A PROJECT REPORT ON

CLASSIFICATION OF HIGHLY INTERACTING REGIONS OF THE GENOME WITH EXPLAINABLE AI

SUBMITTED TO THE

CUMMINS COLLEGE OF ENGINEERING FOR WOMEN, KARVENAGAR, PUNE
(an autonomous institute affiliated to Savitribai Phule Pune University.)
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF TECHNOLOGY (COMPUTER ENGINEERING)

SUBMITTED BY

SHREYA PAWASKAR	**C22018881961**	**4947**
RADHIKA SETHI	**C22018221381**	**4456**
AANCHAL TULSIANI	**C22018881901**	**4964**

DEPARTMENT OF COMPUTER ENGINEERING

MKSSS'S CUMMINS COLLEGE OF ENGINEERING FOR WOMEN

KARVENAGAR, PUNE 411052

SAVITRIBAI PHULE PUNE UNIVERSITY

2021-2022

CERTIFICATE

This is to certify that the project report entitled

"CLASSIFICATION OF HIGHLY INTERACTING REGIONS OF THE GENOME WITH EXPLAINABLE AI "

Submitted by

SHREYA PAWASKAR	**C22018881961**	**4947**
RADHIKA SETHI	**C22018221381**	**4456**
AANCHAL TULSIANI	**C22018881901**	**4964**

is a bonafide student of this institute and the work has been carried out by her under the supervision of **Prof. Pranjali Deshpande** and it is approved for the partial fulfillment of the requirement of Cummins College of Engineering for Women, Karvenagar, Pune (an autonomous institute affiliated to Savitribai Phule Pune university.) , for the award of the degree **Bachelor of Technology** (Computer Engineering)

(Prof. Pranjali Deshpande)
Guide
Department of Computer Engineering

(Dr. Supriya Kelkar)
Head
Department of Computer Engineering

(Dr. M. B. Khambete)
Principal,
Cummins College of Engineering for Women Pune – 52

Place: Pune

Date: 02/06/2022

भारतीय विज्ञान शिक्षा एवं अनुसंधान संस्थान पुणे
INDIAN INSTITUTE OF SCIENCE EDUCATION AND RESEARCH PUNE
डॉ. होमी भाभा मार्ग, पुणे 411008, महाराष्ट्र, भारत | Dr. Homi Bhabha Road, Pune 411008, Maharashtra, India
T +91 20 2590 8001 W www.iiserpune.ac.in

02 February 2022

Project Certificate

To Whom it May Concern

This is to certify that the following students enrolled in the Bachelors program of **MKSSS's Cummins College of Engineering for Women** are working on a research project at the Indian Institute of Science Education and Research, Pune as their Final Year Project (B.Tech Project).

1. Shreya Pawaskar
2. Radhika Sethi
3. Aanchal Tulsiani

The duration of the project is from **August 2021 to June 2022** and is being carried out under my guidance. The students are working on SEQUENCE LEVEL UNDERSTANDING OF THE 3D ORGANIZATION OF THE GENOME: Investigating potential roles of DNA signals in the 3D organization of the genome.

Narlikar

Leelavati Narlikar, PhD
Associate Professor
Department of Data Science

ACKNOWLEDGEMENT

Our heartfelt gratitude to our project guide Prof. Pranjali Deshpande, Comp. Dept. CCOEW, for her valuable suggestions and stimulating continuous guidance during the course of the project.

We would like to thank our external guide, Prof.Dr. Leelavati Narlikar for her constant guidance and support. We would also like to thank our mentor, Miss. Anushua Biswas for her continuous valuable guidance, suggestions, and precious time

We would like to express my sincere gratitude to the respected Principal, Dr. Mrs. Madhuri Khambete, and Dr. Mrs. Supriya Kelkar, Head of Department of Computer Engineering, Cummins College of Engineering for Women. We are also grateful to all the staff members of the Computer Engineering Department for their support and encouragement and guidance.

Last but not least we express our thanks to our friends and our parents who guided us in every step which we took.

SHREYA PAWASKAR	**C22018881961**	**4947**
RADHIKA SETHI	**C22018221381**	**4456**
AANCHAL TULSIANI	**C22018881901**	**4964**

ABSTRACT

The human body consists of 37.2 trillion cells. All cells in our bodies have the same genomic content, but they have various capabilities and phenotypes due to the diverse proteins synthesized by the cell, accomplished through the regulation of the expression of different genes. Proteins generated from distinct sets of active genes in cells determine these features. Gene expression refers to the process of promoting or suppressing the expression of specific genes. Certain sections of the genome are extremely interacting and are generally in charge of a collection of genes, termed TADs. Highly interacting regions are self-interacting domains found in the 3D genomic organization. Highly interacting region disruption can result in altered gene expression, linked to genetic diseases and cancer. In a diseased cell, these regions may shift, resulting in the activation of different genes and the synthesis of incorrect proteins. Our goal is to identify sequence signatures, classify them into highly interacting region or non-highly interacting region regions. We would also investigate these sequences present in the boundaries of the highly interacting region regions, which would allow us to classify genomic sequences as potentially interacting regions or non-interacting regions and determine which genomic sequence properties are influential in the transformation of a cell into a diseased cell.

For a long time, chromosomal conformation capture-based techniques have been practiced to capture the 3D organization configuration of the genome. With advancements in sequencing, 3C based experiments coupled with high throughput sequencing resulted in Hi-C experimental procedures to identify genome-wide chromatin interactions. Hi-C is a technique used to detect gene proximity and chromosomal rearrangements. Highly interacting regions were discovered for the first time in 2D chromatin interaction maps using Hi-C and 5C data from populated cells as interacting squares along the diagonal, representing local contacts. They are crucial in limiting promoter–enhancer interactions.

Their boundaries are preferentially stable across cell types, with only a proportion displaying cell-type specificity. Highly interacting regions record the local connections between different genomic profiles. They assist us in determining which genome regions are nearby, which helps to infer the 3D organization of the genome.

To discover what types of patterns or mutations result in a diseased condition, as compared to normal, we will use Deepshap, which provides information on precise patterns of the DNA structure. We will also make use of deep learning to distinguish Highly interacting regions and their boundaries and determine which traits in these regions make them more interactive, allowing us to obtain a deeper knowledge of how these specific regions influence gene regulation.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ABBREVIATION **ILLUSTRATION**

TAD	Topologically associating domains
FASTA	fast-all
Deepshap	Deep Learning SHAP (SHapley Additive exPlanations)
TF	transcription factor

LIST OF FIGURES

LIST OF TABLES

01 INTRODUCTION

3D organization of the gene and why is it important?

The three-dimensional configuration of the genome is crucial for gene regulation. The genome's 3D structure has been proven to operate in a range of biological processes, and the DNA inside the nucleus of mammalian cells is hierarchically packaged to create chromatin fibres. Higher-order chromatin organizations, for example, are frequently linked to long-distance gene regulation, which controls cell development. Furthermore, proper chromosome segregation during mitosis and meiosis is dependent on chromatin condensation and decondensation, and defects in higher-order chromatin organization can lead to developmental abnormalities and human diseases.

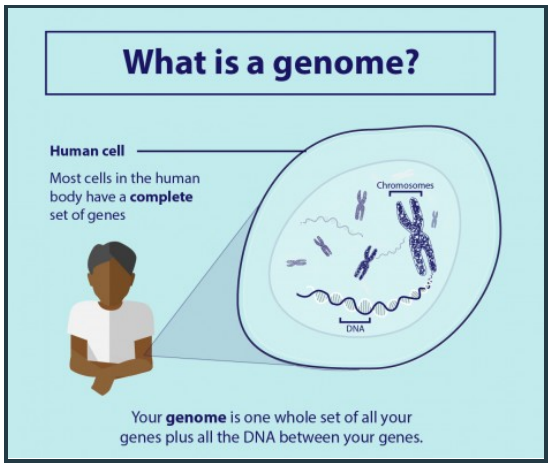


Fig1.1 - What is a genome?

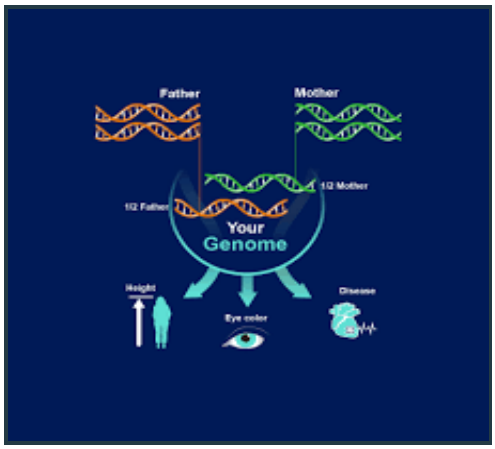


Fig1.2 - Genome Formation

Connecting it to Hi-C and its applications

Due to the limited availability of tools to study the 3D genome, unraveling the molecular basis of the chromatin architecture underlying these crucial developmental events has been historically challenging. The development of cutting-edge technology for probing chromatin organization

has made impressive strides in recent years. For example, chromosome conformation capture and its derivative technologies (such as circularised chromosome conformation capture (4C), chromosome conformation capture carbon copy (5C), and Hi- C) have been useful in detecting 3D chromatin organizations at the DNA level. These new technologies have significantly increased the toolbox for 3D genome research and advanced our understanding of higher-order chromatin organization.

Hi-C can be used to measure genome-wide interaction intensities between any pair of regions in the genome, and classify them as 'Compartment A' (Active), 'Compartment B' (Inactive)

Methods to identify changes in the 3-D organization using Hi-C

1. **TAD:** The self-interacting domain, also known as the **topologically associated domain** (TAD) or contact domain, is another major type of chromatin organization. Highly interacting regions were first identified in 2D chromatin interaction maps using Hi- C and 5C data from populated cells as interacting squares along the diagonal, which represent local contacts. Highly interacting regions have been proposed to have a major function in limiting promoter–enhancer interactions. Highly interacting region boundaries preferentially remain stable across cell types, with only a small subset exhibiting cell-type specificity. Highly interacting region boundaries are typically delineated in mammals by the chromatin architectural proteins CCCTC-binding factor (CTCF) and cohesin. CTCF and cohesin were proposed to promote 'loop extrusion', which contributes to highly interacting region formation. In this way, loop-like structures that promote interactions within TADs, but insulate regions across TAD boundaries are formed, but other factors besides CTCF and cohesin may contribute to the formation of TADs.

2. When compared to Highly interacting regions, chromatin structures may be reorganized more extensively locally. Cell type-specific interactions between genes and cis-regulatory

elements such as enhancers are examples of this reorganization. However, aside from using ultra-deep Hi- C datasets with billions of contact reads, Hi-C's relatively low resolution makes detecting promoter–enhancer interactions difficult. Protein-centric or region-centric chromatin interaction analyses, such as **chromatin interaction analysis by paired-end tag** sequencing(ChIA-PET), (ChIA-PET sequencing allows for the analysis of chromatin interactions between genomic regions bound to specific factors understudy) capture Hi- C, HiChIP, and proximity ligation- assisted chromatin immunoprecipitation and sequencing (PLAC–seq), are alternatives to Hi- C. A study using promoter capture Hi- C yielded high-resolution interaction data on cis-regulatory elements in 17 human haemato-poietic cell types. Such abundant resources can also be used to discover new cis-elements and connect non-coding disease variants to their target genes. (promoter–enhancer loops are termed as 'interaction loops')

1.1 MOTIVATION

For a long time, chromosomal conformation capture-based techniques have been practiced to capture the 3D organization configuration of the genome. With advancements in sequencing, 3C based experiments coupled with high throughput sequencing resulted in Hi-C experimental procedures to identify genome-wide chromatin interactions. Hi-C is a technique used to detect gene proximity and chromosomal rearrangements. Highly interacting regions were discovered for the first time in 2D chromatin interaction maps using Hi-C and 5C data from populated cells as interacting squares along the diagonal, representing local contacts. They are crucial in limiting promoter–enhancer interactions. Their boundaries are preferentially stable across cell types, with only a proportion displaying cell-type specificity. Highly interacting regions record the local connections between different genomic profiles. They assist us in

determining which genome regions are nearby, which helps to infer the 3D organization of the genome.

To discover what types of patterns or mutations result in a diseased condition, as compared to normal, we will use Deepshap, which provides information on precise patterns of the DNA structure. We will also make use of deep learning to distinguish Highly interacting regions and their boundaries and determine which traits in these regions make them more interactive, allowing us to obtain a deeper knowledge of how these specific regions influence gene regulation.

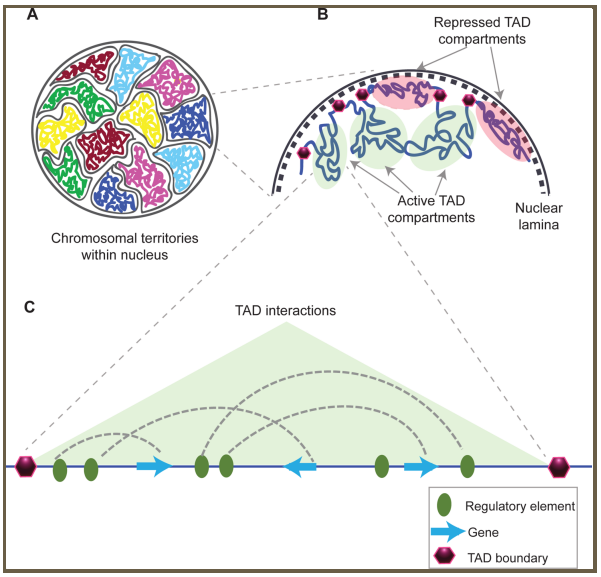


Fig 1.3 - Highly interacting regions & their interactions

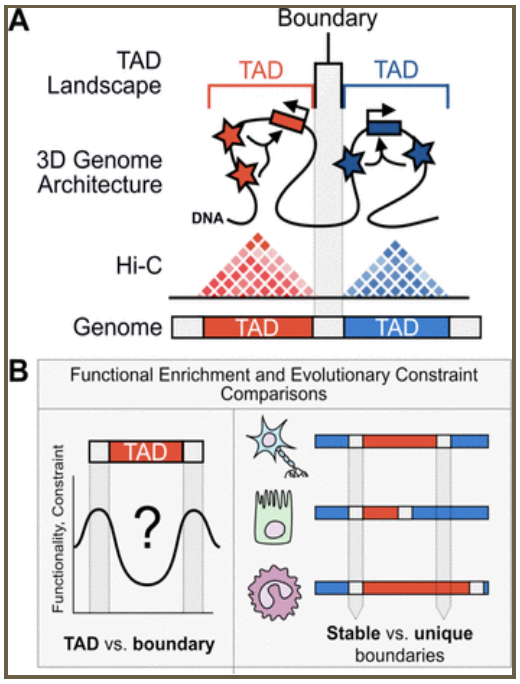


Fig 1.4 - Highly interacting region & it's boundaries.

1.2 PROBLEM DEFINITION

Thus our problem statement is "To identify the regions governing gene regulation and classify them as highly interacting and non-interacting regions using Explainable AI"

Our goal is to identify sequence signatures, classify them into highly interacting region or non-highly interacting region regions. We would also investigate these sequences present in the boundaries of the highly interacting region regions, which would allow us to classify genomic sequences as potentially interacting regions or non-interacting regions and determine which genomic sequence properties are influential in the transformation of a cell into a diseased cell.

➢ The Objectives being covered are :

- Learn how to model biological sequencing and DNA data. Understand their underlying properties and how to find patterns in them

- Efficiently use deep learning algorithms to distinguish highly interacting regions and their boundaries.
- Classify genomic sequences as potentially interacting regions or non-interacting regions
- Find which traits of highly interacting regions in these regions make them more interactive.
- Understand how these specific regions influence gene regulation.
- Find genomic sequence properties which influence the transformation of a cell into a diseased cell.

02 LITERATURE SURVEY

2.1 BACKGROUND OR DOMAIN

Topologically associating domains (TADs) are fundamental units of three-dimensional (3D) nuclear organization. TAD boundaries— regulate gene expression by restricting interactions of cis-regulatory sequences to their target genes. TAD and TAD boundary disruption has been implicated in rare disease pathogenesis; however, we have a limited framework for integrating cross-cell-type TAD maps into the study of genome evolution and interpretation of common trait-associated variants. Here, we investigate an unexplored attribute of 3D genome architecture—the stability of TAD boundaries across cell types—and demonstrate its relevance to understanding how genetic variation in TADs contributes to complex disease.

Schematic of our exploration into 3D chromatin TAD boundary stability and functionality. Chromatin is organized in 3D space into TADs which are experimentally determined by Hi-C experiments. Regions within a TAD are much more likely to interact with one another than regions outside of the TAD. Genes are represented by boxes with right-angled arrows, while gene regulatory elements, such as enhancers, are represented by stars. (B) This research focuses on two primary issues: (1) Are functional annotations, evolutionary conservation, and complex trait heritability more enriched in TADs or TAD boundaries? (2) Are stable TAD borders (i.e., those found in numerous tissues) more or less functionally enriched than TAD boundaries found only in one tissue?

2.2 Comparison between various techniques :

Method	Description	Advantages	Disadvantages
3C	• Used to study chromatin structure • Basis for several derivative techniques	• Highly Quantitative • Easy Data Analysis	• Large amount of input cells

4C	• A derivative 3C method • Designed to search the genome for sequences contacting a selected genomic site of interest	• Modified and better protocol than 3C	• Chromatin interactions must be close and stable to be detected
5C	• Known as 3C-Carbon Copy. • Detects interactions between all restriction fragments within a given region	• High efficiency	• Cannot detect contact larger than a few MBs
Hi-C	• Capable of identifying long range interactions • Used to analyze genome-wide chromatin organization	• Interactions can be detected even over relatively large genomic-distances • Can detect in-cis interactions and trans-interactions	• Low resolution • Deep sequencing needed
Statistical Techniques	• Used in statistical analysis of raw research data. • Provides different ways to assess the robustness of research outputs.	• Flexibility - Fits data Better • Can handle inputs of variable lengths	• High memory and compute time
Deep Learning	• Class of machine learning algorithms • Create models with several hidden layers of neural networks to make accurate predictions.	• Efficient at Delivering High-quality Results • Best Results with Unstructured Data • No Need for Feature Engineering	• Requires very large amount of data • Extremely expensive to train due to complex data models

Table 1 Comparision of Methods

2.3 LITERATURE REVIEW

Paper Name	Author Names	Year	Details
Methods for mapping 3D chromosome architecture	Rieke Kempfer & Ana Pombo	2019	Discusses the Chromatin Conformation Capture 3C approach needed to study the chromatin structure. They reveal the chromosome organization.
3C and 3C-based techniques: the powerful tools for spatial genome organization deciphering	Jinlei Han, Zhiliang Zhang & Kai Wang	2018	Discusses how Hi-C has helped in visualizing the abstract 3D structure of the genome. High-throughput and long sequencing read, single-cell sequencing, and epigenomics data provide us with more insights into the 3D genome.
Comparative Hi-C Reveals that CTCF Underlies Evolution of Chromosomal Domain Architecture	Matteo Vietri Rudan, Christopher Barrington, Stephen Henderson, Christina Ernst, Duncan T.Odom, Amos Tanay, Suzana Hadjur	2015	Tells us how Hi-C is used to detect gene proximity and chromosomal rearrangements. The Hi-C approach extends 3C-Seq to map chromatin contacts genome-wide, and it has also been applied to studying in situ chromatin interactions.
A decade of 3C technologies: insights into nuclear organization	Elzo de Wit and Wouter de Laat	2012	Discusses how current 3C, 4C [chromosome conformation capture-on-chip], 5C [chromosome conformation capture carbon copy], HiC, and ChIA-PET), have contributed to our current understanding of genome structure

Table 2 Traditional Methods Literature Survey

Statistical Based Methods:

Paper Name	Author Names	Year	Details
TopDom: an efficient and deterministic method for identifying topological domains in genomes	Hanjun Shin, Yi Shi, Chao Dai, Harianto Tjong, Ke Gong, Frank Alber, Xianghong Jasmine Zhou	2016	Does not identify highly interacting region hierarchies or overlapping highly interacting regions.
Identification of hierarchical chromatin domains	Caleb Weinreb, Benjamin J. Raphael	2015	Limited by high requirements of data resolution.
Two-dimensional segmentation for analyzing Hi-C data (HiCseg)	C. L´evy-Leduc, V. Brault, M. Delattre, E. Lebarbier, T. Mary-Huard and S. Robin	2014	Does not identify highly interacting region hierarchies or overlapping highly interacting regions.
Topological Domains in Mammalian Genomes Identified by Analysis of Chromatin Interactions	Jesse R. Dixon, Siddarth Selvaraj, Feng Yue, Audrey Kim, Yan Li, Yin Shen, Ming Hu, Jun S. Liu & Bing Ren	2012	Higher time complexity (takes longer to run compared to HiCseg & TopDom)

Table 3 Statistical Methods Literature Survey

Deep learning:

Paper Name	Author Names	Year	Details
PredTAD: A machine learning framework that models 3D chromatin organization alterations leading to oncogene dysregulation in breast cancer cell lines	Jacqueline Chyr, Zhigang Zhang, Xi Chen, Xiaobo Zhou	2021	The need for experimentally generated ChIPseq data, since generation of Hi-C and other high-throughput chromatin conformation capturing assays are expensive, time consuming, and not readily available.
DeepCLIP: predicting the effect of mutations on protein–RNA binding with deep learning	Alexander Gulliver Bjørnholt Grønning, Thomas Koed Doktor, Simon Jonas Larsen, Ulrika Simone Spangsberg Petersen, Lise Lolle Holm, Gitte Hoffmann Bruun, Michael Birkerod Hansen, Anne-Mette Hartung, Jan Baumbach, Brage Storstein Andresen	2020	DeepCLIP and DeepRAM can become very long as the number of sites increases. Thus, has a clear disadvantage regarding run time.
Deep highly interacting region: Visualizing Genomic Sequence Classifications	Jack Lanchantin, Ritambhara Singh, Zeming Lin, Yanjun Qi	2016	The results are not guaranteed on larger datasets.

Table 4 Deep Learning Literature Survey

03 REQUIREMENTS

3.1 Problem Statement

TADs record the local connections between different genomic profiles. They assist us in determining which genome regions are nearby, which helps to infer the 3D organization of the genome. The problem to be solved is the differentiation between highly interactive regions and non-interactive regions in TADs. Since DNA is just a sequence of ACGT, it is difficult to differentiate features between two regions. This is the classification problem - the problem is the DNA is difficult to classify which regions are high interactive regions and which are non-interactive regions.

Problem Statement:

"Identify regions governing gene regulation and classify them into highly interacting and non interacting regions with Explainable AI"

3.2 SRS

3.2.1 Scope

The Objectives being covered are:

- Learn how to model biological sequencing and DNA data. Understand their underlying properties and how to find patterns in them
- Efficiently use deep learning algorithms to distinguish between highly interacting and non interacting regions
- Find which traits of highly interacting regions make them more interactive.
- Understand how these specific regions influence gene regulation.
- Find genomic sequence properties which influence the transformation of a cell into a diseased cell.

3.2.2 Features

1. Data Fetching - We need to fetch the data since we will be provided with only the coordinates. To fetch the required data, we will make use of BED Tools
2. Cell Classification -Understand how TAD regions differ in different types of cells
3. Discover Sequence Signatures -To identify the sequence signatures specific to these regions and understand the difference between the TAD region and its boundary

3.2.3 Functional Requirements

3.2.3.1. Accurate Data Reading - The system should be able to read the HiC Data From NCBI.

3.2.3.2. Suitable Dataset Generation - The system should be able to generate a suitable Fasta File using data preprocessing.

3.2.3.3. Data Classification using Markov Models - The system should be able to correctly classify the regions into TAD/Non TAD using Markov Models

3.2.3.4. Data Classification using Deep Learning
The system should be able to correctly classify the regions into TAD/Non TAD using Deep Learning models - Deepshap and CNN.

3.2.3.5. Data Visualizations - The system should be able to generate data visualizations like

- The accuracy of the classifier
- AUC RoC Curve of the classifier
- Precision-Recall Curve of the classifier

- Confusion Matrix of the classifier
- The highest frequency element from the Markov Model

3.2.4 Nonfunctional Requirements

3.2.4.1 Performance Requirements - The system should be able to perform all the functions with least time complexity. the software's speed of response, throughput, and execution time should be high.

3.2.4.2 Safety Requirements - Data integrity is the overall accuracy, completeness, and consistency of data. The data should be retained by the system safely.

3.2.4.3 Security Requirements - The system should be secure enough for the data & results. It should be operable only by authorized users.

3.2.4.4 Software Quality Attributes

- Reliability - The likelihood that a product will run without failure for a specific number of uses (transactions) or for a specified amount of time is known as reliability. The system should be accurate enough for all kinds of data sets.
- Flexibility - The system should be flexible enough for developers to make changes as and when necessary.

3.2.4.5 Hardware Requirements

1. GPU access for training models
2. Linux Kernel for bioinformatics tools (BED Tools)
3. Intel i3 or higher
4. 4GB RAM

5. 1 Gb hard free drive space

3.3 Use Cases

1. Fasta file generation

Name	Generate Fasta File
Description	A Fasta file is generated from the HiC Data From NCBI
Pre condition	The dataset should not be empty
Sequence	• User upload HiC data from NCBI • System fetching the coordinates • The data is pre processed into a Fasta File Format • The resulting file is displayed on the screen
Post Condition	The user can see the Fasta File that is to the classified.

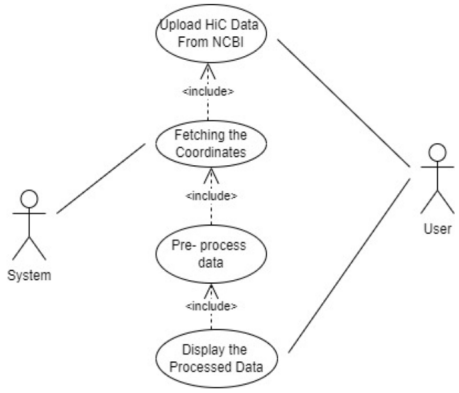


Fig 3.3.1 - Use Case 1

2. Data Visualisation

Name	Generate Data Visualization
Description	Data visualizations are generated from the classifier output
Pre condition	The outputs should not be empty
Sequence	• The classifier predicts the output • The accuracy is displayed • AUC ROC and Precision Recall Curves are shown • Confusion Matrix can be seen • The highest frequency element of the Markov Model is displayed
Post Condition	The user can see the data visualizations of the result generated

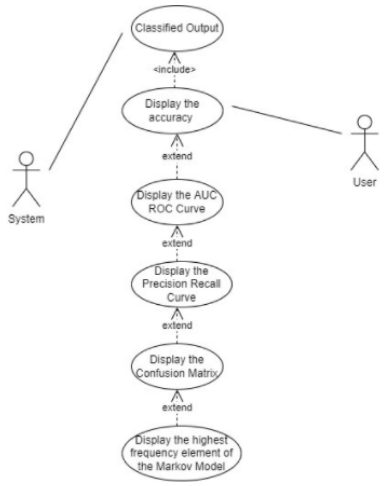


Fig 3.3.2 - Use Case 2

3. Complete system use case - Markov Models

Name	Predict using Markov Models
Description	The region is classified in TAD/Non TAD using Markov Models
Pre condition	The dataset should not be empty
Sequence	• The dataset is cleaned and converted • The inputs regions are classified • The output is explained • The results are displayed on the screen
Post Condition	The user knows the classification and the parameters which lead to the classification

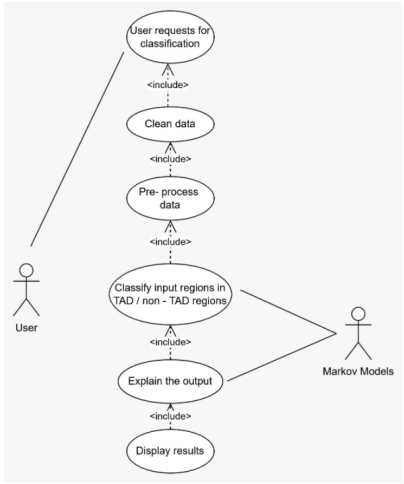


Fig 3.3.3 - Use Case 3

4. Complete System Use case - Deep Learning

Name	Predict using CNN and DeepShap
Description	The region is classified in interacting and non interacting regions using CNN and the output is explained using DeepShap
Pre condition	The dataset should not be empty
Sequence	• The dataset is cleaned and converted • The inputs regions are classified • The output is explained • The results are displayed on the screen
Post Condition	The user knows the classification and the parameters which lead to the classification

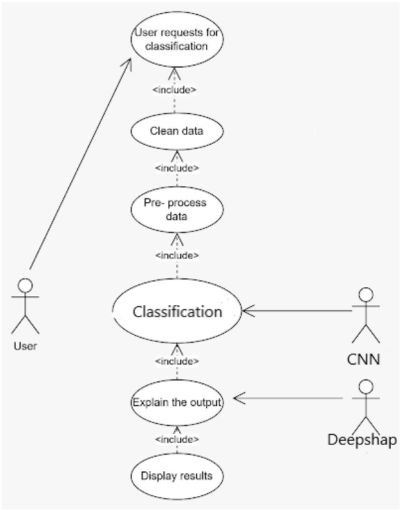


Fig 3.3.4 - Use Case 4

04 System Design

4.1 Algorithms:

A. Deepshap: In many cases, understanding why a model generates a particular prediction is just as important as the accuracy of the prediction. However, complex models that even experienced struggle to read, such as ensemble or deep learning models, often obtain the maximum accuracy for huge datasets, creating a tension between accuracy and interpretability. SHAP (SHapley Additive exPlanations) is a game-theoretic method to understanding any machine learning model's output. It has two innovative components: (1) the finding of a new class of additive feature significance measures, and (2) theoretical results indicating that this class has a single solution with a set of desirable characteristics.

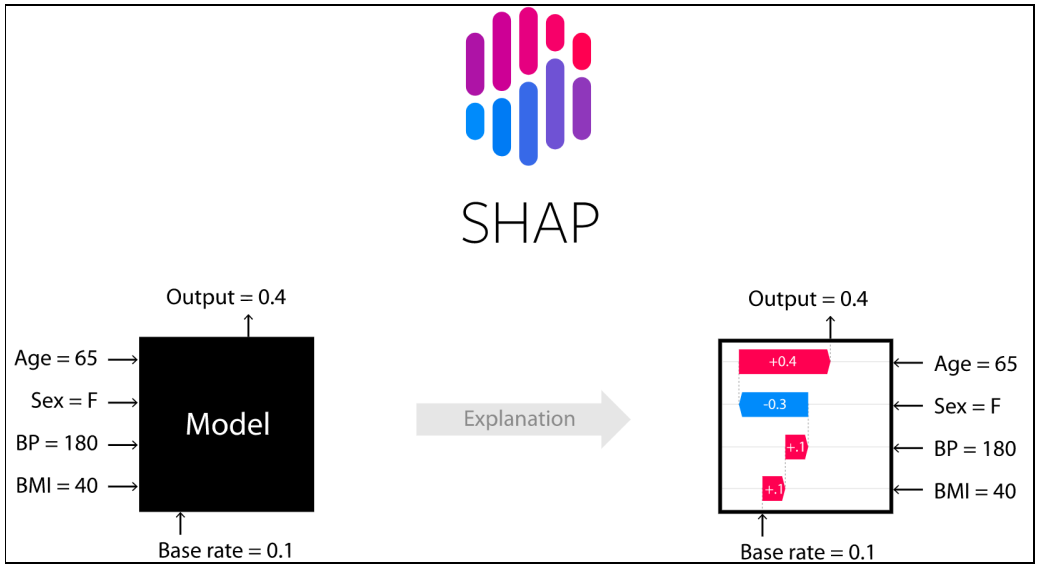


Fig 4.1.3 - DeepShap

4.2 UML Diagrams

- Activity Diagram

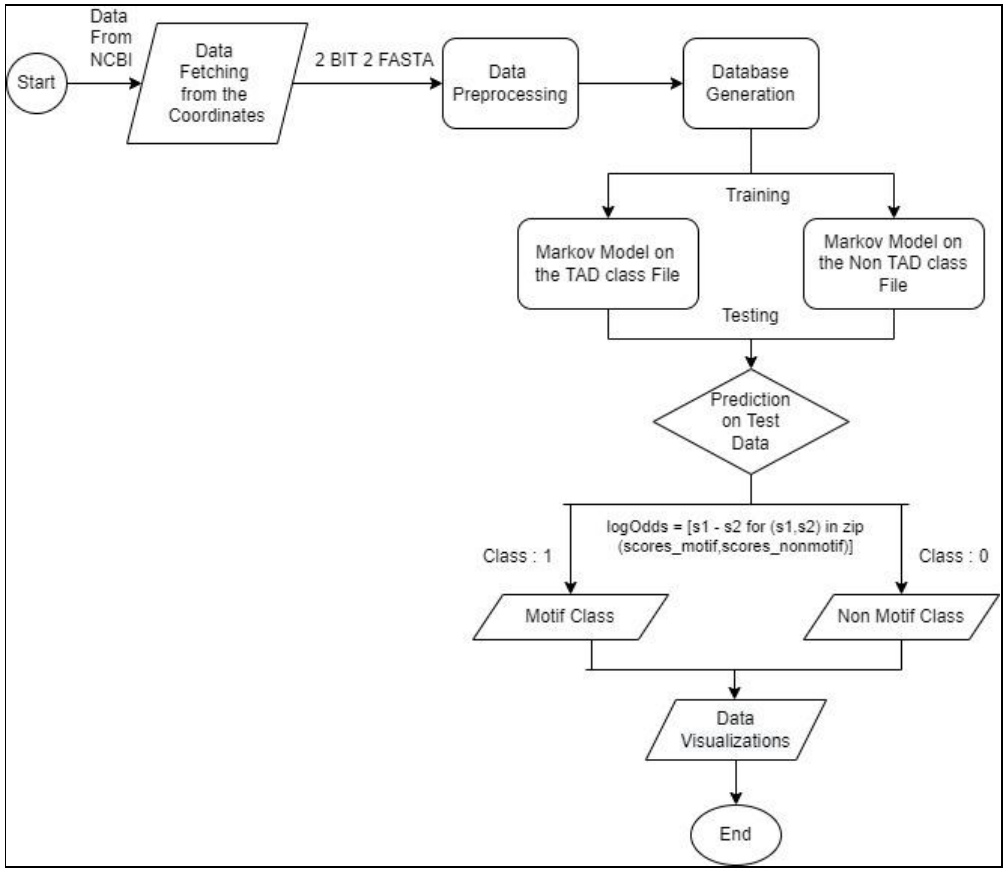


Fig 4.2.1 Activity Diagram

4.3 Architecture :

A) Markov Models :

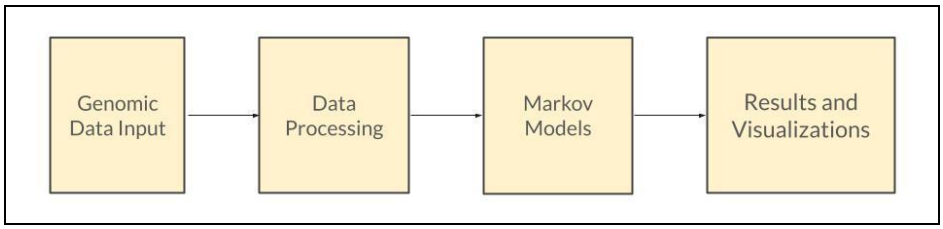


Fig 4.3.1 Markov Model Flow Diagram

B) Deep Learning :

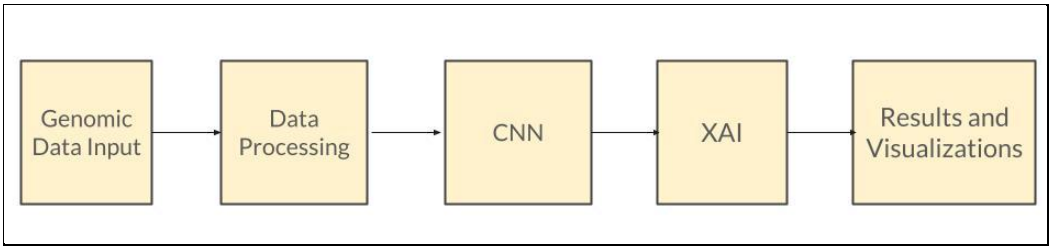


Fig 4.3.2 Deep Learning Flow Diagram

Preprocessing flow :

A) Markov Models :

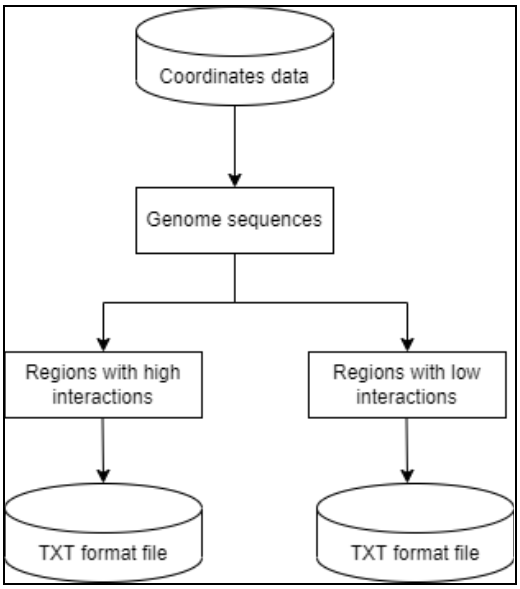


Fig 4.3.3 Markov Model Preprocessing Diagram

Training and testing flow:

A) Markov Models :

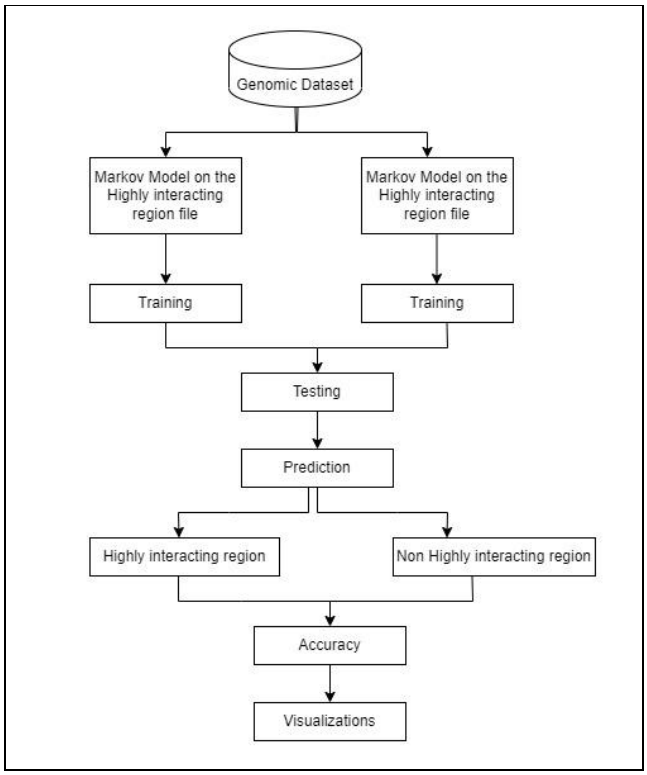


Fig 4.3.4 Markov Model Training and testing Diagram

B) **Deep Learning :**

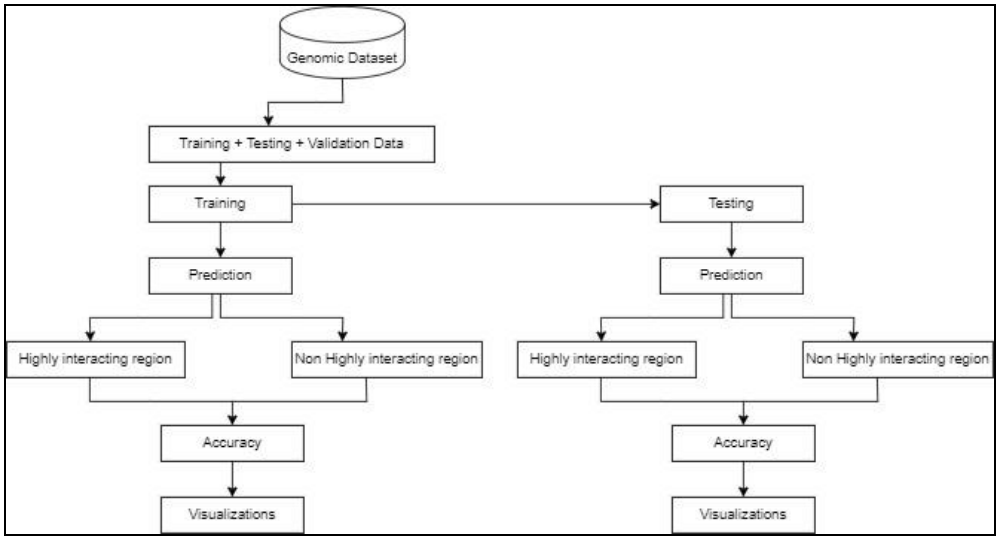


Fig 4.3.5 Deep Learning Training and testing Diagram

05 Technology

5.1 Platform used

Language

- Python 3.7
- Bash

IDE

- Google Colab
- Linux Terminal
- Visual Studio Code

Development Tools

- Bed Tools (2bit2fasta)

5.2 Test Plans

Sr.no	Test Case Name	Purpose of the test	Input	Expected Output	Test Result
1	Generate a Fasta File	To validate a Fasta file is generated from the HiC Data From NCBI	HiC Data From NCBI	The user can see the Fasta File that is classified.	Pass
2	Predict using Markov Models	To validate the region is classified in TAD/Non TAD using Markov Models	Preprocessed Fasta File	The user can see the classification & the parameters which lead to the classification using Markov Models	Pass
3	Predict using Deep Learning	To validate the region is classified in TAD/Non TAD usingCNN and Deepshap	Preprocessed Fasta File	The user can see the classification & the parameters which lead to the classification using Deep Learning	Pass
4	Generate Data Visualizatio ns	To validate that data visualizations are generated by the classifier output	Output from Classifier Data	The user can see various data visualizations.	Pass

Table 5.1 - Test Cases

06 Implementation Aspects

A) Statistical Models

We have worked on 3 scenario implementations so far.

6.1. Markov models on Dummy Data and highly interacting regions :

We generated dummy data with 0.25 probability and divided into 2 files. We embedded a highly interacting region using the random function to each sequence in one file (highly interacting region file), and the other file being the non-highly interacting region file. We trained a highly interacting region Markov model on the highly interacting region file, a non-highly interacting region Markov model on the non-highly interacting region file.

6.2. Markov models on Dummy Data and JASPAR highly interacting regions

We generated dummy data with 0.25 probability. They are using the probability frequency matrix from Jaspar, generated a Probability Weight Matrix, to generate real-like highly interacting regions. We divided into 2 files and embedded a highly interacting region using the random function to each sequence in one file (highly interacting region file), and the other file being the non-highly interacting region file.

6.3. Markov Models with Cross-Validation

6.3.1 Data Preparation

We Generated dummy data with 0.25 probability. Using the probability frequency matrix from Jaspar, generated a Probability Weight Matrix, to generate real-like highly interacting regions. We divided into 2 files and embedded a highly interacting region using the random function to each sequence in one file (highly interacting region file), and the other file being the non highly interacting region file.

6.3.2 Flow Diagram

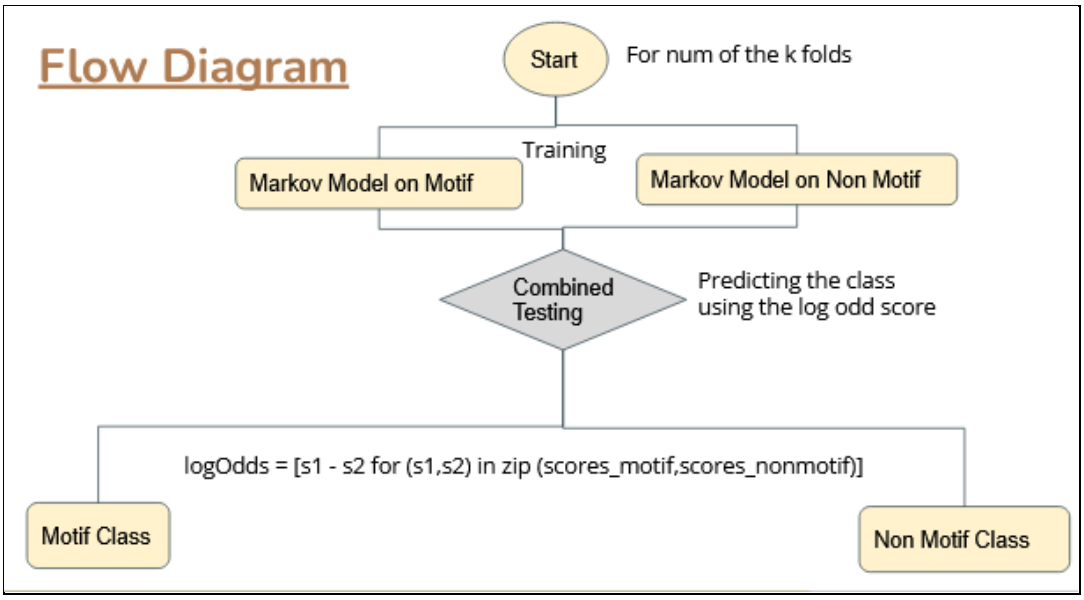


Fig 6.3 Flow Diagram

6.3.3 Algorithm

1. Generate dummy background data of 0.25 AGCT
2. Develop PWM from JASPER PFM
    - For every background sequence in the highly interacting region file generate a highly interacting region
    - using random.randint() select an index to insert the highly interacting region
    - save the index to a list
    - Generate highly interacting region database

3. Generate non highly interacting region database

- Train highly interacting region Markov model on highly interacting region database
- Train non highly interacting region Markov model on non highly interacting region database

4 . Combine test sequences of the highly interacting region and non highly interacting

region and give to both models

5 . Compare the log-odds score of the sequence belonging to the highly interacting region Markov model and the non highly interacting region Markov mode, accordingly assign the class it belongs to

6. Generate Data visualization such as AUC ROC curves, accuracy, and F1 scores.

6.4 Markov models with Cross Validation on Real Data

6.4.1 Dataset Used:

We referred the paper - Sub-kb Hi-C in D. melanogaster reveals conserved characteristics of TADs between insect and mammalian cells. We downloaded the data files with domain coordinates from National Center for Biotechnology Information (NCBI).

	A	B	C	D
1	n G1/S-arrested S2R+ cells			
2	**chromosome**	**Start Position**	**End Position**	
3	chr2L	7716	17449	
4	chr2L	21643	66422	
5	chr2L	66422	71492	
6	chr2L	71492	84867	
7	chr2L	86316	94100	
8	chr2L	94100	102022	
9	chr2L	102022	109431	
10	chr2L	109431	114582	
11	chr2L	117130	122075	
12	chr2L	122075	131214	
13	chr2L	131214	143261	
14	chr2L	143261	155437	
15	chr2L	155437	159411	
16	chr2L	159411	180711	
17	chr2L	180711	203046	
18	chr2L	206959	223445	
19	chr2L	223445	250672	
20	chr2L	250672	272225	
21	chr2L	272225	277446	
22	chr2L	277446	283328	
23	chr2L	283328	294987	
24	chr2L	294987	298022	
25	chr2L	298022	305030	
26	chr2L	305030	312470	
27	chr2L	312470	348254	

Fig 6.4.1 Dataset Obtained

6.4.2 Shift of 200

To get the coordinates of the left and right boundaries, we take a shift of 200 coordinates and generate 3 files accordingly. The Sequence length is of 200 chars. After generating the right bed files, we used 2BitToFa to generate it's fasta file, and then converted the fasta file into a suitable database for our processing

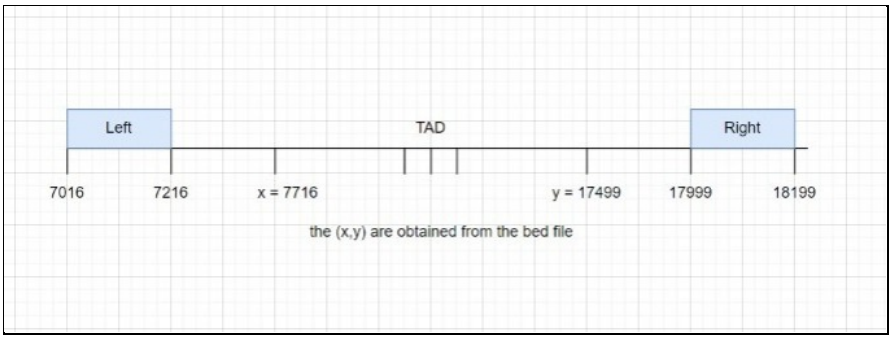


Fig 6.4.2 Shift of 200

```
chr2L   7316    7516    chr2L:7316-7516
chr2L   21243   21443   chr2L:21243-21443
chr2L   66022   66222   chr2L:66022-66222
chr2L   71092   71292   chr2L:71092-71292
chr2L   85916   86116   chr2L:85916-86116
```

Fig 6.4.3 Bed File containing Shift of 200

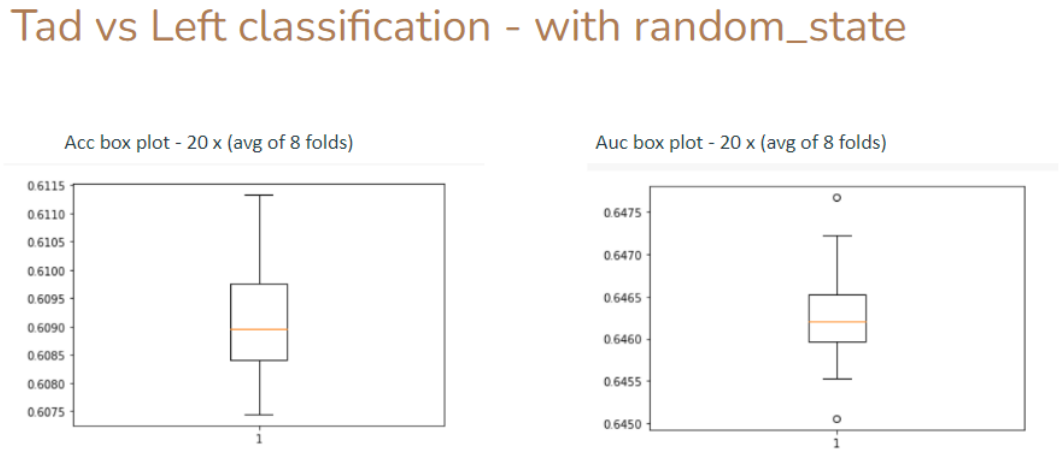
```
>chr2L:7316-7516
TAGAGAGGAGAGGACAATATTATAATTGTAGACCGTTTTAAACACTTTAA
AATGTTTAACCATTTATCAATTATTCTACTAAATGTAGGTGATTTTATTT
ATTAGAATACGAATTCTTTATCTGAATCGAACTAAGTAAGCCTAAGCGCT
TAGGAAAAATACATACTTGACGAGTAGAGTGAAATAATTACAAATATTAG
>chr2L:21243-21443
CACAGATAATAATACGACCGGTCAGAGCTTAACCGTGTATCTGTTTTATA
AAACGTGAACAATATTATTAGCCCAAAACGATATGCGCGTCATTTTAACA
CACACAAAGTCGCGATCGGTGGGTCTAGTGTGCCCGTGTATCTCTATCGA
AAAAATCATATATTTTTTAGAAGGTATATTTACCATGACTGACTGGAATG
```

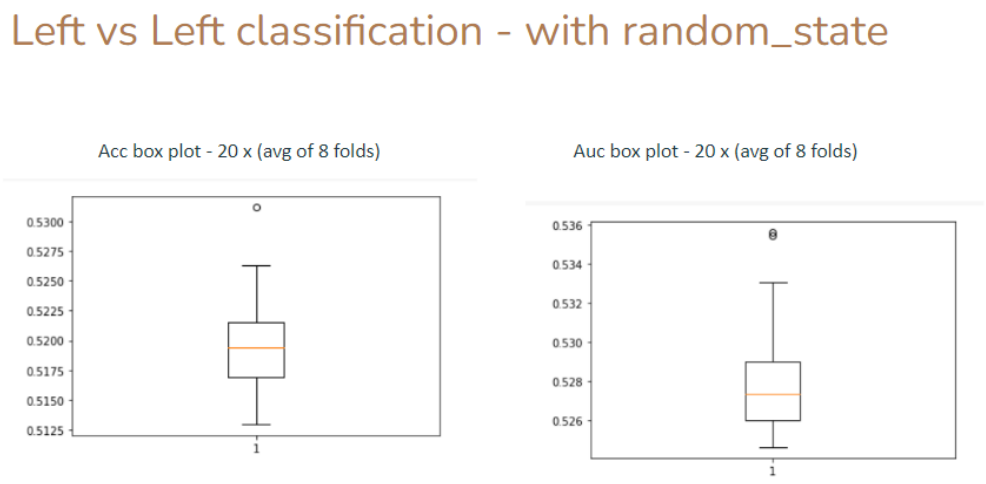
Fig 6.4.4 Fasta File containing Shift of 200

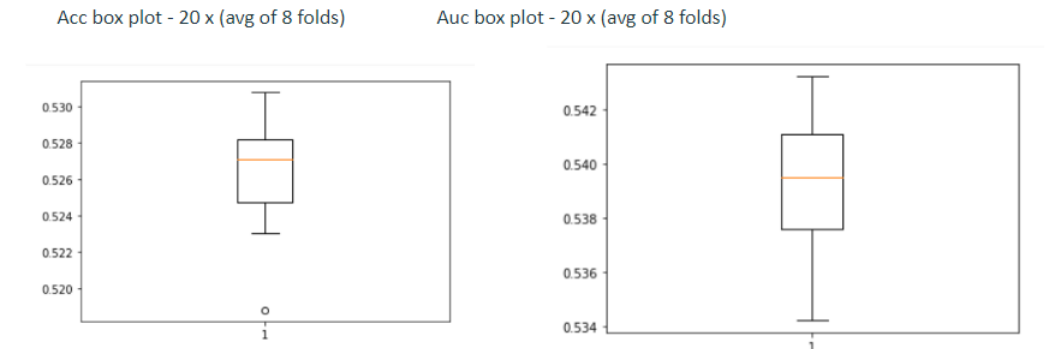
	FoldID	EventId start_index	seq	Bound
2	A	chr2L:7316-7516 -5	TAGAGAGGAGAGGACAATATTATAATTGT	
3	A	chr2L:21243-21443	-5	CACAGATAATAATACGACCGG

Fig 6.4.5 Text File containing Shift of 200

Results:







Shift of 500

1. To get the coordinates of the left and right boundaries, we take a shift of 200 coordinates and generate 3 files accordingly.
2. Sequence length of 500 char
3. After generating the right bed files, we used 2BitToFa to generate it's fasta file, and then converted the fasta file into a suitable database for our processing

1	chr2L	7016	7216	chr2L:7016-7216
2	chr2L	20943	21143	chr2L:20943-21143
3	chr2L	65722	65922	chr2L:65722-65922
4	chr2L	70792	70992	chr2L:70792-70992
5	chr2L	85616	85816	chr2L:85616-85816

Fig 6.4.6 Bed File containing Shift of 500

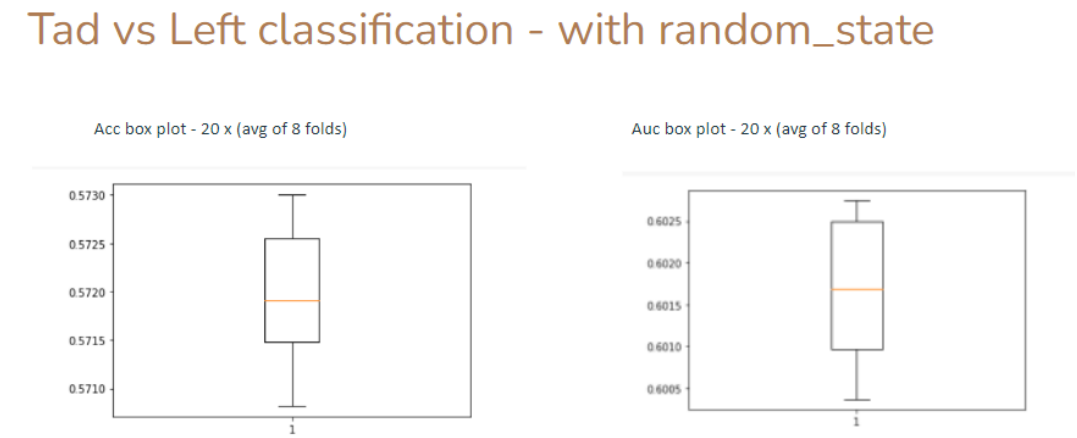
```
>chr2L:7016-7216
TGTTTAATACCTATTTGCGCATATGCGTTTATTTTTGGGATTTAATTTTA
ACATTTTTCAACAAAACCGTTACAAATGTAATTTTAAATCAGGAAACGAC
TTTGGTATGAAAATATGTTTTTTTGTGCGCTTTTAAACATGTAACTGCTC
TTTTGTGCTGTTTTATTGAATGCTATCACAGCGTAAAATTTTAGTTTTAA
>chr2L:20943-21143
CGTGGCTGACTCACCGACTACGCCTCTATGCTAAGAACATACATATTTGT
GGACACTTTATTTAAGAAGTTTGAAATATAAATCAATTGCCTAATCAGTA
ATACTGTTGTTGAGCCCTTTACCTTTATGTATTTCCGTTTGTACGGTTAA
GGCGGTGGCCGAGTAATTTTTTGAACTATTTTATTTGCTCACCATCACGC
```

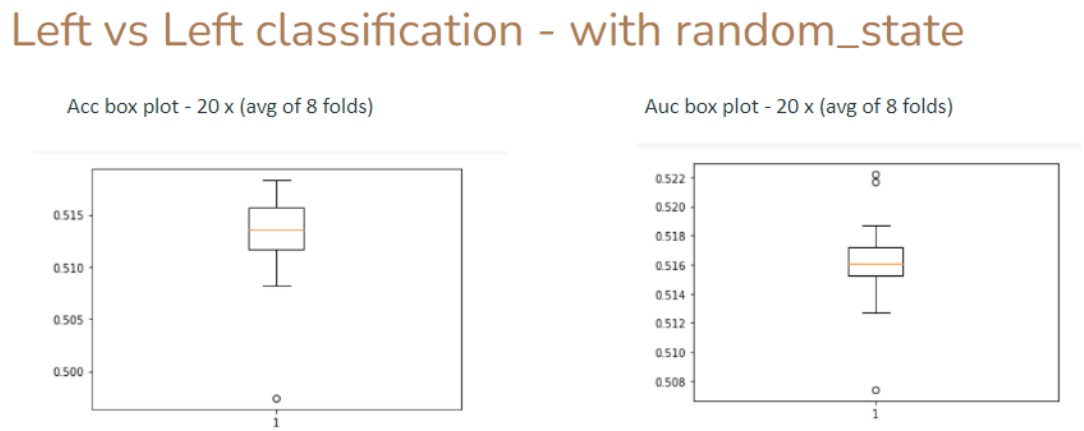
Fig 6.4.7 Fasta File containing Shift of 500

FoldID	EventId	start_index	seq	Bound
A	chr2L:7016-7216	-5	TGTTTAATACCTATTTGCGCATATGCGTTTATTTTTGGGATTTAATTTTA	
A	chr2L:20943-21143	-5	CGTGGCTGACTCACCGACTACGCCTCTATGCTAAGAACATAC	

Fig 6.4.8 Text File containing Shift of 500

Results





6.5 DNA Complement

After performing the three experiments on the data of shift 200 and data of shift 500, we realised that on classification of left vs right boundary, there were some signals, as the ideal accuracy should have been 0.50. For each input sequence, we took the complement of it and added it to our training dataset.

A-> T, T-> A, G-> C, C-> G

We found out the box plots of the AUC and Accuracy values of all the combinations of data.

Results:

Visualization labels:

- m - mixing the left and right datasets, dividing them into 2, and training separately.
- mc - the same, but while also taking the complement of the sequences while training
- lr - classification between left and right boundary
- lrc - classification between left and right boundary (including their complements while training)
- lt - classification between left and tad region
- ltc - classification between left and tad region (including their complements while training)

1. Box plot of ROC:

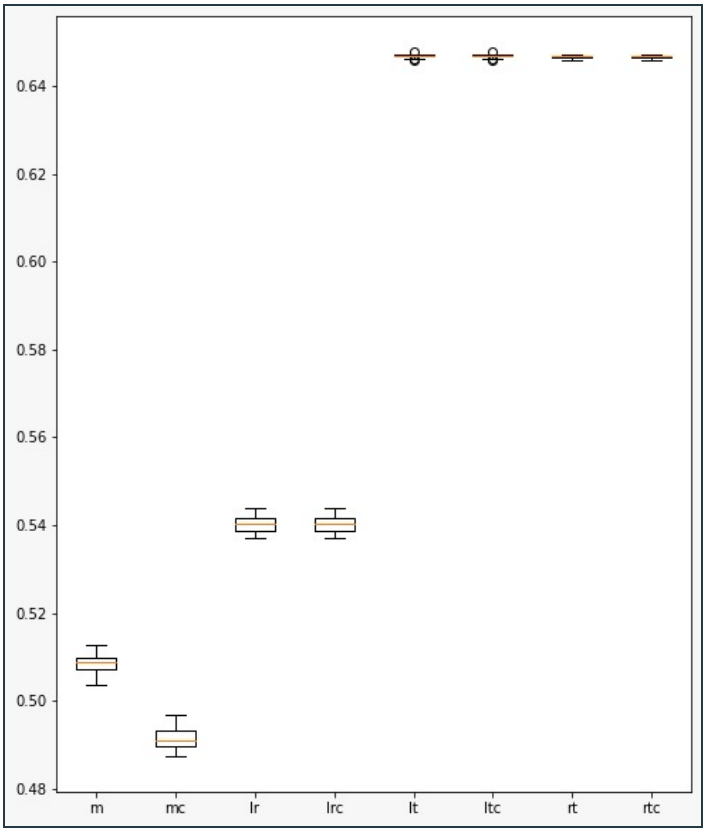


Fig 6.5.1 Box plot of AUC

1. **Box plot of ACC**

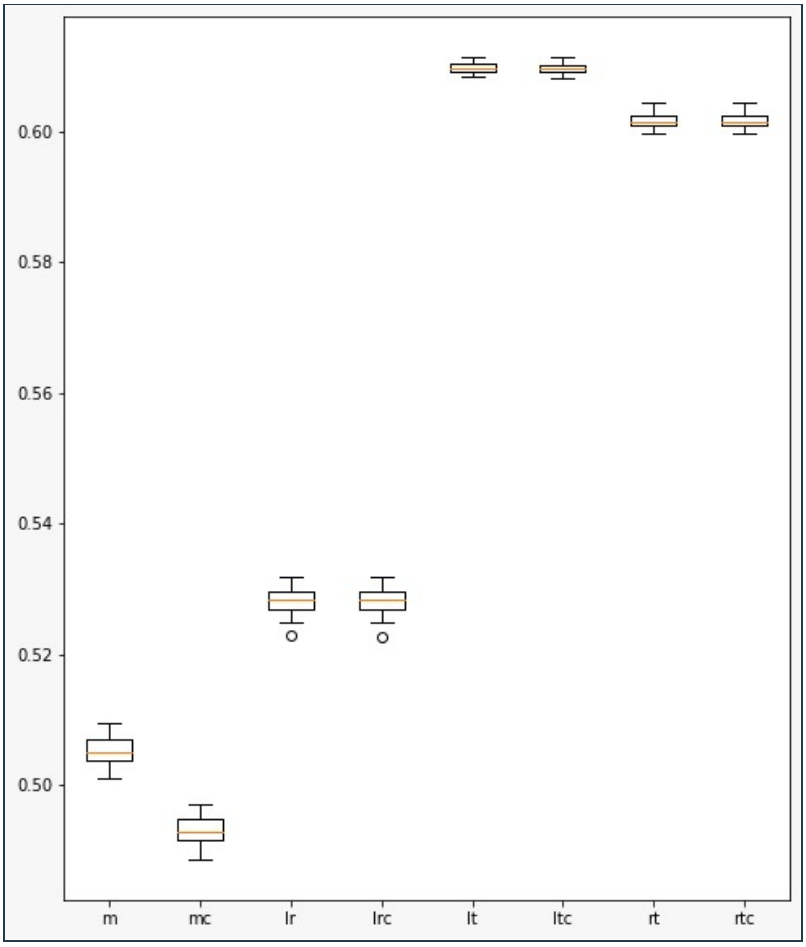


Fig 6.5.2 Box plot of ACC

B) Deep Learning

6.1 Why is Deep Learning important in Genomics?

Deep Learning methods are capable of working on large datasets. It capable of identifying highly complex patterns. It can be used for the interpretation of regulatory control.It can build generative models (generating their own data and feeding to the model). CNN's can discover new patterns even when the locations of patterns within sequences are unknown. It can learn from millions of sequences through parallel implementation on a GPU. It generalizes well across technologies, even without

correcting for technology-specific biases. It can tolerate a moderate degree of noise and mislabeled training data.

6.2 CNN Model Architecture

The output will have 2 classes - 0 and 1 to classify between a TAD and boundary region.

Libraries used:

a. Keras

b. Tensorflow

Model Summary :

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 192, 480)          17760

 batch_normalization (BatchN  (None, 192, 480)         1920
 ormalization)

 leaky_re_lu (LeakyReLU)     (None, 192, 480)          0

 max_pooling1d (MaxPooling1D  (None, 62, 480)          0
 )

 dropout (Dropout)           (None, 62, 480)           0

 conv1d_1 (Conv1D)           (None, 59, 480)           922080

 batch_normalization_1 (Batc  (None, 59, 480)          1920
 hNormalization)

 leaky_re_lu_1 (LeakyReLU)   (None, 59, 480)           0

 max_pooling1d_1 (MaxPooling  (None, 28, 480)          0
 1D)

 dropout_1 (Dropout)         (None, 28, 480)           0

 conv1d_2 (Conv1D)           (None, 25, 240)           461040
```

```
 max_pooling1d_2 (MaxPooling  (None, 8, 240)           0
 1D)

 dropout_2 (Dropout)         (None, 8, 240)           0

 conv1d_3 (Conv1D)           (None, 8, 320)           77120

 batch_normalization_2 (Batc  (None, 8, 320)          1280
 hNormalization)

 leaky_re_lu_2 (LeakyReLU)   (None, 8, 320)           0

 max_pooling1d_3 (MaxPooling  (None, 3, 320)          0
 1D)

 flatten (Flatten)           (None, 960)              0

 dense (Dense)               (None, 180)              172980

 dense_1 (Dense)             (None, 2)                362

=================================================================
Total params: 1,656,462
Trainable params: 1,653,902
Non-trainable params: 2,560
_________________________________________________________________
```

6.3 Results Obtained :

1. **CROSS VALIDATION on Simulated Data**

   Kfold = 10

   Training Accuracy obtained:

```
------------------------------------------------------------------------
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.23009233176708221 - Accuracy: 93.9393937587738%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.21601657569408417 - Accuracy: 95.15151381492615%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.09989245980978012 - Accuracy: 97.45454788208008%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.19407188892364502 - Accuracy: 95.99999785423279%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.0978783518075943 - Accuracy: 97.93939590454102%
------------------------------------------------------------------------
> Fold 6 - Loss: 0.12726952135562897 - Accuracy: 97.93939590454102%
------------------------------------------------------------------------
> Fold 7 - Loss: 0.05214976891875267 - Accuracy: 98.90776872634888%
------------------------------------------------------------------------
> Fold 8 - Loss: 0.1217418983578682 - Accuracy: 98.05825352668762%
------------------------------------------------------------------------
> Fold 9 - Loss: 0.14774739742279053 - Accuracy: 97.69417643547058%
------------------------------------------------------------------------
> Fold 10 - Loss: 0.058431413024663925 - Accuracy: 99.02912378311157%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 97.21135675907135 (+- 1.5705969767735715)
> Loss: 0.13452916070818902
------------------------------------------------------------------------
```

Testing accuracies obtained:

- fold 1 : 0.942708087854
- fold 2 : 0.957537466941
- fold 3 : 0.965992203234
- fold 4 : 0.953881012837
- fold 5 : 0.960375427825
- fold 6 : 0.970837064618
- fold 7 : 0.964799104513
- fold 8 : 0.956217566108
- fold 9 : 0.972143665158
- fold 10 : 0.962032255783

Mean (Average)	0.96065238548708
Median	0.96120384180405
Range	0.029435577304292

Geometric Mean	0.96061683091567
Largest	0.9721436651583711
Smallest	0.9427080878540792

Testing AUCs obtained:

- fold 1 : 0.94303030303
- fold 2 : 0.957575757576
- fold 3 : 0.966060606061
- fold 4 : 0.953939393939
- fold 5 : 0.96
- fold 6 : 0.970909090909
- fold 7 : 0.964805825243
- fold 8 : 0.956310679612
- fold 9 : 0.972087378641
- fold 10 : 0.962378640777

Mean (Average)	0.9607097675787
Median	0.96118932038835
Range	0.029057075610474
Geometric Mean	0.96067479223173
Largest	0.9720873786407767
Smallest	0.943030303030303

2. CROSS VALIDATION on Drosophila Data

Training accuracy obtained:

```
------------------------------------------------------------------------
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 0.9264867901802063 - Accuracy: 68.48484873771667%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.37268972396850586 - Accuracy: 86.90909147262573%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.45120540261268616 - Accuracy: 83.99999737739563%
------------------------------------------------------------------------
> Fold 4 - Loss: 0.5648748278617859 - Accuracy: 84.72727537155151%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.4073570966720581 - Accuracy: 88.48484754562378%
------------------------------------------------------------------------
> Fold 6 - Loss: 0.38833341002464294 - Accuracy: 89.21211957931519%
------------------------------------------------------------------------
> Fold 7 - Loss: 0.38002294301986694 - Accuracy: 89.19903039932251%
------------------------------------------------------------------------
> Fold 8 - Loss: 0.47175219655036926 - Accuracy: 88.10679316520691%
------------------------------------------------------------------------
> Fold 9 - Loss: 0.410195916891098 - Accuracy: 88.95630836486816%
------------------------------------------------------------------------
> Fold 10 - Loss: 0.5044685006141663 - Accuracy: 87.01456189155579%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 85.50948739051819 (+- 5.930888894166661)
> Loss: 0.4877386808395386
------------------------------------------------------------------------
```

Testing accuracies obtained:

- fold 1 : 0.748133999412
- fold 2 : 0.852353549722
- fold 3 : 0.821079727776
- fold 4 : 0.829497354497
- fold 5 : 0.845548045842
- fold 6 : 0.829829856331
- fold 7 : 0.828588102143
- fold 8 : 0.830505177912
- fold 9 : 0.815827613633
- fold 10 : 0.856858694369

Mean (Average)	0.82582221216374
Median	0.82966360541416
Range	0.10872469495674
Geometric Mean	0.82529943796934

Largest	0.8568586943690237
Smallest	0.7481339994122834

Testing AUCs obtained:

- fold 1 : 0.749090909091
- fold 2 : 0.850909090909
- fold 3 : 0.820606060606
- fold 4 : 0.831515151515
- fold 5 : 0.844848484848
- fold 6 : 0.833939393939
- fold 7 : 0.831310679612
- fold 8 : 0.827669902913
- fold 9 : 0.816747572816
- fold 10 : 0.853155339806

Mean (Average)	0.82597925860547
Median	0.8314129155634
Range	0.10406443071492
Geometric Mean	0.82548129825759
Largest	0.8531553398058253
Smallest	0.7490909090909091

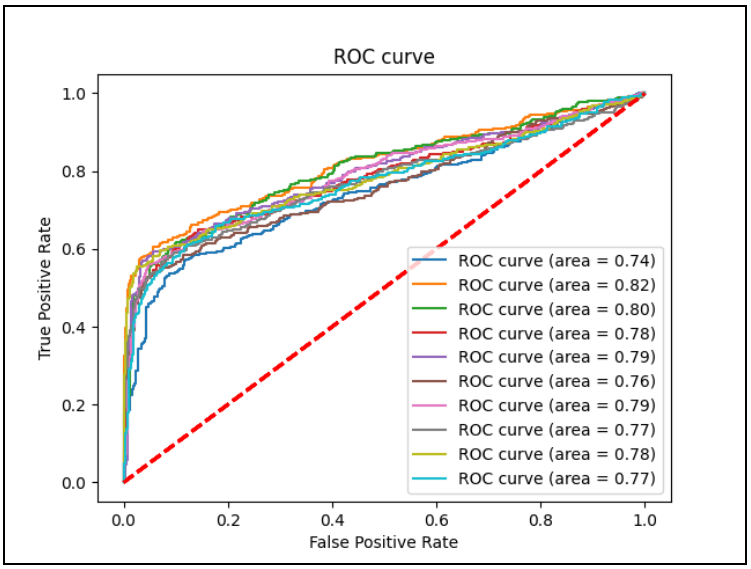


Fig 6.3 ROC Curve of 11 folds

6.4 Monte Carlo Simulations:

```
Avg Test ACC: 0.99509090909091
Avg Test AUC: 0.80331029128835
Time taken: 92 minutes
```

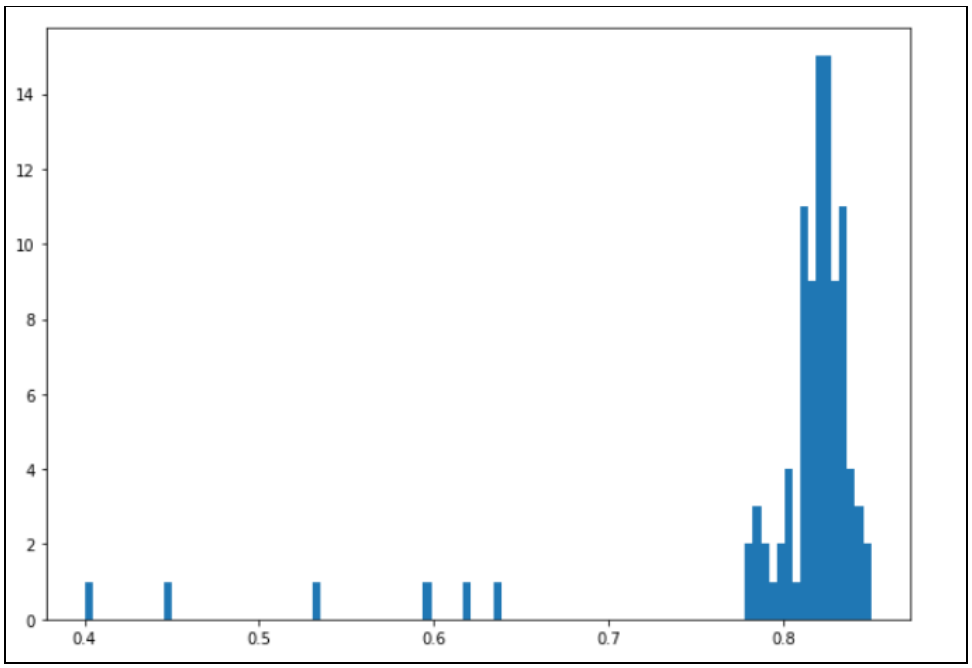


Fig 6.4 Monte Carlo Simulation

6.5 X-AI Results :

The 4 features are :
Feature 0 = “A”
Feature 1 = “C”
Feature 2 = “G”
Feature 3 = “T”

1. **Summary plots :** It shows the positive and negative relationships of the predictors with the target variable. The horizontal location shows whether the effect of that value is associated with a higher or lower prediction. Variable is high (in red) or low (in blue)

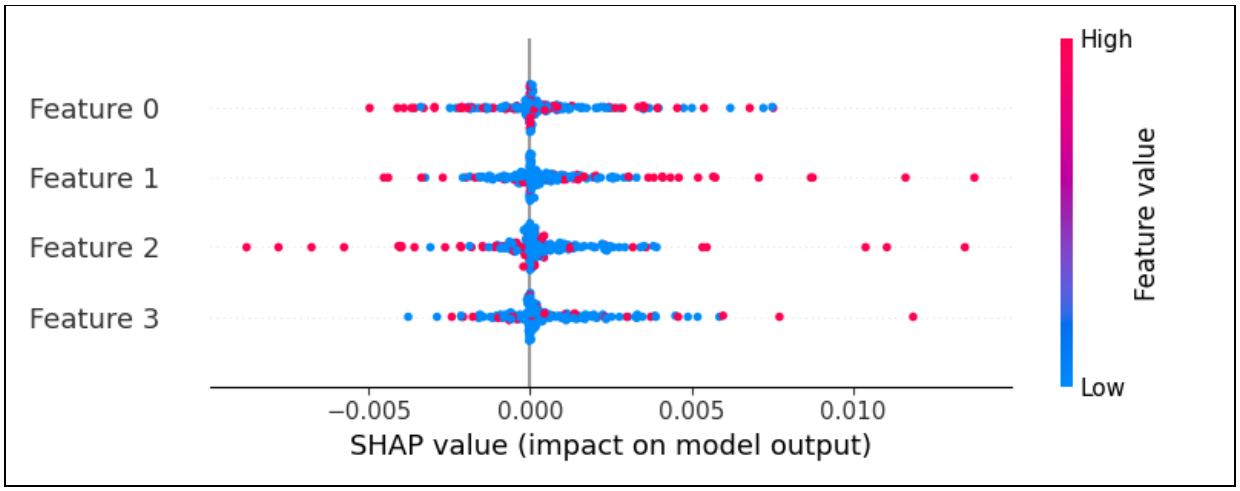


Fig 6.5.1 XAI Summary Plot

2. **Decision Plot :**

The x-axis represents the model's output. The y-axis lists the model's features. The features are ordered by descending importance.

For 10 sequences:

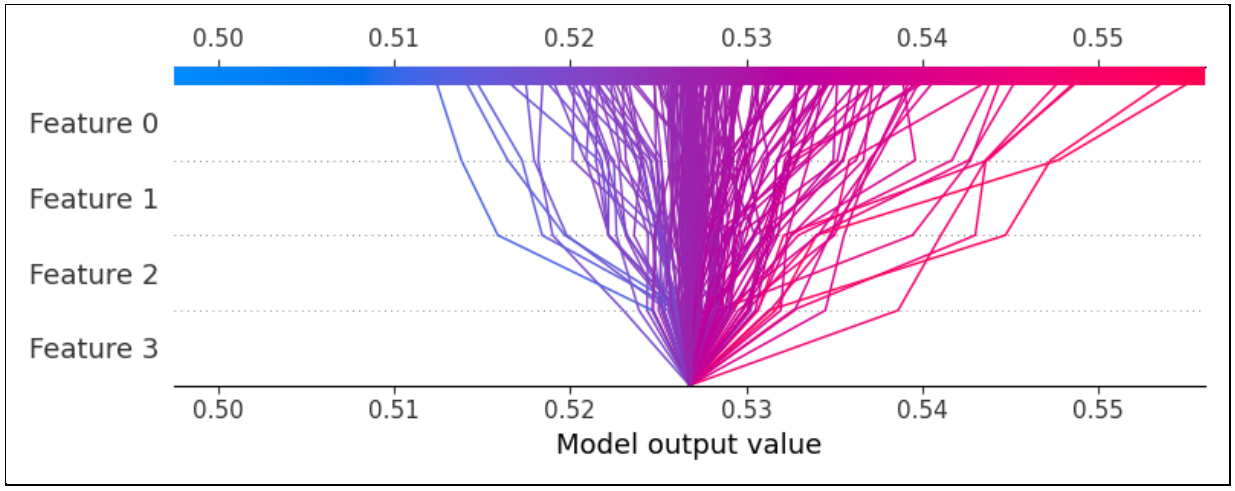


Fig 6.5.2 XAI Decision Plot for 10 sequences

For 1st sequence:

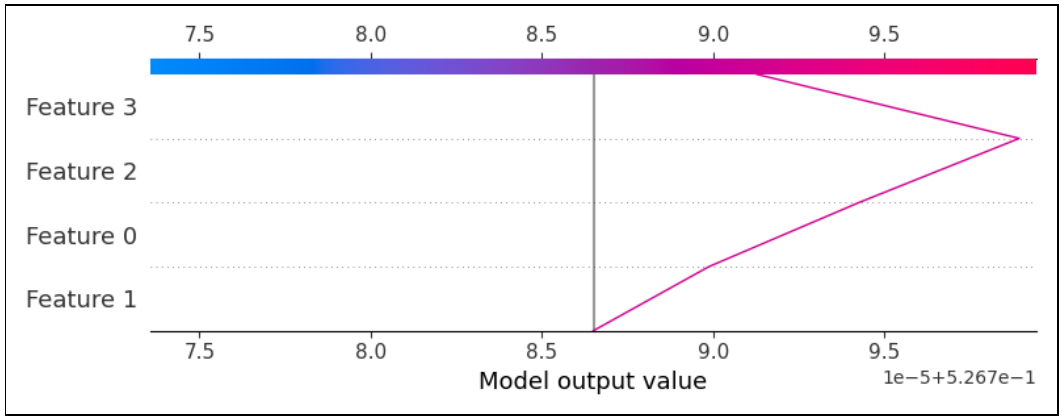


Fig 6.5.3 XAI Decision Plot for the 1st sequences

3. **Individual Force Plots:**

This plot was used to visualize the given SHAP values with an additive force layout. It indicates how features contributed to the model's prediction for a specific observation.
For the 1st observation:

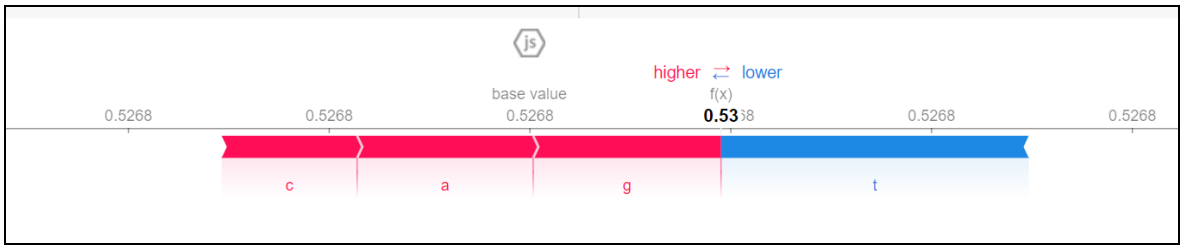


Fig 6.5.4 XAI Individual Force Plot

The base value is the value that would be predicted if we did not know any features for the current output. The prediction higher are shown in red, and those pushing the prediction lower are in blue. So here the output value which is the prediction for that observation is 0.53. A higher-than-the-average 'C' pushes the prediction to the right.

4. Collective Force Plots:

Effect of A:

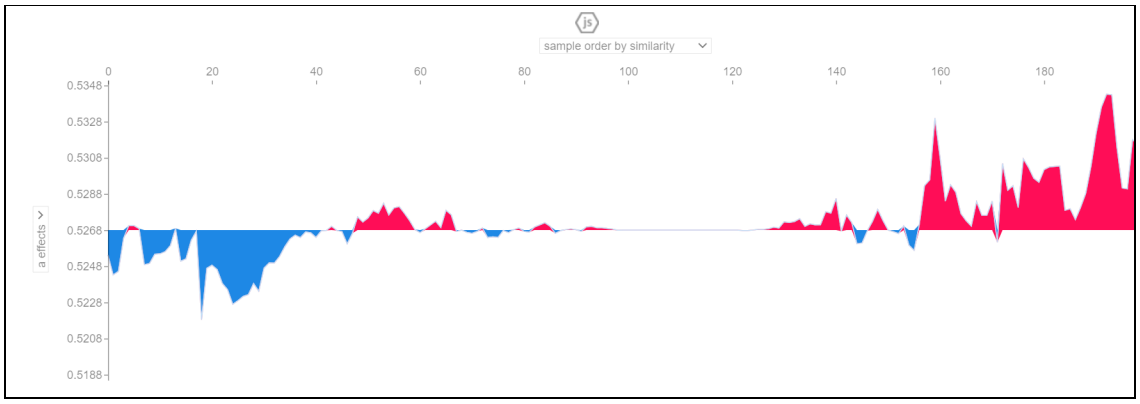


Fig 6.5.5 XAI Force Plot - Effect of A

Effect of C:

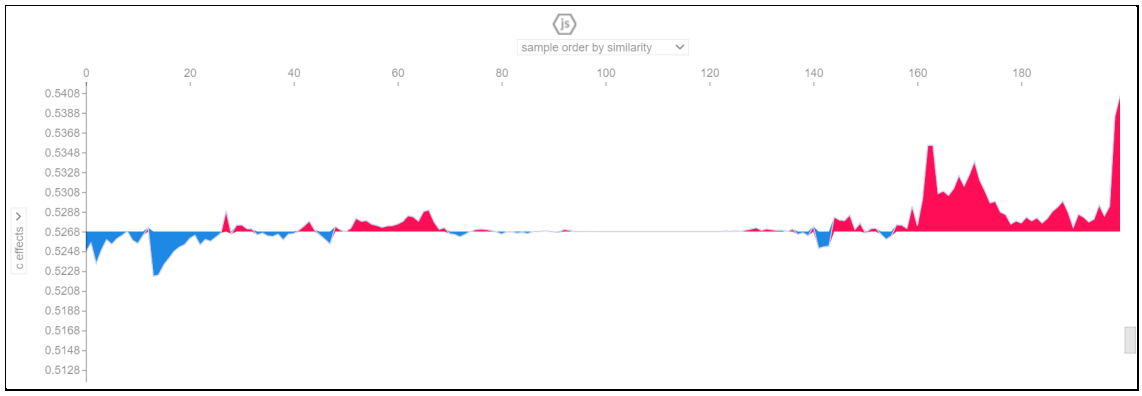


Fig 6.5.6 XAI Force Plot - Effect of C

Effect of G:

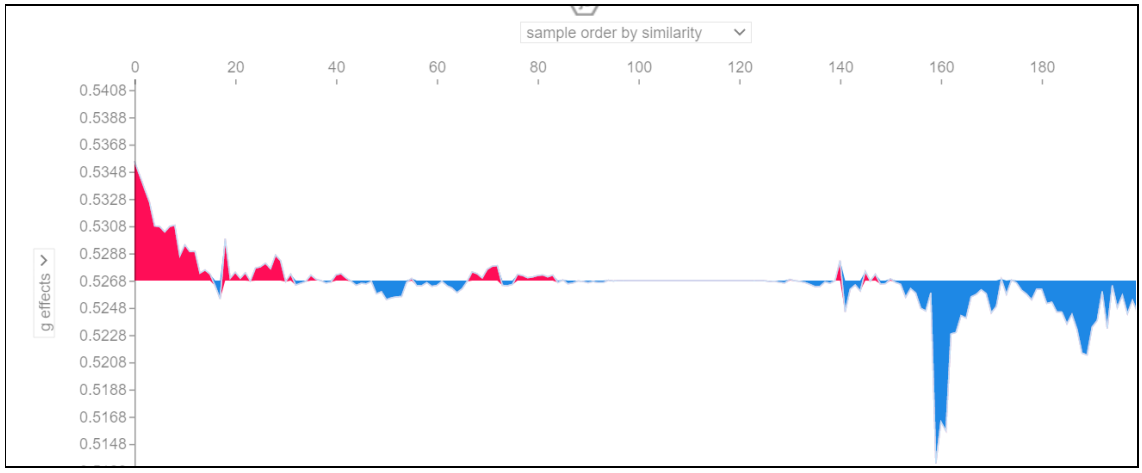


Fig 6.5.7 XAI Force Plot - Effect of G

Effect of T:

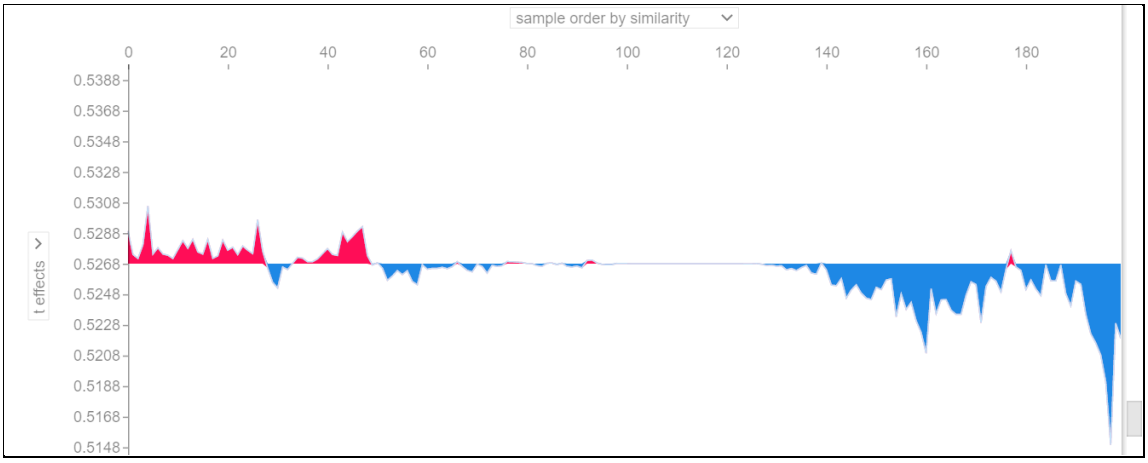


Fig 6.5.8 XAI Force Plot - Effect of T

07 Conclusion and Future Work

Deep learning models have outperformed traditional statistical models in extracting features from the genome sequences. They are computationally faster to execute. This generic pipeline can be open-sourced and useful for multiple uses under the domain of precision medicine. We explored what parts of the DNA are critical for the transformation of the cell into a diseased cell using Explainable AI. This model can be used in areas of precision medicine. Right now precision medicine is only bioinformatics & biological approaches but this model is faster & cheaper than traditional sequencing. It makes the use of explainable ai to understand which parts of the DNA are critical for the mutation occourance and why

The outputs from the explainable AI tell about the WHY & WHAT, then we need to solve the question of HOW!

“ How does that particular region affect the gene regulation?”

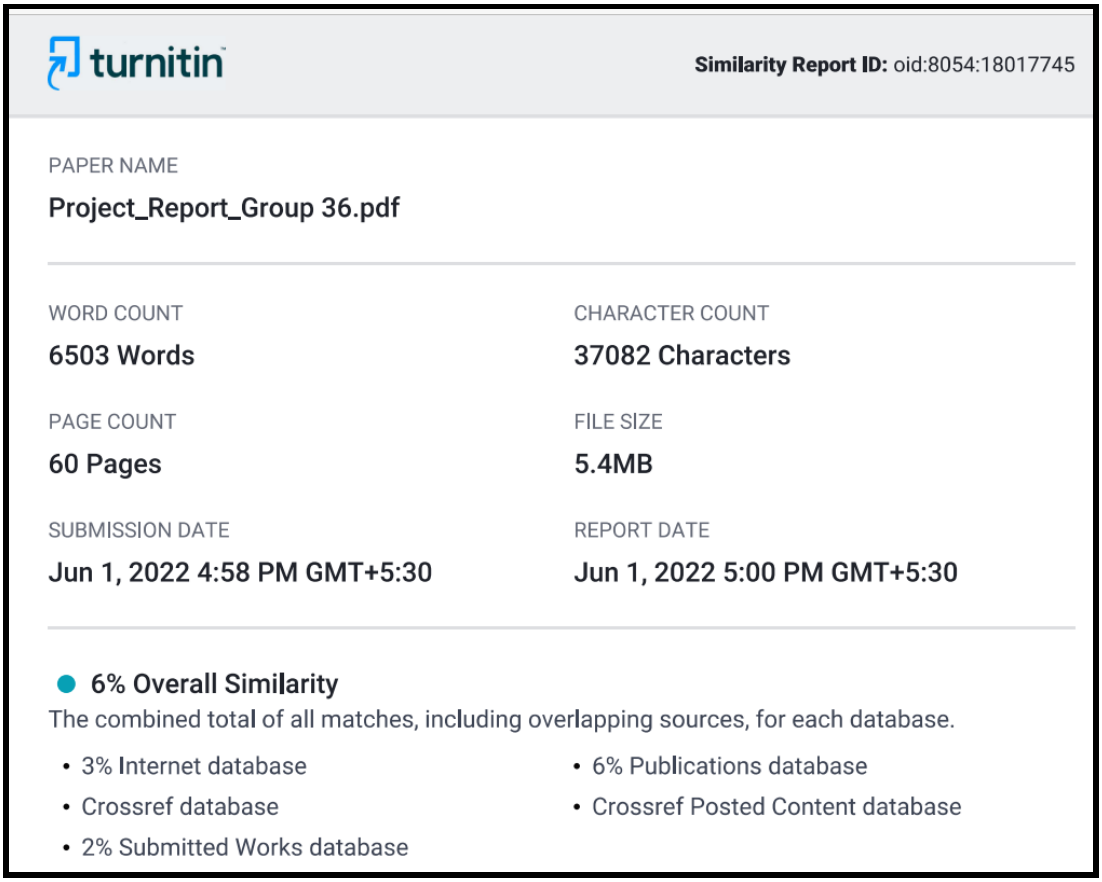
Since, it’s expensive, time consuming and not easy to generate high throughput data, we wish to build a generative model that will generate its own data and feed it to the model.

Appendix A: Details of paper publication

We have submitted a research paper based on our work to the Seventh International Conference on Data Management, Analytics and Innovation - ICDMAI 2023, being held during 20-22 January, 2023 in Pune India.

We are awaiting the results regarding the same. If accepted, the research paper will be published in the Springer series "Advances in Intelligent Systems and Computing" . The books of this series are submitted to ISI Proceedings, EICompendex, DBLP, SCOPUS, Google Scholar and Springerlink

Appendix B: Plagiarism Report



turnitin

Similarity Report ID: oid:8054:18017745

PAPER NAME

Project_Report_Group 36.pdf

WORD COUNT	CHARACTER COUNT
6503 Words	37082 Characters
PAGE COUNT	FILE SIZE
60 Pages	5.4MB
SUBMISSION DATE	REPORT DATE
Jun 1, 2022 4:58 PM GMT+5:30	Jun 1, 2022 5:00 PM GMT+5:30

● 6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 3% Internet database
- Crossref database
- 2% Submitted Works database
- 6% Publications database
- Crossref Posted Content database

Appendix C: User Manual

The full source code of this project is available as two GitHub repositories.

A) Markov Models

https://github.com/radhikasethi2011/btechproj

B) Deep Learning

https://github.com/radhikasethi2011/SilencerEnhancerPredict

REFERENCES

1. Kempfer, R., Pombo, A. Methods for mapping 3D chromosome architecture. Nat Rev Genet 21, 207–226 (2020). https://doi.org/10.1038/s41576-019-0195-2

2. Han, J., Zhang, Z. & Wang, K. 3C and 3C-based techniques: the powerful tools for spatial genome organization deciphering. Mol Cytogenet 11, 21 (2018). https://doi.org/10.1186/s13039-018-0368-2

3. Vietri Rudan M, Barrington C, Henderson S, Ernst C, Odom DT, Tanay A, Hadjur S. Comparative Hi-C reveals that CTCF underlies evolution of chromosomal domain architecture. Cell Rep. 2015 Mar 3;10(8):1297-309. DOI: 10.1016/j.celrep.2015.02.004. Epub 2015 Feb 26. PMID: 25732821; PMCID: PMC4542312.

4. Lanchantin, Jack, Ritambhara Singh, Zeming Lin, and Yanjun Qi. "Deep motif: Visualizing genomic sequence classifications." arXiv preprint arXiv:1605.01133 (2016).

5. de Wit E, de Laat W. A decade of 3C technologies: insights into nuclear organization. Genes Dev. 2012 Jan 1;26(1):11-24. doi: 10.1101/gad.179804.111. PMID: 22215806; PMCID: PMC3258961.

6. Alipanahi, B., Delong, A., Weirauch, M. et al. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat Biotechnol 33, 831–838 (2015). https://doi.org/10.1038/nbt.3300

7. U of T news (2016, May 17). DeepBind crunches data to find patterns behind origins of disease. https://www.utoronto.ca/news/deepbind-crunches-data

8. Crowley C., Yang Y., Qiu Y., Hu B., Abnousi A., Lipiński J., Plewczyński D., Wu D., Won H., Ren B., et al. FIREcaller: Detecting frequently interacting regions from Hi-C data. Comput. Struct. Biotechnol. J. 2021;19:355–362. doi: 10.1016/j.csbj.2020.12.026.

9. Kenlon, Seth (2021, August 12). A guide to the Linux terminal for beginners. https://opensource.com/article/21/8/linux-terminal

10. Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17). JMLR.org, 3145–3153.

11. Mohammadreza (Reza), Salehi (2020, Jan 19). A Review of Different Interpretation Methods (Part 2: Pixel-wise Decomposition, DeepLift LIME) https://mrsalehi.medium.com/a-review-of-different-interpretation-methods-in-deep-learning-part-2-input-gradient-layerwise-e077609b6377

12. Fernando, Zeon Trevor, Jaspreet Singh, and Avishek Anand. "A study on the Interpretability of Neural Retrieval Models using DeepSHAP." Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019.
https://arxiv.org/abs/1907.06484

13. Wang, Q., Sun, Q., Czajkowsky, D.M. et al. Sub-kb Hi-C in D. melanogaster reveals conserved characteristics of TADs between insect and mammalian cells. Nat Commun 9, 188 (2018). https://doi.org/10.1038/s41467-017-02526-9

14. Yu, H., Samuels, D.C., Zhao, Yy. et al. Architectures and accuracy of artificial neural network for disease classification from omics data. BMC Genomics 20, 167 (2019). https://doi.org/10.1186/s12864-019-5546-z

15. Kim, YG., Kim, S., Cho, C.E. et al. Effectiveness of transfer learning for enhancing tumor classification with a convolutional neural network on frozen sections. Sci Rep 10, 21899 (2020). https://doi.org/10.1038/s41598-020-78129-0

16. Matteo Vietri Rudan, Christopher Barrington, Stephen Henderson, Christina Ernst, Duncan T. Odom, Amos Tanay, Suzana Hadjur, Comparative Hi-C Reveals that CTCF Underlies Evolution of Chromosomal Domain Architecture, Cell Reports, Volume 10, Issue 8, 2015, Pages 1297-1309, ISSN 2211-1247, https://doi.org/10.1016/j.celrep.2015.02.004.

17. Dixon, J., Selvaraj, S., Yue, F. et al. Topological domains in mammalian genomes identified by analysis of chromatin interactions. Nature 485, 376–380 (2012). https://doi.org/10.1038/nature11082

18. Barutcu AR, Maass PG, Lewandowski JP, Weiner CL, Rinn JL. A TAD boundary is preserved upon deletion of the CTCF-rich Firre locus. Nat Commun. 2018 Apr 13;9(1):1444. doi: 10.1038/s41467-018-03614-0. PMID: 29654311; PMCID: PMC5899154.

19. Chyr J, Zhang Z, Chen X, Zhou X. PredTAD: A machine learning framework that models 3D chromatin organization alterations leading to oncogene dysregulation in breast cancer cell lines. Computational and structural biotechnology journal. 2021;19:2870–80. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8965109/

20. Alexander Gulliver Bjørnholt Grønning, Thomas Koed Doktor, Simon Jonas Larsen, Ulrika Simone Spangsberg Petersen, Lise Lolle Holm, Gitte Hoffmann Bruun, Michael Birkerod Hansen, Anne-Mette Hartung, Jan Baumbach, Brage Storstein Andresen, DeepCLIP: predicting the effect of mutations on protein–RNA binding with deep learning, Nucleic Acids Research, Volume 48, Issue 13, 27 July 2020, Pages 7099–7118, https://doi.org/10.1093/nar/gkaa530

21. Spiro C. Stilianoudakis, Maggie A. Marshall, Mikhail G. Dozmorov. preciseTAD: A transfer learning framework for 3D domain boundary prediction at base-pair resolution
doi: https://doi.org/10.1101/2020.09.03.282186

22. López-García G, Jerez JM, Franco L, Veredas FJ (2020) Transfer learning with convolutional neural networks for cancer survival prediction using gene-expression data. PLOS ONE 15(3): e0230536. https://doi.org/10.1371/journal.pone.0230536

23. R. K. Sevakula, V. Singh, N. K. Verma, C. Kumar and Y. Cui, "Transfer Learning for Molecular Cancer Classification Using Deep Neural Networks," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 16, no. 6, pp. 2089-2100, 1 Nov.-Dec. 2019, doi: 10.1109/TCBB.2018.2822803.