BONAFIDE CERTIFICATE

Certified that this Project report titled “**PATTERN MATCHING IN BIOLOGICAL DATA TO DETECT ABNORMALITIES**” is the bonafide work of “**VINAY KUMAR REDDY K, ARVIND SRINATH K, ELAVARASAN A J”** under my supervision in partial fulfilment for the award of the degree of Bachelor of Engineering in Computer Science and Engineering. Certified further that to the best of my knowledge the work reported here in does not form part or full of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion to this or any other candidate.

Signature	**Signature**
Dr.P.JAYASHREE	Dr.P.JAYASHREE
PROJECT COORDINATOR	**SUPERVISOR**
Associate Professor	Associate Professor
Dept. of Computer Technology	Dept. of Computer Technology
Anna University, MIT Campus	Anna University, MIT Campus

ACKNOWLEDGEMENT

We are highly indebted to our respectable Dean, **Dr A. RAJADURAI** and to our reputable Head of the Department **Dr P.ANANDHAKUMAR**, Department of Computer Technology, MIT, Anna University for providing us with sufficient facilities that contributed to success in this endeavour.

We would like to express our sincere thanks and deep sense of gratitude to our Supervisor, **Dr.P.JAYASHREE** for her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of this phase of project work.

We sincerely thank all the Panel members **Dr. P. Jayashree, Dr. S. Muthuraj Kumar** and **Mr. V. Muthumanikandan** for the valuable suggestions.

We would be failing in our duty, if we forget to thank all the teaching and non-teaching staff of ours department, for their constant support throughout the course of our project work.

Vinay Kumar Reddy K (2014503560)
Arvind Srinath K (2014503507)
Elavarasan A J (2014503514)

TABLE OF CONTENTS

INTRODUCTION

In the past decade though whole genome sequencing is completed to analyse the genome due its vast requirement of memory for storing the genome while processing the genome to exploit the various properties of DNA. Now, with the introduction of Big Data it became easier to handle the date and many algorithms have been proposed for analysing the genome which led to detection that the tandem repeats play an important role in abnormality. If we can detect the abnormalities earlier and accurately it would aid for getting treatment which could save many lives.

DEOXYRIBONUCLEIC ACID (DNA)

Deoxyribonucleic acid is molecule that contains the genetic instructions used in the growth, development, functioning and reproduction of all known living organisms and many viruses. It is made up of two strands called polynucleotides. Since they are made of simple monomers called nucleotides. Nucleotides are mainly made up of four components Adenine (A), Guanine (G), Cytosine (C) and Thymine (T).

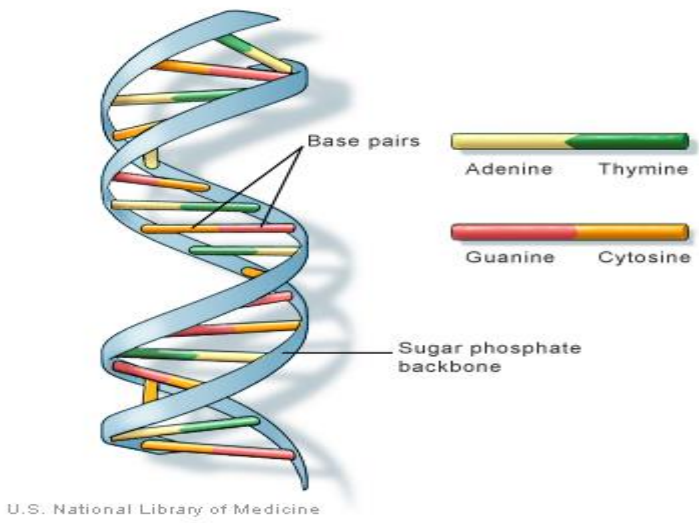


Fig. 1 - Sample DNA Strand

TANDEM REPEATS:

Tandem repeats occur in DNA when one or more nucleotides are repeated and the repetitions are adjacent to each other. In most organisms, the bulk of the intergenic DNA is made up of repeated sequences. Repetitive DNA can be divided into two categories: interspersed repeats, where repeat units are distributed in the genome in an apparently random fashion, and tandem repeats, where repeat units are placed next to each other in an array. The functions of DNA repeats are still largely unclear, but, in recent years, it has been found that a number of tandem repeats are related to diseases and play an important role in gene regulation.
Next-generation sequencing (NGS) machines do not read out an entire genomic template. Instead, they produce short random fragments of nucleotide sequences known as sequence reads. A quality score, expressing the confidence that a particular nucleotide has actually been found at a specific location, is associated with each nucleotide in a sequence read. This raw sequencing data generated by NGS machines is commonly stored in FASTQ files. Recently,

several tools for FASTQ compression have been developed, which – without performing any kind of internal alignment or assembly – are approaching the Kolmogorov complexity. FASTQ data can be aligned with tools such as BWA, SAMtools or Bowtie2. Subsequently, the data is stored in Sequence Alignment/Map (SAM) format files which, compared to FASTQ files, contain the alignment/mapping information as well as additional metadata. Mapped sequencing data represented in SAM files contains more redundancy, as multiple sequence reads are typically mapped to the same location on the donor genome. The average amount of reads mapping to the same location is referred to as coverage.

VARIABLE NUMBER TANDEM REPEATS

Is a location in an genome where a small nucleotide is organised as a tandem repeat. They often show variations in length among each individual.

QUALITY SCORES

The FastQ format provides a simple extension to the FastA format, and stores a simple numeric quality score with each base position. Despite being a “standard” format, FastQ has a number of variants, deriving from different ways of calculating the probability that a base has been called in error, to different ways of encoding that probability in ASCII, using one character per base position.

PHRED scores

Quality scores were originally derived from the PHRED program which was used to read DNA sequence trace files, and linked various sequencing metrics such as peak resolution and shape to known sequence accuracy. The PHRED program assigned quality scores to each base, according to the following formula:

$$Q\_PHRED=-10\log10(pe)$$

where Pe is the probability of erroneously calling a base. PHRED put all of these quality scores into another file called QUAL (which has a header line as in a FastA file, followed by whitespace-separated integers. The lower the integer, the higher the probability that the base has been called incorrectly.

While scores of higher than 50 in raw reads are rare, with post-processing (such as read mapping or assembly), scores of as high as 90 are possible. Quality scores for NGS data are generated in a similar way. Parameters relevant to a particular sequencing chemistry are analyzed for a large empirical data set of known accuracy. The resulting quality score lookup tables are then used to calculate a quality score for de novo next-generation sequencing data.

Solexa scores

The Solexa quality scores, which were used in the earlier Illumina pipelines, are calculated differently from the PHRED scores:

$$Q\text{-}SOLEXA=-10\log10(pe/1-pe)$$

CLASSIFICATION

Instead of processing the genome sequence which is made of millions of base pairs it would be better if it is processed once the repetitive parts are removed which would increase efficiency in both time and cost.

PROBLEM DEFINITION

Though human genome is more than 99 percent common in all humans the current available methods of processing biological data reprocess the complete sequence (on average of 3 million base pairs) each time. Human analysis of the genome is more error prone.

OBJECTIVE

When processing the data it is enough to consider only the parts of genome that has changes when compared to reference genome. Integration of intelligence in order to analyse the genome in order to find any abnormalities.

LITERATURE SURVEY

The authors **"Zhang, Jingsong, Yinglin Wang, Chao Zhang, Yongyong "** described a paper titled **"Mining contiguous sequential generators in biological sequences"**.In this paper they have designed an algorithm to remove the repeated sequences. The advancement in the big data field has led to design of algorithms to process the sequential data but they were not considering the fact that the genome is made up of repetitive segments called tandem repeats. To overcome this situation they have designed an algorithm called ConSgen [1] an effective algorithm for discovering contiguous sequential generators. It adopts the n-gram model, called shingles, to generate potential frequent subsequence's and leverages several pruning techniques unpromising parts of search space. The algorithm works on three-step manner. In the first step, each sequence is split into a series of snippets which preserve the original ordering of items. The length of the snippets is fixed within one complete scan, while it incrementally increases by step length 1 according to n-gram model in subsequent scans. In the next step, by exploring the some properties of contiguous sequential generators, three effective pruning techniques, redundant snippet pruning, max-prefix-suffix pruning and support pruning are proposed. In the third step, all contiguous sequential non-generators are distinguished successively from the set of contiguous sequential patterns by performing the lower-cluster checking. The problem is to check whether there exist a super-pattern absorbing smaller patterns. **Redundant snippet pruning** In real-world datasets, some snippets often appear multiple times, not only among different sequences but also inside one sequence. For each newly split snippet, we check the previous snippets to see whether the new one already exists. The repeated snippets can be easily identified and discarded on-the-fly. **Max-prefix-suffix pruning** Unlike classic candidate enumerate-and-test paradigm, max-prefix-suffix pruning, for a new length-k snippet s, does not need to check all its subsequences if there exist an infrequent one. Instead, it checks only the frequency of the max-prefix-suffixes of s. The length-k snippet is pruned immediately if its max-prefix or max-suffix is infrequent. **Support pruning** A snippet is called a promising candidate if it meets: (1) it is distinct from the previous snippets (already split snippets); and (2) both the max-prefix and the max-suffix of the snippet are frequent. Such snippets satisfying the above two conditions can be shifted to count their supports and check whether they are frequent.

The authors **“Zhou, Hongxia, Liping Du, Hong Yan”** described a paper titled **"Detection of tandem repeats in DNA sequences based on parametric spectral estimation".** In this paper they explain an algorithm using a spectral method. Tandem repeats [2], which occur frequently in genomes, are related to the gene regulatory function and various diseases. Since a DNA sequence consists of ordered nucleotides represented as letters A, C, T, and G, string matching methods including those based on the Hamming or edit distance, data compression, and color coding, are useful for tandem repeat analysis. The algorithms that have been proposed used Hamming distance and edit distance. However, the main limitation of these algorithms is excessive running time since a genome sequence can easily consist of millions of neucleotides. In order to reduce the run-time, heuristic algorithms have been proposed however these algorithms are not sufficient by themselves because they require a period of the repeat or the basic pattern of the region to be specified in advance. To overcome this problem a computer program known as tandem repeat finder (TRF) has been proposed which works based on probabilistic model. It consists of a screening phase and a verification phase. Algorithms, other than these measuring the sequence edit distance, may also be used. If the four letters A, C, T, and G, in a DNA sequence are assigned numerical values, then signal processing operations can be used for tandem repeats. In the tetrahedral representation, each of the four letters is assigned to a vertex of a regular tetrahedron in space. An optimal criteria like use of a weight vector to obtain the sequence can also be used. Numerical representation of DNA sequences opens up the possibility of applying signal processing techniques to the analysis of genomic data. Recently, signal processing methods have also been used for DNA repeat identification. These techniques include the discrete Fourier transformation (DFT), the short time periodicity transfer (STPT), and the exact periodic subspace decomposition (ESPD). A computer program called spectral repeat finder is also developed for this purpose. An efficient method for tandem repeat detection using spectrogram of a DNA sequence is analysed based on the autoregressive model. Most signal -processing-based methods use the FT in spectral analysis for tandem repeats detection since they are able to process a large amount of data quickly. However, the FT is well known to produce data truncation articrafts and poor resolution for spectral analysis. To overcome this problem a repeat detection method based on autoregressive (AR) model. It can offer a high resolution since it has the capability of extrapolating the autocorrelation function of the input.

The authors **“Deng, Yue, Zhiquan Ren, Youyong Kong, Feng Bao, Qionghai Dai”** described a paper titled **"A hierarchical fused fuzzy deep neural network for data classification".** In this paper they have designed a DNN for learning deep features of DNA. Over the past few years there is so much growth in the field of bioinformatics with the completion of genome sequencing. As the analysis of genome is unveiling various abnormalities. deep learning method has been widely used in many fields of science researches. Many different deep learning models are applied for different applications, e.g., deep neural networks (DNN)[3] and convolutional neural network (CNN) . DNN has a classical auto encode (AE) structure which can be used to extract deep feature automatically. An AE is composed with several stacked RBM, The RBM is an undirected graphical model that defines the distribution of visible units using binary hidden units. In training process, each RBM can be trained independently and the output of the bottom RBM can be used as the input data of the upper RBM. After getting all the parameters for each layer, they can get a DNN to carry out feature extraction and dimension reduction. Because the input unit is binary, the input data sent to the input unit should be bounded from 0 to 1. This peculiarity just adapts the characteristic of DNA methylation data. The output getting from equation 4 are also bounded, which makes the features finally extracted from the top layer are still in accordance with the characteristics of methylation data. In order to visually assess the effect of feature selection, they adopted the distributed stochastic neighbour embedding (t-SNE)[3] method to visualize the high-dimensional data in two-dimensional space. t-SNE is a nonlinear dimensionality reduction technique which has been used in a wide range of applications, including computer security research, music analysis, cancer research, and bioinformatics. It is particularly well suited for embedding high-dimensional data into a space of two or three dimensions, which can then be visualized by a scatter plot. Especially, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modelled by nearby points and dissimilar objects are modelled by distant points. In order to make fair comparisons, they also applied the k-means, the Gaussian mixture model (GMM), and the SOM method to cluster the features extracted by PCA, NMF, and DNN methods, respectively.

The authors **“Gosztolya, Gábor, László Tóth”** described a paper titled **"DNN-based Feature Extraction for Conflict Intensity Estimation from Speech"**. In this paper they have designed a DNN for Speech recognition. The standard computational approach, developed over the years on various paralinguistic tasks[4], is to extract several thousand general, utterance-level features from the speech excerpts, and use these to train general machine learning methods such as Support-Vector Machines (SVM) or Deep Neural Networks (DNNs) to perform classification or regression. In the DNN each neuron corresponds to a particular person. They have proposed to use only two classes, corresponding to a zero-or-one speaker, and a two-or-more speaker case. Next, they extracted features from the frame-level DNN outputs, and these features will be used for a specific time window instead of the whole utterance. Although using the amount of speaker overlap may prove to be beneficial for conflict intensity estimation, we should not discard all other features.

The authors **“Ramakrishnan, Nithya, Ranjan Bose”** described a paper titled **"Analysis of healthy and tumour DNA methylation distributions in kidney-renal-clear-cell-carcinoma using Kullback–Leibler and Jensen–Shannon distance measures".** They have praposed an algorithm for detecting kidney related cancer using kullback-leibler and jensen-shannon distance measurement. DNA methylation [5] ia an epigenetic phenomenon in which methyl groups gets bounded to the 5-carbon of the cytosines ring to result in 5-methyl cytosine (5-mC) of the DNA molecule altering the expression of the associated genes. Abnormalities like cancer is linked with hypo or hyper-methylation of specific genes as global changes in DNA methylation. In human somatic cells, 5-mC occurs in CpG sites and islands. A CpG site in a location within a DNA sequence in which a cytosine and guanine appear consecutively. When a CpG island island in the promoter region of a gene is methylated, the gene expression is switched off. DNA methylation patterns are either inherited (maintenance) or developed newly (de-novo methylation). Irregular changes in DNA methylation have been linked with many abnormalities. Based on the kullback-leibler and jensen-shannon distance measurement a popular abnormality kidney-renal-clear-cell-carcinoma is detected.

The authors **“Voges, Jan, Marco Munderloh, Jörn Ostermann”** described a paper titled **"Predictive coding of aligned next-generation sequencing data".** They have designed an algorithm for high throughput next-generation sequencing. Due to novel high-throughput next-generation sequencing technologies, the sequencing of huge amounts of genetic information has become affordable. On account of this flood of data, IT costs have become a major obstacle compared to sequencing costs. High-performance compression of genomic data is required to reduce the storage size and transmission costs. The high coverage inherent in next-generation sequencing technologies produces highly redundant data. Due the above reason there is a need for the design of a compression algorithm for aligned sequence reads. They have proposed an algorithm TCS[6]. The algorithm is solely requires SAM files which contain regions that are sorted by their mapping positions. The algorithm is based on the unwinding of the sequence reads by using the mapping positions and the additional alignment information provided by the CIGAR strings. The process of expanding the nucleotide sequence yields a modified CIGAR string, which they call stogy, and the expanded nucleotide sequence, called exs. We call this mapped file as tcs file. After encoding the first read, the mapping position pos and the expanded read exe are pushed to a circular buffer which implements the sliding window. The tsc records are a new sequence read representation are ultimately passed to a dictionary-based code in order to exploit the redundancy along the record stream. Specifically, error-free and properly aligned sequence reads produce a tsc record stream which is highly suitable for dictionary-based compression because of repetitions in the data stream and the compression is based on the DEFLATE algorithm which uses Huffman coding and LZ77 algorithm.

The authors **“Deng, Yue, Zhiquan Ren, Youyong Kong, Feng Bao, Qionghai Dai”** described a paper titled **"A hierarchical fused fuzzy deep neural network for data classification".** In this paper they have described a fuzzy logic and deep learning to extract more information and to provide more accurate results. Typical deep learning is a fully deterministic model that sheds no light on data uncertainty reductions. In this paper, they show how to introduce the concepts of fuzzy learning[7] into DL to overcome the shortcomings of fixed representation. The bulk of the proposed fuzzy system is a hierarchical deep neural network that derives information from both fuzzy and neural representations. Then, the knowledge learned from these two respective views are fused altogether forming the final data representation to be classified.

The authors **“Azim, Md Aashikur Rahman, Costas S. Iliopoulos, M. Sohel Rahman, M. Samiruzzaman”** described a paper titled **"A Simple, Fast, Filter-Based Algorithm for Approximate Circular Pattern Matching".** In this paper they have explained an algorithm forn circular pattern matching. They have also explained other algorithms for this purpose and their advantaged and disadvantages. String matching or pattern matching problem is a classical problem in computer science with extensive applications in different branches of science and engineering. This problem is interesting as a fundamental computer science problem and is a basic requirement of many practical applications. In this paper, they present a fast and efficient algorithm for the approximate circular pattern matching problem based on some filtering techniques. They presented SimpLiFiCPM that is a filter-based algorithm to handle the exact version of the circular pattern matching problem. Given a pattern P of length m and a text T of length n, both being sequences of characters drawn from a finite character set , the goal of this problem is to find all the occurrences of P in T. They use SFF (Simple Fast Filter-based Algorithm) [8] to match the sequence.

The authors **"Pérez, Sandino Vargas, Fahad Saeed"** described a paper titled "**A parallel algorithm for compression of big next-generation sequencing datasets".** In this paper they have described a method for compressing the huge amount of DNA sequence. With the advent of high-throughput next-generation sequencing (NGS) [9] techniques, the amount of data being generated represents challenges including storage, analysis and transport of huge datasets. One solution to storage and transmission of data is compression using specialized compression algorithms. They have designed an parallel processing algorithm for compressing the DNA files. However, these specialized algorithms suffer from poor scalability with increasing size of the datasets and best available solutions can take hours to compress Gigabytes of data. In this paper we introduce paraDSRC, a parallel implementation of DSRC using a message passing model that presents reduction of the compression time complexity by a factor of O(1/p). Our experimental results show that paraDSRC achieves compression times that are 43% to 99% faster than DSRC and compression throughputs of up to 8.4GB/s on a moderate size cluster. For many of the datasets used in our experiments super-linear speedups have been registered, making the implementation strongly scalable. They also show that paraDSRC is more than 25.6x faster than comparable parallel compression algorithms.

The authors **"Ku, Chee Seng, En Yun Loy, Agus Salim, Yudi Pawitan, Kee Seng Chia"** described an article titled **"The discovery of human genetic variations and their use as disease markers: past, present and future"**. In this article they have explained the genetic markers that were used in the past, in the present and in future. The field of human genetic variations has progressed rapidly over the past few years. It has added much information and deepened our knowledge and understanding of the diversity of genetic variations in the human genome. This significant progress has been driven mainly by the developments of microarray and next generation sequencing technologies. The array-based methods have been widely used for large-scale copy number variation (CNV) [10] detection in the human genome. The arrival of next generation sequencing technologies, which enabled the completion of several whole genome resequencing studies, has also resulted in a massive discovery of genetic variations. These studies have identified several hundred thousand short indels and a total of thousands of CNVs and other structural variations in the human genome. The discovery of these 'newer' types of genetic variations, indels, CNVs and copy neutral variations (inversions and translocations) has also widened the scope of genetic markers in human genetic and disease gene mapping studies. In this article they have summarize the latest developments in the discovery of human genetic variations and address the issue of inadequate coverage of genetic variations in the current genome-wide association studies, which mainly focuses on common SNPs.

PROPOSED WORK

1. Design and Implementation of an algorithm to remove the tandem repeats that make the most part of the genome which in turn reduces the memory and time while analysing the genome for abnormalities.
2. To find the count of tandem repeats in a genome by using an open source tool.
3. Design and Implementation of an algorithm to detect whether there are abnormalities.
4. Design and Implementation of a Deep Neural Network(DNN) to find the type of abnormality and the stage of abnormality (if any).

TOOLS USED

- **Galaxy**
- **RepeatExplorer**

Introduction

Galaxy is a web based analysis and workflow platform designed for biologists to analyse their own data. It comes with most of the popular bioinformatics tools already installed and ready for use. There are many Galaxy servers around the world and some are tailored with specific toolsets and reference data for analysis of human genomics, microbial genomics, proteomics etc.

Upload Data into Galaxy

There are two main ways to get your data into Galaxy

1. **Upload a file from your own computer**
   - From the Galaxy tool panel, click on **Get Data -> Upload File**
   - Click the Choose File button
   - Find and select the *Contig_stats.txt.gz* file you downloaded and click **Open**
   - **Set the "file format"** to tabular
   - Click the **Start** button
   - Once the progress bar reaches 100%, click the **Close** button
2. **Upload a file from a URL**
   - From the Galaxy tool panel, click on **Get Data -> Upload File**
   - Enter the URL into the Text box
   - Set the file format to fastqsanger
   - Click **Start**
   - Once the progress bar has reached 100%, click **Close**
3. **Convert Fastq file to sangerFastq sequences**
   - Goto the **TOOLS** panel
   - Select **NGS:QC and Manipulation**

- Select **Fastq groomer,** Select input file and click **Execute.**

4. **Quality Check**

   - Goto the **TOOLS** panel
   - Select **NGS:QC and Manipulation**
   - Select **FastQC,** Select input file and click **Execute.**

5. **Quality Filter**

   - Go to **NGS:QC and Manipulation**
   - Select **Filter FastQ,** Select input file and click **Execute**

6. **Mapping/Short Read Alignment**

   - Go to **NGS:Mapping**
   - Select **Bowtie2**
     - Select weather the data is single or paired end
     - Select the filtered fastq file
     - Click **yes** on **write unaligned reads (in fastq format) to separate file(s)** to separate unaligned reads to separate file
     - Select **Build-in genome index** option and select corresponding reference genome name.
     - Specify the read group for the file by clicking on yes and specify sample name, library, readgroup id, and platform .
     - Select analysis mode as default settings.
   - Click on Execute

7. **SAM/BAM File Manipulation**

   - Filter **SAM/BAM** file using **SAMTools**
     - Go to **NGS:SAMTools**
     - Select **Filter SAM/BAM**
     - Select input file and click **Execute.**
   - Mark duplicates using picard
     - Go to **NGS:picard**
     - Select **Mark Duplicates**
     - Select file from history, and click **yes** in **do not write duplicates to the output file** option
     - Click on **Execute.**
   - Remove PCR duplicates from SAM/BAM file

- Go to **NGS:SAMTools**
- Select **RmDup**
- Select BAM dataset and specify whether data is single or paired end
- Click on **Execute.**

8. VCF File Creation

- Go to **SAMTools**
- Select **MPileup**
- Select Reference genome
- Select BAM file.
- If Perform **Genotype Likelihood Computation** is selected, output will be in BCF/VCF. If do not perform **Genotype Likelihood Computation** is selected, output will be in pileup format.
- If we select **Perform INDEL calling** or **do not Perform INDEL calling** , we can perform INDEL calling and SNP calling together or SNP calling alone respectively.

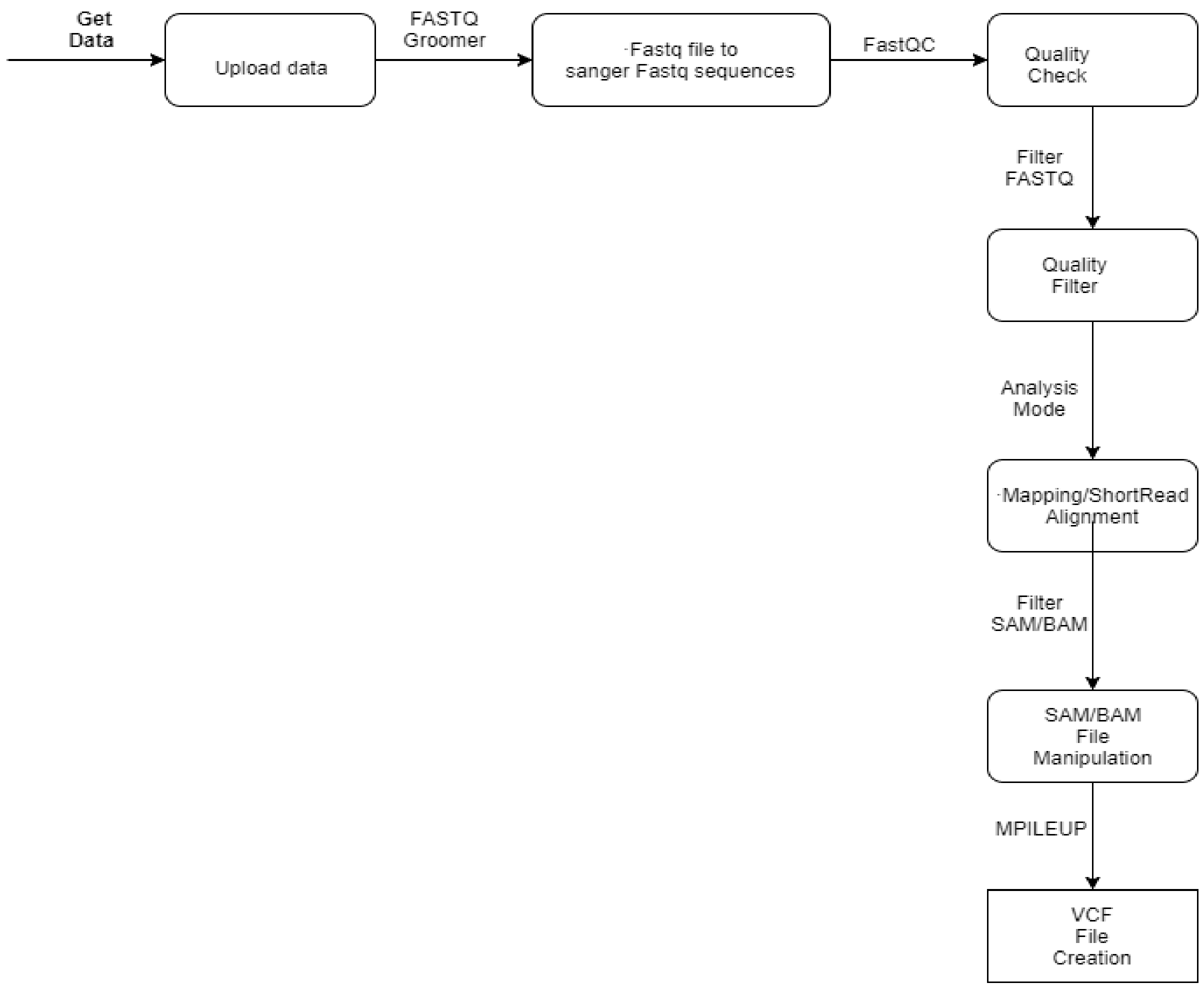


Fig. 2 – Flow diagram of removing repeating patterns

RepeatExplorer

RepeatExplorer is a computational pipeline for discovery and characterization of repetitive sequences in eukaryotic genomes. The pipeline uses high-throughput genome sequencing data as an input and performs graph-based clustering analysis of sequence read similarities to identify repetitive elements within analyzed samples. It should be noted that although the repeat identification algorithm generally works for any genome, some parts of the pipeline (e.g. protein domain-based classification of mobile elements) were primarily developed for application to plant genomics. However, there is a possibility to supply a custom repeat database to improve sensitivity in classification of non-plant repeats.

Steps

- **Getting your data to server**
  - ➢ Datasets can be downloaded directly from the EBI Short Read Archive using **Get Data → EBI SRA** tool. Enter the ENA accession number in the search window, locate the corresponding dataset and select download link in the "Galaxy" column
- **Pre-processing of sequence reads**
  - ➢ The clustering analysis requires a single file containing read sequences in FASTA format as an input.
  - ➢ If such a file can be uploaded by the user, no pre-processing is required
  - ➢ However, data obtained from sequencing facilities or downloaded from public archives are usually in FASTQ format combining nucleotide sequence information with sequencing quality scores.
  - ➢ There is a number of programs for analysing and pre-processing raw sequence reads in **Tools → NGS: QC and manipulation → FastQC:Read QC**
- **Grooming the data**
  - ➢ The FASTQ illumine format must be converted to FASTQ sanger format using **Tools → NGS: QC and manipulation → FASTQ Groomer**
- **Trim the data**
  - ➢ If there is any discrepancy in the sequence then the sequence should be trimmed using **Tools → NGS: QC and manipulation → FASTQ Trimmer**
- **FASTQ to FASTA**
  - ➢ The FASTQ format is converted into FASTA using **Tools → NGS: QC and manipulation → FastQC:Read QC → FASTQ to FASTA converter**
- **TAREAN**
  - ➢ Tandem repeat analyser (TAREAN), a computational pipeline which was built on the principles of graph-based repeat clustering, enhanced and supplemented with additional tools facilitating unsupervised identification and characterization of satellite repeats from unassembled sequence reads
  - ➢ The pre-processed FASTA file is passed into **Tools → TAREAN → Tandem Repeat Analyser**

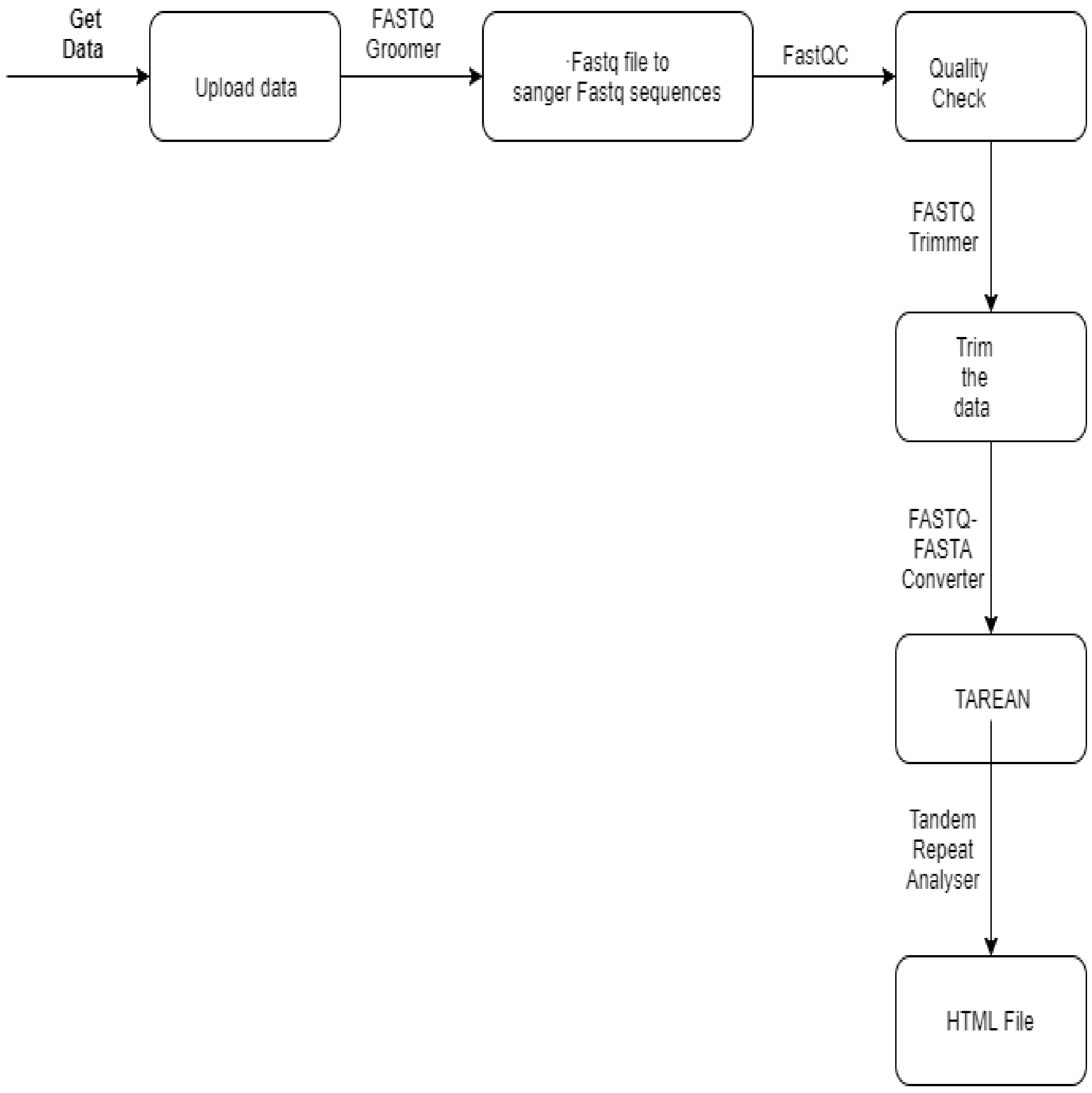


Fig. 3 – Flow diagram of tandem repeat analysing

PROGRAMING LANGUAGE USED

Java

JAVA

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems(which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them. Excel is the very popular file format created by Microsoft. Although it is not an opened file format, Java applications can still read and write Excel files using the **Apache POI - the Java API for Microsoft Documents**, because the development team uses reverse-engineering to understand the Excel file format. Hence the name POI stands for Poor Obfuscation Implementation.

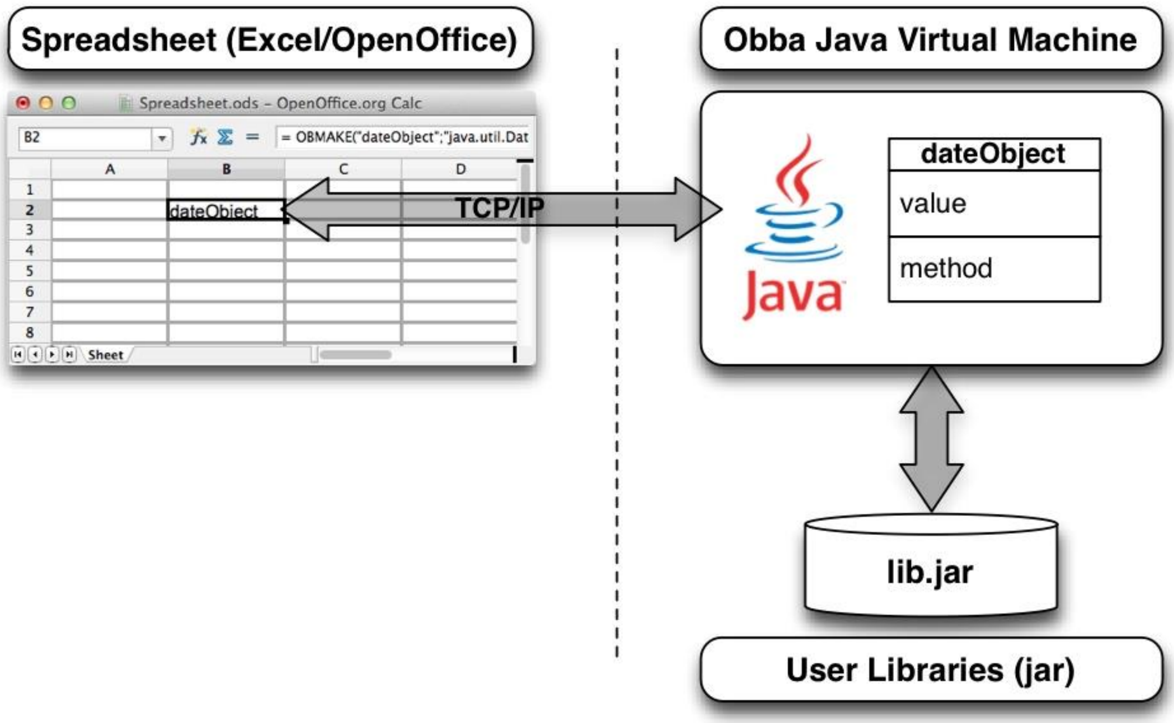

Spreadsheet (Excel/OpenOffice)
Spreadsheet.ods - OpenOffice.org Calc
B2
= OBMAKE("dateObject";"java.util.Dat
A
B
C
D
dateObject
Sheet
TCP/IP
Obba Java Virtual Machine
Java
dateObject
value
method
lib.jar
User Libraries (jar)

Fig. 4 – Architecture of Java integration with Excel Sheet

ARCHITECTURE DIAGRAM

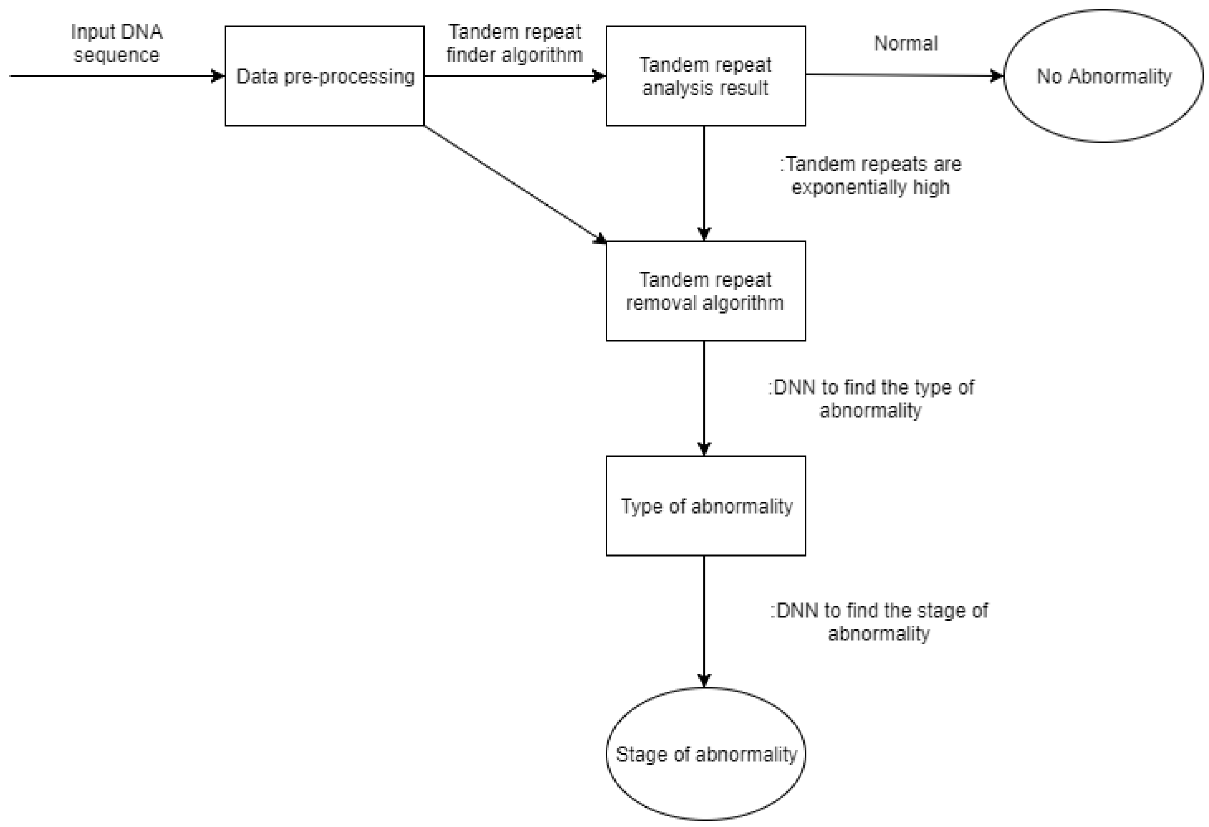

Input DNA sequence
Data pre-processing
Tandem repeat finder algorithm
Tandem repeat analysis result
Normal
No Abnormality
:Tandem repeats are exponentially high
Tandem repeat removal algorithm
:DNN to find the type of abnormality
Type of abnormality
:DNN to find the stage of abnormality
Stage of abnormality

Fig. 5 – Architecture diagram of the proposed project

ALGORITHM

Abnormality Type Detection algorithm

Step 1: Extract the features from the vcf file.

Step 2: For each record in the vcf file read the different parameters.

Step 3: Once the parameters are extracted based on the values of the different parameters the records are compared with the reference records of the human genome (Since we are trying to detect the abnormalities in human genome).

Step 4: For each record assign the reference gene.

Step 5: Write the record into an flat file.

Step 6: Open the flat file.

Step 7: For each record in the flat file read the gene.

Step 8: Compare it with the reference Genes of abnormalities which is stored in another flat file.

Step 9: If any matching is found then display the abnormality.

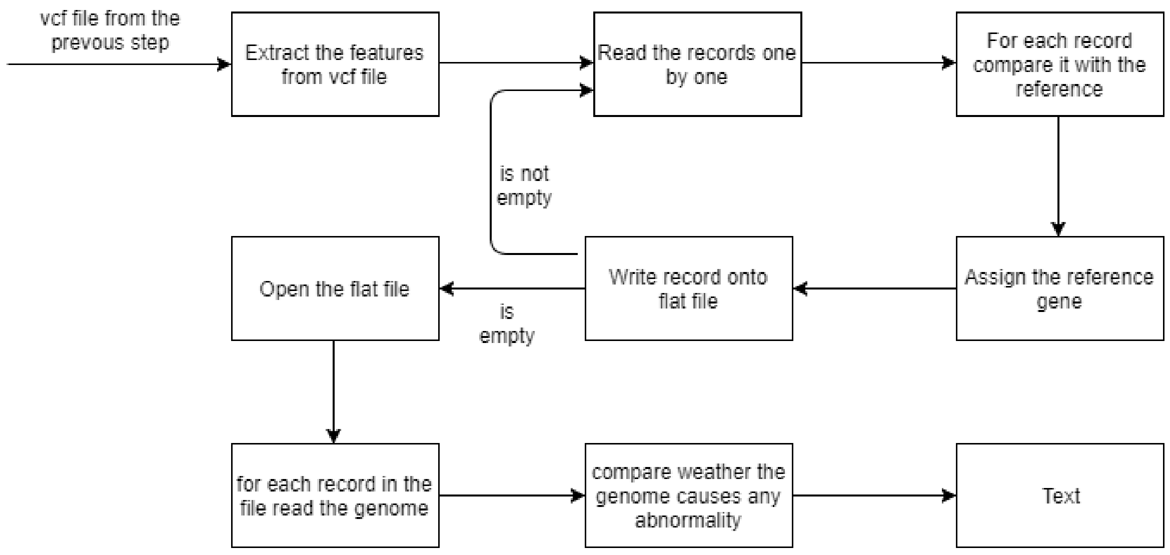

vcf file from the prevous step
Extract the features from vcf file
Read the records one by one
For each record compare it with the reference
is not empty
Open the flat file
is empty
Write record onto flat file
Assign the reference gene
for each record in the file read the genome
compare weather the genome causes any abnormality
Text

Fig 6 – Flow diagram of abnormality type detection algorithm

IMPLEMENTATION

The input is a DNA sequence which is generated based on one of the standard method of DNA sequencing. This file contains a lot of unnecessary data that need to be removed for this purpose the DNA sequence need to be filtered before doing any processing. By using the galaxy tool this can be done. Once the useful sequence is checked for quality and the base pairs that have a low quality are discarded which leads to better accuracy.

Quality Score is assigned to each base pair using any one of the Scoring methods as follows.

Quality Score Q_PHRED=-10log10(pe)
Or
Q-SOLEXA=-10log10(pe/1-pe)

Then the Sequence is aligned according to Bowtie2 algorithm. Galaxy an online tool is used to perform the quality check and bowtie2. The steps for processing is shown in tools used.

The sequence is then sorted, marking and removing of duplicates are performed subsequently. In order to remove the tandem repeats by which unnecessary processing of data is eliminated.

Samtools is used to perform the sorting, removing duplicated and piling up the input sequence with the reference genome by using the following commands.

```
samtools sort filename.bam
samtools rmdup filename.bam
```

Once the data has got rid of repeats it is compared with the reference genome by using the following command.

```
samtools mpileup input.bam output.bcf
```

This file needs to be converted to vcf format because the bcf format is compressed for this purpose another tool called bcftools is used. The command for this is

```
bcftools view -c -v -g input.bcf ouput.vcf
```

On the other hand we need to check for tandem repeats for the basic confirmation of any abnormality. For this purpose we use the Galaxy Repeat Explorer. By analysing the result we know whether there is an exponential growth in tandem repeats.

The vcf file is then compared with the reference human data of chromosomes inorder to know the gene expression based on the features of vcf file. Now this reference Gene is compared with that of abnormal genes so that we can find what kind of abnormality it is.

MODULES

Phase - I

1. Study on DNA properties and their use in bioinformatics
2. Data Pre-processing and deletion of tandem repeats.
3. Detection of abnormality.

Phase - II

1. Study on the design and training of DNN’s.
2. Design of Deap Neural Network (DNNs) to find the type and stage of abnormality.

RESULTS AND ANALYSIS

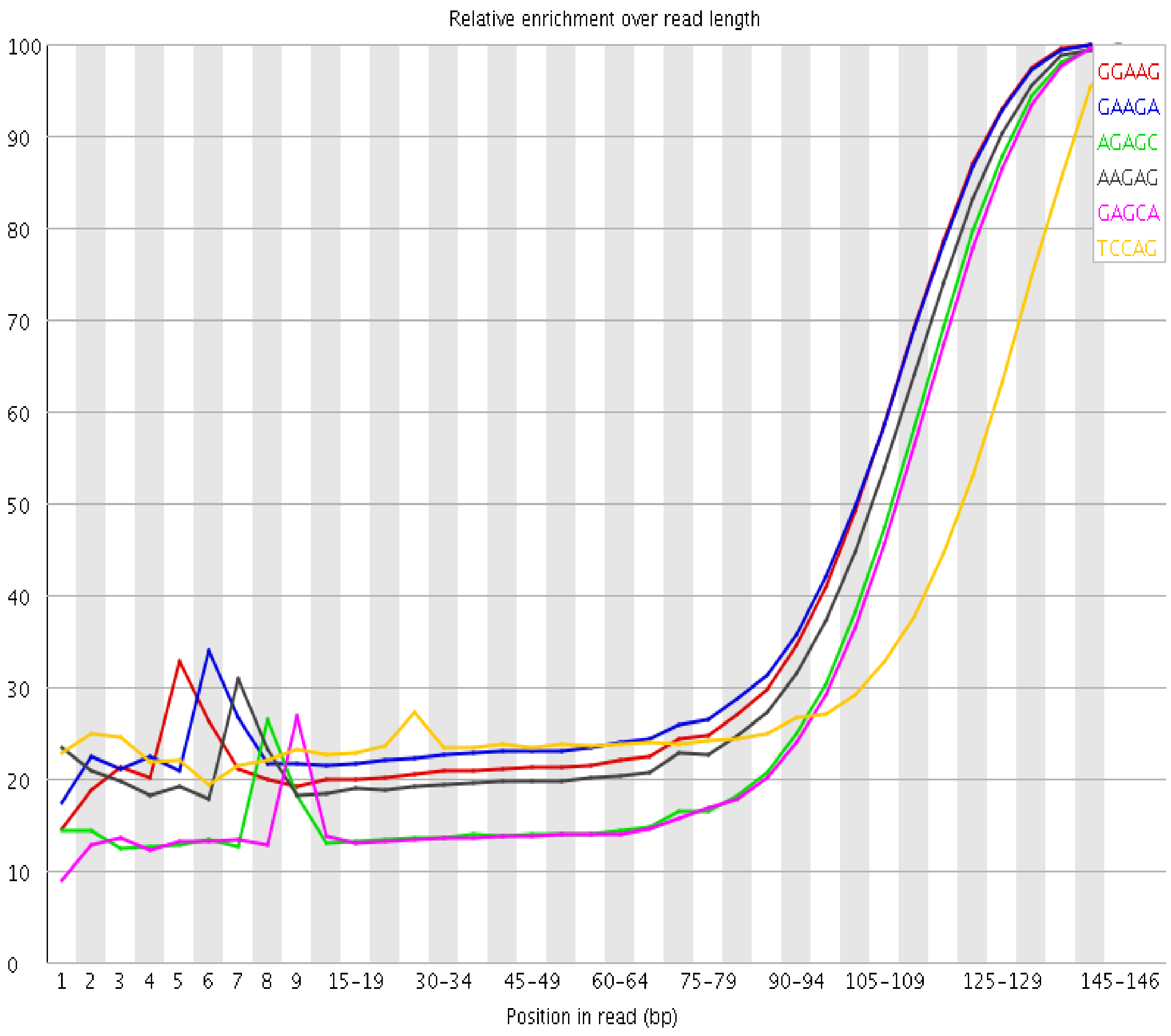


Fig. 7 – K-Mer of the input data

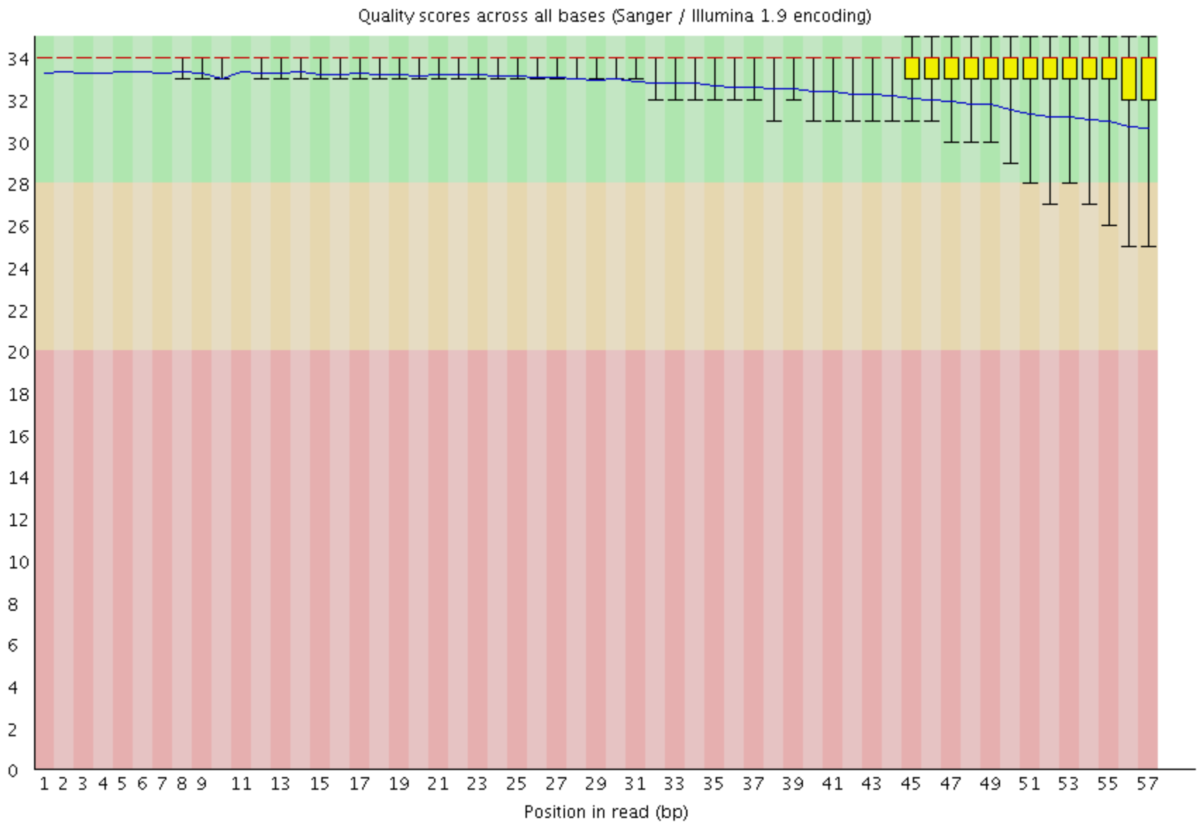


Fig. 8 – Per base quality of input

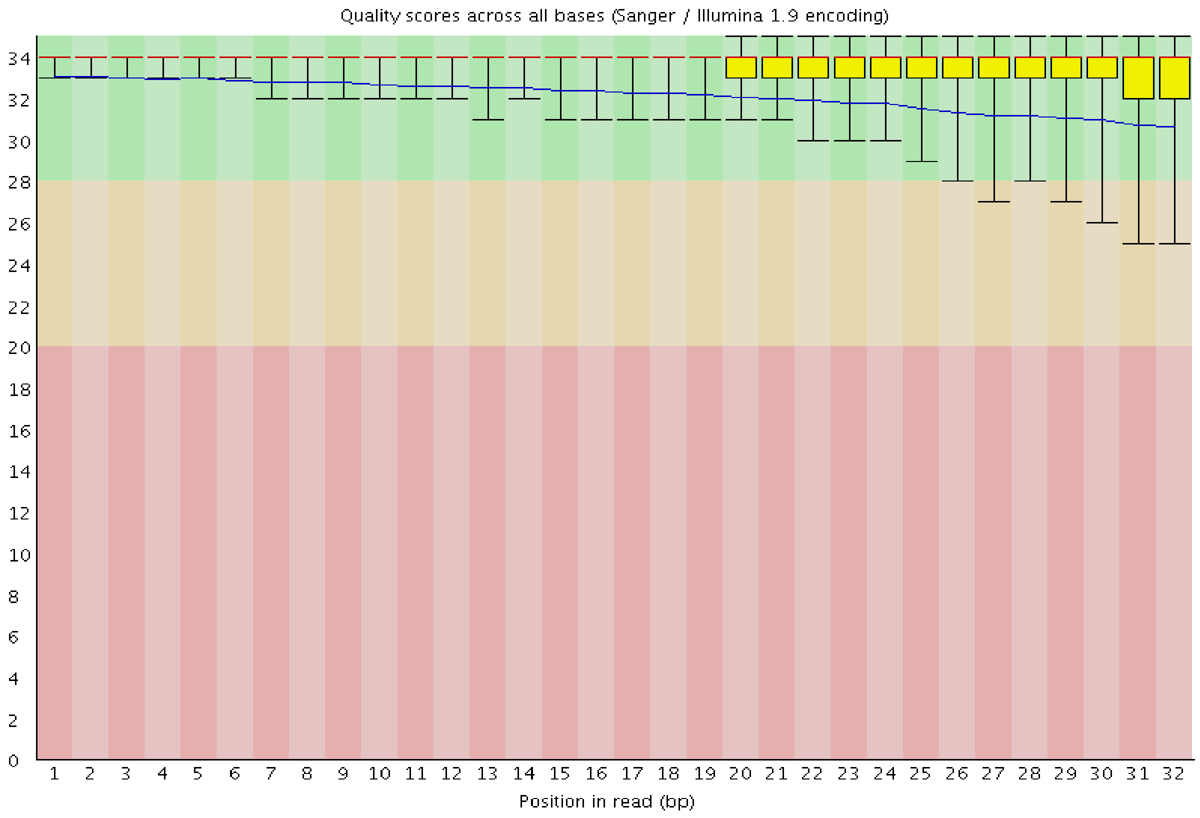


Fig. 9 – per base quality after pre-processing

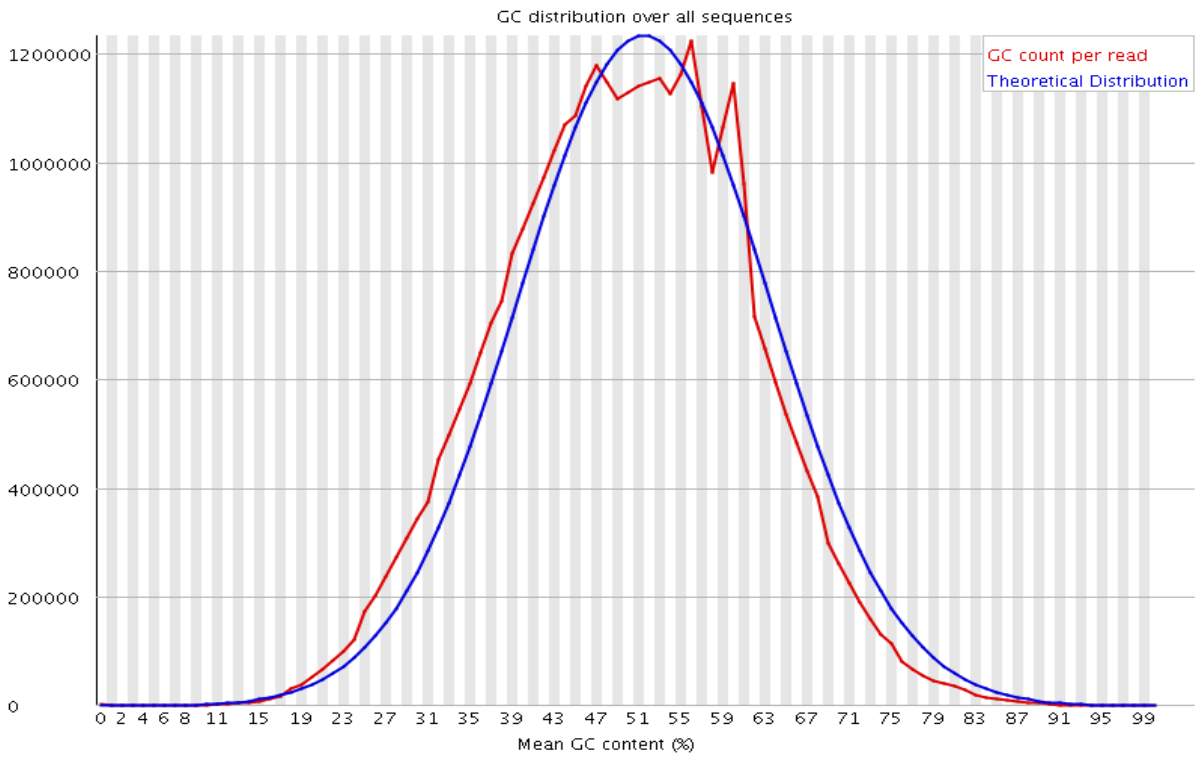


Fig. 10 – per sequence gc count of input

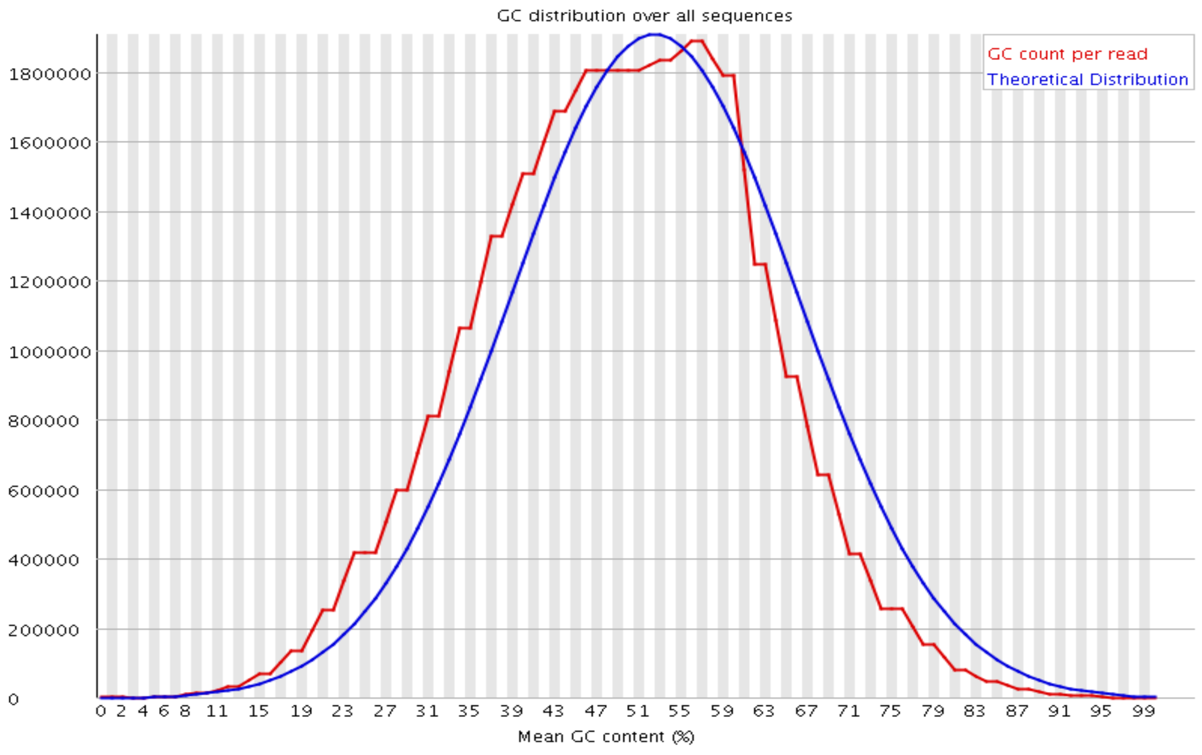


Fig. 11 – Per sequence gc count after pre processing

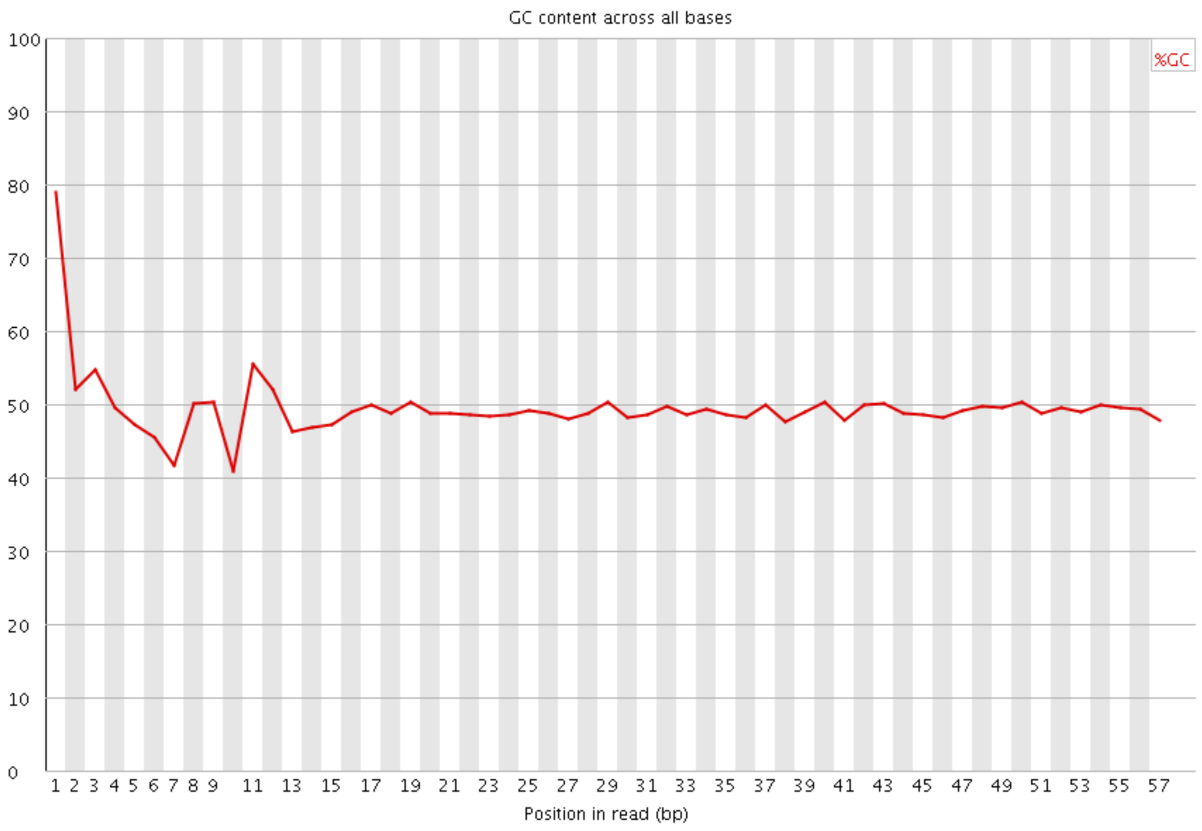


Fig. 12 – per base gc count of input data

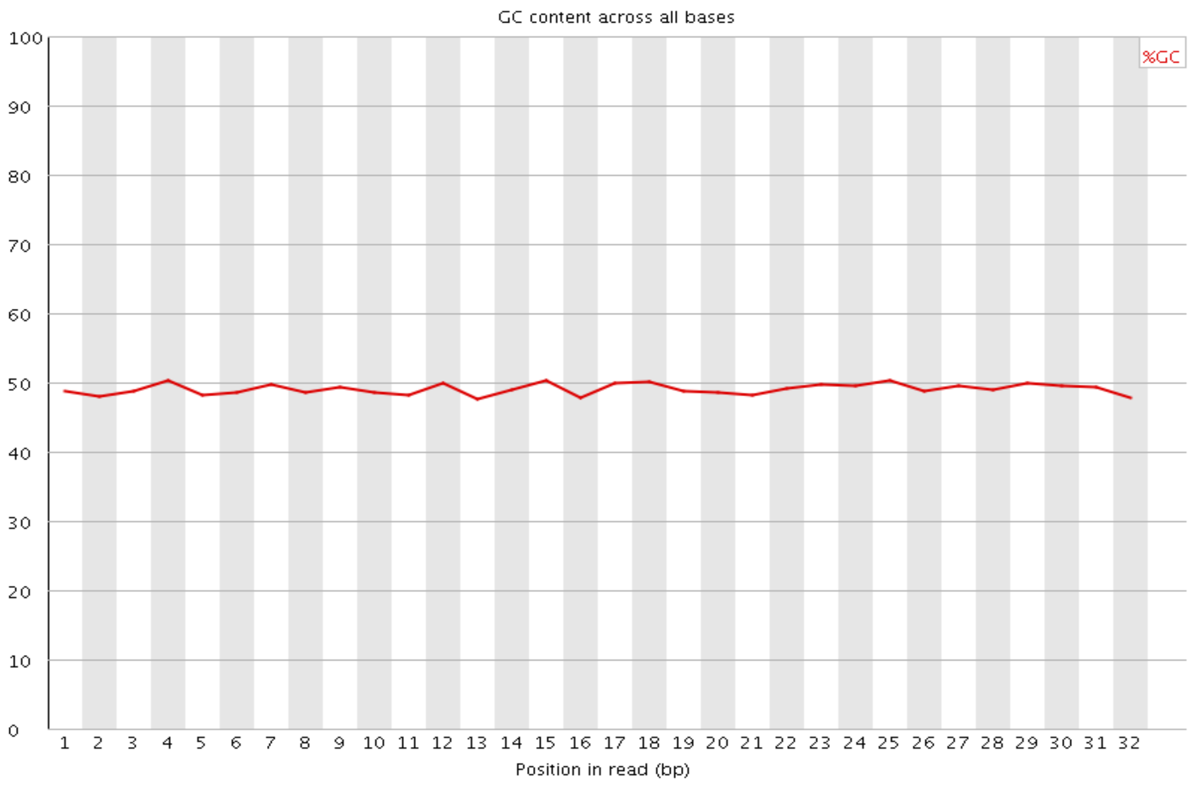
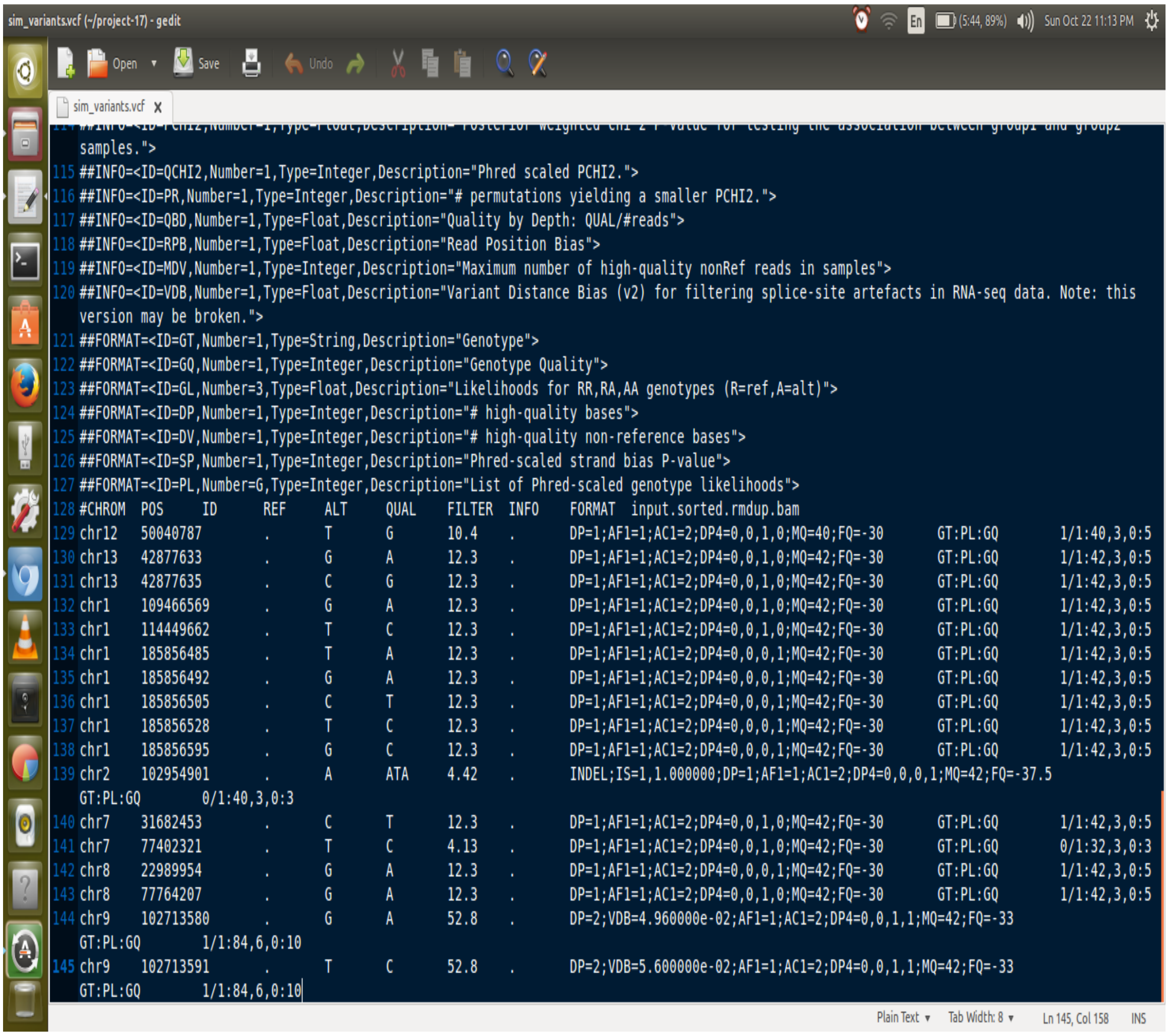
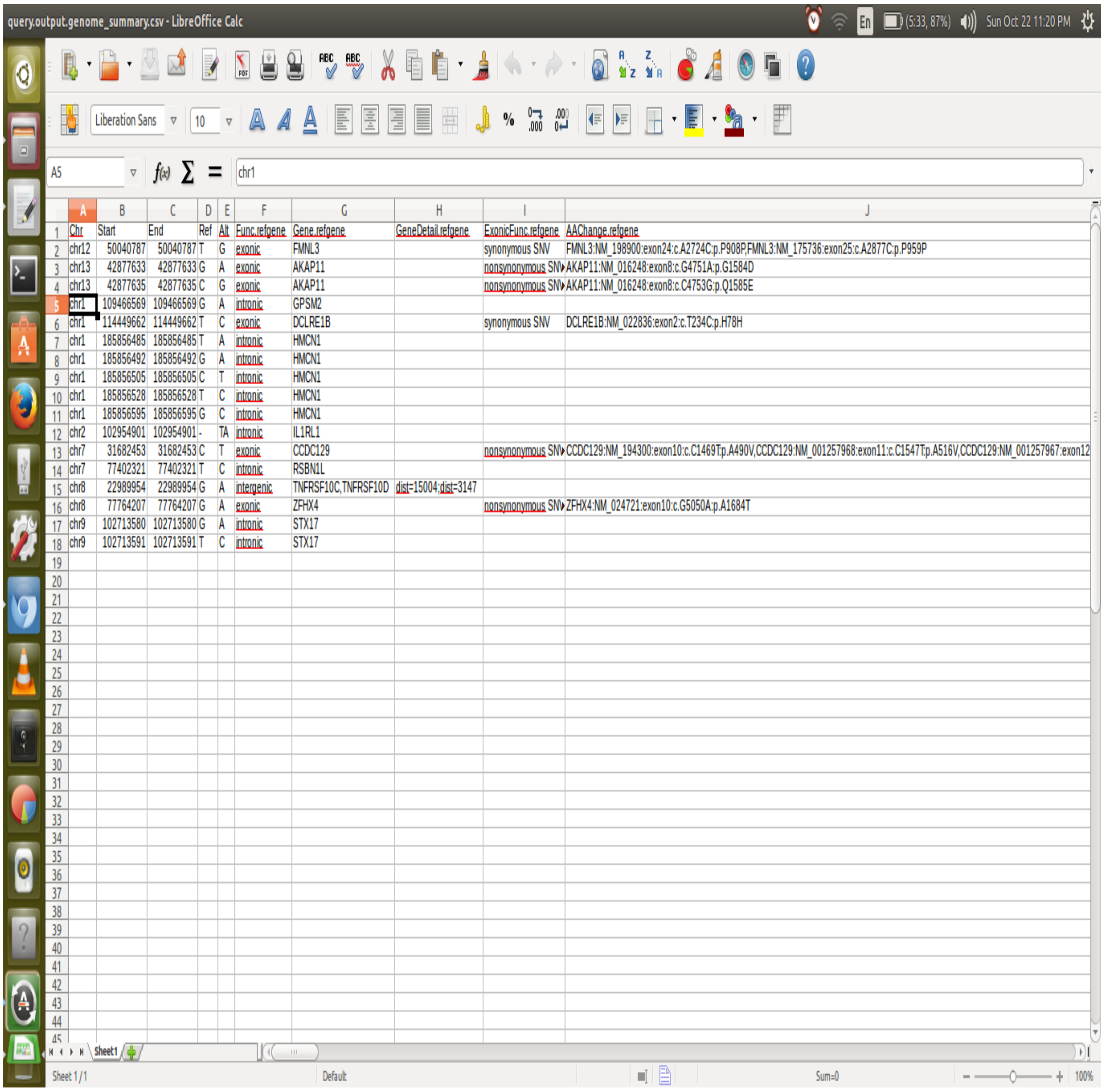


Fig. 13 – per base gc count after pre processing



```
samples.">
##INFO=<ID=QCHI2,Number=1,Type=Integer,Description="Phred scaled PCHI2.">
##INFO=<ID=PR,Number=1,Type=Integer,Description="# permutations yielding a smaller PCHI2.">
##INFO=<ID=QBD,Number=1,Type=Float,Description="Quality by Depth: QUAL/#reads">
##INFO=<ID=RPB,Number=1,Type=Float,Description="Read Position Bias">
##INFO=<ID=MDV,Number=1,Type=Integer,Description="Maximum number of high-quality nonRef reads in samples">
##INFO=<ID=VDB,Number=1,Type=Float,Description="Variant Distance Bias (v2) for filtering splice-site artefacts in RNA-seq data. Note: this
version may be broken.">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##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="# high-quality bases">
##FORMAT=<ID=DV,Number=1,Type=Integer,Description="# high-quality non-reference bases">
##FORMAT=<ID=SP,Number=1,Type=Integer,Description="Phred-scaled strand bias P-value">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="List of Phred-scaled genotype likelihoods">
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  input.sorted.rmdup.bam
chr12   50040787        .       T       G       10.4    .       DP=1;AF1=1;AC1=2;DP4=0,0,1,0;MQ=40;FQ=-30       GT:PL:GQ        1/1:40,3,0:5
chr13   42877633        .       G       A       12.3    .       DP=1;AF1=1;AC1=2;DP4=0,0,1,0;MQ=42;FQ=-30       GT:PL:GQ        1/1:42,3,0:5
chr13   42877635        .       C       G       12.3    .       DP=1;AF1=1;AC1=2;DP4=0,0,1,0;MQ=42;FQ=-30       GT:PL:GQ        1/1:42,3,0:5
chr1    109466569       .       G       A       12.3    .       DP=1;AF1=1;AC1=2;DP4=0,0,1,0;MQ=42;FQ=-30       GT:PL:GQ        1/1:42,3,0:5
chr1    114449662       .       T       C       12.3    .       DP=1;AF1=1;AC1=2;DP4=0,0,1,0;MQ=42;FQ=-30       GT:PL:GQ        1/1:42,3,0:5
chr1    185856485       .       T       A       12.3    .       DP=1;AF1=1;AC1=2;DP4=0,0,0,1;MQ=42;FQ=-30       GT:PL:GQ        1/1:42,3,0:5
chr1    185856492       .       G       A       12.3    .       DP=1;AF1=1;AC1=2;DP4=0,0,0,1;MQ=42;FQ=-30       GT:PL:GQ        1/1:42,3,0:5
chr1    185856505       .       C       T       12.3    .       DP=1;AF1=1;AC1=2;DP4=0,0,0,1;MQ=42;FQ=-30       GT:PL:GQ        1/1:42,3,0:5
chr1    185856528       .       T       C       12.3    .       DP=1;AF1=1;AC1=2;DP4=0,0,0,1;MQ=42;FQ=-30       GT:PL:GQ        1/1:42,3,0:5
chr1    185856595       .       G       C       12.3    .       DP=1;AF1=1;AC1=2;DP4=0,0,0,1;MQ=42;FQ=-30       GT:PL:GQ        1/1:42,3,0:5
chr2    102954901       .       A       ATA     4.42    .       INDEL;IS=1,1.000000;DP=1;AF1=1;AC1=2;DP4=0,0,0,1;MQ=42;FQ=-37.5
GT:PL:GQ        0/1:40,3,0:3
chr7    31682453        .       C       T       12.3    .       DP=1;AF1=1;AC1=2;DP4=0,0,1,0;MQ=42;FQ=-30       GT:PL:GQ        1/1:42,3,0:5
chr7    77402321        .       T       C       4.13    .       DP=1;AF1=1;AC1=2;DP4=0,0,1,0;MQ=42;FQ=-30       GT:PL:GQ        0/1:32,3,0:3
chr8    22989954        .       G       A       12.3    .       DP=1;AF1=1;AC1=2;DP4=0,0,0,1;MQ=42;FQ=-30       GT:PL:GQ        1/1:42,3,0:5
chr8    77764207        .       G       A       12.3    .       DP=1;AF1=1;AC1=2;DP4=0,0,1,0;MQ=42;FQ=-30       GT:PL:GQ        1/1:42,3,0:5
chr9    102713580       .       G       A       52.8    .       DP=2;VDB=4.960000e-02;AF1=1;AC1=2;DP4=0,0,1,1;MQ=42;FQ=-33
GT:PL:GQ        1/1:84,6,0:10
chr9    102713591       .       T       C       52.8    .       DP=2;VDB=5.600000e-02;AF1=1;AC1=2;DP4=0,0,1,1;MQ=42;FQ=-33
GT:PL:GQ        1/1:84,6,0:10
```

Fig. 14 – VCF file generated after removal and mpileup

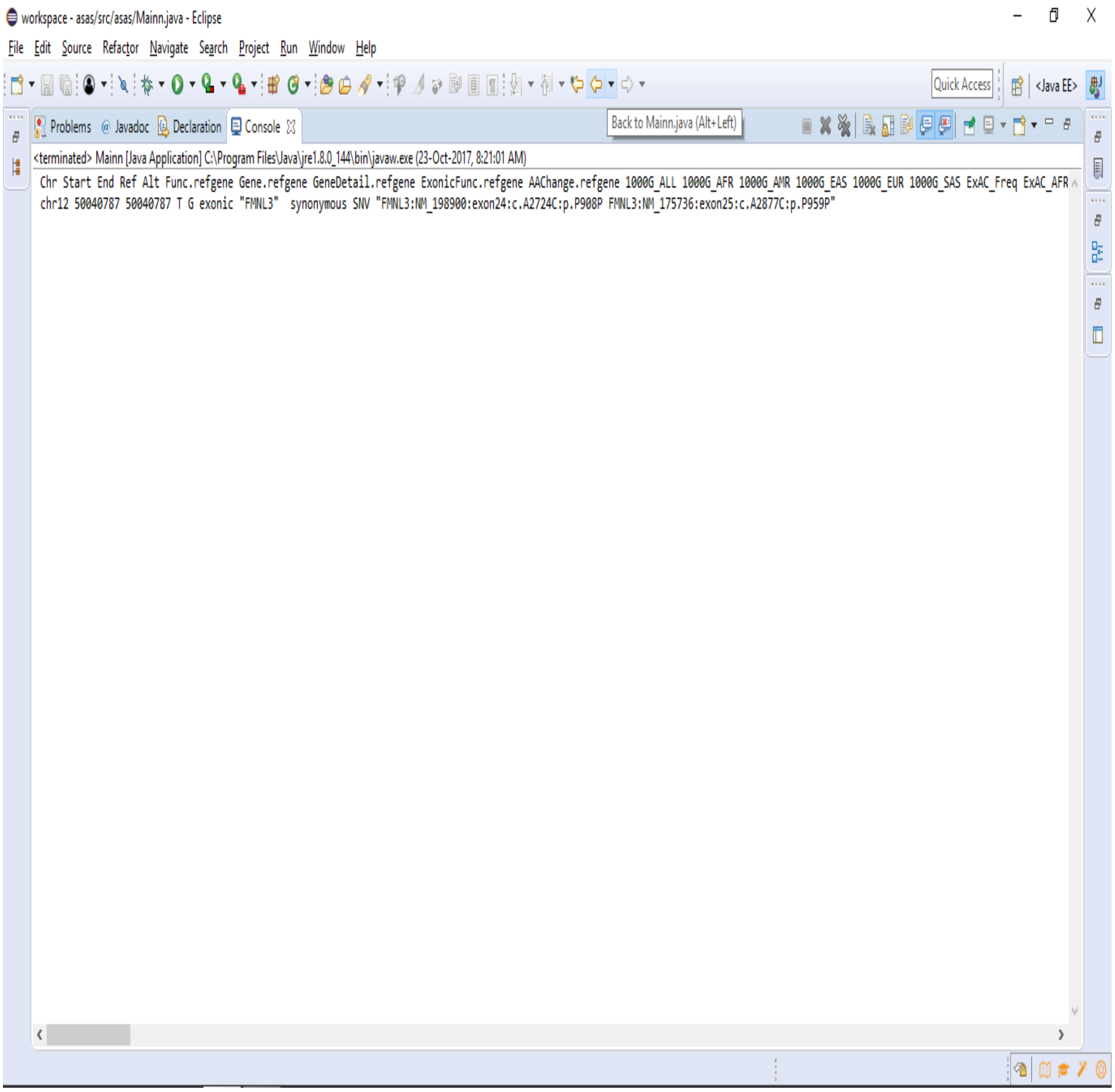

query.output.genome_summary.csv - LibreOffice Calc

A5 chr1

Chr	Start	End	Ref	Alt	Func.refgene	Gene.refgene	GeneDetail.refgene	ExonicFunc.refgene	AAChange.refgene
chr12	50040787	50040787	T	G	exonic	FMNL3		synonymous SNV	FMNL3:NM_198900:exon24:c.A2724C:p.P908P,FMNL3:NM_175736:exon25:c.A2877C:p.P959P
chr13	42877633	42877633	G	A	exonic	AKAP11		nonsynonymous SN	AKAP11:NM_016248:exon8:c.G4751A:p.G1584D
chr13	42877635	42877635	C	G	exonic	AKAP11		nonsynonymous SN	AKAP11:NM_016248:exon8:c.C4753G:p.Q1585E
chr1	109466569	109466569	G	A	intronic	GPSM2			
chr1	114449662	114449662	T	C	exonic	DCLRE1B		synonymous SNV	DCLRE1B:NM_022836:exon2:c.T234C:p.H78H
chr1	185856485	185856485	T	A	intronic	HMCN1			
chr1	185856492	185856492	G	A	intronic	HMCN1			
chr1	185856505	185856505	C	T	intronic	HMCN1			
chr1	185856528	185856528	T	C	intronic	HMCN1			
chr1	185856595	185856595	G	C	intronic	HMCN1			
chr2	102954901	102954901	-	TA	intronic	IL1RL1			
chr7	31682453	31682453	C	T	exonic	CCDC129		nonsynonymous SN	CCDC129:NM_194300:exon10:c.C1469T:p.A490V,CCDC129:NM_001257968:exon11:c.C1547T:p.A516V,CCDC129:NM_001257967:exon12
chr7	77402321	77402321	T	C	intronic	RSBN1L			
chr8	22989954	22989954	G	A	intergenic	TNFRSF10C,TNFRSF10D	dist=15004;dist=3147		
chr8	77764207	77764207	G	A	exonic	ZFHX4		nonsynonymous SN	ZFHX4:NM_024721:exon10:c.G5050A:p.A1684T
chr9	102713580	102713580	G	A	intronic	STX17			
chr9	102713591	102713591	T	C	intronic	STX17			

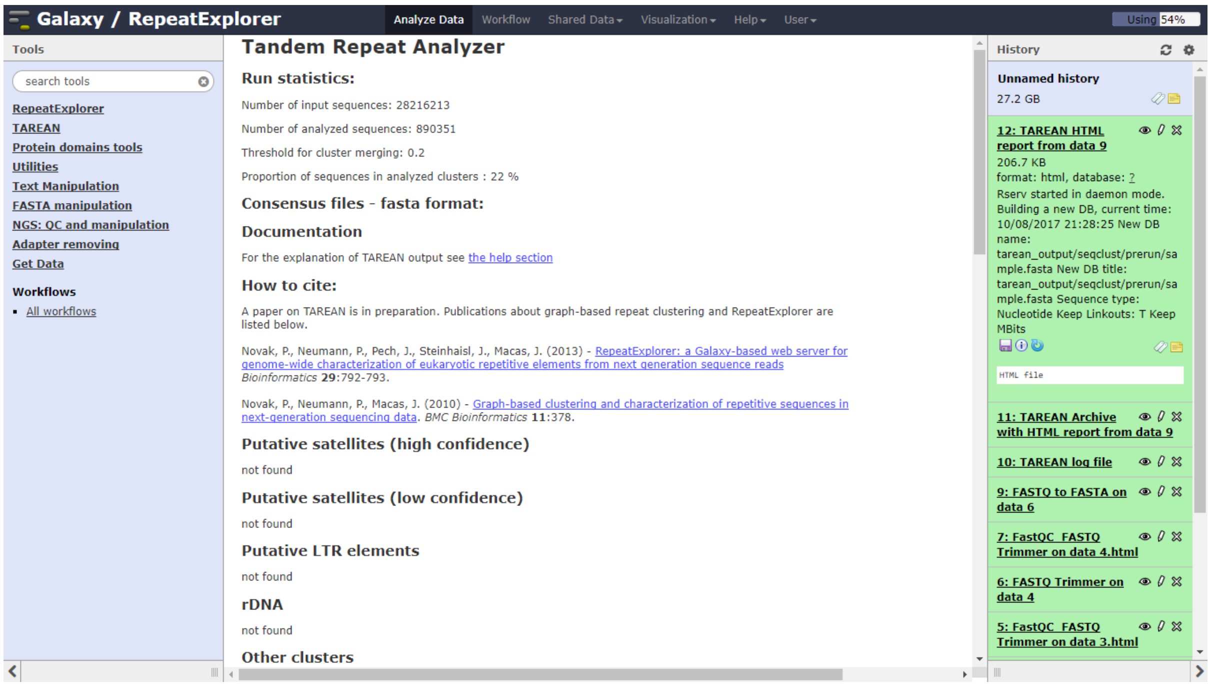
Sheet 1 / 1 Default Sum=0 100%

Fig. 15 – Output of the type of gene for each record in the VCF file


workspace - asas/src/asas/Mainn.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
<Java EE>
Problems
Javadoc
Declaration
Console
Back to Mainn.java (Alt+Left)

```
<terminated> Mainn [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (23-Oct-2017, 8:21:01 AM)
Chr Start End Ref Alt Func.refgene Gene.refgene GeneDetail.refgene ExonicFunc.refgene AAChange.refgene 1000G_ALL 1000G_AFR 1000G_AMR 1000G_EAS 1000G_EUR 1000G_SAS ExAC_Freq ExAC_AFR
chr12 50040787 50040787 T G exonic "FMNL3"  synonymous SNV "FMNL3:NM_198900:exon24:c.A2724C:p.P908P FMNL3:NM_175736:exon25:c.A2877C:p.P959P"
```

Fig. 16 – Output of abnormality detection



Galaxy / RepeatExplorer — Analyze Data | Workflow | Shared Data | Visualization | Help | User — Using 54%

Tools

search tools

RepeatExplorer
TAREAN
Protein domains tools
Utilities
Text Manipulation
FASTA manipulation
NGS: QC and manipulation
Adapter removing
Get Data

Workflows
- All workflows

Tandem Repeat Analyzer

Run statistics:

Number of input sequences: 28216213

Number of analyzed sequences: 890351

Threshold for cluster merging: 0.2

Proportion of sequences in analyzed clusters : 22 %

Consensus files - fasta format:

Documentation

For the explanation of TAREAN output see the help section

How to cite:

A paper on TAREAN is in preparation. Publications about graph-based repeat clustering and RepeatExplorer are listed below.

Novak, P., Neumann, P., Pech, J., Steinhaisl, J., Macas, J. (2013) - RepeatExplorer: a Galaxy-based web server for genome-wide characterization of eukaryotic repetitive elements from next generation sequence reads *Bioinformatics* **29**:792-793.

Novak, P., Neumann, P., Macas, J. (2010) - Graph-based clustering and characterization of repetitive sequences in next-generation sequencing data. *BMC Bioinformatics* **11**:378.

Putative satellites (high confidence)

not found

Putative satellites (low confidence)

not found

Putative LTR elements

not found

rDNA

not found

Other clusters

History

Unnamed history
27.2 GB

12: TAREAN HTML report from data 9
206.7 KB
format: html, database: ?
Rserv started in daemon mode. Building a new DB, current time: 10/08/2017 21:28:25 New DB name: tarean_output/seqclust/prerun/sample.fasta New DB title: tarean_output/seqclust/prerun/sample.fasta Sequence type: Nucleotide Keep Linkouts: T Keep MBits

HTML file

11: TAREAN Archive with HTML report from data 9

10: TAREAN log file

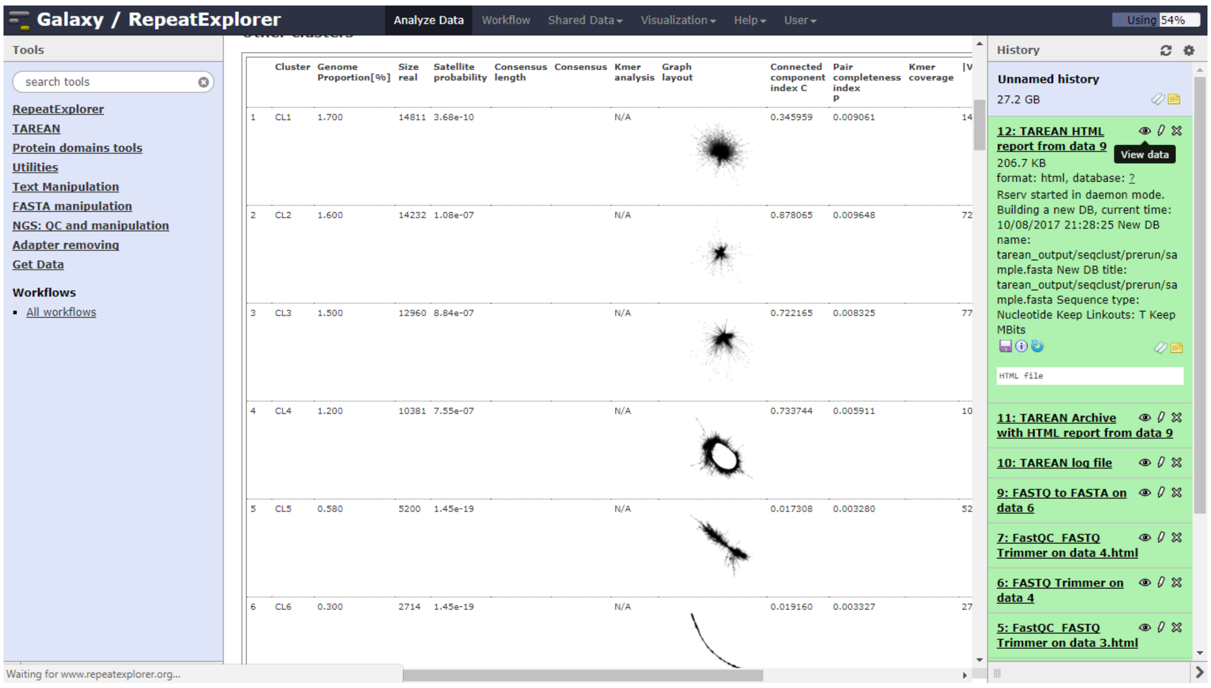
9: FASTQ to FASTA on data 6

7: FastQC FASTQ Trimmer on data 4.html

6: FASTQ Trimmer on data 4

5: FastQC FASTQ Trimmer on data 3.html

Fig. 17(a) – Result of tandem repeats



Galaxy / RepeatExplorer — Analyze Data | Workflow | Shared Data | Visualization | Help | User — Using 54%

Tools

search tools

RepeatExplorer
TAREAN
Protein domains tools
Utilities
Text Manipulation
FASTA manipulation
NGS: QC and manipulation
Adapter removing
Get Data

Workflows
- All workflows

	Cluster	Genome Proportion[%]	Size real	Satellite probability	Consensus length	Consensus	Kmer analysis	Graph layout	Connected component index C	Pair completeness index P	Kmer coverage	|V
1	CL1	1.700	14811	3.68e-10			N/A		0.345959	0.009061		14
2	CL2	1.600	14232	1.08e-07			N/A		0.878065	0.009648		72
3	CL3	1.500	12960	8.84e-07			N/A		0.722165	0.008325		77
4	CL4	1.200	10381	7.55e-07			N/A		0.733744	0.005911		10
5	CL5	0.580	5200	1.45e-19			N/A		0.017308	0.003280		52
6	CL6	0.300	2714	1.45e-19			N/A		0.019160	0.003327		27

History

Unnamed history
27.2 GB

12: TAREAN HTML report from data 9 View data
206.7 KB
format: html, database: ?
Rserv started in daemon mode. Building a new DB, current time: 10/08/2017 21:28:25 New DB name: tarean_output/seqclust/prerun/sample.fasta New DB title: tarean_output/seqclust/prerun/sample.fasta Sequence type: Nucleotide Keep Linkouts: T Keep MBits

HTML file

11: TAREAN Archive with HTML report from data 9

10: TAREAN log file

9: FASTQ to FASTA on data 6

7: FastQC FASTQ Trimmer on data 4.html

6: FASTQ Trimmer on data 4

5: FastQC FASTQ Trimmer on data 3.html

Waiting for www.repeatexplorer.org...

Fig. 17(b) – Result of tandem repeats

REFERENCES

[1] Zhang, Jingsong, Yinglin Wang, Chao Zhang, and Yongyong Shi. "Mining contiguous sequential generators in biological sequences." *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 13, no. 5 (2016): 855-867.

[2] Zhou, Hongxia, Liping Du, and Hong Yan. "Detection of tandem repeats in DNA sequences based on parametric spectral estimation." *IEEE transactions on information technology in biomedicine* 13, no. 5 (2009): 747-755.

[3] Deng, Yue, Zhiquan Ren, Youyong Kong, Feng Bao, and Qionghai Dai. "A hierarchical fused fuzzy deep neural network for data classification." *IEEE Transactions on Fuzzy Systems* 25, no. 4 (2017): 1006-1012.

[4] Gosztolya, Gábor, and László Tóth. "DNN-based Feature Extraction for Conflict Intensity Estimation from Speech." *IEEE Signal Processing Letters* (2017).

[5] Ramakrishnan, Nithya, and Ranjan Bose. "Analysis of healthy and tumour DNA methylation distributions in kidney-renal-clear-cell-carcinoma using Kullback–Leibler and Jensen–Shannon distance measures." *IET Systems Biology* 11, no. 3 (2017): 99-104.

[6] Voges, Jan, Marco Munderloh, and Jörn Ostermann. "Predictive coding of aligned next-generation sequencing data." In *Data Compression Conference (DCC), 2016*, pp. 241-250. IEEE, 2016.

[7] Deng, Yue, Zhiquan Ren, Youyong Kong, Feng Bao, and Qionghai Dai. "A hierarchical fused fuzzy deep neural network for data classification." *IEEE Transactions on Fuzzy Systems* 25, no. 4 (2017): 1006-1012.

[8] Azim, Md Aashikur Rahman, Costas S. Iliopoulos, M. Sohel Rahman, and M. Samiruzzaman. "A Simple, Fast, Filter-Based Algorithm for Approximate Circular Pattern Matching." *IEEE transactions on nanobioscience* 15, no. 2 (2016): 93-100.

[9] Pérez, Sandino Vargas, and Fahad Saeed. "A parallel algorithm for compression of big next-generation sequencing datasets." In *Trustcom/BigDataSE/ISPA, 2015 IEEE*, vol. 3, pp. 196-201. IEEE, 2015.

[10] Ku, Chee Seng, En Yun Loy, Agus Salim, Yudi Pawitan, and Kee Seng Chia. "The discovery of human genetic variations and their use as disease markers: past, present and future." *Journal of human genetics* 55, no. 7 (2010).