Development of Machine Learning Algorithm for Proteomic Analysis of Rheumatoid Arthritis (RA) Disease

Arun Kumar I A[1], K.B.Ramesh[2], Vidya Niranjan[3], B.P.Mallikarjunaswamy[4]

PG Student[BMSPI], Dept. of EIE, R V College of Engineering, Bengaluru, India[1]

Associate Professor, Dept. of EIE, R V College of Engineering, Bengaluru, India[2]

Professor and HoD, Dept. of BT, R V College of Engineering, Bengaluru, India[3]

Professor, Dept. of CSE, Siddhartha Institute of Technology, Tumkur, India[4]

ABSTRACT: Bioinformatics is an interdisciplinary field of science combines biology, computer and information engineering, mathematics and statistics to analyze and interpret the biological data like DNA, RNA and Protein set and it has been used for insilico analysis of biological queries using mathematical and statistical techniques. Proteomics is the part of bioinformatics which deals with the proteins which are expressed by the genomes. Rheumatoid Arthritis is a chronic inflammatory disorder that can affect joints. In some cases, the condition can damage a wide variety of body systems, including the skin, eyes, lungs and blood vessels. The statistics reveals that the patients who suffer from Rheumatoid Arthritis (RA) are 41 for every 1,00,000 members. Machine learning is a field of computer science which deals with the development of algorithms for performing prediction analysis based on data, has a number of emerging applications in the field of bioinformatics. Machine learning techniques have been applied to analyze proteomics and for classification of unknown samples and identification of genes relevant to the disease state. K-means clustering is one of the simplest and popular unsupervised machine learning algorithms. The correlation formula is used for finding out the risk genes and magnitude of the risk genes are calculated by Relative genetic risk factor.In this paper, it is explained the characteristics of Rheumatoid arthritis disease and basics of proteomics and discussed that the applications of machine learning techniques to analyze the protein data set of Rheumatoid arthritis (RA) disease. In the proposed project work, the K-means clustering algorithm has been implemented in python programming language. The implemented clustering algorithm performs the classification operation based on the threshold value so that it differentiates the query protein data sequence whether it belongs to normal or abnormal category. The motif patterns are stored in the database tables are created using sqlite3 and graphical user interface (GUI's) are provided in the front end applications which are developed using python language. The techniques used in the literature are studied and the results are validated in the proposed work

KEYWORDS: RA,CCA,CED,EED,sqlite3,$L^2(R)$,$L^2(R^2)$

I.INTRODUCTION

The early identification of the RA is still underway. Twenty two patients were included in the study of RA. Among these twenty two patients, fifteen were comprised of late RA. Eighteen patients without inflammatory arthropathy formed the control group [1]. Analysis of B-lymphocytes, myeloid was made using cytofluorimeter and the FacsDiva software. The plasmacytoid DCs was statistically significant overhelmed with early and late RA compared with the control group. The difference was found in the percent of cells with phenotyped B-lymphocytes. The dynamics was detected by the decrease in the percentage of plasmacytoid dendritic cells and B-lymphocytes in patients with the group of early RA. The data demonstrates the difference in the pereipheralled blood DCs subtyped ratio in group with early and late RA compared with other patients [2]. These markers which are cellular can be used for early diagnosis, evaluation of the activity and the treatment effectiveness in patient with RA.

The data used for the analysis of RA are kept in the database. The database is a structure set of data held in a computer, especially one that is accessible in various ways. Here sqlite3 database is used. The connect inbuilt function used here is used to combine both the database and user queries. The inbuilt function cursor is used to mark the database, another inbuilt function called the execute, executes the user related queries [3]. The last inbuilt function commit used to combine the executed user related queries with the database. Here these database related functions are coded at the back end inorder to store the sequences and retrieve whenever needed at times[4].

Inorder to predict the RA, Machine learning concept in used. K-means clustering is the one of the method in machine learning which is an Unsupervised learning. Unsupervised learning is where you only have input data, say X and no corresponding output variables like Y [5]. The objective for unsupervised learning is to model the underlying structure or distribution in the data inorder to learn more about the data. These are called unsupervised learning because unlike supervised learning above there is no accurate answers and there is no teacher. Algorithm are left to their own devices to find and present the interesting structure in the data. Unsupervised learning problems can be further divided into clustering and association problems. Association rule learning problem is where you want to discover rules that describe large portions of your data, such as person who buy X also tend to buy Y. Clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by inducing behavior [6]. K-means algorithm is an algorithm to cluster n objects based on attributes into k partitions where k is less than n.It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of naturally occuring clusters in the data. It assumes that the object attributes form a vector space [7].An algorithm for partitioning (or clustering) N data points into K disjoint subsets and Sj containing data points so as to minimize the sum-of-squares criterion [8]. K-means clustering is an algorithm to classify or to group the objects based on features into K number of group [6]. The grouping is done by minimizing the sum of squares of distances between data and the corresponding centroidof the cluster [9]. Figure 1.1 shows the K-means clustering.

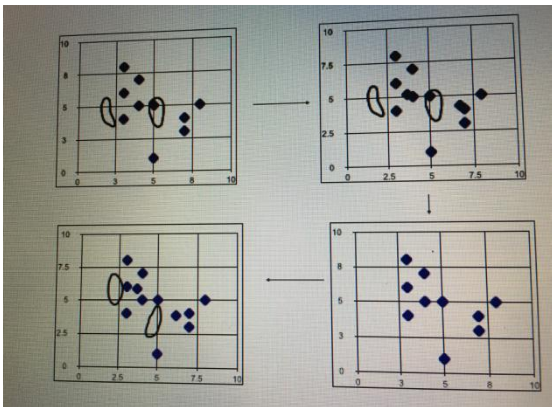


Figure 1.1: K-means clustering

II. LITERATURE SURVEY

The search for new biomarkers that will allow the diagnosis of arthritis in the early, pre-destructive phase of the disease is still underway. Twenty two patients with early rheumatoid arthritis (RA), duration of the disease up to 12 months) were included in the study. Fifteen patient patients comprised a group of late rheumatoid arthritis. Eighteen patients with OA and without inflammatory arthropathy had formed the control group. Analysis of the B-lymphocytes, myeloid, and plasmacytoid DCs count was carried out using a flow cytofluorimeter (BD FACSCantoII, USA) and FacsDiva software. The percent of plasmacytoid DCs was statistically significant predominated in the group of patients with early and late RA in comparison with the control group - 3.8 * 10% and 9.0 * 10% vs 1.0 * 10%, respectively (p = 0.0042). Furthermore, the difference was found in the percent of cells with the phenotype B-lymphocytes: 7.95 * 10% and 7.7 * 10% vs 3.3* 10%, respectively (p = 0.014). The dynamics was detected due to a decrease in the percent of plasmacytoid dendritic cells and B-lymphocytes in patients in the group with early rheumatoid arthritis: respectively 3.5 * 10% vs 0.6 * 10% (statistically not significant, p> 0.05), and 6.9 * 106 / l vs 4.9 * 10% (p= 0.045). These data demonstrate the difference in the peripheral blood DCs subtypes ratio in group with early and late RA compared with OA- patients. These cellular markers can be used for early diagnosis, evaluation the activity and treatment effectiveness in patient with RA.

DNA repeats are believed to play significant roles in genome evolution and manifestation of severe diseases. Many of the methods for finding repeated sequences use distances, similarities and consensus sequences to generate candidate sequences. This paper presents results obtained using a dedicated numerical representation with a mapping algorithm (using DNA distances and consensus types) and an in-memory dot-plot analysis combined with image processing techniques, to visual isolate the positions of DNA repeats with different lengths.

We have performed large-scale analysis of RNADNA contacts in chromatin based on our own experimental data for K562 human cells and GRID-Seq. We have devised RNA-Seq controls and validated the resulting interactions with known chromatin-associated RNAs. We propose a pipeline for the analysis of RNA-DNA interactions, a noise correction procedure and clustering by the distance preference approach. We have compared the obtained interactomes with DNA-DNA contacts (Hi-C) for the same cell line and report the association of contacts with active compartment A and boundaries of topologically associating domains (TADs). We have detected a fraction of chromatin-enriched RNAs and report their chromatin properties. We also describe classes of RNAs by their contact preferences with DNA. Optofluidic laser that has a single layer of DNA molecules on the ring resonator surface is proposed. A target DNA can be detected in truly digital manner only with a single pulse of laser excitation. B DNA structures have a great potential to form and influence various genomic processes including transcription. One of the mechanisms of transcription regulation is nucleosome positioning. Even though only B-DNA can be wrapped around a nucleosome, non-B DNA structures can compete with a nucleosome for a genomic location. Here we used permanganate/S1 nuclease footprinting data on non-B DNA structures, such as Z-DNA, H-DNA, G-quadruplexes and stress-induced duplex destabilization (SIDD) sites, together with MNase-seq data on nucleosome positioning in the mouse genome. We found three types of patterns of nucleosome positioning around non-B DNA structures: a structure is surrounded by nucleosomes from both sides, from one side, or nucleosome free region.

Machine learning models based on random forest and XGBoost algorithms were constructed to recognize DNA regions of 1kB length containing a particular pattern of nucleosome positioning for four types of DNA structures (Z-DNA, H-DNA, G-quadruplexes and SIDD sites) based on statistics of di- and tri-nucleotides. The best performance (94% of accuracy) was reached for Gquadruplexes while for other types of structures the accuracy was under 70%. We conclude that 1kB regions containing G quadruplexes have distinct compositional properties, and this fact points to preferential locations of such pattern in the genome and requires further investigation. Gene ontology analysis revealed that the genes intersecting with the discovered patterns are enriched in channel and transmembrane activity, transcription factor and receptor binding. The direction for further research is to study the distribution of the discovered patterns in different tissues to identify well-positioned and dynamic nucleosomes and reveal genes, regulated via DNA structures and nucleosome positioning.

III.DESIGN AND IMPLEMENTATION

The front end coding flowchart is shown in Figure 2.1 for the front end coding in Figure 2.2 where the user defined functions are used for the buttons for create, read, update, delete operations. The inbuilt functions are used for the labels and the frames. The labels are for loading the name, pattern and addresses, the frames are of two kinds, mainframe and the subframe [10].

The back end coding flowchart is shown in Figure 2.3 for the back end coding in Figure 2.4 where the inbuilt functions are used for connecting the user related queries with the database sqlite3, keeping the mark on the database the function cursor is used [11]. The user related queries at last are executed and commited with the database.

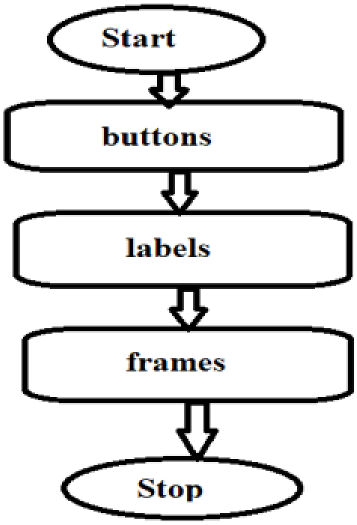
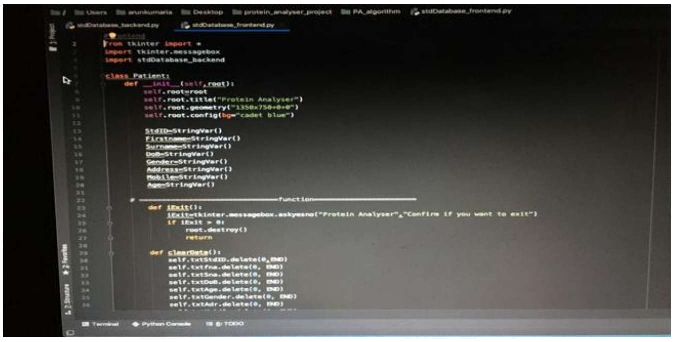


Figure 2.1: Flowchart of front end coding



/ Users arunkumaria Desktop protein_analyser_project PA_algorithm stdDatabase_frontend.py
stdDatabase_backend.py
stdDatabase_frontend.py
Terminal
Python Console
6: TODO

```
#frontend
from tkinter import *
import tkinter.messagebox
import stdDatabase_backend

class Patient:
    def __init__(self,root):
        self.root=root
        self.root.title("Protein Analyser")
        self.root.geometry("1350x750+0+0")
        self.root.config(bg="cadet blue")

        StdID=StringVar()
        Firstname=StringVar()
        Surname=StringVar()
        DoB=StringVar()
        Gender=StringVar()
        Address=StringVar()
        Mobile=StringVar()
        Age=StringVar()

      # ----------------------------------function----------------------------------
        def iExit():
            iExit=tkinter.messagebox.askyesno("Protein Analyser","Confirm if you want to exit")
            if iExit > 0:
                root.destroy()
                return

        def clearData():
            self.txtStdID.delete(0,END)
            self.txtfna.delete(0, END)
            self.txtSna.delete(0, END)
            self.txtDoB.delete(0, END)
            self.txtAge.delete(0, END)
            self.txtGender.delete(0, END)
            self.txtAdr.delete(0, END)
```

Figure 2.2: front end coding

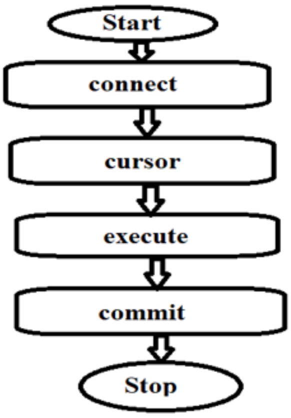
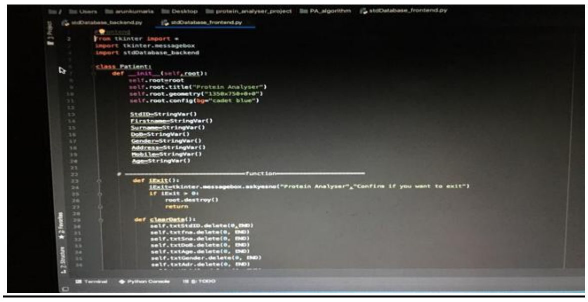


Figure 2.3: Flowchart of back end coding



/ Users arunkumaria Desktop protein_analyser_project PA_algorithm stdDatabase_frontend.py
stdDatabase_backend.py
stdDatabase_frontend.py
Terminal
Python Console
6: TODO

```
#frontend
from tkinter import *
import tkinter.messagebox
import stdDatabase_backend

class Patient:
    def __init__(self,root):
        self.root=root
        self.root.title("Protein Analyser")
        self.root.geometry("1350x750+0+0")
        self.root.config(bg="cadet blue")

        StdID=StringVar()
        Firstname=StringVar()
        Surname=StringVar()
        DoB=StringVar()
        Gender=StringVar()
        Address=StringVar()
        Mobile=StringVar()
        Age=StringVar()

      # ----------------------------------function----------------------------------
        def iExit():
            iExit=tkinter.messagebox.askyesno("Protein Analyser","Confirm if you want to exit")
            if iExit > 0:
                root.destroy()
                return

        def clearData():
            self.txtStdID.delete(0,END)
            self.txtfna.delete(0, END)
            self.txtSna.delete(0, END)
            self.txtDoB.delete(0, END)
            self.txtAge.delete(0, END)
            self.txtGender.delete(0, END)
            self.txtAdr.delete(0, END)
```

Figure 2.4: back end coding

IV.METHODOLOGY

From the Figure 3.1, the methodology using the machine learning concept of K-means clustering is as follows-:
a)At first begin with decision where value of k is equal to number of clusters.
b) Put the initial partition which classifies the data into k clusters [12]. Training samples can be assigned randomly or systematically as the following-:
i) First k training sample can be taken as single element clusters
ii) Assign to the nearest centroid, the remaining training samplerecomputes the centroid of the gaining cluster after each assignment.
c) Taking the each sample in sequence, compute the centroid of each clusters from its distance. Update the centroid of the cluster gaining the new sample, if the sample is not in the cluster with related and close to centroid. Update the centroid of cluster gaining new sample with cluster losing the sample after switching this sample to the cluster [13].
d) Step(c) is repeated until convergence is achieved and until a pass through training sample causes no new assignments [14].

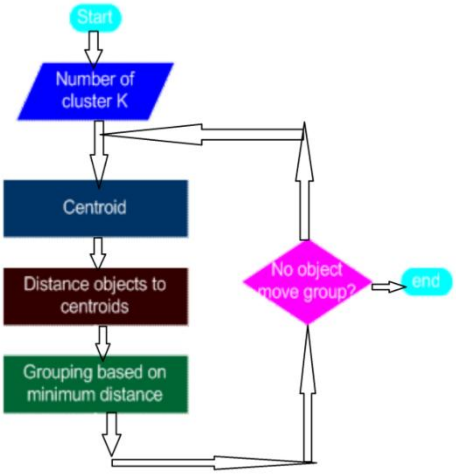

Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
end

Figure 3.1: Methodology using K-means clustering

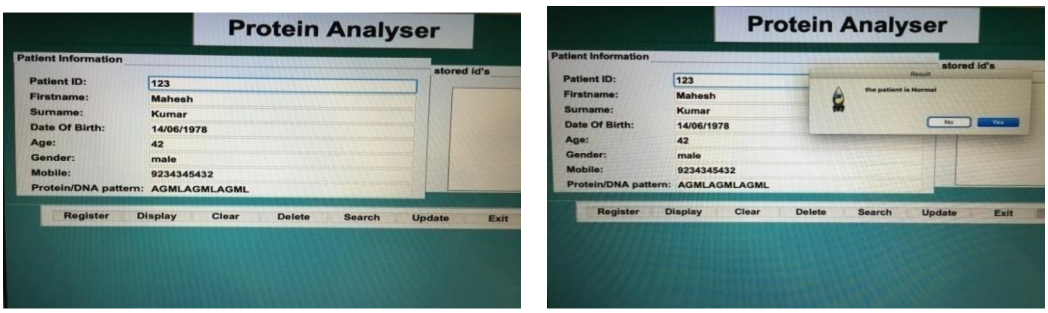
V.FUNCTIONAL BLOCK DIAGRAM AND FLOWCHART

Figure 2.1 consists of the front end functional block diagram where the buttons,labels and frames are well set. The user defined functions are set for the buttons to do the create, read, update, delete operations [15]. The labels are used for determining the Names, Patterns or signals and addresses. The frames are of two kinds, mainframe and the subframe. Inbuilt functions are used as the functional properties of labels and frames.
Figure 2.3 consists of the back end functional block diagram where the database is well set. It consists of the connect, cursor, execute and commit inbuilt functions where the connect combines the user related queries with the data, cursor keeps a mark on the database, execute which executes the user related queries and commits it to the database [16].

VI.RESULT AND DISCUSSION

The standard data is compared with the obtained calculated data to get the graph of following pattern. The importance of machine learning in the disease identification is mainly highlighted. For identifying the disease from coding using Python [17]. We have the different sorts of data where the presence of cytosine is seen in which we are able to get the result of cytosine replaced by thymine instantly represented by the diagram using the K-means clustering where in case the missed alleles are found [18]. Plenty of diseases like cancer,sickle-cell anaemia, beta-thalassaemia and cystic fibrosis are identified [19].

The datasets are categorized as two types, the known datasets or signals and unknown datasets or signals. From Figure 5.1, the patient of age 42 is normal. Since the essential amino acid and non essential amino acid is in equal quantity in the body which is the known dataset [20].



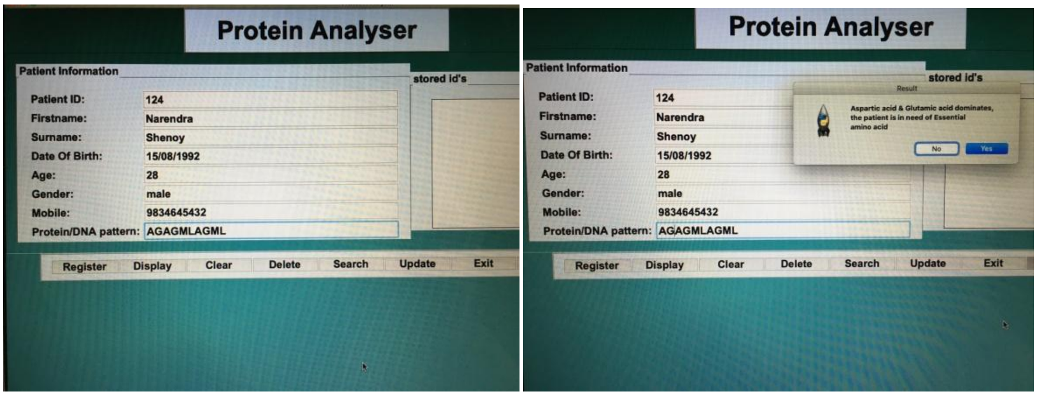
Patient ID:	123
Firstname:	Mahesh
Surname:	Kumar
Date Of Birth:	14/06/1978
Age:	42
Gender:	male
Mobile:	9234345432
Protein/DNA pattern:	AGMLAGMLAGML

Result

the patient is Normal

Figure 5.1: Sample For Getting Normal Result

From Figure 5.2,the patient of age 28 is still normal and in need of essential amino acid to the body which is also a product of known dataset [21].



Patient Information

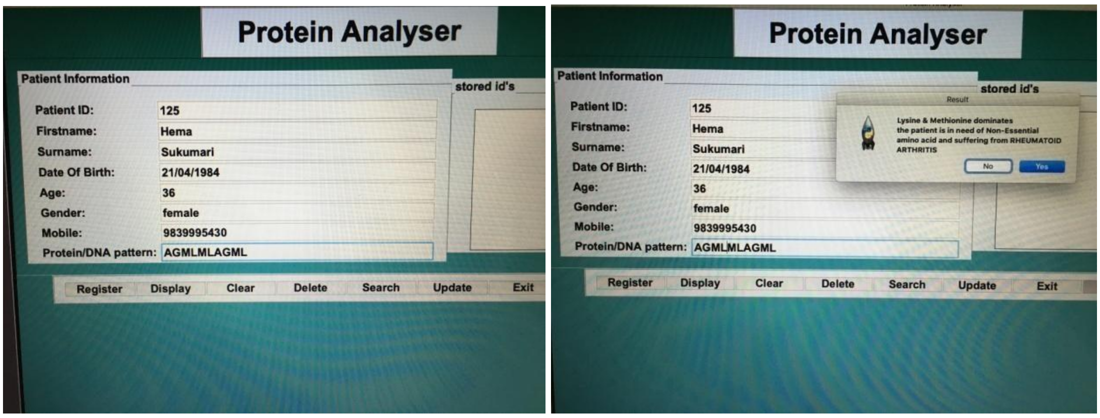
Patient ID: 124
Firstname: Narendra
Surname: Shenoy
Date Of Birth: 15/08/1992
Age: 28
Gender: male
Mobile: 9834645432
Protein/DNA pattern: AGAGMLAGML

Result

Aspartic acid & Glutamic acid dominates,
the patient is in need of Essential
amino acid

Figure 5.2: Sample for the need of essential amino acid

From Figure 5.3, The patient of age 36 is having more essential amino acid in the body and is said to be suffering from Rheumatoid Arthritis which is also a product of known dataset [22].



Patient Information

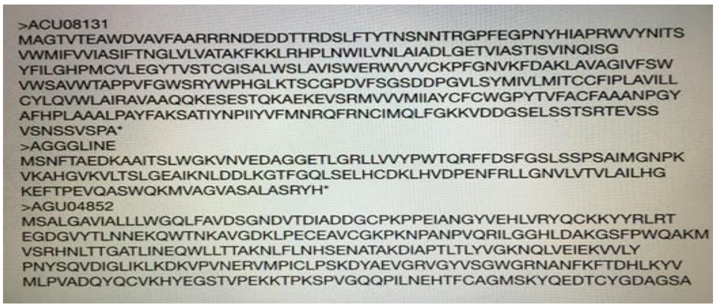
Patient ID: 125
Firstname: Hema
Surname: Sukumari
Date Of Birth: 21/04/1984
Age: 36
Gender: female
Mobile: 9839995430
Protein/DNA pattern: AGMLMLAGML

Result

Lysine & Methionine dominates
the patient is in need of Non-Essential
amino acid and suffering from RHEUMATOID
ARTHRITIS

Figure 5.3: Sample for getting Rheumatoid Arthritis

The Figure 5.4 shows how the unknown dataset looks like and is the dataset from the person suffering from Rheumatoid Arthritis. The translated form of this dataset is evaluated in Figure 5.5.



```
>ACU08131
MAGTVTEAWDVAVFAARRRNDEDDTTRDSLFTYTNSNNTRGPFEGPNYHIAPRWVYNITS
VWMIFVVIASIFTNGLVLVATAKFKKLRHPLNWILVNLAIADLGETVIASTISVINQISG
YFILGHPMCVLEGYTVSTCGISALWSLAVISWERWVVVCKPFGNVKFDAKLAVAGIVFSW
VWSAVWTAPPVFGWSRYWPHGLKTSCGPDVFSGSDDPGVLSYMIVLMITCCFIPLAVILL
CYLQVWLAIRAVAAQQKESESTQKAEKEVSRMVVVMIIAYCFCWGPYTVFACFAAANPGY
AFHPLAAALPAYFAKSATIYNPIIYVFMNRQFRNCIMQLFGKKVDDGSELSSTSRTEVSS
VSNSSVSPA*
>AGGGLINE
MSNFTAEDKAAITSLWGKVNVEDAGGETLGRLLVVYPWTQRFFDSFGSLSSPSAIMGNPK
VKAHGVKVLTSLGEAIKNLDDLKGTFGQLSELHCDKLHVDPENFRLLGNVLVTVLAILHG
KEFTPEVQASWQKMVAGVASALASRYH*
>AGU04852
MSALGAVIALLLWGQLFAVDSGNDVTDIADDGCPKPPEIANGYVEHLVRYQCKKYYRLRT
EGDGVYTLNNEKQWTNKAVGDKLPECEAVCGKPKNPANPVQRILGGHLDAKGSFPWQAKM
VSRHNLTTGATLINEQWLLTTAKNLFLNHSENATAKDIAPTLTLYVGKNQLVEIEKVVLY
PNYSQVDIGLIKLKDKVPVNERVMPICLPSKDYAEVGRVGYVSGWGRNANFKFTDHLKYV
MLPVADQYQCVKHYEGSTVPEKKTPKSPVGQQPILNEHTFCAGMSKYQEDTCYGDAGSA
```

Figure 5.4: Unknown dataset of a patient suffering from Rheumatoid Arthritis

Figure 5.5 shows the distribution of adenine, guanine, thymine, cytosine for a given set of data.
Yellow colored data shows the replacement of the alleles [23], i.e.., thymine and cytosine.
Red colored data is the centroid point.
Blue colored data is the distribution of adenine & guanine.

	X(standard in counts)	Y(calculated in counts)
Adnine	736902	245056
Guanine	763077	252562
Thymine	702601	253079
Cytosine	689613	248551

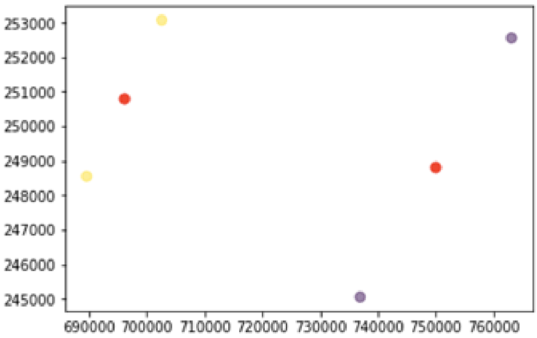


Figure 5.5: distribution of adenine, guanine, thymine, cytosine

VII. CONCLUSION

In this paper, the importance of machine learning in the disease identification is mainly highlighted. For identifying the disease here from coding using Python we have the different sorts of data where the presence of cytosine is seen in which we are able to get the result of cytosine replaced by thymine instantly represented by the diagram using the K-means clustering where in case the missed alleles are found. Plenty of diseases like cancer,sickle-cell anaemia, beta-thalassaemia and cystic fibrosis are identified where here the Rheumatois Arthritis disease is more stressed.

For the future enhancement it can be made to directly analyze the graph with different diseases by assigning the threshold which help to segregate the diseasesmore clearly.

REFERENCES

1. X. Li, B. Liao, L. Cai, Z. Cao, and W. Zhu, “Informative SNPs selection based on two-locus and multilocus linkage disequilibrium: Criteria of max-correlation and min-redundancy,” *IEEE/ ACM Trans. Comput. Biol. Bioinformat.*, vol. 10,(3), pp. 688–695, 2013.
2. P.I.DeBakker,R.R.Graham,D.Altshuler, B. E.Henderson, and C. A.Haiman,“TransferabilityoftagSNPstocapturecommongenetic variation in DNA repair genes across multiple populations,” in Proc.Pac.Symp.Biocomput.,2006,vol.11,pp.478–486.
3. C. S. Carlson, M. A. Eberle, M. J. Rieder, Q. Yi, L. Kruglyak, and D. A. Nickerson, “Selecting a maximally informative set of single nucleotide polymorphisms for association analyses using linkage disequilibrium,” Amer. J. Human Genetics, vol. 74, no. 1, pp. 106– 120, 2004.
4. R. C. Hardison and G. A. Blobel, “GWAS to therapy by genome edits?” Science, vol. 342,(6155), pp. 206–207, 2013.
5. W. Yang and C. C. Gu, “A whole-genome simulator capable of modeling high-order epistasis for complex disease,” Genetic Epidemiol., vol. 37,(7), pp. 686–694, 2013.
6. Gyenesei, J. Moody, C. A. Semple, C. S. Haley, and W. H. Wei, “High-throughput analysis of epistasis in genome-wide association studies with BiForce,” Bioinformatics, vol. 28,(15), pp. 1957– 1964, 2012.
7. Liao, X. Li, W. Zhu, R. Li, and S. Wang, “Multiple ant colony algorithm method for selecting tag SNPs,” J. Biomed. Informat., vol. 45,(5), pp. 931–937, 2012.
8. K. Ting, W.T. Lin, and Y. T. Huang, “Multi-objective tag SNPs selection using evolutionary algorithms,” Bioinformatics, vol. 26,(11), pp. 1446–1452, 2010.
9. J. He and A. Zelikovsky, “Informative SNP selection methods based on SNP prediction,” *IEEE Trans. Nanobiosci.*, vol. 6,(1), pp. 60–67, Mar. 2007.
10. Z. Q. Liu and S. L. Lin, “Multilocus LD measure and tagging SNP selection with generalized mutual information,” Genetic Epidemiol., vol. 29,(4), pp. 353–364, 2005.

11. Alasdair Gilchrist, ”Industry 4.0”, IoT., vol. 29,(4), pp. 353–364, 2019.
12. Alp Ustundug and EmreCevikcan, ”Industry 4.0 managing the digital transformation”, cloud computing., vol. 44,(4), pp. 353–364, 2019.
13. Machine Learning, Tom Mitchell, McGraw-Hill International Editions
14. Digital Image Processing and Analysis-byB.Chanda and D.DuttaMajumdar.
15. H. Zha, C. Ding, M. Gu, X. He and H.D. Simon. "Spectral Relaxation for K-means Clustering", Neural Information Processing Systems vol.14 (NIPS 2001). pp. 1057-1064, Vancouver, Canada. Dec. 2001.
16. J. A. Hartigan (1975) "Clustering Algorithms". Wiley.
17. J. A. Hartigan and M. A. Wong (1979) "A K-Means Clustering Algorithm", Applied Statistics, Vol. 28, No. 1, p100-108.
18. D. Arthur, S. Vassilvitskii (2006): "How Slow is the k-means Method?,"
19. D. Arthur, S. Vassilvitskii: "k-means++ The Advantages of Careful Seeding" 2007 Symposium on Discrete Algorithms (SODA).
20. Jiawei Han and MichelineKamber, “Data Mining: Concepts and Techniques”, 2 edition (4 Jun 2006).
21. Gordon S. Linoff and Michael J. Berry, “Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management”, 3rd Edition edition (1 April 2011)
22. Vipin Kumar and Mahesh Joshi, “Tutorial on High Performance Data Mining ”, 1999Rakesh Agrawal, RamakrishnanSrikan, “Fast Algorithms for Mining Association Rules”, Proc VLDB, 1994.