CNV detection using RF Classifier

A PROJECT REPORT
Submitted By

Prasoon Goswami, Roll No - 12616002034

Under the supervision of
Prof. Rituparna Sinha
Department of Information Technology

In partial fulfilment for the award of the degree
Of
BACHELOR OF TECHNOLOGY
In
INFORMATION AND TECHNOLOGY

HERITAGE INSTITUTE OF TECHNOLOGY, KOLKATA
An autonomies Institute under
Maulana Abul Kalam Azad University of Technology
Formerly Known as a
West Bengal University of Technology
July 2020

ACKNOWLEDGEMENT

I would take this opportunity to thank Prof. Pranay Chaudhury, Principal Heritage Institute of Technology for providing me with all the necessary facilities to make our project work a success.

I would like to thank our Head of the Department Prof. Siuli Roy for her kind assistance as and when required.

I will be thankful to Prof. Rituparna Sinha my project coordinator for constantly supporting and guiding me for giving me invaluable insights. Her guidance and her words of encouragement motivated me to achieve my goal and impetus to excel.

I thank my Faculty members and Laboratory assistants at the Heritage Institute of Technology for paying a pivotal and decisive role during the development of the project. Last but not least I thank all my friends for their cooperation and encouragement that they have bestowed on me.

HERITAGE INSTITUTE OF TECHNOLOGY, KOLKATA
An autonomous Institute under
Maulana Abul Kalam Azad University of Technology
Formerly Known as a
West Bengal University of Technology

BONAFIDE CERTIFICATE

Certified that this project report "CNV DETECTION USING RF CLASSIFIER" is the bonafide work of PRASOON GOSWAMI, who carried out under my supervision.

SIGNATURE
PROF. SIULI ROY
HEAD OF THE DEPARTMENT
Department of Information Technology

Heritage Institute of Technology

SIGNATURE
PROF. RITUPARNA SINHA
PROJECT GUIDE
Department of Information Technology

Heritage Institute of Technology

SIGNATURE
EXTERNAL EXAMINER

ABSTRACT

Copy number variants (CNV) are associated with phenotypic variation in several species. However, properly detecting changes in copy numbers of sequences remains a difficult problem, especially in lower quality or lower coverage next-generation sequencing data. Here, inspired by recent applications of machine learning in genomics, we describe a method to detect duplications and deletions in short-read sequencing data. In low coverage data, machine learning appears to be more powerful in the detection of CNVs than the gold-standard methods of coverage estimation alone, and of equal power in high coverage data. We also demonstrate how replicating training sets allows more precise detection of CNVs, even identifying novel CNVs in two genomes previously surveyed thoroughly for CNVs using long-read data.

List of tables, graphs and figures

FIGURES

- Random forest Chapter 3 Section 3.1
- Scatter plots Chapter 5 Section 5.1
- Box plots Chapter 5 Section 5.2
- Heat map Chapter 5 Section 5.3

TABLES

- Confusion matrix Chapter 5 Section 5.4
- RF classifier output Chapter 5 Section 5.5

Table of Contents

CHAPTER NO	TITLE	PAGE NO.
	Abstract	**iv**
	List of tables and figures	**v**
1.	**Project Introduction**	**1**
	1.1 Project objective	**2**
	1.2 Project description	**2**
	1.3 Hardware and software requirements	**3**
	1.4 Copy Number Variation	**3**
2.	**Literature Review**	**5**
3.	**Introduction to RF Classifier**	**11**
	3.1 Introduction	**12**
	3.2 Algorithm	**12**
	3.3 Advantages	**13**
	3.4 Disadvantages	**13**
4.	**Design and Implementation**	**14**
	4.1 The dataset	**15**
	4.2 Generating the dataset	**15**
	4.3 Method	**17**
	4.4 Algorithm implementation	**18**
5.	**Results and Analysis**	**20**
	5.1 Scatter Plot	**21**
	5.2 Box Plot	**22**
	5.3 Heat Map	**23**
	5.4 Confusion Matrix	**24**
	5.5 Classifier Output	**26**

Conclusion **27**
Future Scope and Enhancement **28**
References **29**

CHAPTER 1

PROJECT INTRODUCTION

Project Introduction

1.1 Project objective

This project aims to deliver a highly scalable and useful software solution to detect Copy Number Variations[9] (CNVs) in a human genome. This project uses Random forest classifier to process simulated data set and prepare a model to deliver future predictions

1.2 Project description

Copy number variation[9] (CNV) of DNA sequences is responsible for functional phenotypic variation in many organisms, particularly when it comes to causing or fighting diseases. Also, recent studies have found that duplications and deletions are an important type of mutations with functional consequences for evolution to act upon. Hence, we propose to build a system that will detect CNV of DNA sequences of a human genome with higher efficiency especially in lower quality or lower coverage next-generation sequencing data.

The proposed work is explained in details in the next chapter where the actual implementation and exploratory data analysis (EDA) has been done. Further, a detailed performance study has been done on the results generated by running above mentioned algorithm.

1.3 Hardware and software requirements

Hardware Requirements

- Processor - Intel Core 2 Duo/Quad/hex/Octa or higher-end 64-bit processor PC or Laptop
- High-speed internet connection

Software Requirements

- Operating System - Windows/Linux/Mac
- Programming Languages - Python 3.5 (or above)
- Editors and IDEs - Anaconda, Jupyter Notebook

1.4 Copy Number Variation

Copy number variation (CNV) is a phenomenon in which sections of the genome are repeated and the number of repeats in the genome varies between individuals. Copy number variation is a type of structural variation: specifically, it is a type of duplication or deletion event that affects a considerable number of base pairs. Approximately two-thirds of the entire human genome may be composed of repeats and 4.8–9.5% of the human genome can be classified as copy number variations. In mammals, copy number variations play an important role in generating necessary variation in the population as well as disease phenotype.

Copy number variations can be generally categorized into two main groups: short repeats and long repeats. However, there are no clear boundaries between the two groups and the classification depends on the nature of the loci of interest. Short repeats include mainly bi-nucleotide repeats (two repeating nucleotides e.g. A-C-A-C-A-C...) and trinucleotide repeats. Long repeats include repeats of entire genes. This classification based on the size of the repeat is the most obvious type of classification as the size is an important factor in examining the types of mechanisms that most likely gave rise to the repeats, hence the likely effects of these repeats on phenotype.

CHAPTER 2

LITERATURE REVIEW

Literature Review

2.1 A Deep Learning Approach for Detecting Copy Number Variation in Next-Generation Sequencing Data

Copy number variants (CNV) are associated with phenotypic variation in several species. However, properly detecting changes in copy numbers of sequences remains a difficult problem, especially in lower quality or lower coverage next-generation sequencing data. Here, inspired by recent applications of machine learning in genomics, we describe a method to detect duplications and deletions in short-read sequencing data. In low coverage data, machine learning appears to be more powerful in the detection of CNVs than the gold-standard methods of coverage estimation alone, and of equal power in high coverage data. We also demonstrate how replicating training sets allows more precise detection of CNVs, even identifying novel CNVs in two genomes previously surveyed thoroughly for CNVs using long-read data.

Related Works

.

- **Rastogi, S., and D. Liberles [1]** A set of lattice model genes that fold and bind to two peptide ligands with overlapping binding pockets, but not a third ligand present in the cell was designed. Each gene was duplicated in a model haploid species with small constant population size and no recombination. One set of models allowed subfunctionalization of binding events following duplication, while another set did not allow subfunctionalization. Modelling under such conditions suggests that subfunctionalization plays an important role, but as a transition state to neofunctionalization rather than as a terminal fate of duplicated genes. There is no apparent selective pressure to maintain redundancy.

- **Jensen, J. D., K. R. Thornton, and P. Andolfatto, 2008 [2]** The fixation of beneficial mutations can strongly reduce levels of closely linked neutral variation – the so-called genetic hitchhiking effect. This prediction has been used to search for positive selection by looking for regions of the genome with reduced variability. The hitchhiking model most often used is of a single selective sweep, where the location and timing of selection are assumed to be known. This single sweep model has been of great value in understanding the effect that a single selective event has on patterns of polymorphism, as a function of the strength of selection and location of the beneficial mutation. However, this model is somewhat disconnected from the problem of detecting selective sweeps in the genome, for which locations and timings are not known a priori, and should be treated as random variables.

- **Li, H., 2011 WGsim [3]** Wgsim is a small tool for simulating sequence reads from a reference genome. It is able to simulate diploid genomes with SNPs and insertion/deletion (INDEL) polymorphisms, and simulate reads with uniform substitution sequencing errors. It does not generate INDEL sequencing errors, but this can be partly compensated by simulating INDEL polymorphism. Wgsim outputs the simulated polymorphisms and writes the true read coordinates as well as the number of polymorphisms and sequencing errors in reading names. One can evaluate the accuracy of a mapper or an SNP caller with wgsim_eval.pl that comes with the package

- **Li, H., B. Handsaker, A. Wysoker, T. Fennell, J. Ruan et al 2009 [4]** The Sequence Alignment/Map (SAM) format is a generic alignment format for storing read alignments against reference sequences, supporting short and long reads (up to 128 Mbp) produced by different sequencing platforms. It is flexible in style, compact in size, efficient in random access and is the format in which alignments from the 1000 Genomes Project are released. SAMtools implements various utilities for post-processing alignments in the SAM format, such as indexing, variant caller and alignment viewer, and thus provides universal tools for processing read alignments.

- **Rebekah L. Rogers, Julie M. Cridland, Ling Shao, Tina T. Hu, Peter Andolfatto, Kevin R. Thornton [5] 2014** We have used whole-genome paired-end Illumina sequence data to identify tandem duplications in 20 isofemale lines of Drosophila yakuba and 20 isofemale lines of D. simulans and performed genome-wide validation with PacBio long molecule sequencing. We identify 1,415 tandem duplications that are segregating in D. yakuba as well as 975 duplications in D. simulans, indicating greater variation in D. yakuba. Additionally, we observe high rates of secondary deletions at duplicated sites, with 8% of duplicated sites in D. simulans and 17% of sites in D. yakuba modified with deletions. These secondary deletions are consistent with the action of the large loop mismatch repair system acting to remove polymorphic tandem duplication, resulting in rapid dynamics of gain and loss in duplicated alleles and a richer substrate of a genetic novelty than has been previously reported. Most duplications are present in only single strains, suggesting that deleterious impacts are common. Drosophila simulans shows larger numbers of whole gene duplications in comparison to larger proportions of gene fragments in D. yakuba. Drosophila simulans displays an excess of high-frequency variants on the X chromosome, consistent with adaptive evolution through duplications on the D. simulans X or demographic forces driving duplicates to high frequency. We identify 78 chimeric genes in D. yakuba and 38 chimeric genes in D. simulans, as well as 143 cases of recruited noncoding sequence in D. yakuba and 96 in D. simulans, in agreement with rates of chimeric gene origination in D. melanogaster. Together, these results suggest that tandem duplications often result in complex variation beyond whole gene duplications that offers a rich substrate of standing variation that is likely to contribute both to detrimental phenotypes and disease, as well as to adaptive evolutionary change

- **Ye, K., M. H. Schulz, Q. Long, R. Apweiler, and Z. Ning 2009 [6]** There is a strong demand in the genomic community to develop effective algorithms to reliably identify genomic variants. Indel detection using next-gen data is difficult and identification of long structural variations is extremely challenging. We present Pindel, a pattern growth approach, to detect breakpoints of large deletions and medium sized insertions from paired-end short reads. We use both simulated reads and real data to demonstrate the efficiency of the computer program and the accuracy of the results.

- **Schrider, D. R., D. Houle, M. Lynch, and M. W. Hahn, 2013 [7]** Because spontaneous mutation is the source of all genetic diversity, measuring mutation rates can reveal how natural selection drives patterns of variation within and between species. We sequenced eight genomes produced by a mutation-accumulation experiment in Drosophila melanogaster. Our analysis reveals that point mutation and small indel rates vary significantly between the two different genetic backgrounds examined. We also find evidence that ~2% of mutational events affect multiple closely spaced nucleotides. Unlike previous similar experiments, we were able to estimate genome-wide rates of large deletions and tandem duplications. These results suggest that, at least in inbred lines like those examined here, mutational pressures may result in net growth rather than contraction of the Drosophila genome. By comparing our mutation rate estimates to polymorphism data, we are able to estimate the fraction of new mutations that are eliminated by purifying selection. These results suggest that ~99% of duplications and deletions are deleterious—making them 10 times more likely to be removed by selection than nonsynonymous mutations. Our results illuminate not only the rates of new small- and large-scale mutations, but also the selective forces that they encounter once they arise.

- **Min Zhao, Qingguo Wang, Quan Wang, Peilin Jia & Zhongming Zhao [9]** Copy number variation (CNV) is a prevalent form of critical genetic variation that leads to an abnormal number of copies of large genomic regions in a cell. Microarray-based comparative genome hybridization (array CGH) or genotyping arrays have been standard technologies to detect large regions subject to copy number changes in genomes until most recently high-resolution sequence data can be analyzed by next-generation sequencing (NGS). During the last several years, NGS-based analysis has been widely applied to identify CNVs in both healthy and diseased individuals. Correspondingly, the strong demand for NGS-based CNV analyses has fuelled the development of numerous computational methods and tools for CNV detection. In this article, we review the recent advances in computational methods pertaining to CNV detection using whole genome and whole exome sequencing data. Additionally, we discuss their strengths and weaknesses and suggest directions for future development

CHAPTER 3

INTRODUCTION TO RF CLASSIFIER

Introduction to RF Classifier

Random Forest

3.1 Introduction

Random forest is a supervised learning algorithm which is used for both classifications as well as regression. However, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, the random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

3.2 Algorithm

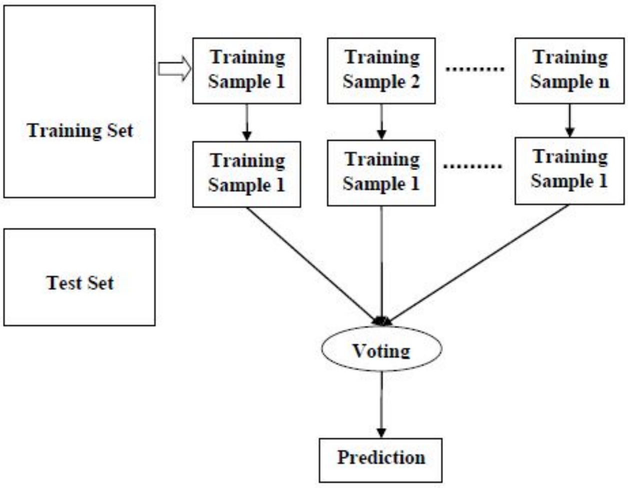
Step 1 − First, start with the selection of random samples from a given dataset.

Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

Step 3 − In this step, voting will be performed for every predicted result.

Step 4 − At last, select the most voted prediction result as the final prediction result.

The following diagram will illustrate its working −



3.3 Advantages

1. The predictive performance can compete with the best-supervised learning algorithms
2. They provide a reliable feature importance estimate
3. They offer efficient estimates of the test error without incurring the cost of repeated model training associated with cross-validation

3.4 Disadvantages

1. An ensemble model is inherently less interpretable than an individual decision tree
2. Training a large number of deep trees can have high computational costs (but can be parallelized) and use a lot of memory

CHAPTER 4

DESIGN AND IMPLEMENTATION

Design and Implementation

4.1 The data set

For this report, the data set is generated using python programming

4.2 Generating the data set

- We consider the coverage of the data be 20
    - So, we generate an array of aligned reads of length 8 Lakh (8,00,000)
    - We provide a count selected randomly in range 19 – 21 counts per b.p which we consider to be normal mapping
    - Store the generated data in an array

- Generating outliers
    - We create two arrays of the length of 2000 bp, one for duplication and one for deletion
    - For duplication we assign a value of count per base pair in range 100-500 randomly
    - For deletion we assign a value of count per base pair in range 0-8 randomly
    - We then insert these arrays at predefined locations in the original array res*
    - For duplication, location of replacement in the original array are [0,15000,45000,110000,550000,795000] index
    - For deletion, location of replacement in original array are [6000,30000,70000,150000,685000] index

- Calculating Mean, median, STD, IQR per sub-window
    - Here we consider the size of per sub-window 10 b.p
    - We use NumPy to calculate Mean, median, STD, IQR over the array res* and store them in arrays

- Forming vectors from above-calculated data per window
    - Here we take window size of 100 b.p
    - We take 10 sub-window and calculate the average Mean, median, STD, IQR per window
    - We store the value in a new array

- Assigning classes to the data based on the locations we decided for inserting outliers in step 2
    - We divide the data into three classes, namely
        - NOR – Normal class
        - DUP – Duplication class
        - DEL – Deletion Class
    - Now the data we have from the above steps are too perfect and correct so we insert random impure values to the array to get a real flavour of data to run ML algorithms
    - So we randomly change the classes at random positions of the array

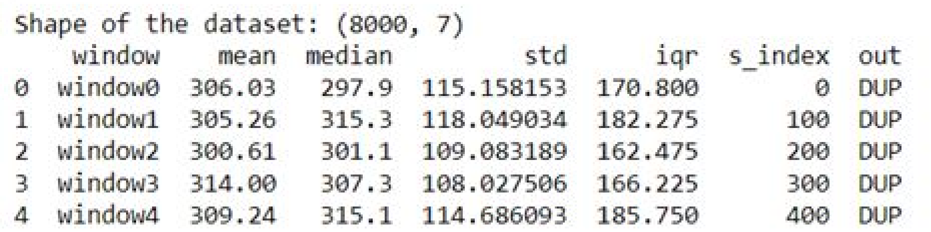
- Creating the CSV file from the above-generated arrays

**** Github Link to the code for generating the data set -**
https://github.com/PraSoonGosWami/CNV-RF/blob/master/generate%20dataset.ipynb

The shape of the data set: 8000 x 7

Data description:

- **window** (string): Describes the name of the window.
- **mean** (numerical): mean per window
- **median**(numerical): median per window
- **std**(numerical): standard deviation per window
- **iqr**(numerical): interquartile distance per window
- **s_index**(numerical): starting index of the given window
- **out**(categorical): NOR | DUP | DEL



```
Shape of the dataset: (8000, 7)
    window    mean  median         std      iqr  s_index  out
0  window0  306.03   297.9  115.158153  170.800        0  DUP
1  window1  305.26   315.3  118.049034  182.275      100  DUP
2  window2  300.61   301.1  109.083189  162.475      200  DUP
3  window3  314.00   307.3  108.027506  166.225      300  DUP
4  window4  309.24   315.1  114.686093  185.750      400  DUP
```

4.3 Method

We identified the number of values that may help determine if duplication or deletion is present in a particular genomic window. We reasoned that both standardized median and mean coverage should indicate if a window is an outlier from the average coverage of a scaffold and that the interquartile range and standard deviation in standardized coverage of a window will increase in regions with higher coverage, decrease in regions with lower coverage and increase dramatically at copy number variant (CNV) edges due to rapid shifts in coverage. Here we define standardization as dividing the coverage by the mean or median of the coverage of all bases on the contig, so the standardized coverage is distributed around 1, with a minimum of 0 and no limit to the maximum. Another component of some CNV detection algorithms is unidirectional split mapped reads and the mapping of improper pairs surrounding or within a CNV which also indicate the breakpoint of a structural variant such as a deletion or tandem duplication. Weusedthesemeasures across a set of sub-windows within the window to define the copy number and CNV class of the focal sub-window at the centre, using the RF classifier.

4.4 Algorithm implementation

We simulated a dataset having read counts per base-pair with a coverage of 20. We then divided the array of reads into window Wi , where each window Wi is considered of equal size of 100 b.p. Now, each window contains 100 read positions referring either to be mapped **Normally (NOR), Duplicated (DUP), Deleted (DEL)** b.p.

In order to detect the abnormality and outliers in the window Wi we calculated mean Mi, median Xi, standard deviation σi, interquartile distance Di for each window Wi

To get more precision we then divide windows Wi of size 100 b.p. into sub-windows Wj of size 10 b.p. We then calculated mean Mj, median Xj, standard deviation σj, interquartile distance Dj for each sub-window Wj

We reasoned that both standardized median and mean coverage should indicate if a window is an outlier from the average coverage of a scaffold and that the interquartile range and standard deviation in standardized coverage of a window will increase in regions with higher coverage, decrease in regions with lower coverage and increase dramatically at copy number variant (CNV) edges due to rapid shifts in coverage. Hence, we take an average of all the parameters calculated for all the sub-windows Wj present in a particular window Wi and hence we have standardized parameter values across each window Wi

$$Mi = Mi/Mj \text{ ——————————— } (1)$$

$$Xi = Xi/Xj \text{ ———————————— } (2)$$

$$\sigma i = \sigma i/\sigma j \text{ ———————————— } (3)$$

$$Di = Di/Dj \text{ ——————————— } (4)$$

We feed the equations **1 2 3 and 4** to the Random forest classifier library from **SciKit Learn [8]** to prepare a model.

Steps

- Importing the data set and describing the data set using python Pandas
- Plotting the scatter plots
- Plotting the box plots to view outliers for all 4 parameters (mean, median, std, iqr)
- Plotting the heat map for the data set
- Creating the dependent variable class
    - Here we change the name of our classes, DUP, NOR, DEL into numbers i.e, 0, 1, 2 respectively for training our model more efficiently
- Splitting the data set into independent and dependent variables
- Creating Train and test data from the dataset and feature scaling
    - RF Classifier works more efficiently on scaled data and yields more accurate prediction
- Training the model, predicting results and generating confusion matrix and classification report

**** Github Link to the code -**
https://github.com/PraSoonGosWami/CNV-RF/blob/master/RandomForestClassifier.ipynb

CHAPTER 5

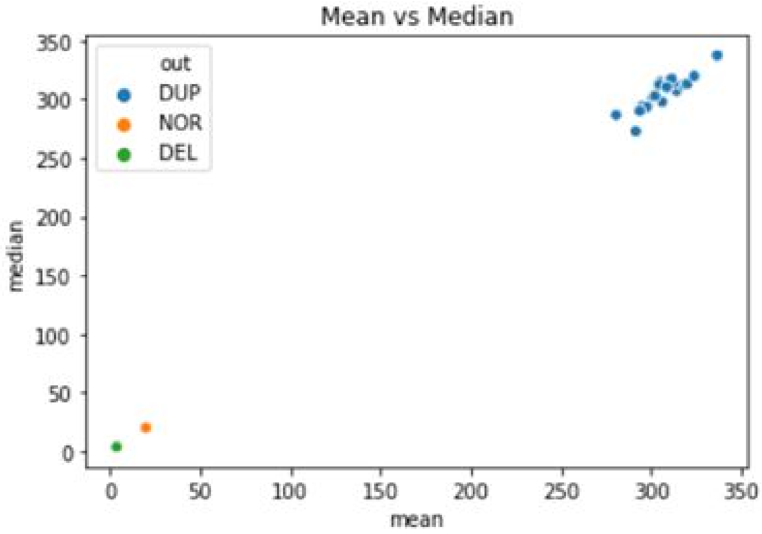
RESULTS AND ANALYSIS

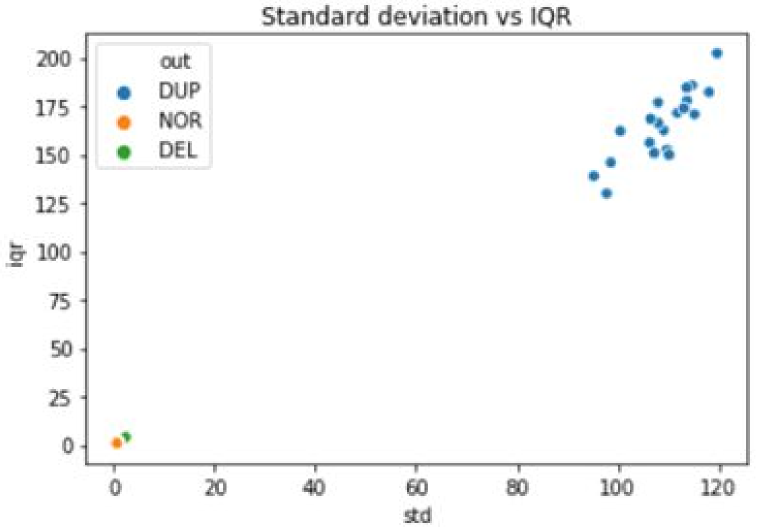
Results and Analysis

5.1 Scatter Plots

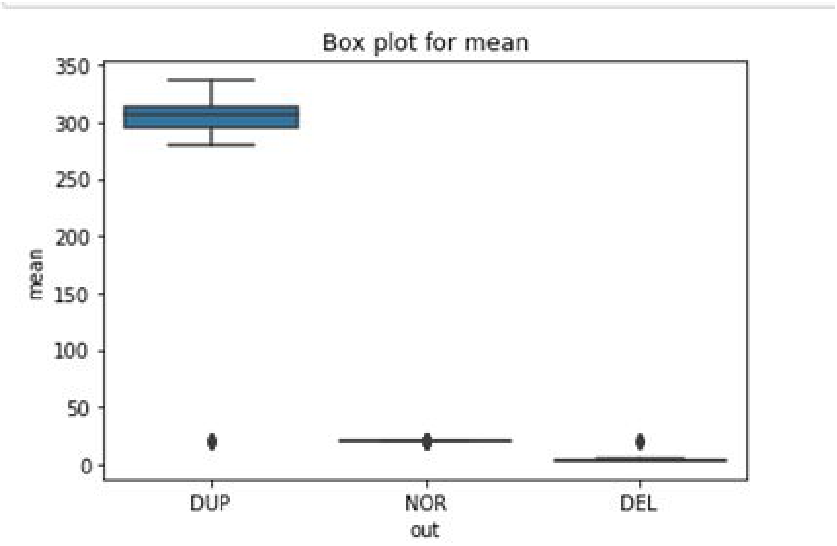
Mean vs Median

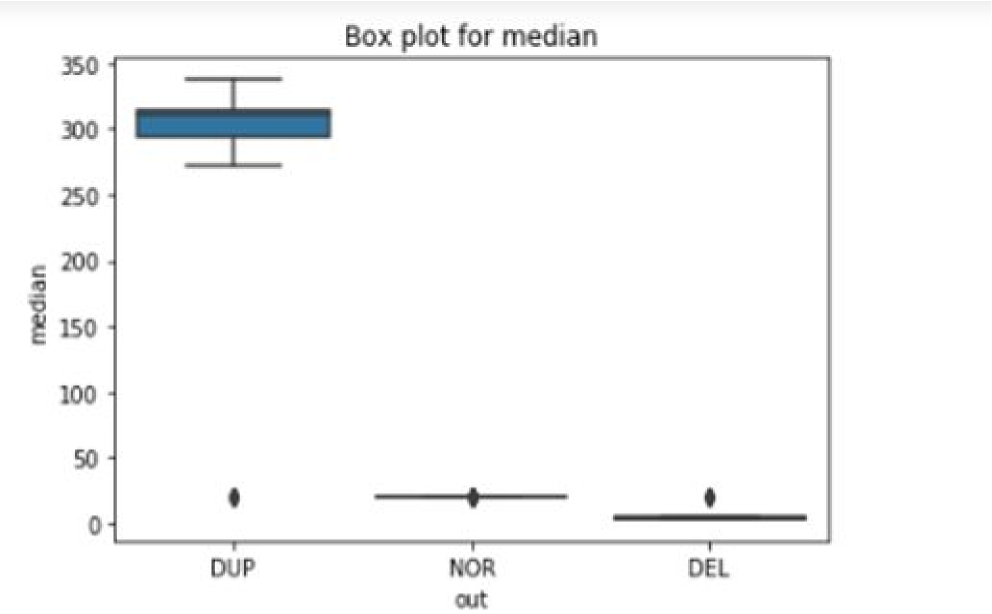
Std vs IQR

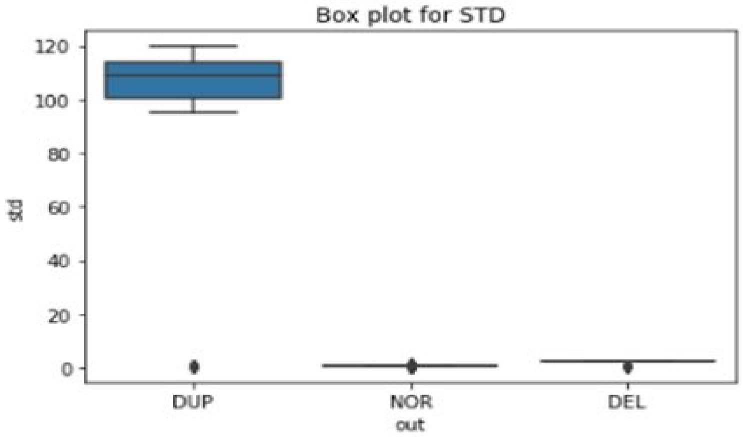


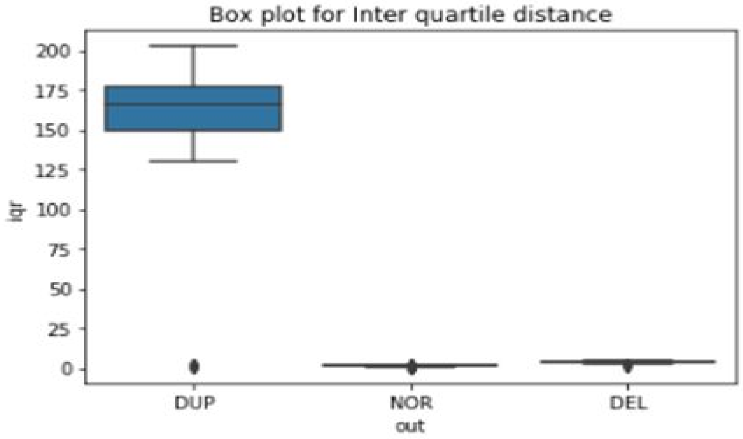


5.2 Box Plots

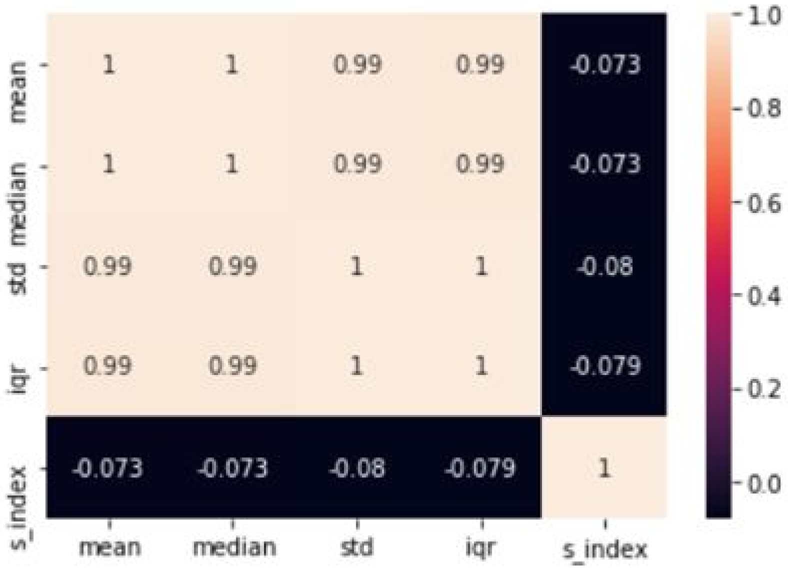

Box plot for mean
350
300
250
200
150
100
50
0
mean
DUP
NOR
DEL
out


Box plot for median
350
300
250
200
150
100
50
0
median
DUP
NOR
DEL
out


Box plot for STD
std
120
100
80
60
40
20
0
DUP
NOR
DEL
out

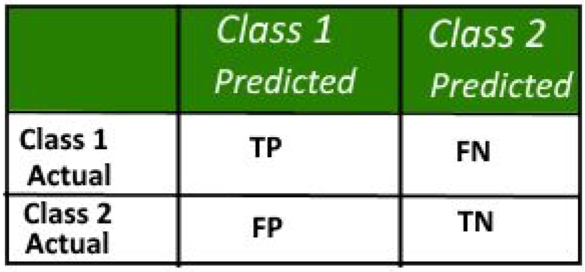

Box plot for Inter quartile distance
iqr
200
175
150
125
100
75
50
25
0
DUP
NOR
DEL
out

5.3 Heat Maps


mean
median
std
iqr
s_index
1 1 0.99 0.99 -0.073
1 1 0.99 0.99 -0.073
0.99 0.99 1 1 -0.08
0.99 0.99 1 1 -0.079
-0.073 -0.073 -0.08 -0.079 1
mean
median
std
iqr
s_index
1.0
0.8
0.6
0.4
0.2
0.0

5.4 Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. It allows easy identification of confusion between classes e.g. one class is commonly mislabeled as the other. Most performance measures are computed from the confusion matrix.



	Class 1 Predicted	Class 2 Predicted
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

Here,

- Class 1: Positive
- Class 2: Negative

Definition of the Terms:

- Positive (P): Observation is positive (for example: is an apple).
- Negative (N): Observation is not positive (for example: is not an apple).
- True Positive (TP): Observation is positive, and is predicted to be positive.
- False Negative (FN): Observation is positive, but is predicted negative.
- True Negative (TN): Observation is negative, and is predicted to be negative.
- False Positive (FP): Observation is negative, but is predicted positive.

Classification Rate/Accuracy:

Classification Rate or Accuracy is given by the relation:

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

However, there are problems with accuracy. It assumes equal costs for both kinds of errors. A 99% accuracy can be excellent, good, mediocre, poor or terrible depending upon the problem.

Recall:

Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized (a small number of FN).

The recall is given by the relation:

$$\textbf{\textit{Recall}} = \frac{TP}{TP + FN}$$

Precision:

To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example of labelled as positive is indeed positive (a small number of FP).

Precision is given by the relation:

$$\textbf{\textit{Precision}} = \frac{TP}{TP + FP}$$

High recall, low precision: This means that most of the positive examples are correctly recognized (low FN) but there are a lot of false positives.

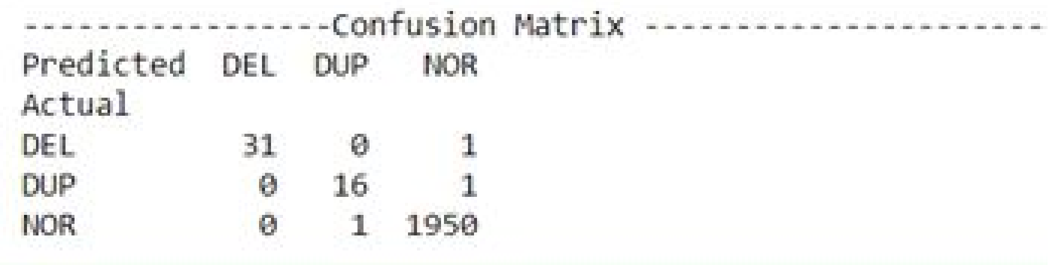
Low recall, high precision: This shows that we miss a lot of positive examples (high FN) but those we predict as positive are indeed positive (low FP)

F-measure:

Since we have two measures (Precision and Recall) it helps to have a measurement that represents both of them. We calculate an F-measure which uses Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more.

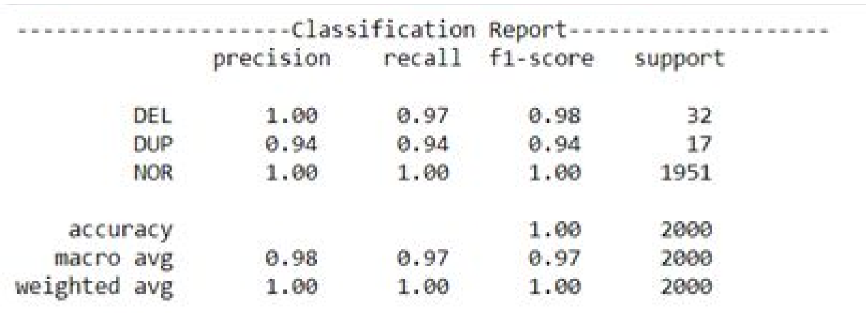
The F-Measure will always be nearer to the smaller value of Precision or Recall.

$$\textbf{\textit{F - measure =}} \quad \frac{2\text{*Recall*Precision}}{\text{Recall + Precision}}$$



```
-----------------Confusion Matrix --------------------
Predicted  DEL  DUP   NOR
Actual
DEL         31    0     1
DUP          0   16     1
NOR          0    1  1950
```

5.5 RF Classifier output



```
--------------------Classification Report-------------------
              precision    recall  f1-score   support

         DEL       1.00      0.97      0.98        32
         DUP       0.94      0.94      0.94        17
         NOR       1.00      1.00      1.00      1951

    accuracy                           1.00      2000
   macro avg       0.98      0.97      0.97      2000
weighted avg       1.00      1.00      1.00      2000
```

Conclusion

We have shown that machine learning classifiers, such as Random forest classifier, perform quite well at detecting copy number variants in comparison to other methods, particularly in samples with reduced coverage or in pools, using statistics easily derived from the sample. These tools are not computationally intensive and can be used across many datasets to detect duplications and deletions for numerous purposes. We expect machine learning to provide powerful tools for bioinformatic use in the future.

Future Scopes and Enhancements

Machine Learning is a very vast field and there is always a scope for enhancement and betterment. This project definitely needs further improvements and enhancements to yield even better and precise results.

This project can be improved and scaled to be used in real case scenario whit original human genomic data. Using big data paradigm and other parallel processing tools we can scale this algorithm to work more efficiently to yield more precise results. A highly improved version of this project can also be proposed for medical and research purposes.

References

Sr No.	Name	Paper Name	Journal Name
1	Rastogi, S., and D. Liberles	Subfunctionalization of duplicated genes as a transition state to neofunctionalization	BMC Evol. Biol. 5: 28 - https://bmcevolbiol.biomedcentral.com/articles/10.1186/1471-2148-5-28
2	G. G. Enas and S. C. Choi	An approximate Bayesian estimator suggests strong, recurrent selective sweeps in Drosophila	PLoS Genet. 4: e1000198. - https://doi.org/10.1371/journal.pgen.1000198
3	Li, H., 2011 WGsim	Read simulator for next-generation sequencing	https://github.com/lh3/wgsim
4	Li, H., B. Handsaker, A. Wysoker, T. Fennell, J. Ruan	The sequence alignment/map format and SAMtools	Bioinformatics 25: 2078–2079. https://doi.org/10.1093/bioinformatics/btp352
5	Rogers, R. L., J. M. Cridland, L. Shao, T. T. Hu, P. Andolfatto	The landscape of Standing Variation for Tandem Duplications in Drosophila yakuba and Drosophila simulans	Mol. Biol. Evol. 31: 1750–1766. https://doi.org/10.1093/molbev/msu124
6	Ye, K., M. H. Schulz, Q. Long, R. Apweiler, and Z. Ning	Pindel: a pattern growth approach to detect breakpoints of large deletions and medium-sized insertions from paired-end short reads.	Bioinformatics 25: 2865–2871. https://doi.org/10.1093/bioinformatics/btp394

7	Schrider, D. R., D. Houle, M. Lynch, and M. W. Hahn, 2013	Rates and genomic consequences of spontaneous mutational events in Drosophila melanogaster.	Genetics 194: 937–954. https://doi.org/10.1534/ genetics.113.151670
8	Pedregosa, F., R. Weiss, and M. Brucher, 2011	Scikit-learn. Machine Learning in Python.	12: 2825–2830 https://scikit-learn.org/
9	Min Zhao, Qingguo Wang, Quan Wang, Peilin Jia & Zhongming Zhao	Computational tools for copy number variation (CNV) detection using next-generation sequencing data: features and perspectives	BMC Bioinformatics volume 14, Article number: S1 (2013) https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-S11-S1