**
SeattleSeq	**X**	**X**	**X**	✓	✓	**X**
SVA	**X**	**X**	**X**	✓	✓	✓
MU2a	**X**	✓	**X**	✓	**X**	**X**
VASST	✓	**X**	**X**	✓	✓	✓
NGS-SNP	**X**	**X**	✓	✓	**X**	**X**
VAT	✓	**X**	✓	✓	✓	✓
AutoAnnotate	**X**	✓	✓	✓	✓	✓

Table 4.13 shows the comparison of different characteristics of annotation tools. For each characteristic that is present, the column for each tool is marked with a tick symbol and if the tool does not have that characteristic then the column are marked with the cross symbol. The characteristics that are taken into account are: the tool obtain the annotation sets from the user, is this platform independent, if it works for the unlimited number of organisms, does it provide the information about the effect of amino acid change on protein function, is this provides annotation for SNPs and INDELs and does it provide the visualization of the annotation results.

CHAPTER 5

CONCLUSION

CHAPTER 5

5. Conclusion

NGS is different from other methods in the case that it has reduced the cost and has made possible the parallel analysis [16]. The huge amount of data is being produced by the NGS platforms [49]. Different bioinformatics tools are available for analyzing the NGS data. Large numbers of tools are available for annotating the genomic variants identified in the NGS datasets. Each annotation tool has its own features. Some tools require the annotation sets to be provided by the user while some require the programming. People sometimes find difficulty to work with these tools so an automated pipeline can be helpful in that case.

In this thesis implementation of an automated pipeline named Autoanntate is discussed. This pipeline takes only the VCF file as an input from the user. It fetches the GTF file programmatically from the web and using this it annotates the SNPs and INDELs in the dataset. It classified SNPs into Non synonymous (nsSNPs), Synonymous (sSNPs), stop gained and stop loss. It also classifies INDELs as frame shift and non-frameshift. It also provides the information about the affect of amino acid change on protein function by fetching the results programmatically from online tools PROVEAN and HOPE. It also provides the different forms of visualization which include different charts for example bar, pie, petal pie and stacked bar charts. It also provides the number of the different variants present in the dataset in the form of tables. It helps the user to perform annotations of genomic variants in an easy way.

5.1 Limitations of AutoAnnoatate

Autoannotate has certain limitations as well. It takes only VCF file as an input. It uses other packages to perform certain tasks. It is dependent on the online tools like PROVEAN and HOPE for identifying the effect of amino acid change on the protein function.

5.2 Future Perspectives

As this is the first version of AutoAnnotate so it can be improved later on. The improvements that can be incorporated are as follows: It should also provide annotation of large structural variants. The current program is written in a sequential manner; it can be made parallel to improve its efficiency. AutoAnnotate can be linked to other databases for example the databases that provide information about SNPs related with the diseases. GUI for this pipeline can be made to make it more convenient for user to use it. It should also generate information about the pathway of the diseases.

CHAPTER 6

USER GUIDE

CHAPTER 6

6. User Guide

In order to use the package user should have R version 3.3.1 installed on the system. It can be downloaded from the link https://cran.r-project.org/bin/windows/base/. Once R is installed user can use the package AutoAnnotate. User has to set the working directory and place the vcf file in that directory. To set the working directory the command used is: setwd("path of directory"). In order to annotate the variants user should run the script named test.R by the following command: source("test.R"). Once the user will run this command, rest of the work will be done by the script and user will have the annotation results in the directory named Results.

Once the package will be available on Comprehensive R Archive Network *(*CRAN) user can use it as described. User can install the package using command install.packages(“AutoAnnotate”). Once the package will be installed user can annotate the vcf file by setting the working directory. The package will read the vcf file and annotate the variants. Results of annotation will be generated in the directory named Results in the working directory.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Hutchison, Clyde A. "DNA sequencing: bench to bedside and beyond."*Nucleic acids research* 35.18 (2007): 6227-6237.
2. Sanger, Frederick. "Sequences, sequences, and sequences." *Annual review of biochemistry* 57.1 (1988): 1-29.
3. Maxam, Allan M., and Walter Gilbert. "A new method for sequencing DNA." *Proceedings of the National Academy of Sciences* 74.2 (1977): 560-564.
4. Sanger, Frederick, Steven Nicklen, and Alan R. Coulson. "DNA sequencing with chain-terminating inhibitors." *Proceedings of the National Academy of Sciences* 74.12 (1977): 5463-5467.
5. Sanger, F. *et al*. "Nucleotide sequence of bacteriophage phi X174 DNA". *Nature* 265, (1977): 687–695.
6. Swerdlow, Harold, *et al*. "Capillary gel electrophoresis for DNA sequencing: laser-induced fluorescence detection with the sheath flow cuvette." *Journal of Chromatography A* 516.1 (1990): 61-67.
7. Hunkapiller, Tom, *et al*. "Large-scale and automated DNA sequence determination." *Science* 254.5028 (1991): 59-67.
8. Shendure, Jay, and Hanlee Ji. "Next-generation DNA sequencing." *Nature biotechnology* 26.10 (2008): 1135-1145.
9. Smith, Lloyd M., *et al*. "Fluorescence detection in automated DNA sequence analysis." *Nature* 321.6071 (1985): 674-679.
10. Gocayne, Jeannine, *et al*. "Primary structure of rat cardiac beta-adrenergic and muscarinic cholinergic receptors obtained by automated DNA sequence analysis: further evidence for a multigene family." *Proceedings of the National Academy of Sciences* 84.23 (1987): 8296-8300.
11. Anderson, Matthew W., and Iris Schrijver. "Next generation DNA sequencing and the future of genomic medicine." *Genes* 1.1 (2010): 38-69.
12. Lander, Eric S., *et al*. "Initial sequencing and analysis of the human genome." *Nature* 409.6822 (2001): 860-921.
13. Venter, J. Craig, *et al*. "The sequence of the human genome." *science* 291.5507 (2001): 1304-1351.

14. Emrich, Charles A., *et al*. "Microfabricated 384-lane capillary array electrophoresis bioanalyzer for ultrahigh-throughput genetic analysis." *Analytical Chemistry* 74.19 (2002): 5076-5083.
15. Metzker, Michael L. "Sequencing technologies—the next generation." *Nature reviews genetics* 11.1 (2010): 31-46.
16. Liu, Lin, *et al*. "Comparison of next-generation sequencing systems." *BioMed Research International* 2012 (2012).
17. Morozova, Olena, and Marco A. Marra. "Applications of next-generation sequencing technologies in functional genomics." *Genomics* 92.5 (2008): 255-264.
18. http://my454.com/products/technology.asp.
19. http://www.roche-applied-science.com/.
20. Mardis, Elaine R. "The impact of next-generation sequencing technology on genetics." *Trends in genetics* 24.3 (2008): 133-141.
21. Bennett, Simon. "Solexa ltd." *Pharmacogenomics* 5.4 (2004): 433-438.
22. Bennett, Simon T., *et al*. "Toward the $1000 human genome." *Pharmacogenomics* 6.4 (2005): 373-382.
23. Adessi, Céline, *et al*. "Solid phase DNA amplification: characterisation of primer attachment and amplification mechanisms." *Nucleic acids research* 28.20 (2000): e87-e87.
24. Fedurco, Milan, *et al*. "BTA, a novel reagent for DNA attachment on glass and efficient generation of solid-phase amplified DNA colonies." *Nucleic acids research* 34.3 (2006): e22-e22.
25. Shendure, Jay, *et al*. "Accurate multiplex polony sequencing of an evolved bacterial genome." *Science* 309.5741 (2005): 1728-1732.
26. “SOLiD system accuray,” http://www.appliedbiosystems.com/ absite/us/en/home/applications-technologies/solid-next-generation- sequencing.html.
27. *Proceedings of the National Academy of Sciences* 100.7 (2003): 3960-3964.
28. Harris, Timothy D., *et al*. "Single-molecule DNA sequencing of a viral genome." *Science* 320.5872 (2008): 106-109.
29. Pushkarev, Dmitry, Norma F. Neff, and Stephen R. Quake. "Single-molecule sequencing of an individual human genome." *Nature biotechnology* 27.9 (2009): 847-850.

30. Huse, Susan M., *et al*. "Accuracy and quality of massively parallel DNA pyrosequencing." *Genome biology* 8.7 (2007): 1.
31. Wang, Zhong, Mark Gerstein, and Michael Snyder. "RNA-Seq: a revolutionary tool for transcriptomics." *Nature Reviews Genetics* 10.1 (2009): 57-63.
32. Ozsolak, Fatih, *et al*. "Amplification-free digital gene expression profiling from minute cell quantities." *nature methods* 7.8 (2010): 619-621.
33. Pickrell, Joseph K., *et al*. "Understanding mechanisms underlying human gene expression variation with RNA sequencing." *Nature* 464.7289 (2010): 768-772.
34. Rozowsky, Joel, *et al*. "AlleleSeq: analysis of allele-specific expression and binding in a network framework." *Molecular systems biology* 7.1 (2011): 522.
35. Montgomery, Stephen B., *et al*. "Transcriptome genetics using second generation sequencing in a Caucasian population." *Nature* 464.7289 (2010): 773-777.
36. Liu, Jinfeng, *et al*. "Genome and transcriptome sequencing of lung cancers reveal diverse mutational and splicing events." *Genome research* 22.12 (2012): 2315-2327.
37. Bahn, Jae Hoon, *et al*. "Accurate identification of A-to-I RNA editing in human by transcriptome sequencing." *Genome research* 22.1 (2012): 142-150.
38. Ramaswami, Gokul, *et al*. "Identifying RNA editing sites using RNA sequencing data alone." *Nature methods* 10.2 (2013): 128-132.
39. Ramaswami, Gokul, *et al*. "Accurate identification of human Alu and non-Alu RNA editing sites." *Nature methods* 9.6 (2012): 579-581.
40. Xu, Fengping, *et al*. "Impact of Next-Generation Sequencing (NGS) technology on cardiovascular disease research." *Cardiovascular diagnosis and therapy* 2.2 (2012): 138-146.
41. Li, Ruiqiang, *et al*. "The sequence and de novo assembly of the giant panda genome." *Nature* 463.7279 (2010): 311-317.
42. Nagalakshmi, Ugrappa, *et al*. "The transcriptional landscape of the yeast genome defined by RNA sequencing." *Science* 320.5881 (2008): 1344-1349.
43. Guttman, Mitchell, *et al*. "Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs." *Nature biotechnology* 28.5 (2010): 503-510.

44. Trapnell, Cole, *et al*. "Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation." *Nature biotechnology* 28.5 (2010): 511-515.
45. Liti, Gianni, *et al*. "Population genomics of domestic and wild yeasts." *Nature* 458.7236 (2009): 337-341.
46. Li, Yingrui, *et al*. "Resequencing of 200 human exomes identifies an excess of low-frequency non-synonymous coding variants." *Nature genetics* 42.11 (2010): 969-972.
47. Durbin, R. M. *et al*. 1000 Genomes Project Consortium. "A map of human genome variation from population-scale sequencing." *Nature* 467.7319 (2010): 1061-1073.
48. Handel, Adam E., Giulio Disanto, and Sreeram V. Ramagopalan. "Next-generation sequencing in understanding complex neurological disease." (2013): 215-227.
49. Kumar, Santosh, Travis W. Banks, and Sylvie Cloutier. "SNP discovery through next-generation sequencing and its applications." *International journal of plant genomics* 2012 (2012).
50. Moorthie, Sowmiya, Alison Hall, and Caroline F. Wright. "Informatics and clinical genome sequencing: opening the black box." *Genetics in Medicine* 15.3 (2012): 165-171.
51. Oliver, Gavin R., Steven N. Hart, and Eric W. Klee. "Bioinformatics for Clinical Next Generation Sequencing." *Clinical chemistry* 61.1 (2015): 124-135.
52. Su, Zhenqiang, *et al*. "Next-generation sequencing and its applications in molecular diagnostics." (2011): 333-343.
53. Altschul, Stephen F., *et al*. "Basic local alignment search tool." *Journal of molecular biology* 215.3 (1990): 403-410.
54. Kent, W. James. "BLAT—the BLAST-like alignment tool." *Genome research* 12.4 (2002): 656-664.
55. Li, Heng, Jue Ruan, and Richard Durbin. "Mapping short DNA sequencing reads and calling variants using mapping quality scores." *Genome research* 18.11 (2008): 1851-1858.
56. Langmead, Ben, *et al*. "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome." *Genome biology* 10.3 (2009): 1.
57. Trapnell, Cole, Lior Pachter, and Steven L. Salzberg. "TopHat: discovering splice junctions with RNA-Seq." *Bioinformatics* 25.9 (2009): 1105-1111.

58. Voelkerding, Karl V., Shale A. Dames, and Jacob D. Durtschi. "Next-generation sequencing: from basic research to diagnostics." *Clinical chemistry* 55.4 (2009): 641-658.

59. Chaisson, Mark J., and Pavel A. Pevzner. "Short read fragment assembly of bacterial genomes." *Genome research* 18.2 (2008): 324-330.

60. Smith, Michael G., *et al*. "New insights into Acinetobacter baumannii pathogenesis revealed by high-density pyrosequencing and transposon mutagenesis." *Genes & development* 21.5 (2007): 601-614.

61. Pagani, Ioanna, *et al*. "The Genomes OnLine Database (GOLD) v. 4: status of genomic and metagenomic projects and their associated metadata." *Nucleic acids research* 40.D1 (2012): D571-D579.

62. Gonzaga-Jauregui, Claudia, James R. Lupski, and Richard A. Gibbs. "Human genome sequencing in health and disease." *Annual review of medicine* 63 (2012): 35.

63. Johnson, Andrew D. "Single-Nucleotide Polymorphism Bioinformatics A Comprehensive Review of Resources." *Circulation: Cardiovascular Genetics*2.5 (2009): 530-536.

64. Soon, Wendy Weijia, Manoj Hariharan, and Michael P. Snyder. "High-throughput sequencing for biology and medicine." *Molecular systems biology* 9.1 (2013): 640.

65. Yi, Xin, *et al*. "Sequencing of 50 human exomes reveals adaptation to high altitude." *Science* 329.5987 (2010): 75-78.

66. Ng, Sarah B., *et al*. "Exome sequencing identifies MLL2 mutations as a cause of Kabuki syndrome." *Nature genetics* 42.9 (2010): 790-793.

67. Ng, Sarah B., *et al*. "Exome sequencing identifies the cause of a mendelian disorder." *Nature genetics* 42.1 (2010): 30-35.

68. Pabinger, Stephan, *et al*. "A survey of tools for variant analysis of next-generation genome sequencing data." *Briefings in bioinformatics* 15.2 (2014): 256-278.

69. Magi, Alberto, *et al*. "Bioinformatics for next generation sequencing data."*Genes* 1.2 (2010): 294-307.

70. Ng, Pauline C., *et al*. "Genetic variation in an individual human exome." *PLoS Genet* 4.8 (2008): e1000160.

71. Schork, Nicholas J., Daniele Fallin, and Jerry S. Lanchbury. "Single nucleotide polymorphisms and the future of genetic epidemiology." *Clinical genetics* 58.4 (2000): 250-264.
72. Taillon-Miller, P., Gu, Z., Li, Q. *et al*. "Overlapping genomic sequences: A treasure trove of single-nucleotide polymorphisms". Genome Res. (1998): 748–754.
73. http://www.hancockfamilyhistory.co.uk/dna.html
74. Mooney, Sean. "Bioinformatics approaches and resources for single nucleotide polymorphism functional analysis." *Briefings in Bioinformatics* 6.1 (2005): 44-56.
75. Johnson, Andrew D. "Single-Nucleotide Polymorphism Bioinformatics A Comprehensive Review of Resources." *Circulation: Cardiovascular Genetics*2.5 (2009): 530-536.
76. http://www.ncbi.nih.nlm.gov/snp/
77. Boeckmann, Brigitte, *et al*. "The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003." *Nucleic acids research* 31.1 (2003): 365-370.
78. Hamosh, Ada, *et al*. "Online Mendelian inheritance in man (OMIM)." *Human mutation* 15.1 (2000): 57.
79. Fredman, D. G. D. F. M. B. H. A. J., et al. "HGVbase: a curated resource describing human DNA variation and phenotype relationships." *Nucleic acids research* 32.suppl 1 (2004): D516-D519.
80. Stenson, Peter D., *et al*. "Human gene mutation database (HGMD®): 2003 update." *Human mutation* 21.6 (2003): 577-581.
81. Clark, Michael J., *et al*. "Performance comparison of exome DNA sequencing technologies." *Nature biotechnology* 29.10 (2011): 908-914.
82. Danecek, Petr, *et al*. "The variant call format and VCFtools." *Bioinformatics*27.15 (2011): 2156-2158.
83. Li, Heng, Jue Ruan, and Richard Durbin. "Mapping short DNA sequencing reads and calling variants using mapping quality scores." *Genome research* 18.11 (2008): 1851-1858.
84. Langmead, Ben, *et al*. "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome." *Genome biol* 10.3 (2009): R25.

85. Li, Heng, *et al*. "The sequence alignment/map format and SAMtools."*Bioinformatics* 25.16 (2009): 2078-2079.

86. Wei, Zhi, *et al*. "SNVer: a statistical tool for variant calling in analysis of pooled or individual next-generation sequencing data." *Nucleic acids research* 39.19 (2011): e132-e132.

87. Chen, Yunqin, and Jia Wei. "Single Nucleotide Variant Calling Tools for RNA-Seq." *Journal of Medical and Bioengineering Vol* 2.2 (2013).

88. McKenna, Aaron, *et al*. "The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data." *Genome research* 20.9 (2010): 1297-1303.

89. Koboldt, Daniel C., *et al*. "VarScan: variant detection in massively parallel sequencing of individual and pooled samples." *Bioinformatics* 25.17 (2009): 2283-2285.

90. McCarthy, Davis J., *et al*. "Choice of transcripts and software has a large effect on variant annotation." *Genome Medicine* 6.3 (2014): 1.

91. Wang, Kai, Mingyao Li, and Hakon Hakonarson. "ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data."*Nucleic acids research* 38.16 (2010): e164-e164.

92. Obenchain, Valerie, *et al*. "VariantAnnotation: a Bioconductor package for exploration and annotation of genetic variants." *Bioinformatics* 30.14 (2014): 2076-2078.

93. Vaser, Robert, *et al*. "SIFT missense predictions for genomes." *Nature protocols* 11.1 (2016): 1-9.

94. Makarov, Vladimir, *et al*. "AnnTools: a comprehensive and versatile annotation toolkit for genomic variants." *Bioinformatics* 28.5 (2012): 724-725.

95. Shetty, Amol Carl, *et al.* "SeqAnt: a web service to rapidly identify and annotate DNA sequence variations." *BMC bioinformatics* 11.1 (2010): 1.

96. Dolled-Filhart, Marisa P., *et al*. "Computational and bioinformatics frameworks for next-generation whole exome and genome sequencing." *The Scientific World Journal* 2013 (2013).

97. Lyon, Gholson J., and Kai Wang. "Identifying disease mutations in genomic medicine settings: current challenges and how to accelerate progress."*Genome medicine* 4.7 (2012): 1.

98. Ge, Dongliang, *et al*. "SVA: software for annotating and visualizing sequenced human genomes." *Bioinformatics* 27.14 (2011): 1998-2000.

99. Garla, Vijay, *et al*. "MU2A—reconciling the genome and transcriptome to determine the effects of base substitutions." *Bioinformatics* 27.3 (2011): 416-418.

100. Yandell, Mark, *et al*. "A probabilistic disease-gene finder for personal genomes." *Genome research* 21.9 (2011): 1529-1542.

101. Grant, Jason R., *et al*. "In-depth annotation of SNPs arising from resequencing projects using NGS-SNP." *Bioinformatics* 27.16 (2011): 2300-2301.

102. Habegger, Lukas, *et al*. "VAT: a computational framework to functionally annotate variants in personal genomes within a cloud-computing environment." *Bioinformatics* 28.17 (2012): 2267-2269.

103. Ali, Asho, *et al*. "Whole genome sequencing based characterization of extensively drug-resistant Mycobacterium tuberculosis isolates from Pakistan." *PLoS One* 10.2 (2015): e0117771.

104. Altmann, André, *et al*. "A beginners guide to SNP calling from high-throughput DNA-sequencing data." *Human genetics* 131.10 (2012): 1541-1554.

105. Technical Note: Informatics. DNA-Seq data processing. An overview of the NextBio DNA sequencing pipeline.

106. http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

107. Choi, Yongwook, and Agnes P. Chan. "PROVEAN web server: a tool to predict the functional effect of amino acid substitutions and indels."*Bioinformatics* (2015): btv195.