Bio-sequence Signatures Using Chaos Game Representation

Achuthsankar S. Nair, Vrinda V Nair, Arun K S
Centre for Bioinformatics, University of Kerala
Krishna Kant, Alpana Dey
Department of Information Technology, Govt of India, New Delhi

Introduction

Computational Biology/Bioinformatics is the application of computer sciences and allied technologies to answer the questions of Biologists, about the mysteries of life. It looks as if Computational Biology and Bioinformatics are mainly concerned with problems involving data emerging from within cells of living beings. It might be appropriate to say that Computational Biology and Bioinformatics deal with application of computers in solving problems of molecular biology, in this context. What are these data emerging from a cell ? Four important data are: DNA, RNA and Protein sequences and Micro array images. Surprisingly, first 3 of them are mere text data (strings, more formally) that can be opened with a text editor. The last one is a digital image which is only indirectly a cellular data. See Fig 1.

(a) DNA Data (4 letter strings)

(b) RNA Data (4 letter strings)

(c) Protein Data (20 letter strings)

(d) Micro Array Image Data (traditional Digital Images)

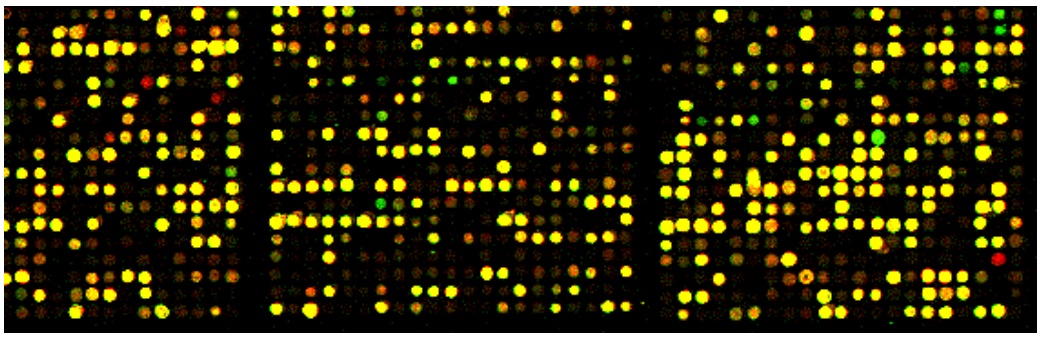


Fig. 1: Four major kinds of data required to be analyzed in Bioinformatics

Our interest is to discuss about deriving signatures for the first three kinds of data. It is well known that the gene regions of the DNA in the nucleus of the cell is copied (*transcribed*) into the RNA and RNA travels to protein production sites and is *translated* into proteins. In short, **DNA→ RNA → Proteins**, is the Central Dogma of Molecular Biology. **Computational Genomics & Proteomics** are fields which encompass various studies of the genome and the proteome, based on their sequences. Both start with sequence data, and attempt to answer questions like this:

Genomics: Given a DNA sequence, where are the genes ? (Gene Finding); How similar is the given sequence with another one ? (Pair-wise Sequence Alignment); How similar are a set of given sequences ? (Multiple Sequence Alignment); Where on this sequence does another given bio-molecule bind ? (Transcription factor binding site identification); How can we compress this sequence ? How can we visualize this sequence insightfully ? (genome browsing)

Proteomics: Given an amino acid sequence data, how similar is it with another one, or how similar are a set of amino acid sequences (pair-wise and multiple sequence alignment); What is the primary, secondary or tertiary structure of the molecule ? (the great protein folding problem); Which part is most chemically active ? (Active site determination problem); How would it interact with another protein ? (protein-protein interaction problem); To which cell compartment is this protein belonging to ? (protein sub-cellular localization or protein sorting problem).

A large number of tools and techniques are available in computational genomics and proteomics, with varying degrees of success. A technique that has been developed successfully and used very widely in both genomics and proteomics, is the sequence alignment technique. This forms the basis for comparative studies of the genome and the proteome. A computationally intensive problem has been addressed to satisfactory level, providing a service which is quite fast and reliable.

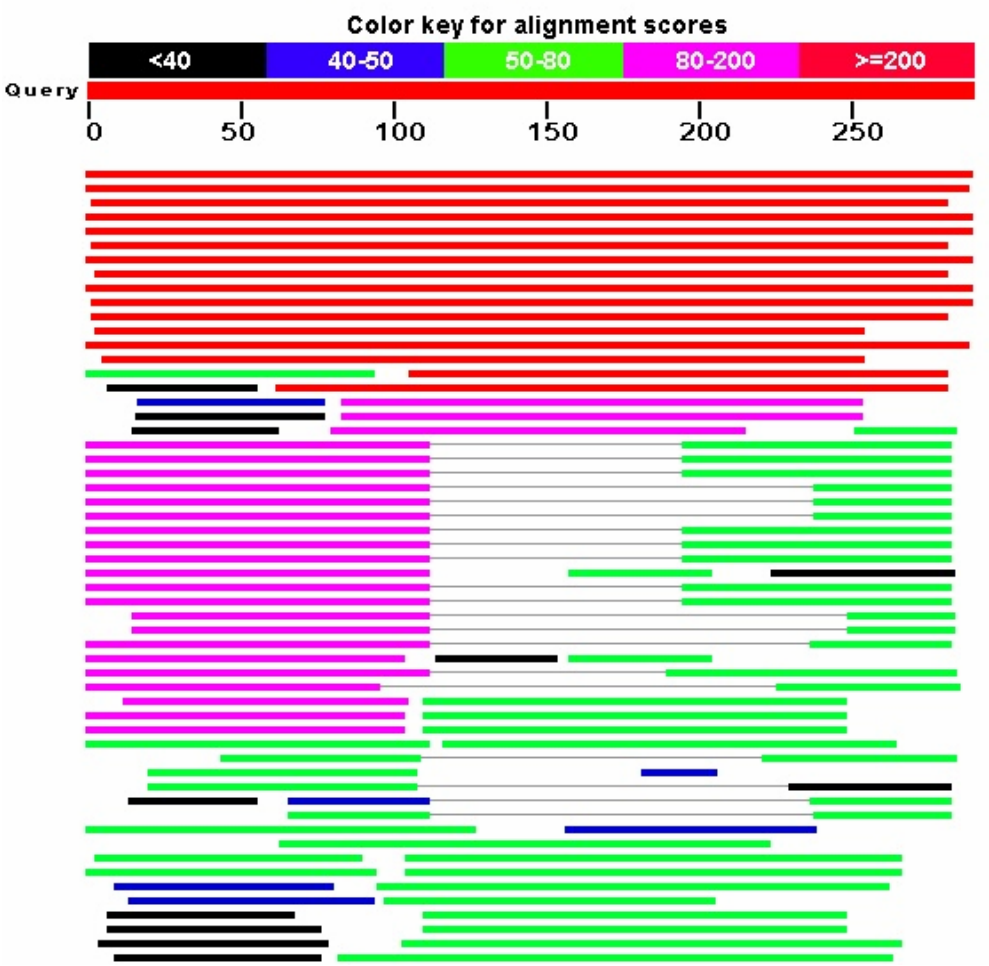


Fig 2. Typical BLAST output for a query

For the modern life scientist, the BLAST service which returns local alignment searches of query sequences has become the Google of Biology. Another example of a successful bioinformatics tool is the UCSC Genome Browser. It would be unthinkable to comprehend the genomic data that is continuously erupting, without a facility such as the genome browser.

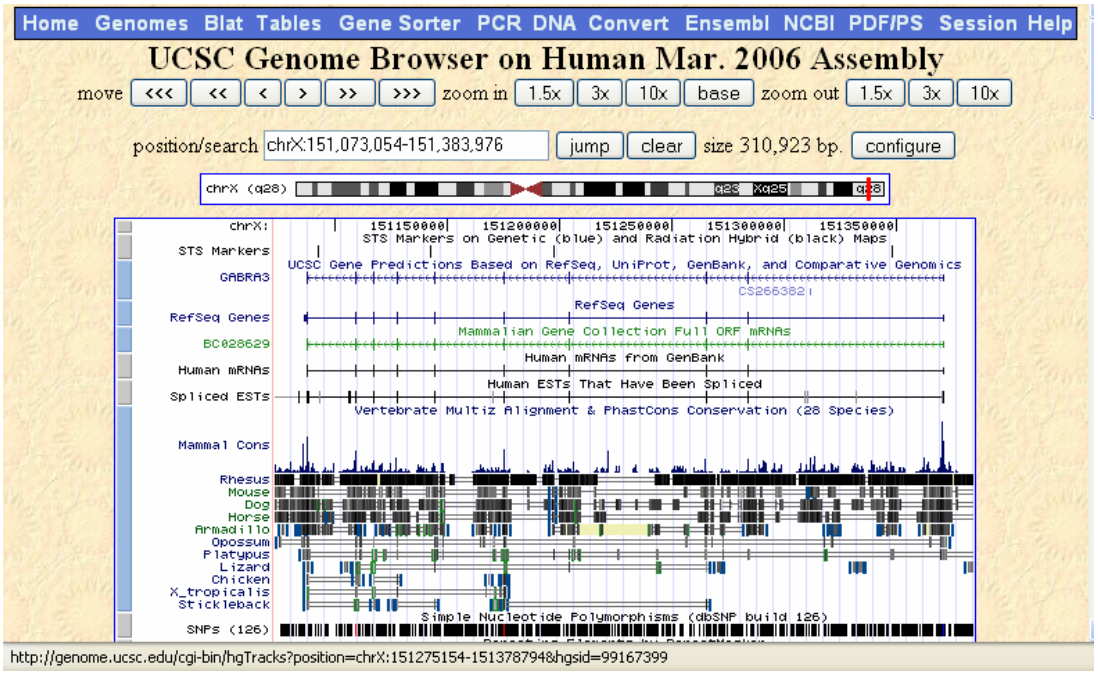


Fig 3. The Genome Browser

Many of the problems in genomics and proteomics have been attacked by various researchers using a plethora of tools, from the field of mathematics, statistics, soft computing and many others. We discuss here yet another method which is useful in analyzing DNA/Protein sequences aimed at solving some of the above problems.

Chaos Game Representation Algorithm

During 1970s, a new field of physics was developed known as *chaotic dynamical systems* or simply *chaos* [1]. This field was closely associated with *fractals.* Fractal geometry, in contrast with Euclidean geometry, deals with objects that possess fractional dimensions like 1.45, 2.79 etc. Fractal geometry considers itself the geometry of the real (rather than the ideal) and consequently treats the objects in nature such as clouds, coastlines, trees, landscapes, lightning etc as possessing fractal dimensions. Among interesting properties of the fractals are their unvarying complexity at varying scales.

The Chaos Game is an algorithm, which is an offshoot of research in the above area. It allows one to produce unique images of fractal nature, known as Chaos game Representation images (CGR images) from symbolic sequences, which can serve as signature images of the sequences. It was originally described by Barnsly in 1988. Chaos Game is an algorithm whose input is a sequence of letters (finite alphabets) and output is an image.

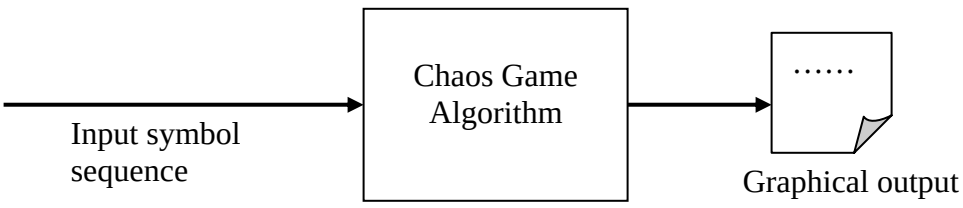


Fig 4: *CGR and it's I/P & O/P*

Even though CGR of any finite sequence with finite set of alphabets is possible, here we are considering only biological sequences like DNA sequence, RNA sequence and amino acid sequences. Life scientists consider that key to life processes is centered around the above mentioned three entities. All these entities can be represented by sequences of finite alphabets.

The use of CGRs as useful signature images of bio-sequences such as DNA has been investigated since early 1990s. CGRs of genome sequences was first proposed by H. Joel Jeffrey [1]. Later other bio-sequences were also explored. We will now briefly introduce the idea of deriving a CGR image of a DNA sequence.

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. Points are marked within the square corresponding to the nucleotides in the sequence. In CGR, the four nucleotides A, G, C, and T are assigned to the corners of a square as in Figure 5. The choice of the corners is not based on any particular criteria, and indeed can be assigned in any other way.

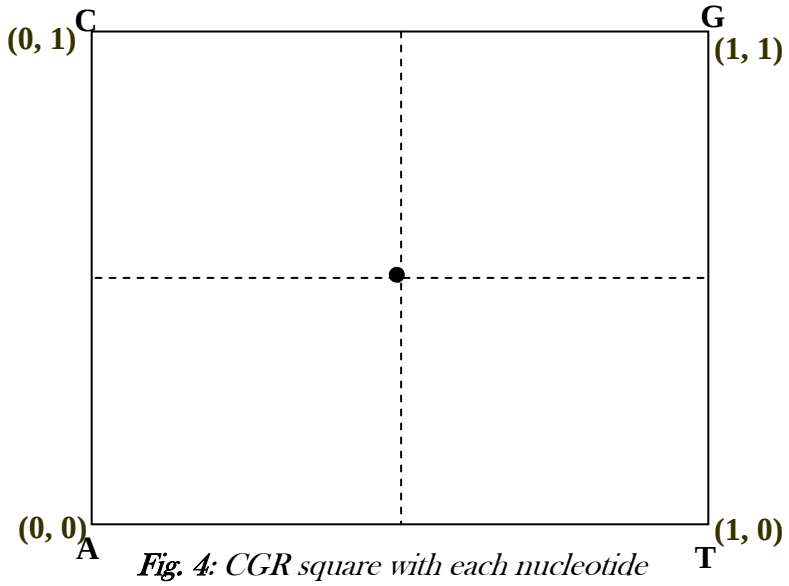


Fig. 4: CGR square with each nucleotide assigned to corner

Nucleotide A has an assigned position (0, 0), T has assigned an position (1, 0), G has an assigned position (1, 1) and C has an assigned position (0, 1). Now we define a procedure for representing any arbitrary nucleotide sequence as a point inside the square. For plotting a given sequence we start from the centre of the square. The first point is plotted halfway between the centre 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 nucleotides. The mid point Pm (x_m, y_m) between two given points P1(x_1, y_1) and P2 (x_2, y_2) can be calculated using the following equation.

i. $x_m = (x_1 + x_2)/2.$

ii. $y_m = (y_1 + y_2)/2.$

These steps for plotting a given sequence are concluded below.

1. Select the first nucleotide from the given sequence.
2. Calculate the mid point between the centre and the corner corresponding to the first nucleotide ($P_N(x_N, y_N)$. Let the mid point be $P_i(x_i, y_i)$. Let (x_c, y_c) be the co-ordinates of the mid point of the square.
    a. $x_i = (x_c + x_N)/2$
    b. $y_i = (y_c + y_N)/2$
3. Do the following steps until all the nucleotides are processed.
    a. Read the next nucleotide in the sequence.
    b. Calculate the mid point between the current point $P_i(x_i, y_i)$ and the corner corresponding to the newly read nucleotide. Let the new mid point be $P_{i+1}(x_{i+1}, y_{i+1})$.
        i. $x_{i+1} = (x_i + x_N)/2$
        ii. $y_{i+1} = (y_i + y_N)/2$

Now using the above procedure let us plot a DNA sequence TACAGA into this square. A square is drawn to any desired scale and corners marked A, T, G and C. Points are marked within the square corresponding to the bases in the sequence, as follows:

1. Plot the first point halfway between the center of the square and the T corner.
2. The next point is plotted halfway between the previous point and the A corner.
3. The next point is plotted halfway between the previous point and the C corner.
4. The next point is plotted halfway between the previous point and the A corner.
5. The next point is plotted halfway between the previous point and the G corner.
6. The next point is plotted halfway between the previous point and the A corner.

Figures 5 to 10 depict the process graphically.

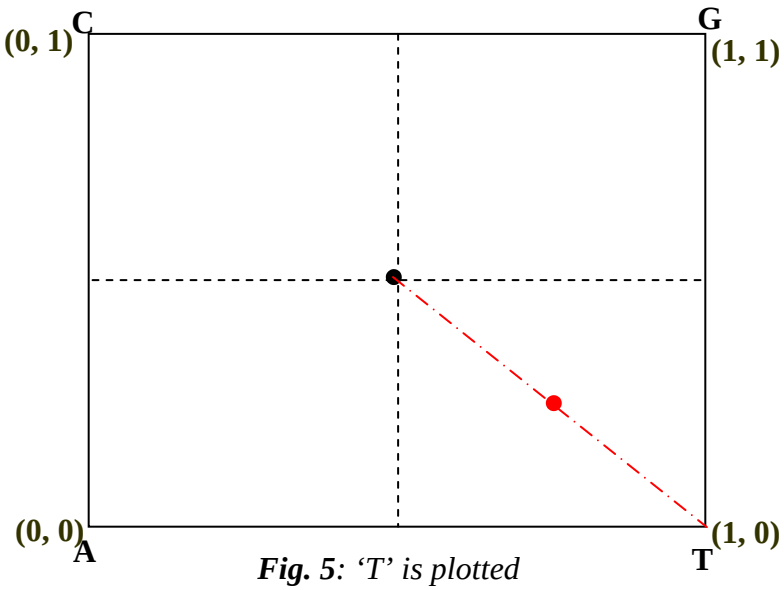


***Fig. 5**: 'T' is plotted*

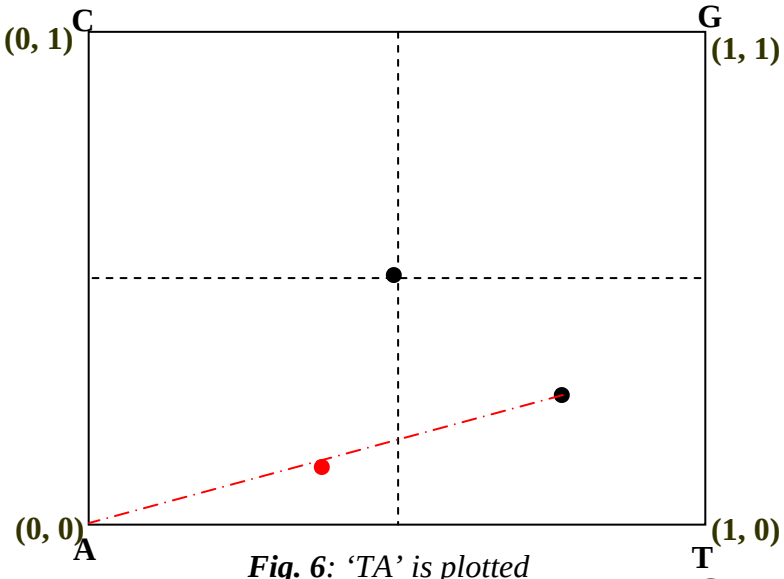


***Fig. 6**: 'TA' is plotted*

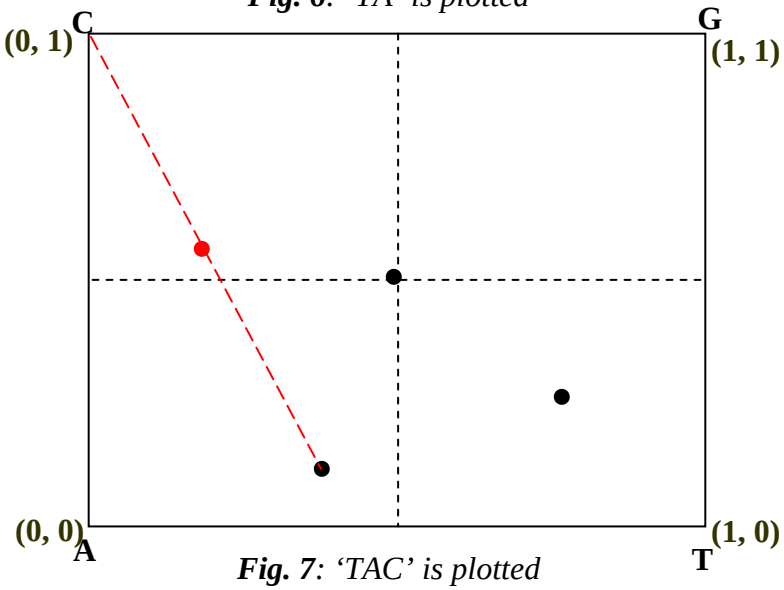


***Fig. 7**: 'TAC' is plotted*

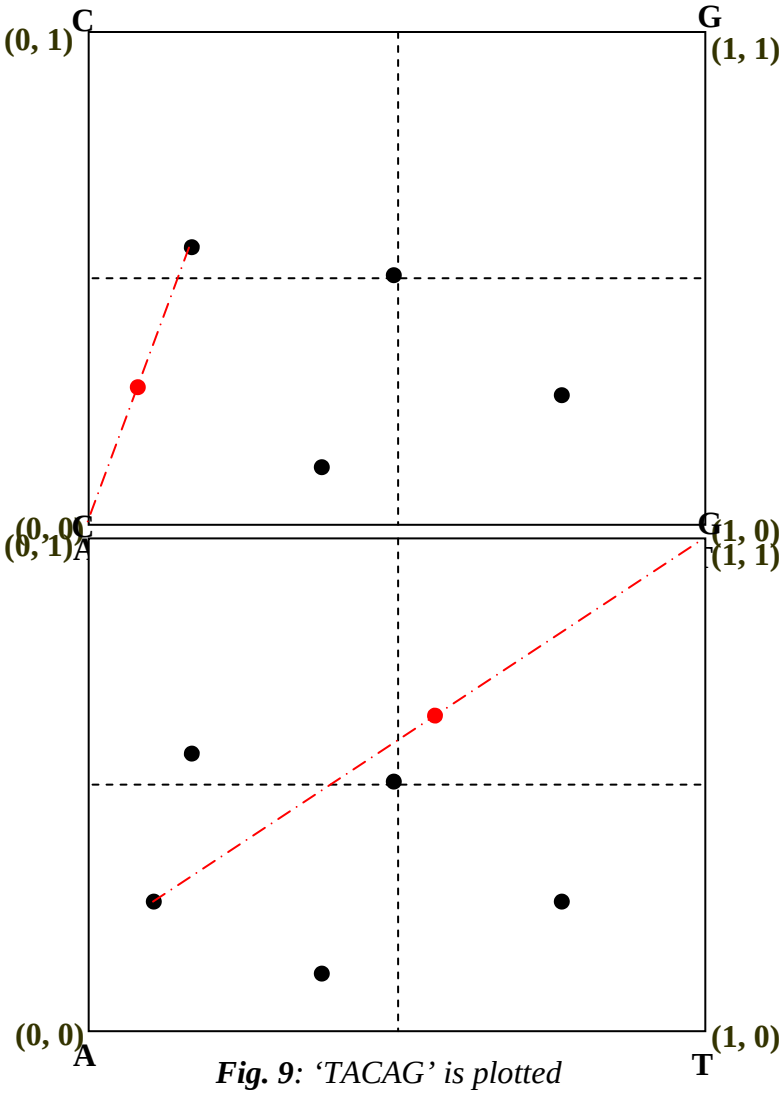


***Fig. 9**: 'TACAG' is plotted*

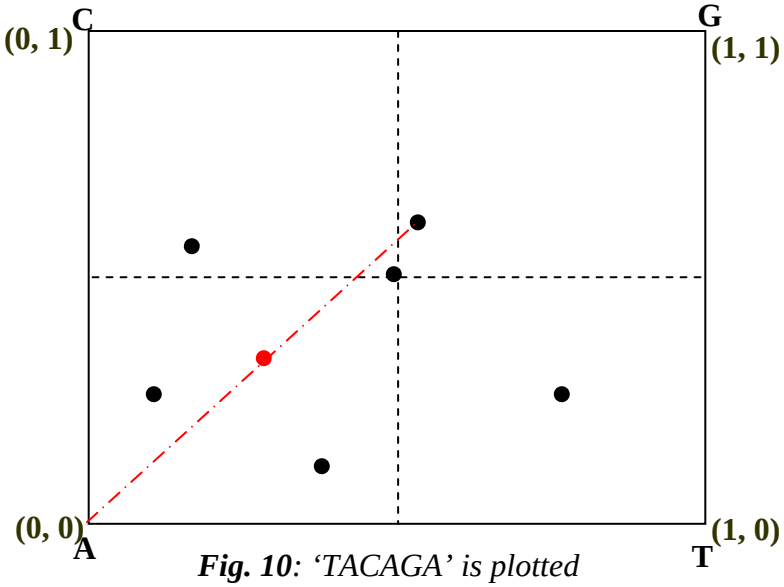


***Fig. 10**: 'TACAGA' is plotted*

Now, let us see how a real CGR would look like. Fig 11 shows CGR of Hepatitis A virus, full genome, plotted using beta version of a tool C-GRex, discussed later in this chapter. Fig 12 and 13 show CGR of full genome of His 2 virus, and Thermosinus carboxydivorans respectively.

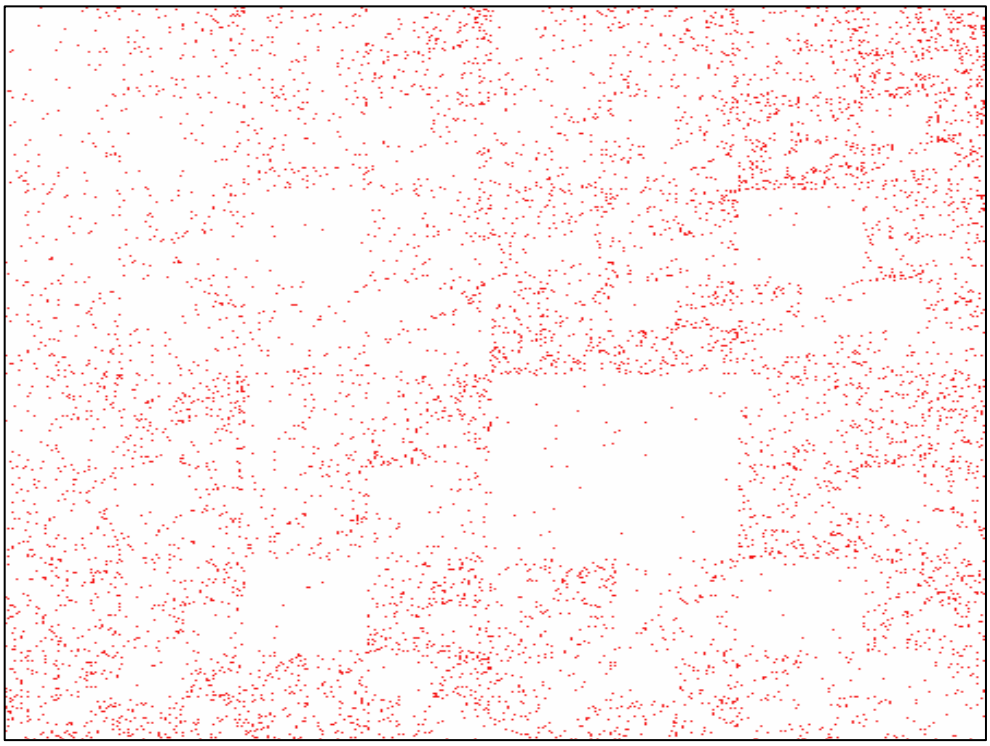


Fig. 11 : *CGR of Hepatitis A virus full genome.*

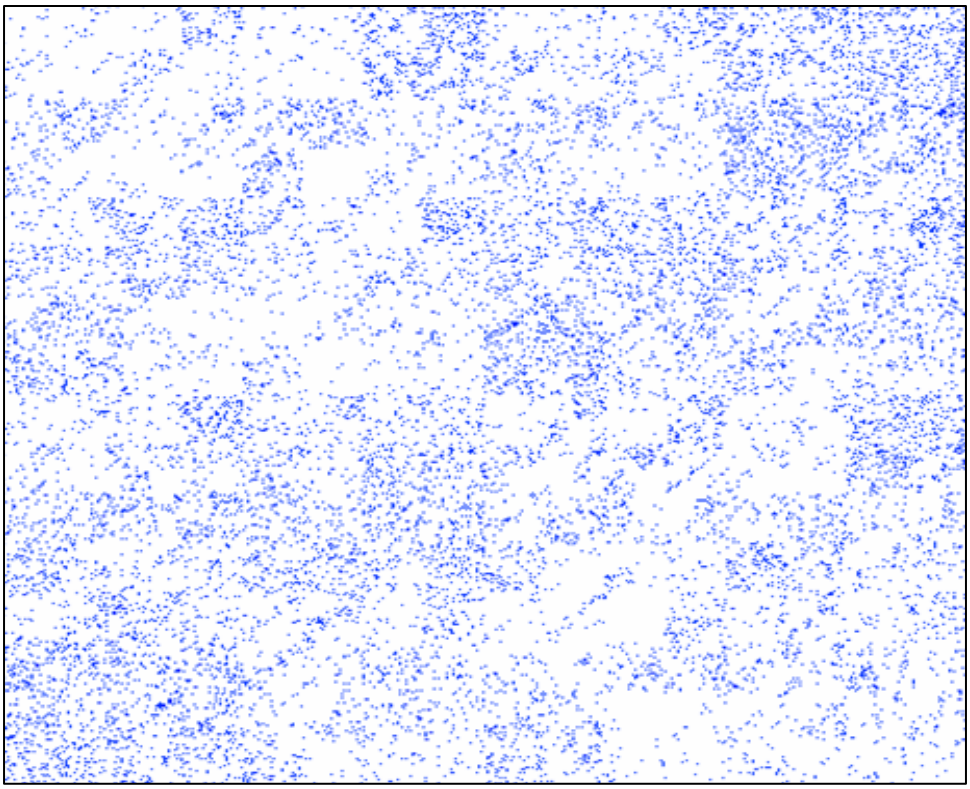


Fig. 12 : *CGR of His 2 virus full genome.*

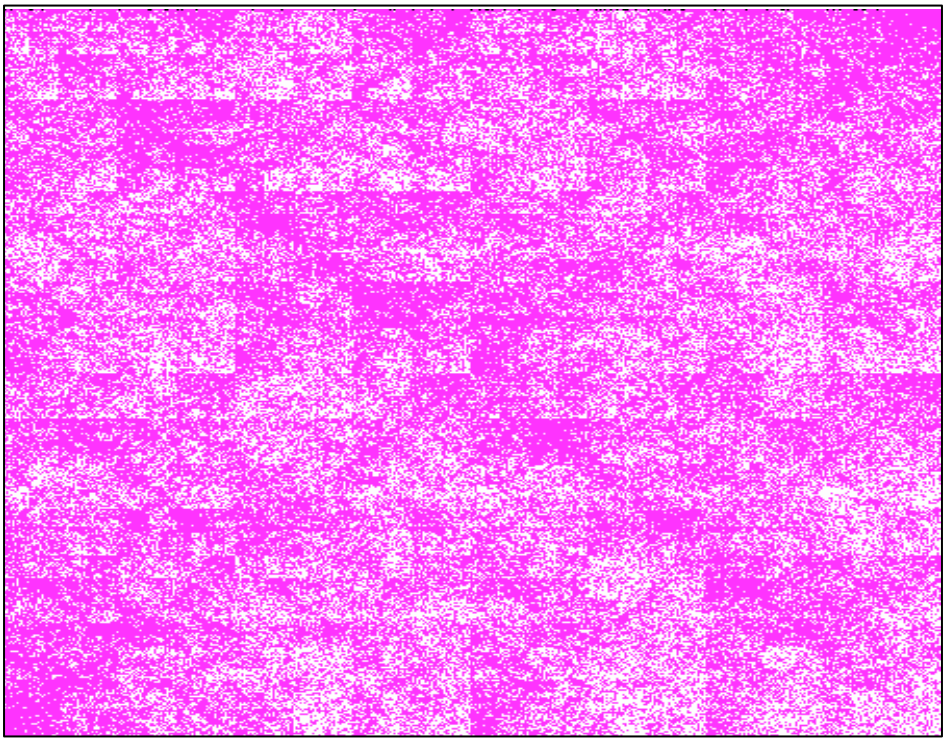


Fig. 13 :*CGR of Thermosinus carboxydivorans full genome.*

The above CGR images clearly indicate that they vary from genome to genome, with characteristic patterns for each. This is what encourages one to investigate if CGRs can indeed be used as unique signature images for genomes and also other bio-sequences.

CGR Properties

A CGR has many properties. Every sequence has a unique CGR. In fact every symbol in a sequence will have a corresponding unique point in the CGR, even though the reverse need not be the case. Every point on the CGR is a representation of all the symbols in the sequence up to that point. For instance, in the CGR of the sequence ATTTGGCCATCG, the fifth point represents the sequence ATTTG.

Each sub-square in a CGR has a special significance. If we divide the CGR into four quadrants, then the top right corner will contain points representing sub-sequences that end with G, as a mid-point between any other point in the square and the G-corner has to fall in this quadrant. Hence if we count the points in this quadrant, it will be equal to the count of the base G in the sequence. If we divide this quadrant into another 4 squares, in the clockwise order, they would represent subsequences that end in GG, TG, AG and CG, making it possible to derive the 2-mer counts by counting the points in these sub-squares. In general, by dividing the CGR square into sub-squares of side 2^{-n}, we can find the number of different n-mers present in the sequence (see Fig 14).

Side ½	- 4 sub-squares	– *monomers*
Side ¼	- 16 sub-squares	– *dimers*
Side 1/8	- 64 sub-squares	– *trimers* and so on

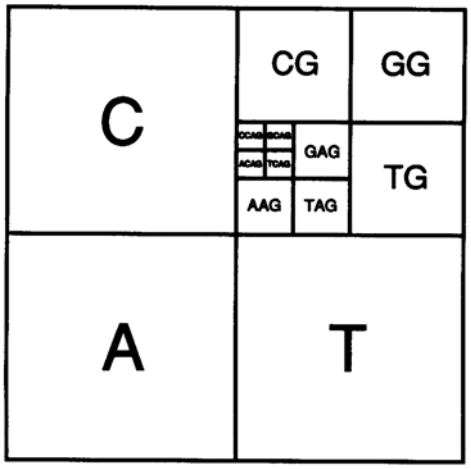
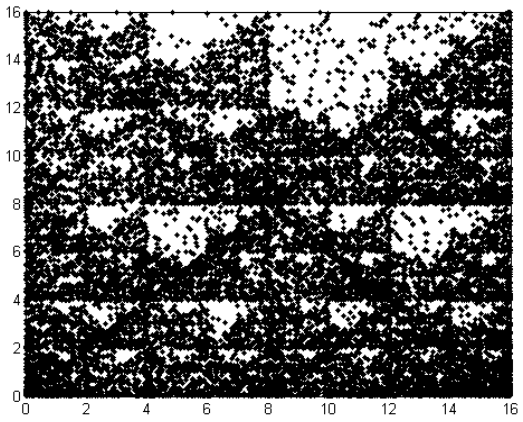


Fig. 14. Correspondence between n-mers and sub-squares of CGR.

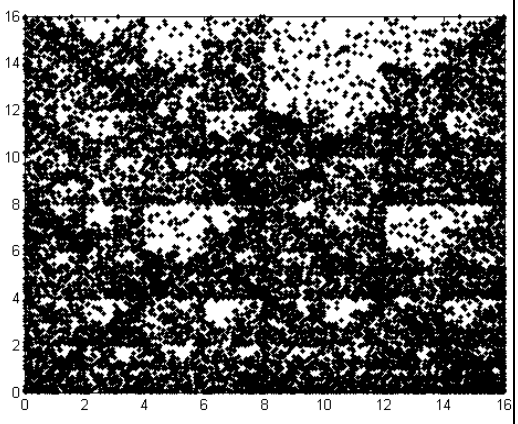
Another interesting characteristics of CGR is that images obtained from parts of a genome show the same pattern as that of the whole genome [2]. Thus analysis of parts of a genome will result in a satisfactory genomic signature. This also helps comparing non homologous genomic sequences when only parts of the genomes are available.

Genomic CGRs

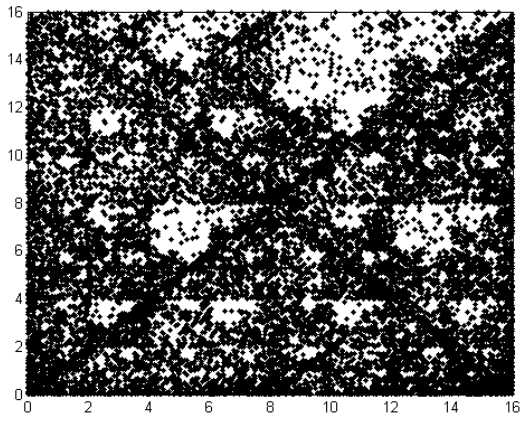
The CGR of various organisms exhibit differing patterns which are characteristic of that species. Here we have obtained 20kbp of a few species which illustrates some interesting patterns and hence features of the group. Main features of CGR images include CG double scoops, diagonals, absence of diagonals, horizontal variation in intensities from top to bottom or reverse, empty patches and word-rich regions of different shapes.



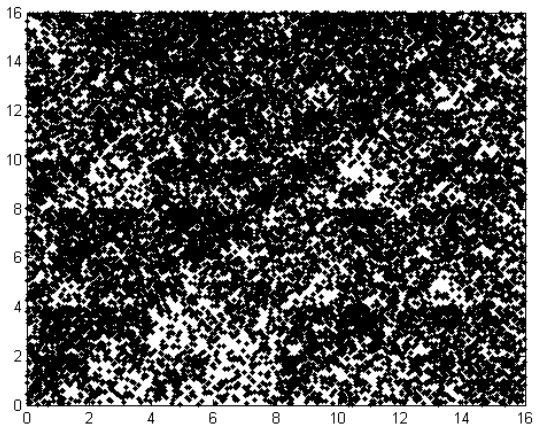
a) Musmusculus (Eukaryote)



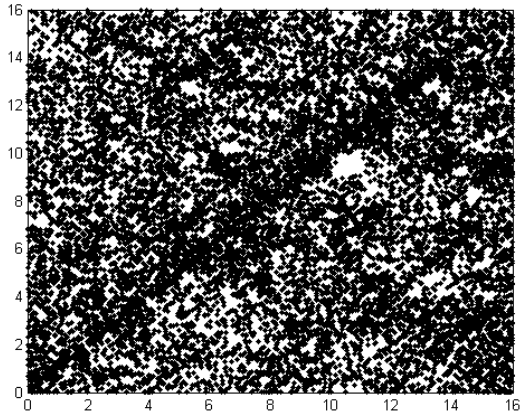
b) Bos Taurus (Eukaryote)



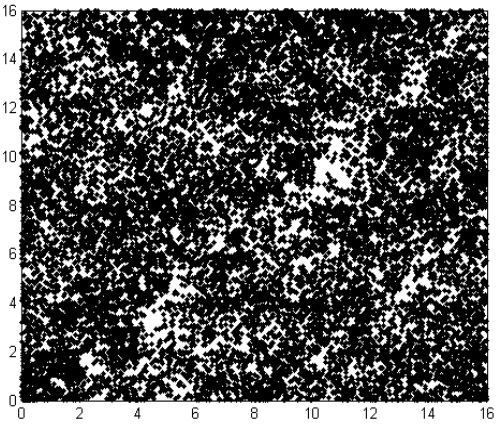
c) Homo Sapiens (Eukaryote)



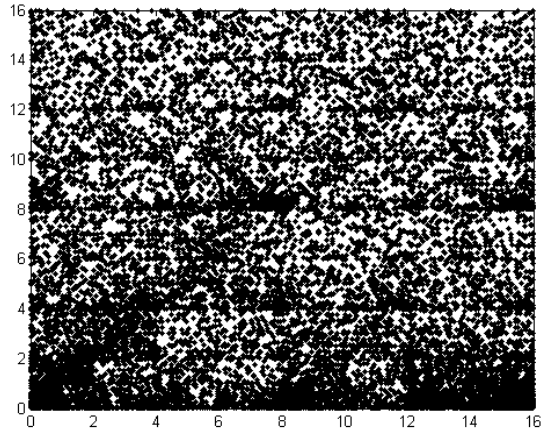
d) Agrobacterium tumefasciens (Bacteria)



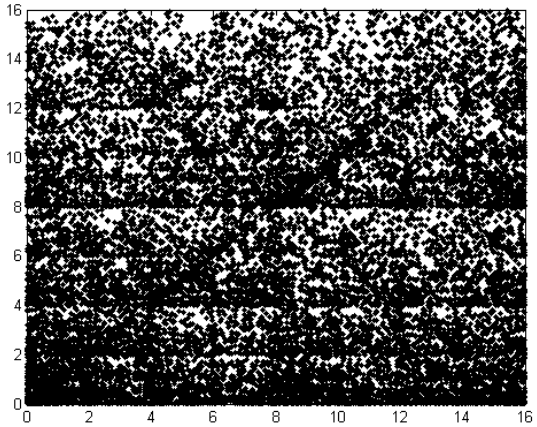
e) A. fulgidus (Archea)



f) E. coli K12 (Bacteria)



g) C. elegans (Eukaryote)



h) Arabidopsis Thaliana (Eukaryote)

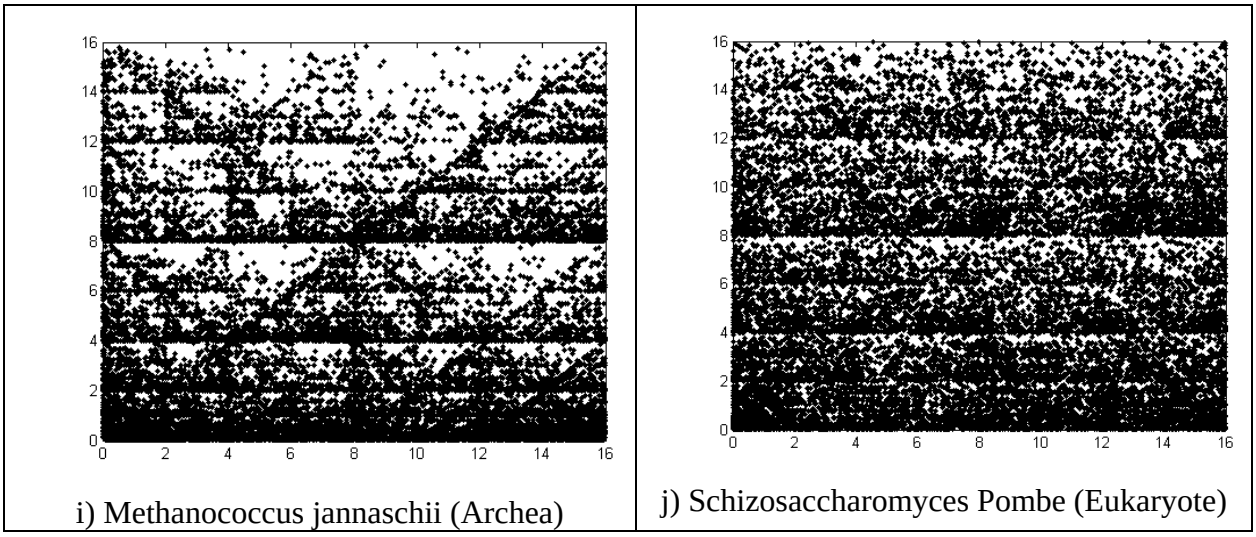


Fig. 15 CGRs for various organisms

Fig. 15 a, b and c are vertebrate images which exhibit CpG depletion manifesting as double scoops. Invertebrates in general have CGRs with uniform distribution of points. However certain patterns are seen as in Fig. 15 g for *C. elegans*, which shows some clustering along AG diagonal and AT line. *Agrobacterium tumefasciens* exhibit a word filled diagonal and upper triangular region whereas *Methanococcus jannaschii* and *Arabidopsis Thaliana* exhibit prominent diagonals. CpG depletion is also seen in some *eubacteria and archeabacteria* (see Fig. 15 i *M. jannaschii*) raising questions about the underlying mechanisms causing this depletion [2]. AT rich regions are exhibited in the form of horizontal lines with decreasing intensity from bottom to top by *Schizosaccharomyces Pombe* belonging to the yeast species. Referring to Fig. 14, the subsquare TAG can be still divided so that the upper left square within TAG represents CTAG. This region is seen empty in a few *eubacteria and archeabacteria* while the rest of the image does not show similarity. Fig 15 e and f illustrate this feature.

The double scoop in CGR image was first reported in human beta globin region. The relatively empty area corresponds to the subsquare CG and hence the inference of the relative sparseness of guanine following cytosine in the gene sequence [1]. The upper right quadrant had a large empty area, which is seen repeated in sub quadrants presenting a *double scoop* appearance. The double scoop points out the relative sparseness of guanine following cytosine in the gene sequence. This is a simple example of using CGRs to make observations of biological relevance. Reference [1] discusses the features of CGRs of vertebrate, invertebrate, plant and slime molds, phages, bacteria and virus.

Studies on CGR of coding regions of Human Globin genes and Alcohol Dehydrogenase genes of phylogenetically divergent species were done by Hill et.al. [3]. They found that CGRs were similar for genes of the same or closely related species but were different for relatively conserved genes from distantly related species. Dutta and Das [4] reported two algorithms that can predict the presence or absence of a stretch of nucleotides in any gene family using CGRs. Nick Goldman [5] showed that simple Markov chain models based solely on dinucleotide and trinucleotide frequencies can account for the complex pattern exhibited in CGRs of DNA sequences. Although later, Almeida et.al. [6] showed that CGR is a generalized scale independent Markov probability table. A very important and useful observation made by Deshavanne et. al. [2] was that

subsequences of a genome exhibit the main characteristics of the whole genome, attesting to the validity of the genomic signature concept. Roschen et. al. [7] have explored the potential of CGR representation for making alignment-based comparisons of whole genome sequences. Classification of CGR images have been done in references [2], [8], [9] and [10].

Proteomic CGRs

Works on CGR used for visualizing amino-acid sequences were done by Andras Fiser, Gabor E. Tusnady and Istav Simon [11]. They demonstrated that CGR can also be used for analyzing protein databases. Suggested applications include investigating regularities and motifs in the primary structure of proteins, analyzing possible structural attachments on the super-secondary structure level of proteins and testing structure prediction methods. Zu-Guo Yu, Vo Anh and Ka-Sing Lau [12] performed multifractal and correlation analyses of the measures based on the CGR of protein sequences from complete genomes and hence attempted to construct a more precise phyogenetic tree of bacteria.

Proteomic CGR requires some tinkering with the sequence (or with the image format) to derive CGR images as amino-acids are 20 in number. We either have to group the amino-acids into 4 categories or change from a square to a polygon. Early approaches were based on groupings. Amino-acid properties have been well studied for long and over 500 properties are known (see AA Index, for instance). This would mean that the grouping itself would have 500 choices. If we decide to change the groupings from 4 to any number from 2 to 20, we can produce corresponding n-sided polygon images. This is a work that is being carried out by some of the authors, and is implemented in an open source tool named C-GRex, discussed next.

C-GRex: A CGR Explorer for Bio-sequences

C-GRex, Chaos Game Representation Explorer is a tool to explore various features of CGR, in such a way that an unbelievable number of CGRs can be derived out of a given sequence, especially an amino-acid sequence, almost resembling a kaleidoscope. It is a handy tool for sequence visualization and analysis of patterns, hot spots and discoveries. C-GRex packs a wide variety of exploration facilities using Chaos Game Representation in DNA, RNA and Protein sequences and any other sequence. The tool comes with a set of functionalities which makes it unique. The software is designed in a way that a person with little knowledge about CGR will able to work with it and manipulate the plot.

The main window of C-GRex is shown in Fig 16. It contains three areas: plot area, settings display area, sequence panel in addition to menu bar and tool bar. The plot area shows the CGR plot of the loaded sequence. The settings applied to the CGR plot is shown in settings display area. The sequence panel contains the loaded sequence. The user can scroll through the sequence. The starting and ending no of the sequence visible in the sequence panel is displayed above the panel.

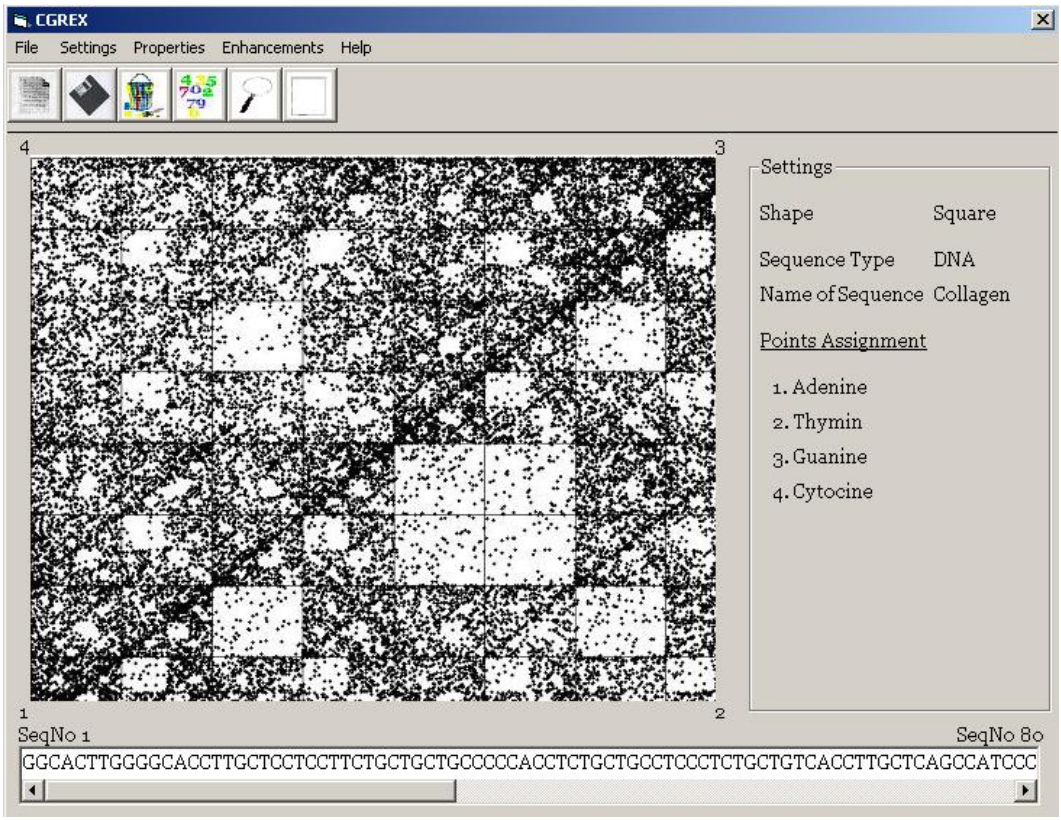


Fig 16: The main window of C-GRex

While it is not the intention here to introduce C-GRex comprehensively, the special feature of C-GRex will be briefly discussed. This relates to generating n-sided polygonal CGRs, with choice of n resting with the user with complete freedom to assign the corners to symbols. Let us assume that a user is interested in exploring various CGRs of given amino-acid sequence. She can choose the plot-style option in CGRex which brings up a dialog box as follows. In the space for corners, user can choose from 5 to 26 corners (3 and 4 are covered by triangle and square, 26 is set as the limit so that even plain English can be accepted by the software).

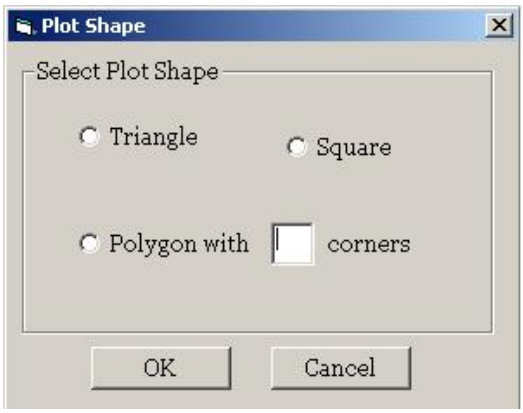


Fig 17: The plot-style dialog box of C-GRex

Suppose the user chooses 8 corners, then 20 amino acids have to be assigned to these 8 corners. C-GRex permits this to be done by a manual assignment or by an automatic assignment based on clustering and a choice of physico-chemical parameter for clustering. For clustering, it uses the well known *k-means* clustering. Fig 18 shows manual selection of amino-acid assignment for 8-cornered polygon. For clustering one of the 500 physico-chemical parameters can be chosen by the user. Fig 19 shows a 8-cornered CGR.

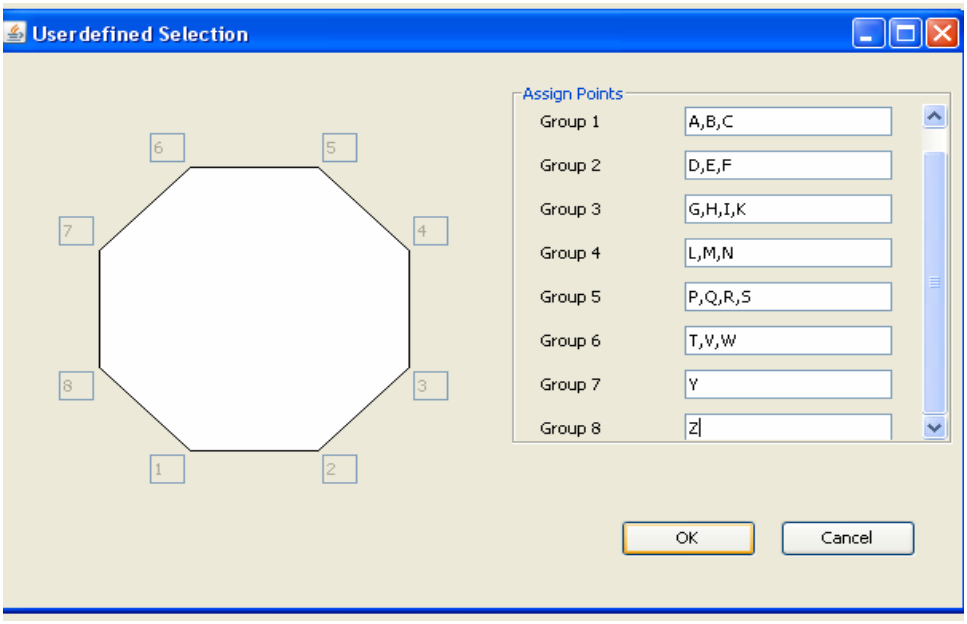


Fig 18: Selection of amino-acid assignment for 8-cornered polygon

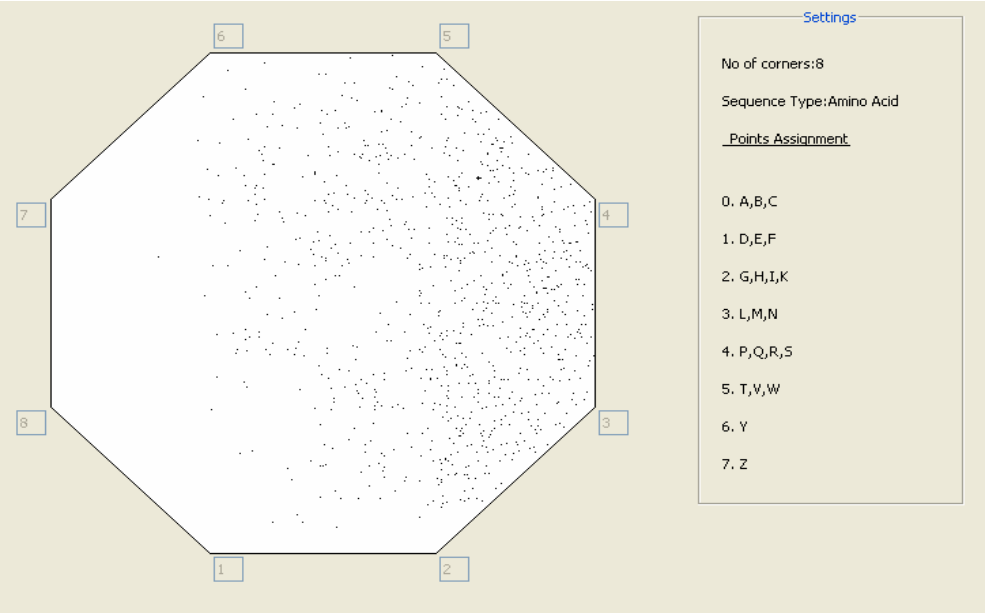


Fig 18: 8-cornered polygonal CGR

Concluding Remarks

We have tried to introduce the concept and application of chaos game representation of bio-sequences. A versatile tool also was introduced. CGRs are a very different way of analyzing bio-sequences. Only a very small number of studies have been conducted in this area. The scope of CGRs is enormous. With the availability of tools such as C-GRex, it is hoped that a lot more of studies can be initiated in this area.

REFERENCES

1. Jeffrey H. J., Chaos game representation of gene structure. *Nucleic Acids Res.* 18:2163–2170, 1990.

2. Deschavanne P. J., Giron A., Vilain J., Fagot G., Fertil B: Genomic signature: characterization and classification of species assessed by chaos game representation of sequences. *Mol. Biol. Evol.*, 16:1391-1399, 1999.

3. Kathleen A. Hill, Nicholas J. Schisler and Shiva M. Singh, Chaos Game Representation of coding regions of human globin genes and alcohol dehydrogenase genes of phylogenetically divergent species. *J Mol Evol* .35:261-269, 1992.

4. Chitra Dutta and Jyotirmoy Das, Mathematical Characterization of Chaos Game Representation New Algorithms for Nucleotide Sequence Analysis. *J Mol Biol.* 228:715-719, 1992.

5. Goldman N., Nucleotide, dinucleotide and trinucleotide frequencies explain patterns observed in chaos game representations of DNA sequences. *Nucleic Acids Res*, 21:2487-2491, 1993.

6. Almeida, J.S, Analysis of genomic sequences by chaos game representation. *Bioinformatics* 17: 429-437, *et al.* 2001.

7. Jijoy Joseph and Roschen Sasikumar, Chaos Game Representation for comparison of whole genomes. *BMC Bioinformatics* 7:243, 2006.

8. Deschavanne P. J., Giron A., Vilain J., Dufraigne C. and Fertil B., Genomic Signature is preserved in short DNA fragments, *IEEE*, 2000.

9. Giron A., Vilain J., Serruys C.,Brahmi D., Deschavanne P. J. and Fertil B., Analysis of parametric images derived from genomic sequences using neural network based approaches. *IEEE,* 1999.

10. Xin Huang, De-Shuang Huang, Hong-Qiang Wang and Xing-Ming Zhao, Representation of DNA sequences with multiple resolutions and BP neural network based classification. *IEEE*, 2004.

11. Andras Fiser, Gabor E. Tusnady and Istvan Simon, Chaos Game representation of protein structures. J. Mol. Graphics, 12:302-304, 1994.

12. Zu-Guo Yu, Vo Anh and Ka-Sing Lau, Chaos game representation of protein sequences based on the detailed HP model and their multifractal and correlation analyses. J. Theor Biol.,226(3):341-8, 2004.