Classification of Eukaryotes Based on Chaos Game Representation using Support Vector Machine

Anoop A Jose, Dr. Smitha Dharan, Joyal John, Shyma S. Nair
Dept. of Computer Science & Engineering
College of Engineering, Chengannur
Alappuzha, India

***Abstract*—Chaos Game Representation is a method of representing bio-sequences as unique images, which is very essential for pattern recognition applications. Some outstanding results were procured by the enormous studies happened in the field and still it is a fast growing research area. In this paper, we aim to produce a species classification system that categorizes the given mitochondrial DNA sequence into one of the seven classes of Eukaryotes. Earlier some classification systems using Artificial Neural Network, Hidden Markov Model were evolved. In this work, a more accurate classification system using Support Vector Machine is proposed.**

Keywords—Bio-sequence; Artificial Neural Network; Hidden Markov Model; Support Vector Machine

I. INTRODUCTION

Earlier there was a belief that everything in the world would be predetermined. There was no chance, no choice and no uncertainty. According to the theory, the future of any system can be predicted by the present state of the system. But as time went on, mathematicians and scientists encountered some very difficult problems to solve using this theory, some in fact were completely unsolvable. This leads to the development of a new field of physics known as chaotic dynamics or simply chaos. The chaos theory deals with nonlinear things that are effectively impossible to predict or control. Turbulence, weather, stock market etc. can be considered as nonlinear chaotic things. Chaotic dynamics is closely related to fractals. The fractals can be considered as the images of chaotic systems. They are self-similar, never ending, infinite complex patterns. Chaos game is an algorithm used to create fractals. Using chaos game algorithm, we can convert anything with chaotic behavior to unique images.

Due to the fractal nature, biological sequences like DNA, RNA and amino acid can be converted in to chaos game representation. The use of Chaos game representation as useful signature images of bio-sequences such as DNA has been investigated since early 1990s. The CGR of bio-sequences was first proposed by H. Joel Jeffry [1]. We will now briefly introduce the method of deriving CGR image from a DNA sequence.

As we all know, Nucleotides are the basic building blocks of DNA sequences. Nucleotides are composed of four bases adenine, thymine, guanine and cytosine. So a DNA sequence can be literally treated as a string composed of four letters A, T, G and C. To derive a Chaos Game Representation of a genome, a square is first drawn to any desired scale and corners marked A, T, G and C. Nucleotide A, T, G and C have assigned positions (0, 0), (1, 0), (1, 1) and (0, 1) respectively. For plotting a given sequence, we start from the center of the square. The first point is plotted halfway between the center of the square, and the corner corresponding to the first nucleotide of the sequence, and successive points are plotted halfway between the previous point, and the corner corresponding to the base of each successive nucleotide. The steps for plotting a given sequence are summarized below.

1. Read the first nucleotide in the given DNA sequence.
2. Calculate the midpoint between the center and the corner corresponding to the first nucleotide and place a mark there. This is the current point.
3. Do the following steps until all nucleotides are processed. Read the next nucleotide in the sequence. Calculate the midpoint between the current point and the corner corresponding to the newly read nucleotide and make a mark there.

Let us illustrate the procedure with an example DNA sequence ATCGTAC. We can make its CGR image by following steps.

1. Plot the first point P1, halfway between the center of the square, and the A corner.
2. The next point P2 is plotted halfway between P1 and the T corner.
3. The next point P3 is plotted halfway between P2 and the C corner.
4. The next point P4 is plotted halfway between P3 and the G corner.
5. The next point P5 is plotted halfway between P4 and the T corner.
6. The next point P6 is plotted halfway between P5 and the A corner.
7. The next point P7 is plotted halfway between P6 and the C corner

Fig 1 depicts the process graphically.

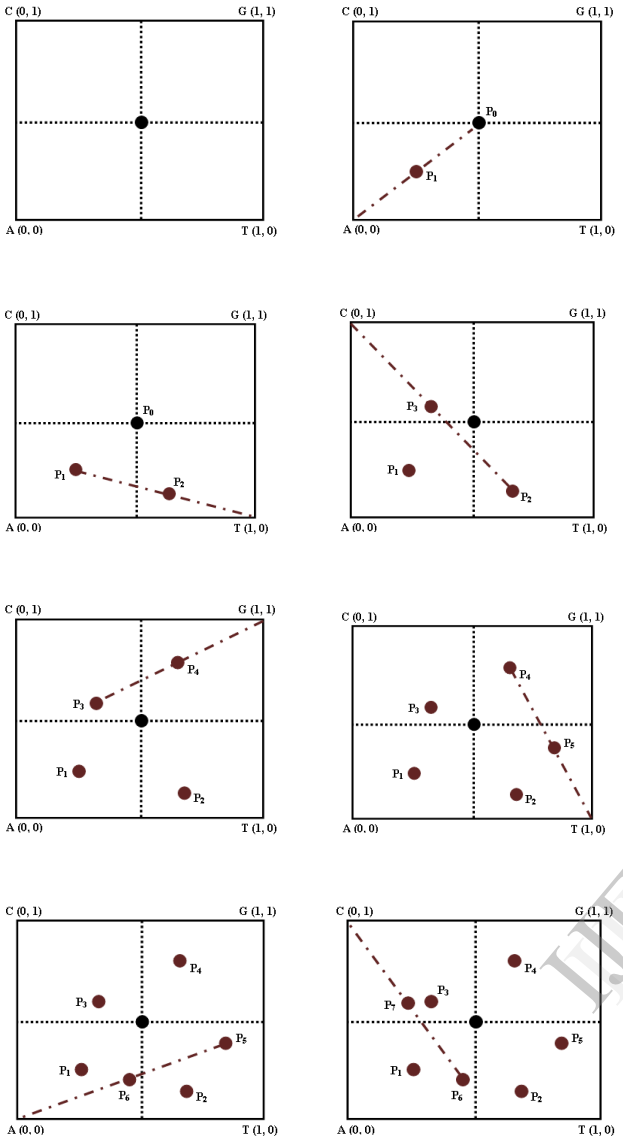

Fig. 1. Chaos Game Representation of 'ATCGTAC'.

A CGR image has so many interesting properties. Every bio-sequence has a unique CGR. The k^{th} point in the CGR corresponds to the first k-long initial subsequence of the given sequence. For example consider the sequence 'ATGTCCA'. Then the first point in the CGR represents the sequence 'A', second point represents 'AT', third point represents the sequence 'ATG' and so on. If two points in the CGR are within the same quadrant, they correspond to sequences with the same last base. If they are in the same sub-quadrant, the sequences have the same last two bases. If they are in the same sub-sub-quadrant the sequence have the same last three bases [2]. The Fig 2 shows the illustration of the above concept.

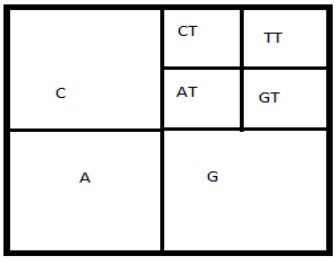

Fig. 2. Relation between nucleotides and areas of CGR

A lot of researches were already done in the field of Chaos Game Representation. The study of fractals and chaotic dynamics gives raise the concept of CGR [3, 4, 5]. In 1990, H. Joel Jeffry converted DNA sequences into Chaos Game Representation [1]. New algorithms for nucleotide sequence analysis were introduced [6]. Later N. Goldman analyzed some patterns in the Chaos Game Representation and found a relation between nucleotides and areas of CGR image [2]. A classification system that discriminates species into prokaryotes and eukaryotes was established [7]. The distribution of positions in CGR plane was shown to be a generalization of Markov chain probability tables [8]. A three dimensional Chaos Game Representation was evolved in 2007 [9]. Classification of Eukaryotes into eight classes using Artificial Neural Network was also performed [10].

II. Materials And Methods

A. *Chaos Game Representation of DNA Sequences*

As we already know, Chaos game representation is a pictorial representation of DNA sequences. In the first step, we have to convert the actual input DNA sequence to corresponding CGR image. The input sequence is so lengthy that we cannot identify any patterns in the sequence. The figure shows just 1/3rd of the sequence of a particular genome. But in the converted CGR form, the patterns can be visualized easily. CGR of Human Beta Globin Region on Chromosome11 (HUMHBB) is shown in Fig 3.

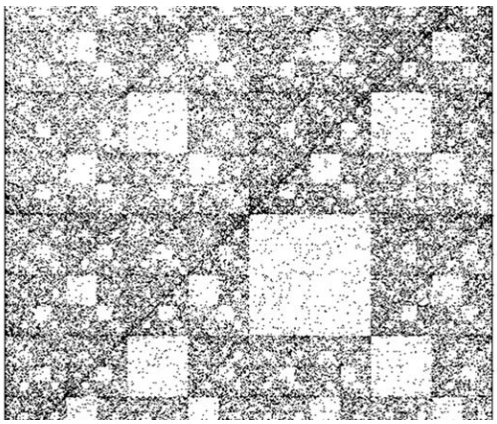

Fig. 3. CGR of HUMHBB Chromosome 11.

If we clearly examine the picture horizontally, we can see that at the top of each quarter-strip there are four copies and at the

top of each eighth-strip there are eight, and so forth. It is the property of self-similarity, a concept very important in the study of fractals. In this paper, we use a tool developed by National Centre for Biotechnology Information (NCBI) known as C-GRex. C-GRex means Chaos Game Representation Explorer. We input original DNA sequence and C-GRex produce corresponding CGR image

B. Obtaining Features from CGR images

The next step is the feature extraction from the CGR images for classification. Here we use dinucleotide frequencies as the features. A dinucleotide means combination of two nucleotide units (AA, AT, GC, TG etc.). We have to extract dinucleotide frequencies from the CGR images. Some properties of Chaos Game Representation help us to derive dinucleotide frequencies easily. We know that if two points are in the same sub-quadrant, they have same last two bases. So the frequency of dinucleotide combinations can be determined by dividing the CGR image using a grid of 4×4 size and counting the dots in each square.

C. Classification using Support Vector Machine

Support Vector Machine is used for classification. We chose Support Vector Machine for classification because of some reasons. It is very suitable for nonlinear classification. Here the basic idea is to map feature vectors nonlinearly to another space and learn a linear classifier there. The linear classifier in new space would be an appropriate nonlinear in the original space. The fig 4 shows the basic idea about Support Vector Machine. Let us consider an example

X=[x1, x2] ⟶Original 2 dimensional (R^2) feature vector

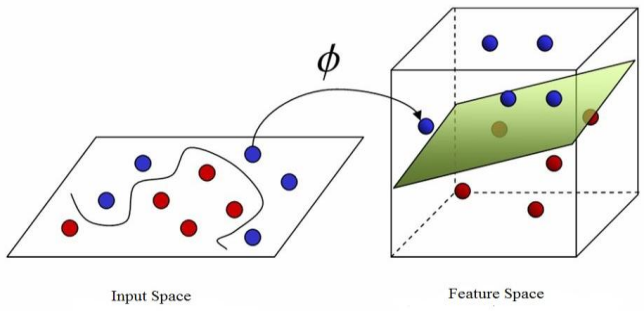

Fig. 4. Principle of Support Vector Machine.

ϕ: $R^2 \longrightarrow R^6$ (Nonlinear function that maps 2 dimensional vectors to 6 dimension)

Let define Z=**ϕ**(X) = [1, x1, x2, $x1^2$, $x2^2$, x1x2]

Then we define a linear classifier in R^6 such as

$g(Z)=a_0z1+a_1z2+a_2z3+a_3z4+a_4z5+a_5z6$.

When g(Z) is converted to original space, it becomes

$g(x) = a_0+a_1x1+a_2x2+a_3x1^2+a_4x2^2+a_5x1x2$, which is a quadratic function in R^2.

There are two major issues in naively using this idea. If we want a p^{th} degree polynomial in the original space (R^m), then the transformed feature vector, Z has dimension $O(m^p)$. Even in the quadratic case, if we have 100 dimensional vectors which is not very uncommon in pattern recognition applications, the transformed vectors should have 10000 components. This results in huge computational cost both for learning and final operation of the classifier. The second problem is about lack of examples for training. We need to learn $O(m^p)$ parameters rather than O(m) parameters. Hence we may need much larger number of examples for achieving proper generalization. But Support Vector Machine offers an elegant solution to both problems.

The Support Vector Machine solves the first problem by using kernel functions. Kernel functions effectively map the original feature vectors into higher dimensional space without explicit calculation. There exist different types of predetermined kernels. In this paper we use three types of kernels i.e. linear kernel, polynomial kernel (2nd degree) and Radial basis function kernel. But in many cases, we cannot get enough examples. For overcome this issue, Support Vector Machine learns not any separating hyper plane but the optimal separating hyper plane. Optimal separating hyper plane is the separating hyper plane that maximizes the separation between classes. Learning optimal hyper plane is essentially what allows Support Vector Machine to learn well with fewer examples even in a very large dimensional space. In this paper, we use an integrated software developed by Chih-Chung Chang and Chih-Jen Lin known as LIBSVM. It is very suitable for multi-class classification.

III. RESULTS AND DISCUSSION

Mitochondrial DNA sequences are downloaded from NCBI database. A total of 696 DNA sequences from each seven classes are taken. The distribution of DNA sequences used for training and testing is shown in table 1. The CGR images are obtained by the tool C-GRex for each DNA sequences. The dinucleotide frequencies are used as features. The dinucleotide frequencies are extracted simply from each CGR image using a 4×4 grid. Support Vector Machine is used for classification. Here LIBSVM is used for classification. Three different kernels were investigated. The accuracy obtained for Linear kernel, Polynomial kernel, RBF kernel is shown in the table 2.

TABLE 1. DISTRIBUTION OF INPUT DATA

Class	Total	Training	Testing
Fungi	55	28	27
Plants	55	27	28
Cnidaria	55	28	27
Platyhelminthes	52	26	26
Porifera	46	23	23
Protostomia	183	91	92
Vertebrata	250	125	125

TABLE 2. ACCURACY OBTAINED FOR DINUCLEOTIDE FREQUENCIES

Kernel used	Accuracy obtained
Linear kernel	95.4023
Polynomial kernel	95.9770
RBF kernel	37.3563

When we take trinucleotide frequencies by using an 8×8 grid and tetranucleotide frequencies by using 16×16 grid, the accuracy is increased. The accuracy obtained for trinucleotide and tetranuleotide frequencies is shown in table 3 and table 4.

TABLE 3. ACCURACY OBTAINED FOR TRINUCLEOTIDE FREQUENCIES

Kernel used	Accuracy obtained
Linear kernel	95.1149
Polynomial kernel	95.1149
RBF kernel	37.3563

TABLE 4. ACCURACY OBTAINED FOR TETRANUCLEOTIDE FREQUENIES

Kernel used	Accuracy obtained
Linear kernel	96.5517
Polynomial kernel	97.1264
RBF kernel	38.2184

IV. CONCLUSIONS

Support Vector Machine is used for classification. Earlier classification systems using Hidden Markov Model and Artificial Neural Network produce an accuracy around 85-90. But here we can easily show that Support Vector Machine is the good choice for a better classification system for CGR images of genome sequence. Three different kernels were investigated here. For dinucleotide, trinucleotide and tetranucleotide frequencies, Linear kernel and 2nd degree polynomial kernel produce very good accuracy more than 90. But the Radial Basis Function kernel responds very poorly. In all cases the accuracy obtained for RBF kernel lies below 50.

ACKNOWLEDGMENT

We would like to take this opportunity to express our gratitude to all those who have guided in the successful completion of this endeavour . We express our sincere thanks to [10] for inspiring us to do this work. We are very thankful to the staffs of College of Engineering, Chengannur for their valuble suggestions.

REFERENCES

[1] H. Joel Jeffrey, “Chaos game representation of gene structure”, Nucleic Acids Research, Vol. 18, No. 8, pp: 2163-2170,1990.

[2] Nick Goldman, “Nucleotide, dinucleotide and tri nucleotide frequencies explain patterns observed in chaos game representations of DNA sequences”, Nucleic Acids Research, 1993, Vol. 21, No. 10, pp: 2487-2491, 1993.

[3] M.F Barnsley, “Fractals everywhere”, springer-verlag, New York, 1998, pp: 118-171.

[4] R. L. Devaney, “An Introduction to Chaotic Dynamical Systems”, Addison Wesley, Redwood City, California, 1989.

[5] S.K. Park and K. W. Miller, “Random Number Generators: Good Ones are Hard to Find”, Communications of the ACM, Vol. 31, No. 10, October, 1988, pp. 1192-1201.

[6] C. Dutha and J. Das, “Mathematical characterization of Chaos Game Representation: new algorithms for nucleotide sequence analysis”, J.Mol.Biol, 1992, pp: 715-719.

[7] P.J Deschavanne, A.Giron, J.Vilain, G.Fagot and B.Fertil, 1999, Mol.Biol.Evol, 16, pp: 1391-1399.

[8] Almeida J S, Joao A. Carrico, Antonio Maretzek, Peter A. Noble and Madilyn Fletcher, “Analysis of genomic sequences by Chaos Game Representation”, BIOINFORMATICS Vol. 17, no. 5, pp: 429-437, 2001.

[9] Iman Tavassoly, Omid Tavassoly, Mohammad Soltany Rezaee Rad, Negar Mottaghi and Dastjerdi, “Three dimensional Chaos Game Representation of genomic sequences”, Frontiers in the Convergence of Bioscience and Information Technologies, 2007.

[10] Vrinda V. Nair, Karthika Vijayan, Deepa P.Gopinath and Achuthsankar S. Nair, “ANN based Classification of Unknown Genome Fragments using Chaos Game Representation”, Second International Conference on Machine Learning and Computing, 2010.