arXiv:2303.09683v1 [q-bio.BM] 16 Mar 2023

Three dimensional chaos game representation of protein sequences

by

Annie Thomas

Graduate Program in Computer Science

Submitted in partial fulfillment
of the requirements for the degree of
Master of Science

Faculty of Graduate Studies
The University of Western Ontario
London, Ontario
May, 2005

© Annie Thomas 2005

THE UNIVERSITY OF WESTERN ONTARIO
FACULTY OF GRADUATE STUDIES

CERTIFICATE OF EXAMINATION

Supervisor	Examiners
______________________________	______________________________
Co-Supervisor	______________________________
______________________________	______________________________
Supervisory Committee	______________________________
______________________________	

The thesis by

Annie Thomas

entitled

THREE DIMENSIONAL CHAOS GAME REPRESENTATION OF PROTEIN SEQUENCES

is accepted in partial fulfillment of the
requirements for the degree of
Master of Science

__________________________	______________________________
Date	Chair of Examining Board

Abstract

A new three dimensional approach to the chaos game representation of protein sequences is explored in this thesis. The basics of DNA, the synthesis of proteins from DNA, protein structure and functionality and sequence alignment techniques are presented. The mathematical background needed for understanding the chaos game representation and fractal analysis are briefly discussed.

An account of the existing literature on the chaos game representation of DNA sequences and a detailed account of the chaos game representation of protein sequences in two dimensions with its advantages and limitations are presented. We explore a new three dimensional approach to the chaos game representation of protein sequences (3D-CGR) and study its ability *a)* to determine protein sequence similarity and differences, *b)* to study the effect of dinucleotide biases at amino acid level on the 3D-CGR derived protein homology, and *c)* to identify sequence similarity based on shuffled motifs that could be used for studying protein evolution due to exon shuffling.

Keywords: Chaos game, dinucleotide bias, protein homology, motifs, exon shuffling, fractal

Acknowledgments

This thesis would not have been possible without the support of many people. I thank my supervisor Dr.Lila Kari for her time, suggestions, patience and encouragement. I thank my co-supervisor Dr.Kathleen Hill for her time, sharing her expert knowledge in Molecular Biology and providing suggestions and interpretations at every stage of the thesis.

I also especially thank Dr.Fernando Sancho for his help with fractal geometry concepts in chapter 3 and reading the thesis. I am grateful to a very special person Dr. Rani Siromoney, for her support, encouragement and prayer. Also, I thank two very special people in my life, my aunt Anne and uncle Daya for being there for me always.

And finally, thanks to my parents for their unconditional love, my husband, brother and uncle Baga for their suggestions, support and encouragement, and numerous friends who endured this long process with me, always offering support and love.

Table of Contents

1 Introduction **1**

2 DNA and protein sequences **4**

- 2.1 DNA 4
  - 2.1.1 Structure 4
  - 2.1.2 Genetic code 5
- 2.2 Proteins 6
  - 2.2.1 Amino acid classification 7
  - 2.2.2 Structure 8
  - 2.2.3 Motif and domain 9
  - 2.2.4 Sequence alignment 10
    - 2.2.4.1 Pairwise alignment 10
    - 2.2.4.2 Alignment score 12
    - 2.2.4.3 Substitution score 12
    - 2.2.4.4 Insertion/deletion score 13
    - 2.2.4.5 Multiple sequence alignment 14
    - 2.2.4.6 Protein classification 15

2.2.5 Phylogenetic tree 16

2.2.5.1 Tree construction 17

3 Chaos and fractals for biological sequences 20

3.1 Fractal 20

3.1.1 Properties of fractals 21

3.1.2 Mathematical fractals 23

3.1.3 Example 24

3.1.4 Fractal dimension 25

3.2 Chaos game 27

4 Chaos game representation of DNA and protein sequences 30

4.1 Chaos game representation of DNA sequences 30

4.2 Chaos game representation of protein sequences 35

4.2.1 Chaos game using a 20 sided polygon 35

4.2.1.1 Method 35

4.2.1.2 Plotting 36

4.2.1.3 Properties 36

4.2.1.4 Limitation 37

4.2.2 Chaos game using a rectangle 38

4.2.2.1 Method 38

4.2.2.2 Plotting 38

4.2.2.3 Properties 39

4.2.3 Chaos game using a 12 sided polygon 39

4.2.3.1 Method 39

4.2.3.2 Plotting 39

4.2.3.3 Properties 39

4.2.4 Chaos game using a square 40

4.2.4.1 Method 41

4.2.4.2 Plotting 41

4.2.4.3 Properties 41

4.2.5 Summary 42

5 Three dimensional CGR of protein sequences 44

5.1 Three dimensional structure and amino acid mapping 45

5.1.1 Mapping 45

5.2 Chaos game on an icosahedron 46

5.3 Distance measure 48

5.4 Experimental objectives 49

5.5 Dataset for protein sequence analysis using 3D-CGR 50

5.6 Software 50

5.7 Results and discussion 51

5.7.1 Tree validation 51

5.7.1.1 Result and interpretation 53

5.7.2 Effect of mapping change 55

5.7.2.1 Results and interpretation 55

5.7.3 3D-CGR and CLUSTALW phylogenetic tree comparison . . . 60

        - 5.7.3.1 Branch length and protein family evolution 60
        - 5.7.3.2 Shuffled Motif detection 62
    - 5.7.4 Distance measure comparison 64
        - 5.7.4.1 Result and interpretation 64
- 5.8 Fractal analysis on protein sequences 67
    - 5.8.1 Octree 67
    - 5.8.2 Fractal dimensions of test sequences 67
        - 5.8.2.1 Result and interpretation 68

6 Conclusion **70**

Appendices **73**

References **77**

Vita **79**

List of Tables

3.1 Dimension of the Koch curve using length of line segments 25

3.2 Dimension of the Koch curve using area of triangles 26

4.1 Summary - Chaos game representation of protein sequences in two dimension 43

5.1 Protein Family, Swiss-Prot ID and Length of test protein sequences . 51

5.2 Protein Family, Swiss-Prot ID and Length of test protein sequences (continued from Table 5.1) 52

5.3 Pairwise comparisons of the three mapping strategies 59

List of Figures

2.1 DNA; A, T, G, C - nucleotides; Anti-parallel strands (5' to 3' and 3' to 5') bonded by base pairs A-T and G-C; 5

2.2 Transcription and translation 6

2.3 Genetic code - Triplet codon (example: UUU); 3 letter representation of amino acid (example: Phe) and corresponding 1 letter representation (example: F) 7

2.4 The basic structure of an amino acid 8

2.5 Protein secondary structure: α- Helix 9

2.6 Protein secondary structure: β- Sheets 10

2.7 Alignment between two sequences 11

2.8 Three possible alignments of two sequences 14

2.9 Phylogenetic tree - (B,C,D,E) - internal nodes; snake, lizard, bird, mouse, fish, frog - leaf nodes 16

2.10 Distance matrices 17

2.11 (a) A star tree (b) a tree with nodes A and B clustered 18

3.1 The middle third Cantor set generated by repeated removal of middle third of interval 21

3.2 The Koch curve generated by replacing the middle third of each interval by the other two sides of an equilateral triangle [3]:(Used) 21

3.3 The Sierpinski triangle generated by repeatedly removing the inverted equilateral triangle from the center of the initial equilateral triangle. . 22

3.4 Topological dimension of the Koch curve [3]:(Adapted) 26

3.5 Box counting dimension of the Koch curve [3]:(Adapted), r - length of the sides and N(r) - no. of boxes needed to cover the fractal 27

3.6 Sierpinski triangle using the chaos Game 29

4.1 Properties of CGR 32

4.2 CGR using 20-gon of DNA Polymerase Human Alpha Chain: Length = 1462 37

4.3 2D point representation using rectangle for sequentiality and composition of amino acids 38

4.4 CGR using 12 sided polygon; HEAT SHOCK PROTEIN 90 (hsp90) family 40

5.1 a) 2D and b) 3D Representation of an icosahedron with first and second nucleotide position of codon and its amino acid mapping respectively 45

5.2 Chaos game of protein sequence in three dimension 47

5.3 Phylogenetic tree validation - a) Sequence derivation b) Known tree c) Tree generated by 3D-CGR 54

5.4 Phylogenetic Tree generated by dinucleotide relatedness mapping using the Image distance 56

5.5 Phylogenetic Tree generated by random mapping 1 using the Image distance 57

5.6 Phylogenetic Tree generated by random mapping 2 using the Image distance 58

5.7 Phylogenetic tree generated by CLUSTALW 61

5.8 Shuffled motif detection 63

5.9 Shuffled motif detection by chaos and CLUSTALW 63

5.10 Phylogenetic tree generated by the Euclid distance 65

5.11 Phylogenetic tree generated by the Pearson distance 66

5.12 Spatial Subdivision using Octree 68

5.13 Fractal curve for the test data set 69

Chapter 1

Introduction

Protein sequence analysis is the key tool for understanding the evolution of proteins, sequence classification and for identifying conserved (they remain same across all species) positions crucial for the function and structure of proteins. This thesis is intended to study the protein sequence similarity using a holistic approach differing from the traditional sequence alignment one which is based on subsequences. The tool used for studying the sequence as a whole is known as *Chaos Game Representation* (CGR).

Jeffrey in 1990 [16] introduced the new tool CGR to visually represent DNA sequences. This new tool stimulated interest among researchers and CGR has since been used to explore the primary sequence organization of DNA and proteins. Although research on CGR has been widely explored, it has been limited to a two dimension representation. This thesis goes a step further to represent the CGR in three dimensions and to understand its potential as a tool in analysing protein sequence similarity.

Following the introduction, Chapter 2 gives the basics of molecular biology: The basic notions of DNA and protein sequences, the synthesis of protein from DNA and the representation of sequence/species relatedness through phylogenetic trees. It also looks into bioinformatic techniques such as sequence alignment and multiple sequence alignment with CLUSTALW, needed for the understanding of the thesis.

Chapter 3 looks into the mathematics behind the chaos game representation. This chapter explains what the chaos game is, how it could reveal the underlying patterns of any sequence. Also, this chapter gives a brief introduction into the mathematics of generating fractals.

In Chapter 4, the usefulness of the CGR explored in the past is explained. The literature on the CGR is grouped into two sections, one on DNA sequences and the other on protein sequences. Since the emphasis of the thesis is on protein sequences, a detailed analysis of all the previous work of the CGR of protein has been provided.

Chapter 5 deals with the new three dimensional approach to the CGR and results of the analysis of protein sequences using 3D-CGR. In the beginning, the objectives of the new approach are presented: to detect protein homology using 3D-CGR, to understand the impact of dinucleotide bias at the amino acid level on the 3D-CGR derived protein homology and to study the sequence relatedness to detect shuffled motifs. Following this, a description of the geometric solid icosahedron used for playing the chaos game, the method of representing amino acids on the icosahedron, and the chaos game in three dimensions are explained. Also, various distance measures used in the thesis and the spatial subdivision method used for determining the fractal dimension are explained in this chapter.

Next, the experimental objectives of the thesis are discussed. They are: (i) the validation of the phylogenetic trees obtained using 3D-CGR for detecting sequence relatedness, (ii) detection of the impact of dinucleotide bias at the amino acid level on the 3D-CGR derived protein homology, (ii) comparison of the trees generated by the 3D-CGR and CLUSTALW for sequence relatedness and shuffled motif detection, (iv) comparison of the effect of using various distance measures on the phylogenetic trees and (v) study the sequence relatedness using fractal patterns.

Following this, the methodologies used for performing the experiments are presented in detail. Our experiments reveal that the 3D-CGR can distinguish protein families and species relatedness within the families of the sequences. The 3D-CGR can detect shuffled motifs that cannot be detected by CLUSTALW. The detection of shuffled motifs by 3D-CGR could be a useful tool in studying protein evolution due to exon shuffling. Also, the significant difference in branch length between closely related sequences on comparison with CLUSTALW indicate that 3D-CGR could be used for measuring the amount of divergence between sequences within a family. The impact of dinucleotide bias at the amino acid level was seen in the branch length between some of the closely related sequences and in the branch order of the families. Finally, the sequence relatedness assessed using fractal curves and its limitation in studying protein homology is explained.

Lastly, Chapter 6 concludes the thesis by briefly presenting the major concepts discussed in each chapter, the novel outcome of the new approach and few words on the future work using 3D-CGR.

Chapter 2

DNA and protein sequences

2.1 DNA

Deoxyribonucleic acid (DNA) stores the genetic information that determines all activities of every living organism. The genetic information stored in DNA is passed from one generation to the next. DNA is made up of four *nucleotides:* guanine, adenine, thymine and cytosine, often referred as G,A,T,C. Nucleotides are organic structures made up of three subunits: phosphate, deoxyribose sugar and a nitrogenous base. The four nucleotides G,A,T and C have the same phosphate and sugar group but differ in their nitrogenous bases (fig 2.1).

2.1.1 Structure

Nucleotides are linked to each other by the phosphate group of one nucleotide with the deoxyribose sugar of another nucleotide forming a strand. The hydroxyl groups on the 5'(5th carbon)- and 3'(3rd carbon) of deoxyribose sugar link to the phosphate groups to form the DNA backbone. DNA is a double stranded molecule with the two strands in *anti-parallel* directions. The 5' end of one strand corresponds to the

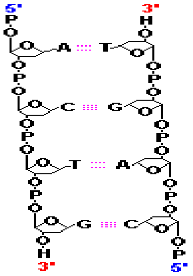


Figure 2.1: DNA; A, T, G, C - nucleotides; Anti-parallel strands (5' to 3' and 3' to 5') bonded by base pairs A-T and G-C;

3' end of the complementary strand and vice versa [17]. The strands are hydrogen bonded together by the base pairs A-T and G-C, the nucleotide A in one strand is hydrogen bonded with nucleotide T in the other strand and similarly nucleotide C in one strand is hydrogen bonded with nucleotide G on the other. For example, if one strand is 5'-ACTG-3' then the other strand is 3'-TGAC-5'. These strands twist together to form a double helical structure.

2.1.2 Genetic code

A strand of DNA is composed of coding and non-coding regions. A coding region refers to part of a DNA strand that contains the genetic information necessary for producing the amino acid chains of proteins responsible for performing many cellular functions. The non-coding regions on the other hand do not participate in amino acid chain formation. The process of generating proteins from DNA is know as *gene expression* and this process involves two stages, *transcription* and *translation* (fig 2.2).

In *transcription*, the nucleotide sequence that ultimately encodes a protein is used as a template to code for RNA (ribonucleic acid), known as mRNA (*messenger RNA*). RNA molecules are similar to DNA except the deoxyribose sugar in DNA is replaced by ribose in RNA, the base Thymine (T) in DNA is replaced by the base Uracil (U)

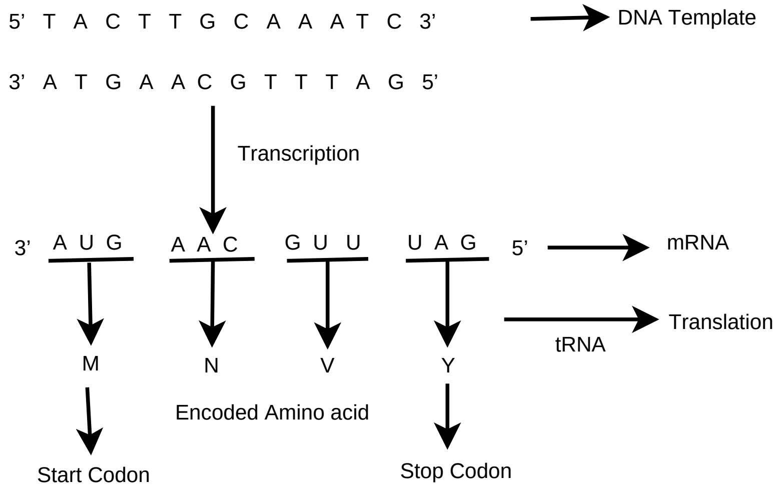
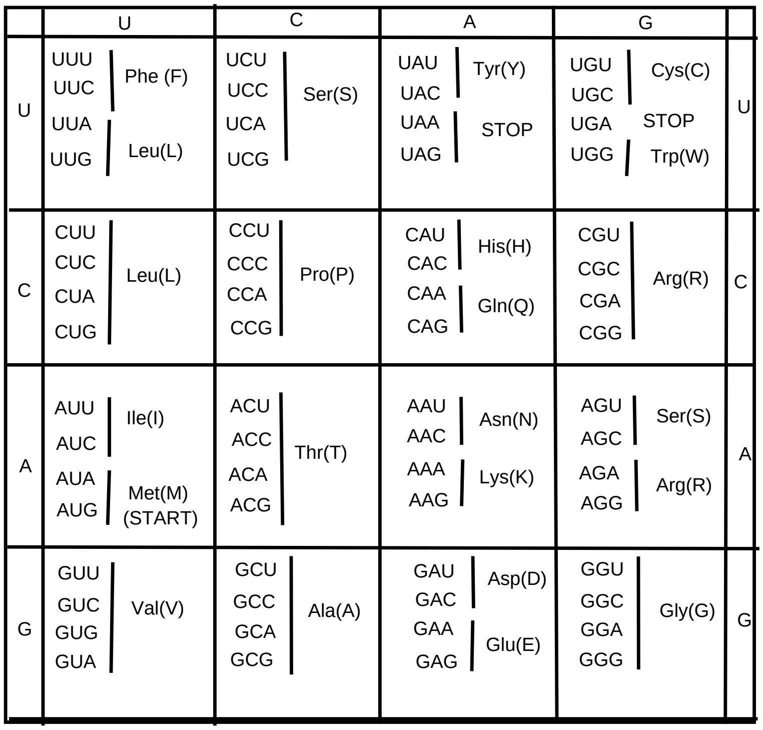

5' T A C T T G C A A A T C 3'
DNA Template
3' A T G A A C G T T T A G 5'
Transcription
3' A U G A A C G U U U A G 5'
mRNA
Translation
tRNA
M
N
V
Y
Encoded Amino acid
Start Codon
Stop Codon

Figure 2.2: Transcription and translation

in RNA and RNA is a single stranded structure. The process by which mRNA codes for protein is translation. The translation process is performed by a special type of RNA called tRNA(*transfer RNA*) that deciphers triplet nucleotide code of mRNA to specific amino acids. These triplet nucleotides are referred as *codons*. The relationship between codon and amino acid is referred as *Genetic code* (fig 2.3). Since DNA is composed of four nucleotides, there are 4^3=64 possible codons. The beginning of translation is signaled by a special codon called *start codon*. There are three codons that do not encode any amino acid but instead signal the end of translation, they are called *Stop Codons*. Since there are only twenty amino acids that make up proteins, more than one codon may refer to a particular amino acid.

2.2 Proteins

Proteins are long chain molecules built from twenty amino acids encoded by *Codons*. The twenty amino acids are Alanine(A), Arginine(R), Asparagine(N), Aspartic acid(D), Cysteine(C), Glutamic acid(E), Glutamine(Q), Glycine(G), Histidine(H), Isoleucine(I), Leucine(L), Lysine(K), Methionine(M),Phenylalanine(F), Proline(P), Serine(S), Thre-



	U	C	A	G	
U	UUU UUC Phe (F) UUA UUG Leu(L)	UCU UCC UCA UCG Ser(S)	UAU UAC Tyr(Y) UAA UAG STOP	UGU UGC Cys(C) UGA STOP UGG Trp(W)	U
C	CUU CUC CUA CUG Leu(L)	CCU CCC CCA CCG Pro(P)	CAU CAC His(H) CAA CAG Gln(Q)	CGU CGC CGA CGG Arg(R)	C
A	AUU AUC Ile(I) AUA AUG Met(M) (START)	ACU ACC ACA ACG Thr(T)	AAU AAC Asn(N) AAA AAG Lys(K)	AGU AGC Ser(S) AGA AGG Arg(R)	A
G	GUU GUC GUG GUA Val(V)	GCU GCC GCA GCG Ala(A)	GAU GAC Asp(D) GAA GAG Glu(E)	GGU GGC GGA GGG Gly(G)	G

Figure 2.3: Genetic code - Triplet codon (example: UUU); 3 letter representation of amino acid (example: Phe) and corresponding 1 letter representation (example: F)

onine(T), Tryptophan(W), Tyrosine(Y) and Valine(V).

Amino acids are small molecules containing an amino group, carboxyl group, hydrogen atom and a side chain (or R group) attached to the central carbon (or alpha carbon) [22]. Amino acids differ only in the side chain R. The amino acids are linked to one another by the central carbon of one amino acid with the amino group of the other amino acid forming a peptide bond. The amino acids in the long chains are referred as *residues*.

2.2.1 Amino acid classification

The similarity of amino acids is classified based on the chemical properties of the side chain (R group). Two major classifications are Hydrophobic (non-polar) - not

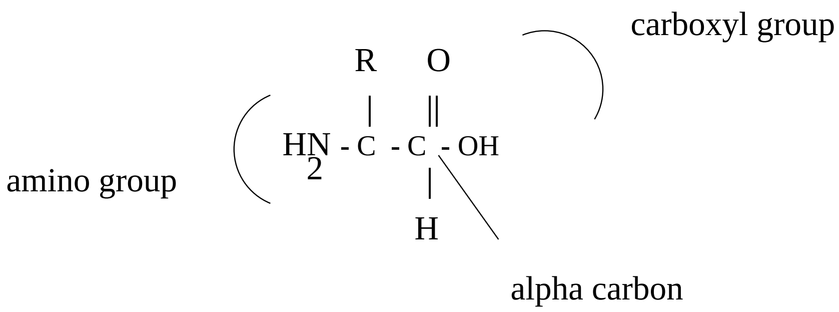


Figure 2.4: The basic structure of an amino acid

soluble in water: A, I, L, M, F, P, W and V and Hydrophilic (polar) - soluble in water: R, H, K, D, E, N, C, Q, G, S, T and Y. The polar class is further divided into: positively charged: R, H and K, negatively charged: D and E and uncharged: N, C, Q, G, S, T and Y. Amino acids are also grouped together based on their physio-chemical properties by which they could be substituted for one another in protein sequences with minimal apparent affect on the functionality of the proteins, known as *Conservative substitutions* [6]. The following are some of the groupings: ILVM, RK, DE, ST, AG and FY.

2.2.2 Structure

The sequence of amino acids forming a protein is known as the primary structure of the protein. In nature, protein molecules collapse and fold into a unique structure known as *native structure.* There are some patterns in the native structure that are quite common and found in many proteins, the location and direction of these patterns are called *secondary structures.* The three main secondary structures are α-helix, β-sheet and random coil. The α-helix is formed by the hydrogen bonds between the carbonyl group of the ith residue and the nitrogen group of the $(i+4)th$ residue (fig 2.5). An α-helix on average has 10 residues having 3.6 residues per turn [17]. β-sheets are strands of amino acid sequences forming hydrogen bonds between them. The bonds are formed between the carbonyl oxygen of amino acids from one strand

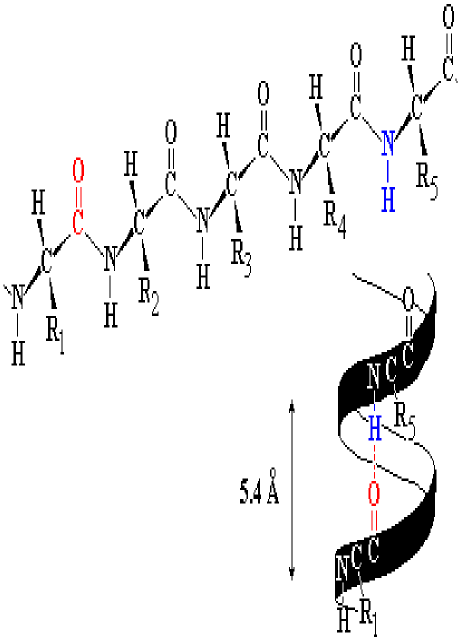


Figure 2.5: Protein secondary structure: α- Helix

with nitrogen groups of the other strand. The strands could be parallel or anti-parallel to each other. A β-sheet can consist of all parallel strands or all anti-parallel strands or can contain both. β-sheets usually consist of 5 to 10 residues [7](fig 2.6). Random coils are sequences of amino acids that connect α- helices and β-sheets and they are not regular structures, both in shape and size.

2.2.3 Motif and domain

A motif is a combination of a few secondary structures [7]. For example, *helix - random coil - helix* is a motif. A domain is a more complex combination of secondary structures having a specific function by binding to external molecules (i.e, DNA), therefore referred to as active site. A domain could maintain its characteristic structure even if separated from the original protein [7]. A protein can have several motifs, which can combine to form specific domains, and one or more domains together form the protein's tertiary structure.

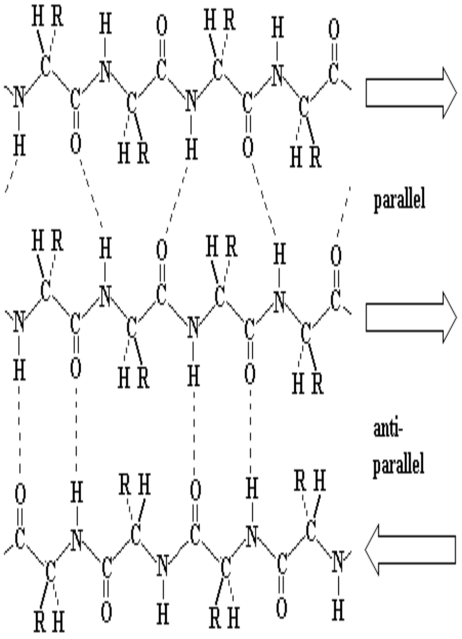


Figure 2.6: Protein secondary structure: β- Sheets

2.2.4 Sequence alignment

2.2.4.1 Pairwise alignment

The number of proteins that exists in nature is very large but these proteins could be classified based on the sequence pattern, structure and their functionality. Proteins that have similarities in their sequences are believed to be derived from an common ancestor. Therefore, determining the similarities of sequences would help to understand the evolution of proteins. Sequence alignment is a technique used to compare biological sequences in order to study the similarities and differences to find the origin of evolution between them. Sequences that share sequence similarity would differ from one another in some sequence positions due to

1. substitution that replaces one nucleotide/amino acid with another.

2. an insertion that adds one or more nucleotides/amino acids.

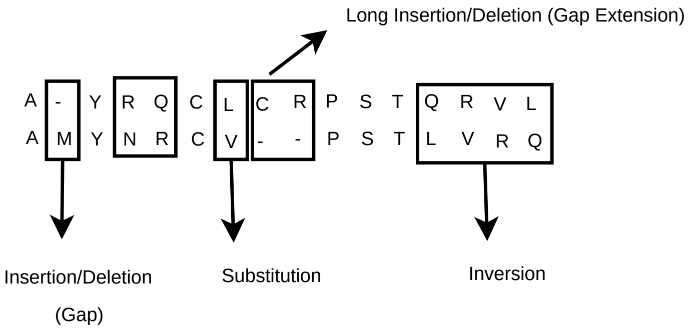

Long Insertion/Deletion (Gap Extension)
A - Y R Q C L C R P S T Q R V L
A M Y N R C V - - P S T L V R Q
Insertion/Deletion
(Gap)
Substitution
Inversion

Figure 2.7: Alignment between two sequences

3. deletion that deletes one or more nucleotides/amino acids.

4. inversion that reverses the orientation of subsequences.

Given two sequences $a = a_1 \ldots a_m$ and $b = b_1 \ldots b_n$ over the alphabet Σ. An alignment of the sequences a and b is a pair of sequences $a_1^{'} \ldots a_l^{'}$ and $b_1^{'} \ldots b_l^{'}$ of equal lengths defined over the extended alphabet $\Sigma' = \Sigma \cup$ {-} containing blank character '-' such that the string a' is derived from a and string b' is derived from b. The alignment is denoted by

$$a'_1 a_2^{'} \ldots a_l^{'}$$

$$b'_1 b_2^{'} \ldots b_l^{'}$$

The length l of an alignment $(a^{'}, b^{'})$ is restricted to $max\{m, n\} \leq l \leq m + n$, since column pairs are not allowed. In sequence alignment, the blank character '-' is referred as a gap denoting insertion/deletion referred to as an *indel* (fig 2.7).

2.2.4.2 Alignment score

Scoring schemes are used to evaluate the alignment between sequences. There are two scores that are used in alignment evaluation:

1. substitution score
2. insertion and deletion score.

2.2.4.3 Substitution score

Substitution scores are matrices developed based on experimental data that encode the expected evolutionary change at the amino acid level. One of the widely used substitution scores in amino acid alignment is Point Accepted Mutation Matrix (PAM) developed by Dayhoff in 1978 [17]. The PAM matrix M contains the probability of amino acid i replaced by amino acid k in a certain evolutionary time period [17]. For example, 1PAM represents, 1 substitution per 100 residues therefore, nPAM is n accepted substitutions in 100 residues (i.e, probability that amino acid i will be replaced by amino acid k in sequences separated by nPAMs of evolutionary distance). 1PAM is generally used for closely related sequences and higher PAM matrices are used for distantly related sequences (highly divergent). 1PAM was obtained by calculating the substitution probabilities based on 71 groups of sequences with $>$ 80% sequence identity [9]. The entries of 1PAM matrix M^1 is calculated as

$$M_{ij} = \log \frac{\frac{m_j * F_{ij}}{\Sigma_i F_{ij}}}{f_i}$$

where, the relative mutability m_j is the number of times the amino acid i is substituted, F_{ij} is the number of times amino acid i is substituted by amino acid j and f_i is the frequency of amino acid i. Once M is known, the matrix M^n gives the probability of any amino acid mutating to any other amino acid in nPAM units. The PAM matrix M^n for $n > 1$ can be obtained by matrix multiplication of M^1.

$$M^2 = M^1 * M^1$$

$$M^3 = M^2 * M^1$$

$$\vdots$$

$$M^n = M^{(n-1)} * M^1$$

Another substitution matrix widely used is BLOSUM (Blocks Substitution Matrix), developed by Henikoff and Henikoff in 1992 based on known alignments of more diverse sequences [9]. The matrix is based on the ungapped alignment (block) from the sequence alignment. Like the PAM matrix, different BLOSUM scoring matrices are obtained for different evolutionary distances. For example, BLOSUM80 matrix represents sequences with approximately 80% identity in sequence alignment.

The relationship between the two substitution matrices is given as, BLOSUM with low percentage corresponds to PAM with large evolutionary distances (i.e PAM250 → BLOSUM45, PAM120 → BLOSUM80). Lower numbered BLOSUM matrices are appropriate for more distantly related sequences and lower numbered PAM matrices are appropriate for more closely related sequences.

2.2.4.4 Insertion/deletion score

Insertion and deletion scores are calculated based on the gap opens (single insertion/deletion) and gap extensions (long insertion/deletion)(fig 2.7). Since long insertions and deletions are expected less than single insertion and deletion, they are penalized less.

```
ALIGN1:
V - E I T G E I S T
P R E - T E R I - T

ALIGN2:
V E I T G E I S T
P R E T - E R I T

ALIGN3:
- V E I T G E - I S T
P R E - T - E R I - T
```

Figure 2.8: Three possible alignments of two sequences

There could be more than one possible alignment between the sequences (fig 2.8) but, the best alignment reflects the evolutionary relationship between homologous sequences. In order to find the best alignment, exhaustive search of all possible alignments are not feasible. Therefore, alignment algorithms use a dynamic program approach to break the problem into subproblems and using partial results to compute the final answer [17].

2.2.4.5 Multiple sequence alignment

Multiple sequence alignment is an extension of pairwise alignment to align more than two sequences simultaneously. Multiple sequences are aligned in order to provide insight into

1. characteristics of protein families
2. identify motifs in sequences with a conserved biological function
3. identify motifs of new proteins that would help to determine biological function.

There are several multiple sequence alignment algorithms. Algorithms with a heuristic approach are more commonly used than the ones that give optimal alignment because optimal alignments are practical only for a handful of sequences. Heuristic algorithms are rapid, require less memory space and offer good performance when used on relatively well conserved homologous sequences. One of the most common heuristic approaches is Progressive alignment used by ClustalW [22]. The Progressive alignment algorithm works as follows:

1. determine pairwise alignment between all pairs of sequences and their alignment scores.
2. construct a guide tree (phylogenetic tree - see section 2.2.5) using the alignment score.
3. Align sequences according to the guide tree by aligning the most closely related sequences using sequence-sequence alignment first, then profile-sequence alignment(between an alignment and a sequence) and finally, profile-profile alignment(between alignments).

In ClustalW substitution matrices and gap penalties vary at different stages of alignment depending on the divergences of the sequences to be aligned. Gap penalties depend on the substitution matrices, the similarity of the sequences, and the length of the sequences in order to introduce new gaps in the coil region rather than in secondary structure regions [22]. One the drawbacks of progressive alignment is that it is unreliable when highly divergent sequences are aligned.

2.2.4.6 Protein classification

A set of proteins that share a common evolutionary origin reflected by their relatedness in function, which is usually demonstrated by similarities in sequence, or in primary, secondary, or tertiary structure is known as Protein Family[1]. Similarly, *superfamiliy* is collection of protein familes that have same overall domain structure (i.e, same domain in same order) [2]. There are several protein family databases such

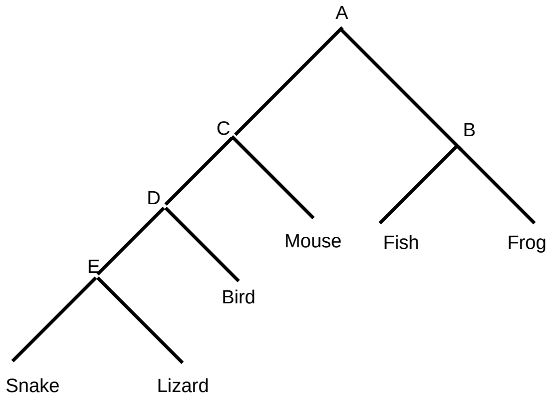


Figure 2.9: Phylogenetic tree - (B,C,D,E) - internal nodes; snake, lizard, bird, mouse, fish, frog - leaf nodes

as Prosite and Pfam. Proteins are also classified based on the secondary structure similarities. Some databases that group proteins based on structure classifications are SCOP (Structural Classification of Proteins) and CATH (Class, Architecture, Topology and Homologous superfamily).

2.2.5 Phylogenetic tree

Evolutionary relationships between species/sequences (taxa) are called phylogenies and they are graphically represented by trees known as *Phylogenetic trees.* A phylogenetic tree is made up of nodes and branches. Nodes represent distinct taxonomical units. Nodes at the tips of the branches are *terminal nodes* and the internal nodes represent an inferred common ancestor (fig 2.9). Branch lengths indicate the amount of divergence between different species/sequences, longer the lines between two species/sequences, the greater the difference between them. Branch order refers to the genealogy of the organism. If two species/sequences are closer to the branch then closer their relationship.

2.2.5.1 Tree construction

One of the widely used tree constructions is the neighbor-joining Method based on distance matrices. The distance matrix consists of estimated distance between all pairs of taxas or operational taxon (OTU) (fig 2.10a) calculated from any method (say sequence alignment) used to find similarities and differences between sequences. The neighbor-joining method starts with a star tree having central node X of degree m (number of neighbor of X). The new internal nodes are successively created and the degree of X is reduced by 1 in each cycle. The iteration stops when the degree of X becomes 3.

OTU

OTU	A	B	C	D	E
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8

(a) Distance matrix of fig 2.11a

OTU

OTU	A	B	C	D	E
B	-13				
C	-11.5	-11.5			
D	-10	-10	-10.5		
E	-10	-10	-10.5	-13	
F	-10.5	-10.5	-11	-11.5	-11.5

(a)

OTU

OTU	U	C	D	E
C	3			
D	6	7		
E	5	6	5	
F	7	8	9	8

(b)

(b) (i) Distance Matrix for fig 2.11a based on neighbor-joining method (ii) Distance matrix for fig 2.11b

Figure 2.10: Distance matrices

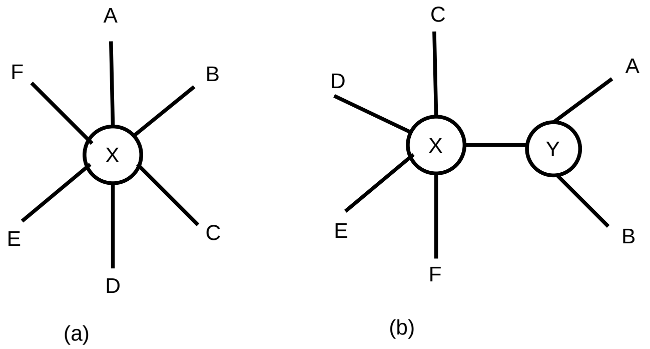


Figure 2.11: (a) A star tree (b) a tree with nodes A and B clustered

The construction of phylogenetic tree by neighbor-joining method ([19]) using distance matrix is explained using the following steps.

1. start with a star tree (fig 2.11a).

2. calculate the net divergence $r(i)$ for each of the OTU's from all the OTU's using the distance matrix from fig 2.10a, net divergence for A is given as

$$r(A) = 5 + 4 + 7 + 6 + 8 = 30$$

3. calculate a new distance matrix (fig 2.10b(i)) using the formula

$$M(ij) = d(ij) - [r(i) + r(j)]/(N - 2)$$

   where N is the number of OTU's, N=6 for fig 2.11a and $d(ij)$ is the distance between i and j, $d(AB) = 5$.

4. Choose two OTU's (A and B from fig:2.11a) from the distance matrix (fig:2.10b(i)) that has the smallest distance and create a new internal node Y that connects A, B and X.

5. calculate new branch length for A and B from Y using

$$S(AY) = d(AB)/2 - [r(A) - r(B)]/2(N - 2), S(BY) = d(AB) - S(AY)$$

also, calculate the distance between Y to all the other nodes. The new distance matrix (fig 2.10 b(ii)) is created for fig 2.11b.

6. repeat process from step 2 until the degree of X becomes 3.

Chapter 3

Chaos and fractals for biological sequences

3.1 Fractal

A *fractal* is a geometric figure that does not become less complex when you break it down into smaller and smaller parts. This implies, a fractal is scale invariant. The word fractal was coined by Mandelbrot from the Latin word *fractus* meaning broken or uneven, to describe objects that are too irregular to fit into traditional geometry [10]. For example, if we take a straight line and remove the middle third from it, we obtain two small straight line segments and if we continue this process repeatedly for smaller segments, in the limit we obtain a fractal called the Cantor set (fig 3.1). Similar examples are the Koch curve and the Sierpinski triangle (fig 3.2, fig 3.3). In all of the above examples, the same structure has been repeated at all scales. Therefore, these fractals are known as *self-similar* fractals. It is not necessary for fractals to be self-similar. For example a coastline, the human body, or the sky on a partly cloudy day are fractals without being self-similar[5].

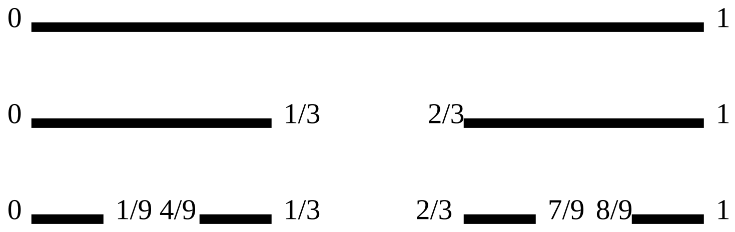


Figure 3.1: The middle third Cantor set generated by repeated removal of middle third of interval

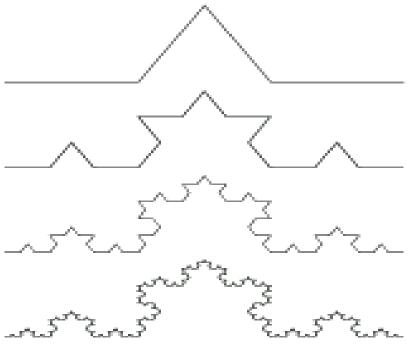


Figure 3.2: The Koch curve generated by replacing the middle third of each interval by the other two sides of an equilateral triangle [3]:(Used)

3.1.1 Properties of fractals

Based on the examples above, we have the following properties of fractals [10]. A fractal set F

1. has a fine structure i.e detail on arbitrarily small scale;
2. is too irregular to be described by traditional geometry;
3. often has some form of self-similarity, perhaps approximate or statistical. For example, any part of the Cantor set F in the interval $[0, \frac{1}{3}]$ and the interval $[\frac{2}{3}, 1]$ are geometrically similar to F. Figure 3.1, 3.2 and 3.3 contain copies of itself at different scales;
4. is in most cases defined recursively. For example, the Cantor set is generated

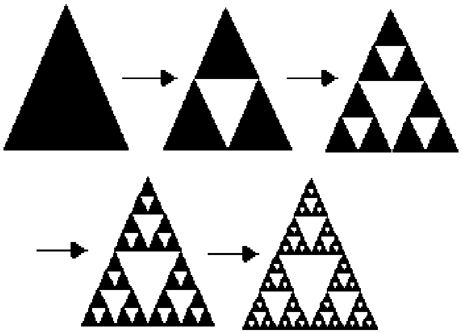


Figure 3.3: The Sierpinski triangle generated by repeatedly removing the inverted equilateral triangle from the center of the initial equilateral triangle.

by repeatedly removing the middle third of intervals and the Sierpinski triangle is obtained by repeatedly removing the inverted triangle.

5. usually has the fractal dimension (see 3.1.1) greater than the topological dimension (see below) or Covering Dimension.

The topological dimension is defined as

A covering of a subset $\mathcal{S}$ of a topological space $\mathcal{X}$ is a collection $\mathcal{C}$ of open subsets in $\mathcal{X}$ whose union contains all of $\mathcal{S}$

A refinement of a covering $\mathcal{C}$ of S is another $\mathcal{C}'$ of $\mathcal{S}$ such that each set $\mathcal{B}$ in $\mathcal{C}'$ is contained in some set A in $\mathcal{C}$. The idea is that the sets in $\mathcal{C}'$ are in some sense "smaller" than those in $\mathcal{C}$ and provide a more finely detailed coverage of $\mathcal{S}$.

A topological space $\mathcal{X}$ has topological dimension m if every covering $\mathcal{C}$ of $\mathcal{X}$ has a refinement $\mathcal{C}'$ in which every point of $\mathcal{X}$ occurs in at most $m + 1$ sets in $\mathcal{C}'$, and m is the smallest such integer.

3.1.2 Mathematical fractals

Mathematically, a great variety of fractals could be generated by iterating a collection of transformations, forming what is known as an Iterated Function System (IFS). If all the transformations in an IFS are *contractive mappings* then iterating these transformations would definitely converge to a unique shape. A contractive mapping is a transformation f that reduces the distance between every pair of points. That is, there is a number s between 0 and 1 and

$$dist(f(x,y), f(x',y')) \leq s * dist((x,y),(x',y'))$$

Formally, a contractive mapping, an IFS and an affine transformation are defined as

Definition 1: A transformation f: $\mathcal{X} \to \mathcal{X}$ on a metric space $(\mathcal{X}, d)$ is called a contractive mapping if there is a constant $0 \leq s \leq 1$ such that $d(f(x), f(y)) \leq s * d(x,y)$ $\forall x, y \varepsilon \mathcal{X}$, where d is the Euclidean distance and any s the contraction factor of f.

Definition 2: Let $\mathcal{T}_1$, $\mathcal{T}_2$, ..,$\mathcal{T}_N$ be a family of contractions on $\Re^k$ and $\mathcal{S}$ be a closed bounded subset of $\Re^k$. Then the system $\mathcal{S}$: $\mathcal{T} = U_{i=1}^{n}\, \mathcal{T}_i$ is called an *iterated function system.*

Definition 3: An *affine transformation* of $\Re^n$ is achieved by applying a linear transformation followed by a translation. An affine transformation $\mathcal{T}$ of $\Re^n$ is represented in matrix-vector form as

$$\mathcal{T}(x) = \mathcal{A}x + b, x \in \Re^n \text{ and } \mathcal{A} \text{ is a transformation matrix}$$

3.1.3 Example

The following example explains the mathematically generated self-similar fractal called the Sierpinski triangle (fig 3.3). Consider $\mathcal{E}_0$ to be a unit triangle. The contractive mapping for producing the Sierpinski Triangle is given by three affine transformations. The three affine transformations for the Sierpinski triangle are

$$\mathcal{T}_1\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = 1/2\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathcal{T}_2\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = 1/2\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1/2 \\ 0 \end{bmatrix}$$

$$\mathcal{T}_3\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = 1/2\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1/4 \\ \sqrt(3)/4 \end{bmatrix}$$

with a contractive factor $\frac{1}{2}$. Applying $\mathcal{T}_1(\mathcal{E}_0)$, $\mathcal{T}_2(\mathcal{E}_0)$ and $\mathcal{T}_3(\mathcal{E}_0)$ to the triangle $\mathcal{E}_0$ produces three smaller equilateral triangles $\mathcal{E}_1$. Similarly, applying all the three transformations to all the three vertices of each of the smaller triangles $\mathcal{E}_1$ produces nine smaller triangles $\mathcal{E}_2$. The iterative application of the three affine transformations produces smaller and smaller triangles resulting in the Sierpinski triangle (fig 3.3). The iterative scheme is

$$\begin{gathered}
\mathcal{E}_0 = \text{a compact set} \\
\mathcal{E}_1 = \mathcal{T}(\mathcal{E}_0) = \mathcal{T}_1(\mathcal{E}_0) \bigcup \mathcal{T}_2(\mathcal{E}_0) \bigcup \mathcal{T}_3(\mathcal{E}_0) \\
\mathcal{E}_2 = \mathcal{T}(\mathcal{E}_1) = \mathcal{T}_1(\mathcal{E}_1) \bigcup \mathcal{T}_2(\mathcal{E}_1) \bigcup \mathcal{T}_3(\mathcal{E}_1) \\
\vdots \\
\mathcal{E}_n = \mathcal{T}(\mathcal{E}_n - 1) = \mathcal{T}_1(\mathcal{E}_n - 1) \bigcup \mathcal{T}_2(\mathcal{E}_n - 1) \bigcup \mathcal{T}_3(\mathcal{E}_n - 1)
\end{gathered}$$

This sequence would converge to a unique shape (the Sierpinski triangle) called an *attractor*. Since all the transformations are applied in each step, this approach is a

n	length of the segment	number of segments	Length of the Koch curve $\mathcal{L}_n$
0	1	1	$\mathcal{L}_n = 1$
1	1/3	4	$\mathcal{L}_n = 4/3$
2	$1/9 = 1/3^2$	$16 = 4^2$	$\mathcal{L}_n = 16/9 = (4/3)^2$
3	$1/27 = 1/3^3$	$64 = 4^3$	$\mathcal{L}_n = 64/37 = (4/3)^3$
...	...	...	...
n	$1/3^n$	4^n	$\mathcal{L}_n = (4/3)^n$

Table 3.1: Dimension of the Koch curve using length of line segments

deterministic approach.

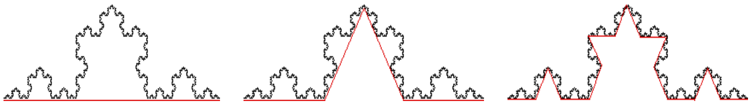
3.1.4 Fractal dimension

The *dimension* is a topological measure of spacial extent. For example, a point has a dimension 0, a line has a dimension 1, a square has a dimension 2 and a cube has a dimension 3. However, topological dimension cannot be used to measure fractals, because, for example, when trying to measure the length of the Koch curve using line segments, as the number of line segments needed to measure the length increases, the length of the Koch curve increases, leading to infinity (fig 3.4). Table 3.1 lists the increasing length of the Koch curve as the number of line segments needed to measure the Koch curve increases (The initial line is of length 1).

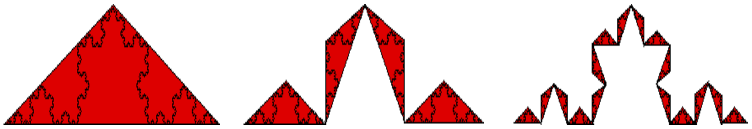
Similarly, when trying to compute the area of the Koch curve by covering it with triangles, as the number of triangles needed to cover the Koch curve increases, the area of the Koch curve decreases leading to zero (fig 3.4). The initial triangle is an isosceles triangle with base 1 and height $\sqrt{(3)}/6$. In the next stage the Koch curve is covered with three smaller triangles whose base and height are reduced by 1/3 compared to the initial triangle. As the process continues, at every stage the area of the triangles are reduced leading to zero. In the above metioned examples the

n	Area of the triangle	number of triangles	Area of the Koch curve $\mathcal{A}_n$
0	$\sqrt(3)/12$	1	$\mathcal{A}_n = \sqrt(3)/12$
1	$(\sqrt(3)/12) * (1/9)$	4	$\mathcal{A}_n = (\sqrt(3)/12) * (4/9)$
2	$(\sqrt(3)/12) * (1/81)$	$16 = 4^2$	$\mathcal{A}_n = (\sqrt(3)/12) * (16/81)$
...	...	...	...
n	$(\sqrt(3)/12) * (1/9)^n$	4^n	$\mathcal{A}_n = (\sqrt(3)/12) * (4/9)^n$

Table 3.2: Dimension of the Koch curve using area of triangles



Topological Dimension of the Koch Curve using Line Segements



Topological Dimension of the Koch Curve using Triangular Area

Figure 3.4: Topological dimension of the Koch curve [3]:(Adapted)

dimension of the Koch curve leads to either infinity or zero producing no limiting value. Measuring an object in an dimension lower than the object produces infinity and higher than the object produced zero. This implies the dimension of the Koch curve is > 1 but < 2 a fractional value. Table 3.2 lists the area of the Koch curve for various sizes of the triangle needed to cover the Koch curve.

Therefore, we use a better dimension called *Box counting dimension* [3] to calculate the dimension of fractals. The box counting dimension of a fractal is calculated by covering the fractal with boxes and calculating the number of boxes $\mathcal{N}_r$ of size r needed to cover the fractal (fig 3.5). The size of a fractal set is measured by its dimension(d_f) given as:

$$d_f = \lim_{r \to 0} \frac{\log \mathcal{N}(r)}{\log r}$$

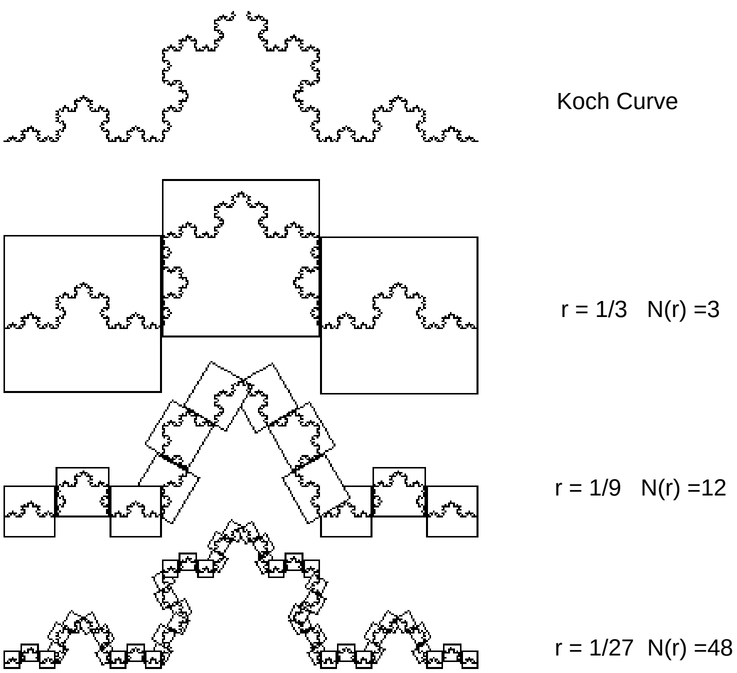


Figure 3.5: Box counting dimension of the Koch curve [3]:(Adapted), r - length of the sides and N(r) - no. of boxes needed to cover the fractal

For better approximation, the number of boxes needed to cover the fractal for various box sizes r is calculated and the fractal dimension is the slope of log-log plot of size of the boxes against the number of such boxes needed to cover the fractal.

Example: Figure 3.5 depicts the number of boxes needed to cover the Koch curve for varying box sizes. The slope of a log-log plot of size of the boxes against the number of boxes needed to cover the Koch curve gives the fractal dimension of the curve.

3.2 Chaos game

Another approach to generate fractals is the random approach of *Chaos Game.* Consider a triangle and the three transformations defined in section 3.1.3. Let the initial set be a single point. Assume that, at each step, one of the three transformations is randomly chosen and applied. Therefore, the output at each stage is a single point. After some transient behavior the points generated form a fractal - Sierpinski triangle. The iteration scheme is

$$y_0 = \text{start point}$$

$$y_1 = \mathcal{T}_1(y_0) \text{ or } \mathcal{T}_2(y_0) \text{ or } \mathcal{T}_3(y_0)$$

$$\vdots$$

$$y_n = \mathcal{T}_1(y_{n-1}) \text{ or } \mathcal{T}_2(y_{n-1}) \text{ or } \mathcal{T}_3(y_{n-1})$$

For example, consider a triangle with vertices (0,0),(1,0) and $(1/2, \sqrt{(3)}/2)$(fig 3.6). The center of the triangle is chosen as the starting point. Choose randomly one of the transformation, say $\mathcal{T}_2$, and apply it to the center point: This would produce a point (say p), which is the midpoint of the center and the vertex (1,0). Again, randomly choose another transformation, say $\mathcal{T}_3$, and apply it to the previously produced point (p): The new point produced is the midpoint of the point p and the vertex $(1/2, \sqrt{(3)}/2)$. As this process is continued for large number of times, the image produced looks like a Sierpinski triangle. The chaos game can be played with any number of vertices, four, five, six and so on. If the vertices are not selected uniformly at random in chaos game then various patterns are produced. This reveals some kind of order in the sequence.

Intuitively, order (non-randomness) means the sequence has a structure. Therefore, the chaos game can be a used as a tool to study the non-randomness of any sequence visually. If the chaos game can be extended to play on DNA or protein sequences then various patterns/structure in them could be revealed.

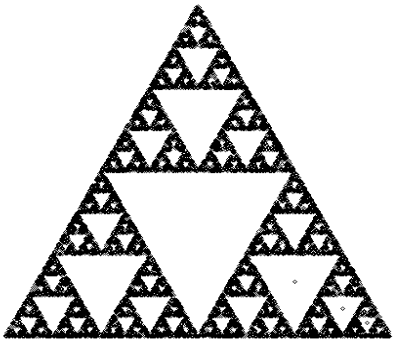


Figure 3.6: Sierpinski triangle using the chaos Game

Chapter 4

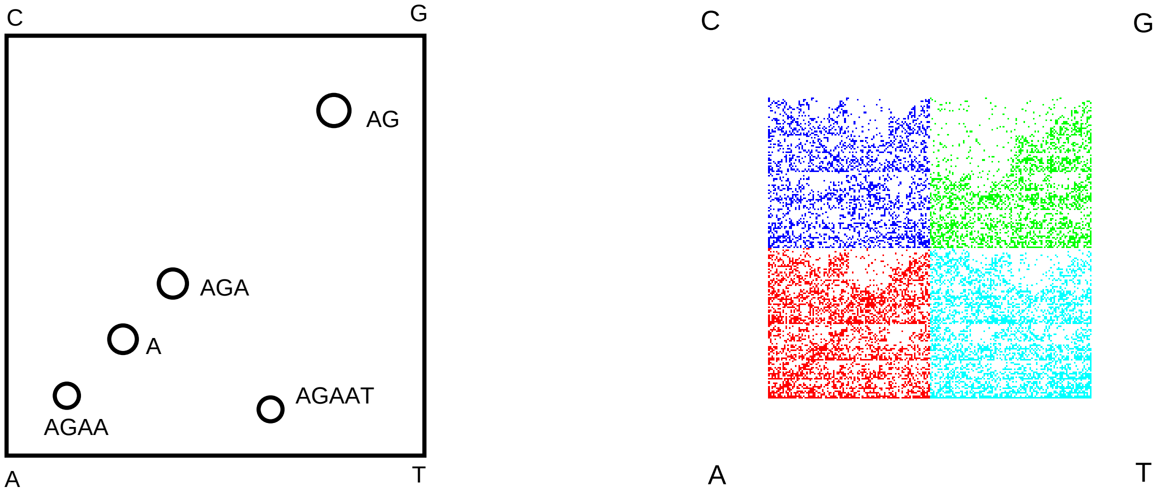
Chaos game representation of DNA and protein sequences

4.1 Chaos game representation of DNA sequences

Chaos game representation (or CGR) is a visual representation technique used, among others to study the patterns in gene structures [16]. The chaos game is played on a square using IFS. The nucleotide bases (A,C,G,T) correspond to the four vertices; the first point is plotted halfway between the center of the square and the corresponding vertex of the first nucleotide base in the sequence, and each subsequent point is plotted halfway between the previous point and the vertex of the subsequent nucleotide base from the sequence (fig 4.1a). The chaos game when applied to DNA sequences showed fractal structures (fig 4.1b , fig 4.1c). Figure 4.1b represents the attractor of chaos game of (*Human Beta Globin-HUMBB*). The 'double scoop'(sparse regions on fig 4.1b) structure in the attractor is due to the paucity of points in various regions of the square [16]. Similarly, when A & T and C & G were plotted opposite to one another, the paucity of points is represented by squares instead of 'double scoop' (fig 4.1c). The patterns are repeated on various scales in the attractor exhibiting the property of self-similarity. These patterns revealed the non-random nucleotide composition in DNA sequences. In a CGR, any *ith* point in the attractor uniquely represents the *ith* long initial subsequence of the sequence. The attractor depicts the base composition of the gene sequence [16]; the square, when divided into four, sixteen, sixty four sub-quadrants and so on represents the mono, di- and tri- etc. nucleotides subsequences (fig 4.1d).

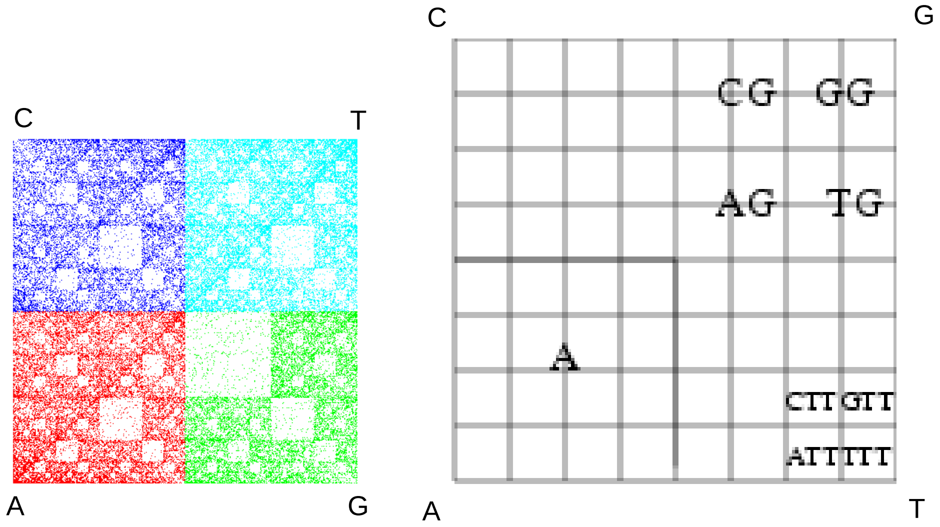
Using a CGR, the presence or absence of a sequence of nucleotides in any DNA sequence can be mathematically characterized [8]. Dutta et.al [8] gave algorithms to find the subsequences corresponding to any given point in CGR, and to simulate CGR patterns of a sequence by predicting the order of nucleotides using the probability of occurrences of di or tri nucleotides [8]. These algorithms are presented in the appendices A and B.

Hill et.al in 1992 [14] examined the coding region of the CGR's of seven *globin* genes from human and the CGRs of 29 closely related *alcohol dehydrogenase* genes from phylogenetically divergent species. CGRs of Human *globin* coding regions and the CGR of the entire Human Globin gene (coding and non-coding) are visually similar to each other but the self-similarity was not readily visible in the individual sequences due to smaller number of points (2000 nucleotides). Also, the di-nucleotide frequencies of the *globin* genes from human are not significantly different from one another. Therefore, the di-nucleotide frequencies partially accounted for the self-similarity in the CGR patterns [14]. CGRs from coding regions of *alcohol dehydrogenase* gene from the same species (ADH1, ADH2 etc) are similar to one another. Also, ADH CGRs of closely related species such as human, rodent, primate were similar to one another[14]. CGRs of unrelated genes from the same species are more similar to one another than sequences from unrelated species. Therefore, Hill et.al in 1992 [14] said that the CGR patterns reflect genome type specificity which could be the result of



(a) CGR-AGAAT; Plotting "AGAAT"- 'A' is plotted half-way between the center and the vertex representing A, 'G' is plotted half-way between the previous point plotted and the vertex representing G, 'A' is plotted half-way between the previous point and the vertex representing A, next 'A' is plotted halfway between the previous point and the vertex representing A and similarly 'T' is plotted

(b) CGR-HumanBetaGlobin; Nucleotides A,C,G,T are represented as red, violet, green and blue; paucity of dinucleotide CG forms a 'Double Scoop' pattern



(c) CGR-HumanBetaGlobin; paucity of dinucleotide CG forms a square pattern as vertices C and G are opposite to each other.

(d) Mono-, di-, and Tri- nucleotide configuration

Figure 4.1: Properties of CGR

mutation rates of mono, di, tri nucleotide bases and so on and said the evolution of a gene and its coding sequences should not be examined in isolation, genome specific differential mutation in di-nucleotides or oligonucleotide should be taken into account [14]. Hill et.al in 1997 [15] studied 28 complete mitochondrial genomes using CGR. They said, the global DNA sequence organizationof mitochondrial genomes is species-type specific. The species-type specific patterns appear primarily due to the dinucleotide composition.

CGRs generated from simulated sequences using a first-order Markov-chain probability matrix for *Human Beta Globin* and second-order Markov-chain probability matrix for *Bacteriophage Lambda* were similar to CGRs of the original Human Beta Globin and Bacteriophage Lambda sequences [12]. The first-order and second-order Markov probability matrices were obtained from calculating the dinucleotide and trinucleotide frequencies directly from the DNA sequences without reference to the CGR. Therefore Goldman in 1993 [12], suggested that CGR is a particular case of Markov-Chain model and CGR is only limited to represent mono, di and tri nucleotide representation of the sequences [12].

The Markov chain model is limited to produce only integer number of bases whereas the frequency matrix obtained from CGR (FCGR) can produce non-integer number of bases [4] in contradiction to statement by Goldman in 1993 [12]. The frequency matrix for oligonucleotides of length n_c is obtained by dividing the CGR into a $2^{n_c} * 2^{n_c}$ grid. Then the Markov chain probability matrix could be obtained from FCGR only if the quadrant k satisfies the condition in the following equation to produce an integer order

$$k = 2^{2n_c}, n_c \text{ is an integer} \geq 1$$

But, if the condition 'n_c is an integer ≥ 1' is removed then

$$n_c = \log_2(k)/2,$$

i.e FCGR can track the frequency of oligonucleotide of non-integer order. Therefore, CGR enables the determination of the frequency of redundant fractionary sequences also, FCGR of non-integer order can be used to calculate global distance and local similarities between sequences [4].

If k=3, then the patterns in CGR are determined by mononucleotide, dinucleotide and trinucleotide frequencies but if $k > 3$, then longer oligonucleotides may influence the CGR patterns. Therefore, Wang et.al [20] said, a CGR of $1/2^k$ resolution is completely determined by all the frequencies of length k when the length of the DNA sequence is longer than k . They also analyzed the relationship between dinucleotide relative abundance profile [1](DRAP) and CGR. DRAP can be computed from second-order or dinucleotide frequency FCGR, but second order FCGR cannot be computed from DRAP. Therefore, DRAP or rFCGR (relative FCGR) is a special case of FCGR. Wang et.al [20] said, an n th order FCGR provides more info than DRAP. However, the second-order FCGR or DRAP is a good choice of genomic signature [2] as the computational cost is higher for higher order FCGRs [20]. A new distance measure called *image distance* used to calculate the distance between genomic signatures of two DNA sequences [20] is given by

$$dI_{\mathcal{R}}\left(\bar{\mathcal{A}}, \bar{\mathcal{B}}\right) = 1/4^k * \textstyle\sum_{i=1}^{2^k} \sum_{j=1}^{2^k} \left| density_{\mathcal{R}}\left(\bar{\mathcal{A}}\right)_{i,j} - density_{\mathcal{R}}\left(\bar{\mathcal{B}}\right)_{i,j} \right|$$

---

[1] ratio of the dinucleotide frequency to the frequency of two single nucleotide composing this dinucleotide

[2] The whole set of short oligonucleotide frequencies observed in a DNA sequence is species-specific and is thus considered as a GENOMIC SIGNATURE

$$\bar{\mathcal{A}} = 4^k / \sum_i \sum_j a_{i,j} * \mathcal{A} \text{ and}$$

$$\bar{\mathcal{B}} = 4^k / \sum_i \sum_j a_{i,j} * \mathcal{B}$$

where $\mathcal{A}$ and $\mathcal{B}$ are frequency matrices of kth order, $\mathcal{R}$ is the radius of the neighborhood centered at (i, j) and $density_{\mathcal{R}}$. The phylogenetic trees built using the Euclid distance, Pearson distance and Image distance between two CGRs have proven to be more compatible with phylogenetic relatedness of species than the tree obtained from ClustalW [20].

4.2 Chaos game representation of protein sequences

4.2.1 Chaos game using a 20 sided polygon

The chaos game representation of protein sequences was used to find the motifs in the sequence, describe regularities in structure elements, and evaluate various secondary structure prediction algorithms [11].

4.2.1.1 Method

Fiser et.al in 1994 [11] applied Chaos Game Representation to protein sequences to investigate the motifs in the protein database and protein sequences. A 20-sided regular polygon was used to represent the 20 different amino acids. The (x, y) coordinates of each of the vertices were given as

$$v_{i,x} = cos(2\pi * i/n)$$

$$v_{i,y} = sin(2\pi * i/n)$$

4.2.1.2 Plotting

The coordinates of the 0th point are [0,0] and the mth point was given by dividing the distance between the $(m-1)$th point and the vertex representing the mth amino acid using the dividing ratio s_1 and s_2.
The coordinates of the points are

$$p_{m,x} = (v_{m,x} - p_{m-1,x}) * s_2 + p_{m-1,x}$$

$$p_{m,y} = (v_{m,x} - p_{m-1,y}) * s_2 + p_{m-1,y}$$

The dividing ratio s_1: s_2 is 0.135:0.865 calculated from

$$s_1 = sin(2\pi * i/n)/(1 + sin(2\pi * i/n))$$

$$s_2 = 1/(1 + sin(2\pi * i/n))$$

A lower dividing ratio was used in order to obtain an unambiguous and decodable fractal for an attractor.

4.2.1.3 Properties

The attractor produced was a 20-gon in which there are small separate 20-gons at every vertex. The small 20-gon can further contain smaller 20-gons in their vertices etc (fig 4.2). For example, an amino acid subsequence IDEAL can be decoded by zooming the 20-gon at vertex L followed by the 20-gon at vertex A of the L polygon, the 20-gon at vertex E of the A polygon, the 20-gon at vertex D polygon and the 20-gon at vertex I of the D polygon. Theoretically, each point represents the preceding sequence motif. Eventhough the attractor can be used for identifying subsequences or motifs, they are indistinguishable as the sequence length increases. Fiser et.al extended the chaos game to study the regularities in secondary structure elements of proteins. The major secondary structures helix, sheet and turn were represented as vertices of a triangle and the random coil as the center. The attractor produced was used to study the frequency of attachment of various secondary structure and evaluate structure prediction methods. Therefore, CGR could be used to study both primary and 3D structures of proteins [11].

4.2.1.4 Limitation

The major drawback of this approach is that as the sequence length becomes larger all the polygons looks equally filled. Therefore, various sequence motifs become indistinguishable.

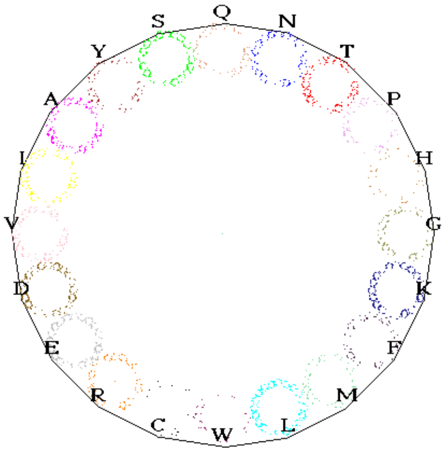
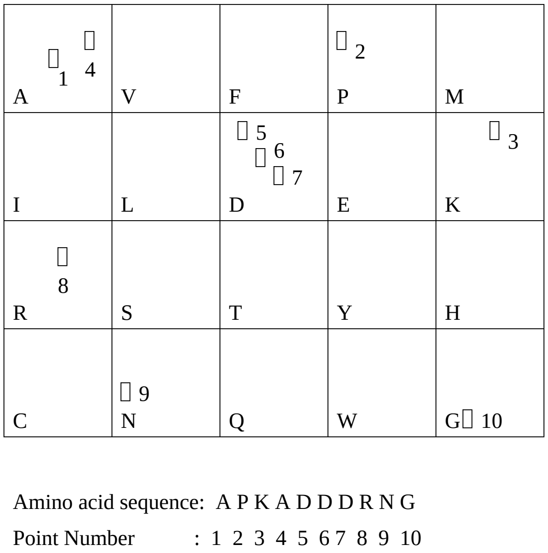


Figure 4.2: CGR using 20-gon of DNA Polymerase Human Alpha Chain: Length = 1462

4.2.2 Chaos game using a rectangle

4.2.2.1 Method

A rectangle was used to represent the sequentiality and composition of amino acids in a sequence. The rectangle was divided into 5 x 4 sub-rectangles representing 20 different amino acids.



A
V
F
P
M
I
L
D
E
K
R
S
T
Y
H
C
N
Q
W
G
1
2
3
4
5
6
7
8
9
10
Amino acid sequence: A P K A D D D R N G
Point Number : 1 2 3 4 5 6 7 8 9 10

Figure 4.3: 2D point representation using rectangle for sequentiality and composition of amino acids

4.2.2.2 Plotting

The chaos game is played as follows: the first point is plotted in the middle of the sub-rectangle labelled with the first amino acid in the sequence. The *ith* point is plotted by scaling the *(i-1)th* point by 1/5 in x -direction and 1/4 in y-direction and moving the point to the sub-rectangle labelled with the *ith* amino acid (fig 4.3).

4.2.2.3 Properties

Some characteristic properties noted were that the points that follow after the insertion and deletion of amino acids are shifted but the degree of shifting is reduced by 5^n in the x-direction and 4^n in the y-direction, where n is the number of letters after the inserted/deleted position. Therefore, insertion/deletion does not change the overall visual impression of the point pattern . The reduction in shift was also noted when there were repeats in amino acids. [18].

4.2.3 Chaos game using a 12 sided polygon

4.2.3.1 Method

CGR can be used to study characteristic patterns of protein families [6]. A 12 sided regular polygon was used to plot a concatenated amino acid sequence of proteins from protein family (fig 4.4). Each vertex of the polygon corresponded to a group of amino acid residues of conservative substitutions (section 3.3) and the amino acids along the vertices of the polygon were placed in the order of decreasing normalized hydrophobicity.

4.2.3.2 Plotting

The chaos game was played as in the case of DNA sequences using the square [16]. A grid counting algorithm was used to quantify the CGR. The 12 sided regular polygon was divided into 24 segments and the number of points in each segment was calculated, the percentage of points in each segment was given by C_j / N* 100(j=1..24).

4.2.3.3 Properties

Even though no fractals were detected in the CGRs, there were specific statistical biases in the distribution of different amino acids, mono-, di-, tri- or higher order

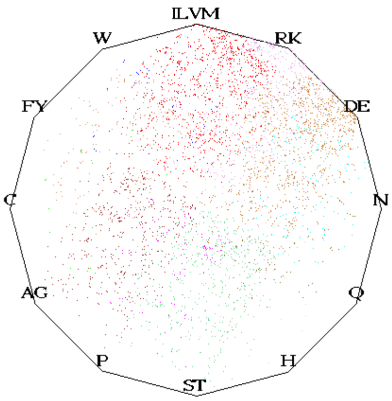


Figure 4.4: CGR using 12 sided polygon; HEAT SHOCK PROTEIN 90 (hsp90) family

peptides in functional classes of proteins [6]. The CGRs of a protein family were dependent on the relative order of the residues. The patterns were insensitive to the shuffling of less abundant residues along the vertices of the polygon, but sensitive to the shuffling of more abundant residues along the vertices. The plots are visually similar for a protein class for many different orientations of the residues. The grid count was also invariant for a particular family of proteins and particular orientation of the residue group along the vertices of the CGR irrespective of the number of sequences concatenated and the order of concatenation. Therefore, the grid count can be used as a diagnostic signature of a protein family for identifying new members of the family and CGR has a potential to reveal evolutionary and functional relationship between proteins having no significant homology [6].

4.2.4 Chaos game using a square

Multi-fractal and correlation analysis were performed on CGR of bacteria families to study the phylogenetic relationship between sequences and sub-families [21].

4.2.4.1 Method

A detailed hydrophobic or non-polar and hydrophilic or polar (HP) model is used to represent the four classes of amino acids - non-polar, uncharged, positively charged and negatively charged as four vertices of the square.

4.2.4.2 Plotting

The chaos game is played as in the case of DNA sequences [16]. The square is divided into equal meshes and a measure μ for each mesh was calculated by dividing the number of points lying in the subset of the CGR by the length of the sequence. This is represented as a measure matrix $\mathcal{A}$. Also, a symbolic sequence is created based on the probability of amino acids in the position of the original sequence and a measure matrix is calculated for it and referred to as a measure of fractal background $\mathcal{A}^f$. A new measure matrix $\mathcal{A}^d$ is obtained by subtracting the measure matrix of the fractal background from the measure matrix of the original sequence and used for calculating the correlation distance.

4.2.4.3 Properties

The phylogenetic tree based on correlation distance is more precise as the mesh size increases [21]. Multifractal analysis ([13]) of the test sequences exhibited multi-fractal like forms indicating that the protein sequences of a complete genome are non-random. Also, said, the correlation analysis is more precise than the multi-fractal analysis for the phylogenetic problem[21].

4.2.5 Summary

The following table summarizes the results of Chaos Game Representation on Protein Sequences in two dimensions.

Approach	Novel Advances	Limitations	Authors	Year
Chaos Game using 20 sided polygon, 20 vertices represents 20 amino acids	motif detection in protein database,regularities in secondary structure elements and evaluation of secondary structure prediction methods	for large sequences, motif detection is not easy due to resolution of the monitor	Fiser et.al	1994
Chaos Game using a 5 x 4 rectangle	sequentiality and composition of amino acids		Pleibner	1997
Chaos Game using 12 sided polygon, 12 vertices represent 12 groups of amino acids	Characteristic patterns of Protein family, measure to detect protein family for a given protein	individual representation of amino acid is lost	Basu et.al	1997
Chaos Game using a square, vertices represent non-polar, uncharged, positively charged and negatively charged amino acids	evaluate phylogenetic tree of bacteria	individual representation of amino acid is lost	Yu et.al	2004

Table 4.1: Summary - Chaos game representation of protein sequences in two dimension

Chapter 5

Three dimensional CGR of protein sequences

The chaos game representation of proteins in two dimensions discussed in the previous chapter helped to identify motifs in the protein databases and to test secondary structure prediction methods [11], reveal patterns that distinguish protein families [6], understand the sequentiality and composition of amino acids [18] and better understanding of the bacterial family homology [21]. In this chapter, a new three dimensional approach to CGR (3D-CGR) as an analysis tool of protein sequence is proposed, with the following objectives:

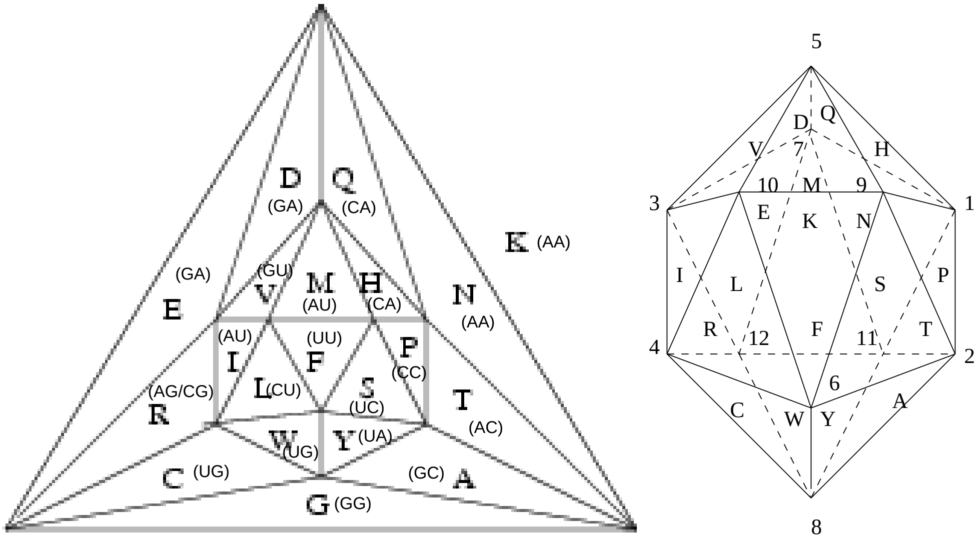
- use the three dimensional approach to detect protein homology
- assess the impact of dinucleotide bias at the amino acid level on 3D-CGR derived protein homology and
- use the three dimensional approach to detect shuffled motifs.

5.1 Three dimensional structure and amino acid mapping

In order to play the chaos game for protein sequences in three dimensions an *icosahedron* was chosen to be the geometric model. An icosahedron is a geometric solid that has twelve vertices, thirty edges and twenty faces. An icosahedron was chosen because the twenty amino acids of a protein can be represented by the twenty faces of an icosahedron.

5.1.1 Mapping

The amino acids are mapped onto the faces of an icosahedron in an order based on the dinucleotide relatedness of codons (see Chapter 2, section 2.1.2). The amino acids that differ by a single nucleotide in the codons are mapped onto the faces of an


D
Q
(GA)
(CA)
K (AA)
(GA)
(GU)
V
M
H
(AU)
(CA)
N
E
(AA)
(AU)
(UU)
I
F
P
(CC)
(AG/CG)
L
(CU)
S
(UC)
T
R
(AC)
W
Y
(UA)
(UG)
C (UG)
(GC) A
G (GG)
5
D
Q
V
7
H
3
10
M
9
1
E
K
N
I
L
S
P
R
12
F
11
T
4
2
6
C
W
Y
A
8

(a) 2D Representation; Amino acid K represents the face at the back

(b) 3D Representation

Figure 5.1: a) 2D and b) 3D Representation of an icosahedron with first and second nucleotide position of codon and its amino acid mapping respectively

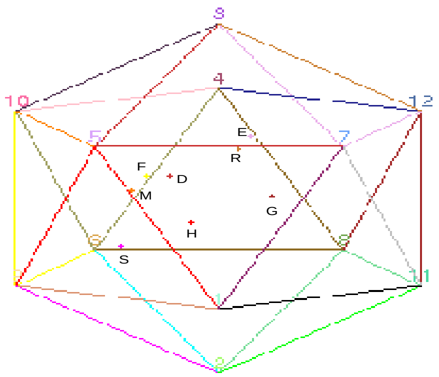
icosahedron that are closer in three dimensional space and the rest of the amino acids were mapped onto the faces that are further apart. The mapping of amino acids onto two dimensional and three dimensional representations of an icosahedron is shown in figure 5.1a and 5.1b.

5.2 Chaos game on an icosahedron

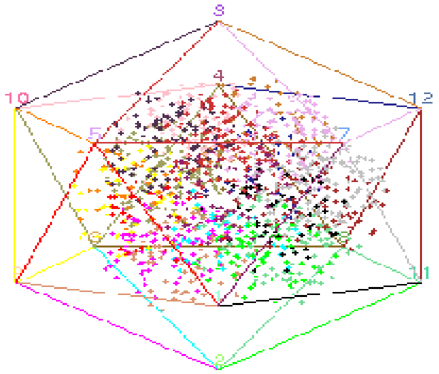
The chaos game was played by taking the center of the icosahedron as the starting point. When the first amino acid is read from a protein sequence, a point is plotted halfway between the center of the icosahedron and the center of the face of the corresponding amino acid. Subsequent points are plotted halfway between the previous point plotted and the center of the face of the amino acid read.

Figure 5.2a shows the chaos game of a sample sequence 'MSDEFGHR' plotted. The first point is plotted halfway between the center of the icosahedron and the center of the face corresponding to the amino acid 'M', the second point is plotted halfway between the first point plotted and the center of the face corresponding to the amino acid 'S', the third point is plotted halfway between the second point and the center of the face corresponding to the amino acid 'D'; similarly E, F, G, H and R are plotted.

The chaos game on a protein sequence produces a *cloud of points* in space. The *cloud of points* did not reveal any obvious patterns to the naked eye. This could be due to the length of protein sequences (approx. < 2000) and also due the points being in 3D space. Figure 5.2b is the output of chaos game using an icosahedron when played on the protein sequence of DNA Polymerase human alpha chain (DPOA_HUMAN) of length 1462. Twenty unique colors have been used to represent the twenty amino acids.



(a) Chaos game on icosahedron for protein sequence 'MSDEFGHR'



(b) Chaos game on icosahedron for protein sequence - DNA Polymerase Human Alpha chain

Figure 5.2: Chaos game of protein sequence in three dimension

3D-CGR was expected to reveal useful patterns in protein sequences, since CGR has been a visual representation technique to study sequence similarities. But, due to the points being in space no patterns were visible to the naked eye. Therefore, we reduced the number of amino acids by grouping them based on conservative substitutions. The groupings were then mapped onto the 12 vertices of the icosahedron and chaos game was played. The points generated by the chaos game showed empty regions near certain groups of aminoacids and more points towards other groups of amino acids indicating the frequency of occurences of the amino acid groups but, failed to reveal any visible patterns to the naked eye. Also, we tried to visually study the points for patterns by rotating the three dimensional figure and projecting the points onto their corresponding faces but, they were not helpful in revealing any visible patterns. Therefore, we decided to quantitatively analyse the similarities and differences between the points produced by different sequences using phylogenetic trees.

5.3 Distance measure

In this section we quantitatively analyse the relationship between protein sequences as well as the membership of a given protein to a protein family by calculating the distance between the clouds of points generated by protein sequences using 3D-CGR. In order to define a distance measure between the clouds of points produced by any two sequences, the points produced by the sequences were enclosed inside a cube. The cube was then subdivided into $n \times n \times n$ small cubes and the density of points in each of the small cubes was calculated. Let P and Q be any two sequences, and s and t be length of the sequences. The density of points in each of the $n \times n \times n$ cubes of the sequences P and Q are represented in matrices

Let $A_{n\times n\times n}$ and $B_{n\times n\times n}$ be the 3 dimensional matrices

$$\text{Density } D_A \text{ of } (a_{ijk}) = (\text{no. of points that fall into the cube } a_{ijk}) \times 1/s$$

$$\text{Density } D_B \text{ of } (b_{ijk}) = (\text{no. of points that fall into the cube } b_{ijk}) \times 1/t$$

Dividing by the length of the sequence is to normalize as the protein sequences compared are of various length. The *image distance* [20] between two sequences P and Q is defined as

$$\sum_{i,j,k=1}^{n} \left| D_A\left(a_{ijk}\right) - D_B\left(b_{ijk}\right) \right|$$

Two other distances used for sequence comparison in this thesis are the *Euclid and Pearson distance.* The Euclid and Pearson distance between sequences P and Q using density matrices A and B is given by

The Euclid distance

$$\sqrt{\sum_{i,j,k=1}^{n} \left(D_A\left(a_{ijk}\right) - D_B\left(b_{ijk}\right)\right)^2}$$

The Pearson distance

$$\frac{\sum_{i,j,k=1}^{n} D_A\left(a_{ijk}\right) \times D_B\left(b_{ijk}\right) - \frac{\sum_{i,j,k=1}^{n} D_A\left(a_{ijk}\right) \times \sum_{i,j,k=1}^{n} D_B\left(b_{ijk}\right)}{\mathcal{N}}}{\sqrt{\left(\sum_{i,j,k=1}^{n} D_A\left(a_{ijk}\right)^2 - \frac{\left(\sum_{i,j,k=1}^{n} D_A\left(a_{ijk}\right)\right)^2}{\mathcal{N}}\right) \times \left(\sum_{i,j,k=1}^{n} D_B\left(b_{ijk}\right)^2 - \frac{\left(\sum_{i,j,k=1}^{n} D_B\left(b_{ijk}\right)\right)^2}{\mathcal{N}}\right)}}$$

$$\mathcal{N} = n \times n \times n$$

5.4 Experimental objectives

To detect protein homology using the 3D-CGR approach, to assess the impact of dinucleotide bias at amino acid sequence level on 3D-CGR derived protein homology and to detect shuffled motifs, the following experiments were performed :

- **validate tree:** The goal of this experiment was to test if the phylogenetic tree constructed using 3D-CGR can detect protein sequence homology.
- **effect of mapping change:** The goal of this experiment was to investigate whether or not varying the mapping of the 20 amino acids on to the 20 faces of

the icosahedron has an effect on the quality of the phylogenetic trees.

- **compare trees:** The goal of this experiment was to compare the phylogenetic trees generated by 3D-CGR with an alignment technique CLUSTALW used for studying sequence relatedness.
- **compare distance measures:** The goal of this experiment was to compare the phylogenetic trees generated by three distance measures and decide which one gives better results in describing the protein sequence homology.
- **assess fractal pattern:** The goal of this experiment was to compare the fractal patterns produced by protein sequences using their fractal dimension.

5.5 Dataset for protein sequence analysis using 3D-CGR

The test data was obtained from the SWISS-PROT Database. Table 5.1 and 5.2 lists all the protein sequences, their length and SWISS-PROT ID used for the test analysis. The protein sequences were selected such that they were of various lengths, and from protein families of diverse functionalities.

5.6 Software

The plotting of amino acids and evaluation of distance measures were performed using Maple 9. X Windows was used for running Maple 9 under the Unix Operating System. Phylip 3.63 package was used for generating phylogenetic trees to determine the sequence similarity and differences. Fractal analysis was performed using Java. CLUSTALW for the test data was run using the default parameter, gap open penalty of 10 and the BLOSUM matrix.

Protein Family	*SWISS_PROTID*	Length
Myoglobin	*MYG_ALLMI*(Alligator)	154
	MYG_CHICK	153
	MYG_HUMAN	153
	MYG_MOUSE	153
Hemoglobin	*HBA_ALLMI*	141
	HBA_CHICK	141
	HBA_HUMAN	141
	HBA_MOUSE	141
	HBA_XENTR(Frog)	141
	HBA_BRARE(Fish)	141
Superoxide dismutase	*SOD1_ORYSA*(Rice)	151
	SODC_DROME(Fruit Fly)	152
	SODC_CHICK	153
	SODC_NEUCR(Fungus)	153
	SODC_XENLA	150
	SODC_BRARE	154
	SODC_HUMAN	153
	SODC_MOUSE	153
	SODC_CAEEL(Worm)	158
	SODC_YEAST	153
Alcohol dehydrogenase	*ADH1_YEAST*	347
	ADH1_NEUCR	353
	ADH1_BACST	337
	ADH1_CAEEL	349
	ADH1_ORYSA	376
	ADHA_HUMAN	374
	ADHA_MOUSE	374
	ADHA_CHICK	375
	ADH1_ALLMI	374

Table 5.1: Protein Family, Swiss-Prot ID and Length of test protein sequences

5.7 Results and discussion

5.7.1 Tree validation

This experiment was performed to test whether the three dimensional CGR could generate a phylogenetic tree that can identify relatedness of sequences and distinguish

Protein Family	*SWISS_PROTID*	Length
Catalase	*CAT1_CAEEL*	524
	CATA_BACSU(Bacteria)	482
	CATA_DROME	506
	CATA_HUMAN	526
	CATA_MOUSE	526
	CATA_BRARE	526
	CATA_ORYSA	491
	CAT1_NEUCR	736
	CATA_YEAST	515
Methionine adenosyltransferase	*METK_BACSU*	400
	METK_CAEEL	404
	METK_DROME	408
	METK_HUMAN	395
	METK_RAT	395
	METK_ORYSA	396
	METK_NEUCR	395
	METK_YEAST	381
6 phosphogluconate dehydrogenase	6*PGD_BACSU*	468
	6*PGD_CANAL*(Yeast)	517
	6*PG1_YEAST*	489
	6*PGD_DROME*	481
	6*PGD_HUMAN*	482
	6*PGD_MOUSE*	482
DNA Polymerase	*DPO1_BACST*	876
	DPOA_DROME	1488
	DPOA_HUMAN	1462
	DPOA_MOUSE	1465
	DPOA_YEAST	1468
	DPOA_ORYSA	1243
	DPOD_CAEEL	1081

Table 5.2: Protein Family, Swiss-Prot ID and Length of test protein sequences (continued from Table 5.1)

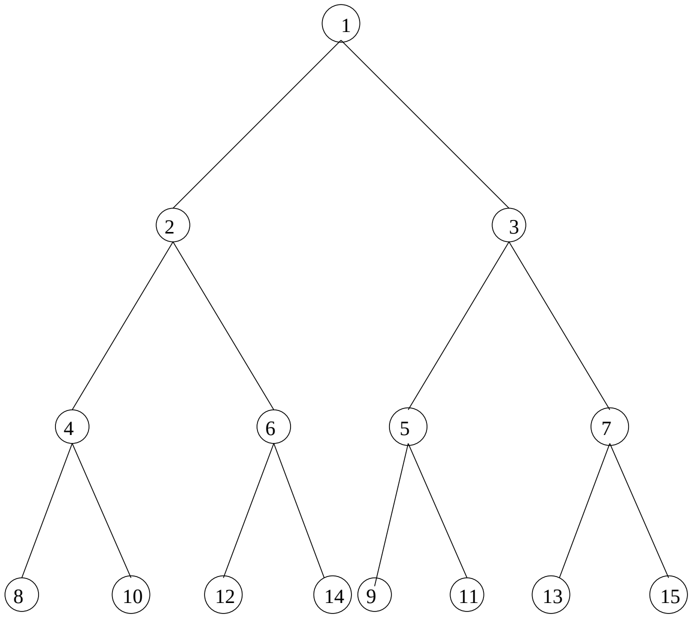
differences between sequences. The method was to use a set of sequences for which the true phylogenetic tree was known and compare the tree obtained by using 3D-CGR with the true phylogenetic tree. As in nature the true phylogenetic tree is never known for sure, we used a starting sequence and simulated its evolution through mutations such that the relatedness of the subsequent sequences was completely transparent.In order to perform this experiment 15 simulated protein sequences of length 352 with known percentage of relatedness between them were created. Sequence 1 was assumed to be the root, sequences 2 and 3 were derived from sequence 1 by 16 amino acid substitutions (in the first half for sequence 2 and second half for sequence 3), sequences 4 and 6 were derived from sequence 2 and sequences 5 and 7 were derived from sequence 3 by the same method, and similarly, sequences 8 and 10 from sequence 4, sequences 12 and 14 from sequence 6, sequences 9 and 11 from sequence 5 and sequences 13 and 15 from sequence 7. At each stage of derivation additional substitutions were made to the derived sequences as in the first step and the substitutions were made such that they do not replace an earlier substitution. Figure 5.3a represents the sample sequence derivation explained above and figure 5.3b represents the true phylogenetic tree that the simulated sequences were expected to generate. In parallel, the chaos game was played on the icosahedron and the image distance was calculated for all pair of sequences from the cloud of points generated by their CGR's. The distances were represented in a distance matrix and the phylogenetic tree was generated.

5.7.1.1 Result and interpretation

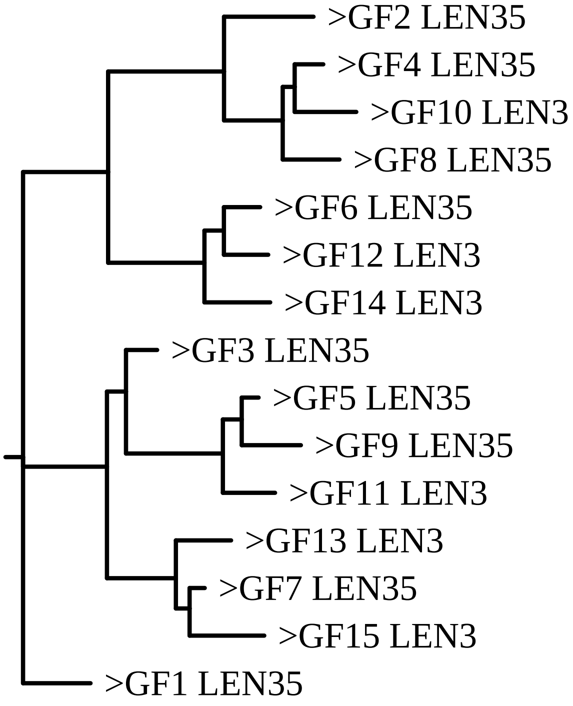
Figure 5.3c represents the phylogenetic tree generated by 3D-CGR. The phylogenetic tree generated by the simulated protein sequences using 3D-CGR was able to establish sequence relatedness based on the mutational difference in the sequences. The hierarchical structure of ancestor and children was exact to that of the known tree. The result shows that the 3D-CGR can identify related sequences and distinguish differences between them. Therefore, 3D-CGR as an analysis tool can be used to study protein sequence relatedness.

M A F T R S Y C N P I Q G D E H K L V W M A F T R S Y C N P I Q G D E H K L V M	SEQUENCE01
M c F T R k Y C W P I R G r E H v L V W M A F T R S Y C N P I Q G D E H K L V M	SEQUENCE02
M c f g R k N C W P I R G r E H v L V W M A F T R S Y C N P I Q G D E H K L V M	SEQUENCE04
M c F T R k Y C W P I R G r E a v L e W M A F T R S Y C N P I Q G D E H K L V M	SEQUENCE06
M A F T R S Y C N P I Q G D E H K L V W M A c T R S Y f N P I Q G D h H K d V M	SEQUENCE03

(a) SEQUENCE02 and SEQUENCE03 derived from first and second half of SEQUENCE01, SEQUENCE04 and SEQUENCE06 derived from SEQUENCE02; substitutions are represented in lowercase letters


1
2
3
4
6
5
7
8
10
12
14
9
11
13
15

(b) True Tree


>GF2 LEN35
>GF4 LEN35
>GF10 LEN3
>GF8 LEN35
>GF6 LEN35
>GF12 LEN3
>GF14 LEN3
>GF3 LEN35
>GF5 LEN35
>GF9 LEN35
>GF11 LEN3
>GF13 LEN3
>GF7 LEN35
>GF15 LEN3
>GF1 LEN35

(c) Validated Tree by 3DCGR

5.7.2 Effect of mapping change

In order to obtain meaningful results we wanted to map the amino acids onto the faces of an icosahedron in a way that is biologically meaningful. Therefore, for our working mapping, we maped the amino acids the differ by a single nucleotide in the codons onto the neighboring faces of an icosahedron and rest of the amino acids were mapped onto the faces that are further apart. We wanted to test whether varying the mapping would change the results of our analysis i.e we wanted the impact of dinucleotide bias at the amino acid level. Consequently, the chaos game was played on the icosahedron with the above mentioned mapping of amino acids and the image distance was calculated between the cloud of points for all pairs of sequences. The image distances between all pairs of sequences was represented as a distance matrix and a phylogenetic tree was generated based on the distance matrix. Similarly, phylogenetic trees were generated for four other random mappings of amino acids onto the faces of an icosahedron. Figures 5.4, 5.5 and 5.6 represents the phylogenetic trees obtained using the dinucleotide related mapping and two random mappings.

5.7.2.1 Results and interpretation

All the trees generated distinguished the protein families of the test sequences from one another. Also, the trees displayed species relatedness within families. The comparison between the three trees is provided in Table 5.3. The dinucleotide related mapping differs from random mapping 1 and random mapping 2 in the branching of ADH sequences, in the branch order of the families and in branch length between closely related species. The difference in branch length and branch order between the mapping based on dinucleotide relatedness of codons and the two random mappings can be attributed to the minor effect of dinucleotide biases at amino acid level.

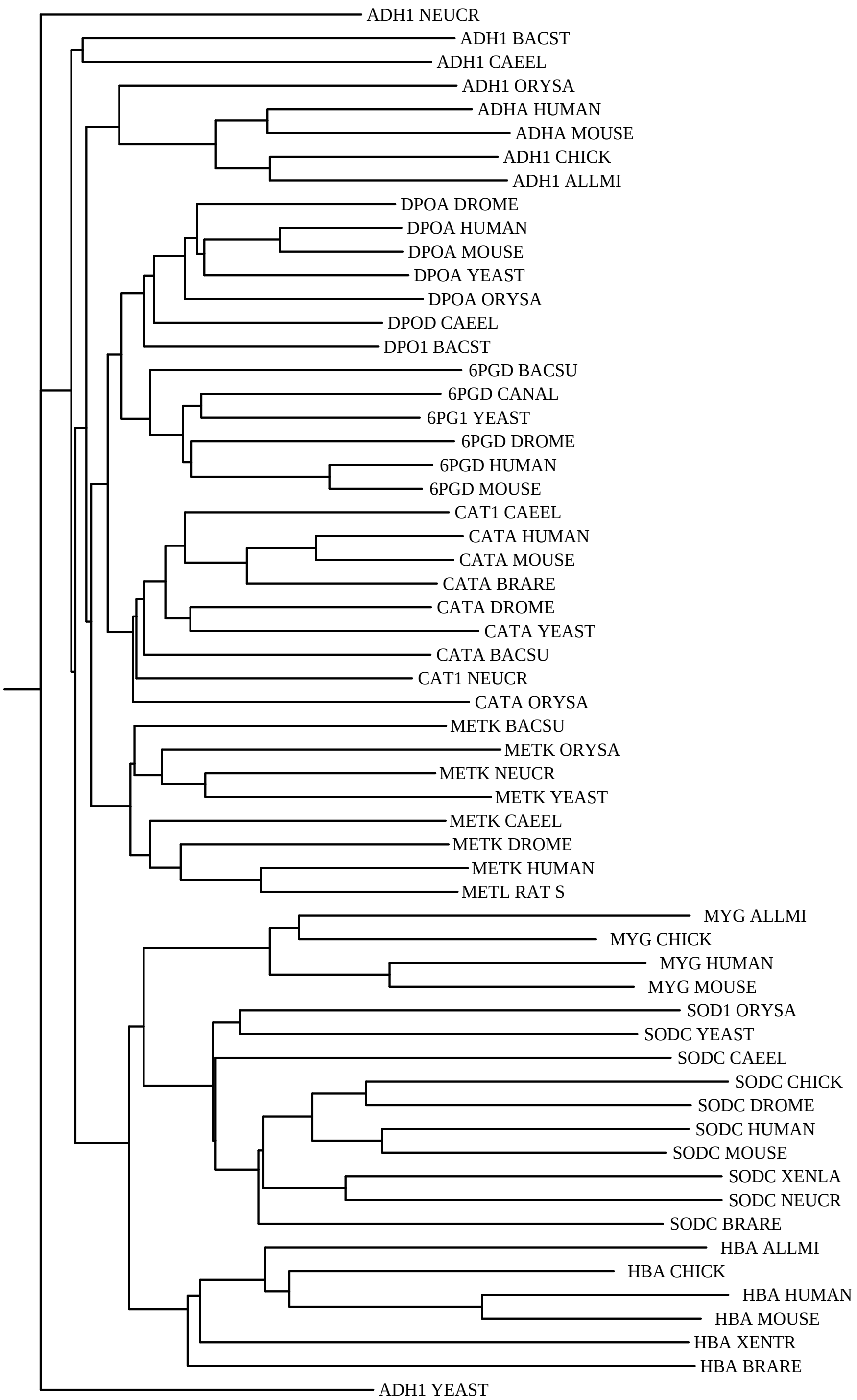

ADH1 NEUCR
ADH1 BACST
ADH1 CAEEL
ADH1 ORYSA
ADHA HUMAN
ADHA MOUSE
ADH1 CHICK
ADH1 ALLMI
DPOA DROME
DPOA HUMAN
DPOA MOUSE
DPOA YEAST
DPOA ORYSA
DPOD CAEEL
DPO1 BACST
6PGD BACSU
6PGD CANAL
6PG1 YEAST
6PGD DROME
6PGD HUMAN
6PGD MOUSE
CAT1 CAEEL
CATA HUMAN
CATA MOUSE
CATA BRARE
CATA DROME
CATA YEAST
CATA BACSU
CAT1 NEUCR
CATA ORYSA
METK BACSU
METK ORYSA
METK NEUCR
METK YEAST
METK CAEEL
METK DROME
METK HUMAN
METL RAT S
MYG ALLMI
MYG CHICK
MYG HUMAN
MYG MOUSE
SOD1 ORYSA
SODC YEAST
SODC CAEEL
SODC CHICK
SODC DROME
SODC HUMAN
SODC MOUSE
SODC XENLA
SODC NEUCR
SODC BRARE
HBA ALLMI
HBA CHICK
HBA HUMAN
HBA MOUSE
HBA XENTR
HBA BRARE
ADH1 YEAST

Figure 5.4: Phylogenetic Tree generated by dinucleotide relatedness mapping using the Image distance

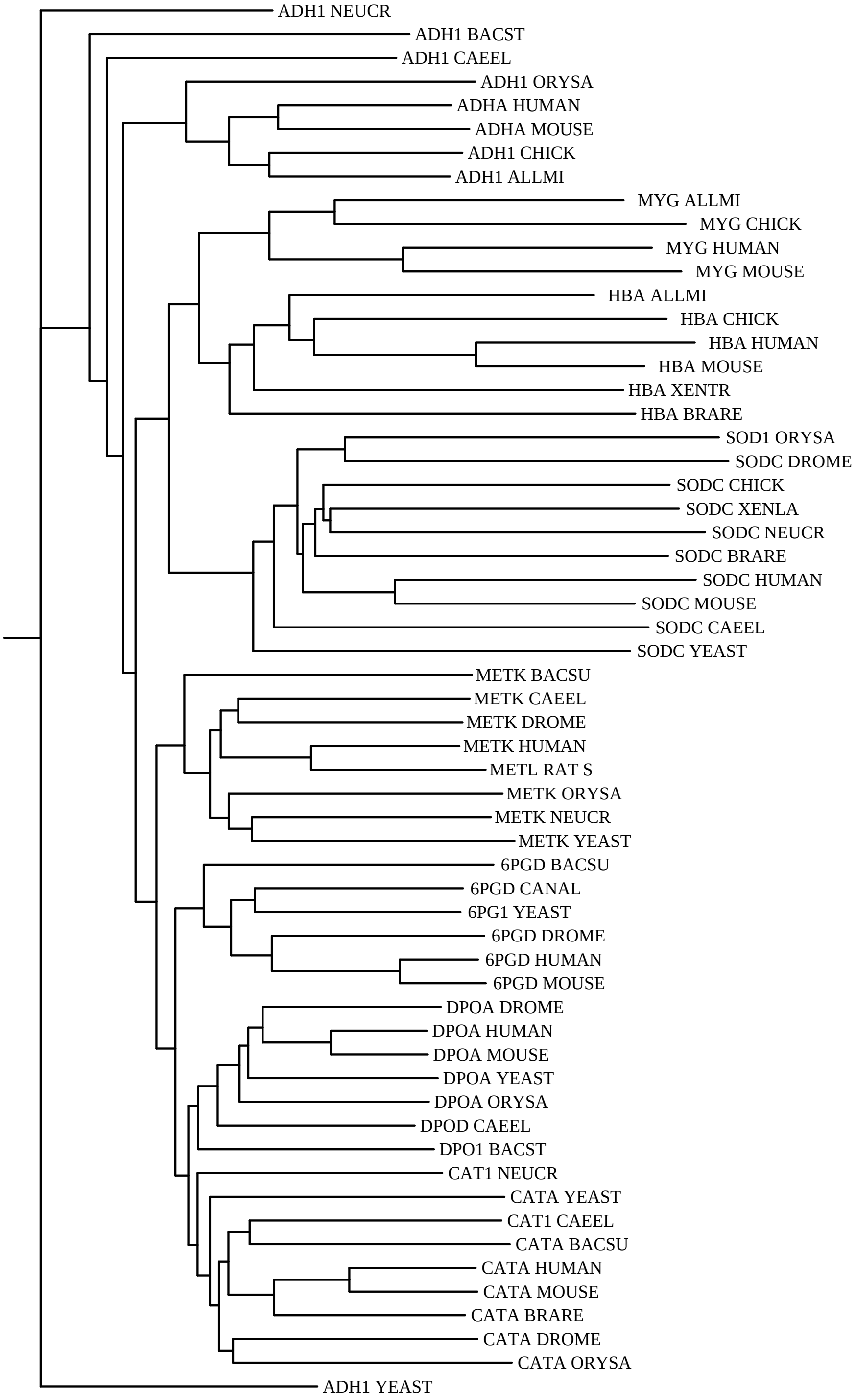

ADH1 NEUCR
ADH1 BACST
ADH1 CAEEL
ADH1 ORYSA
ADHA HUMAN
ADHA MOUSE
ADH1 CHICK
ADH1 ALLMI
MYG ALLMI
MYG CHICK
MYG HUMAN
MYG MOUSE
HBA ALLMI
HBA CHICK
HBA HUMAN
HBA MOUSE
HBA XENTR
HBA BRARE
SOD1 ORYSA
SODC DROME
SODC CHICK
SODC XENLA
SODC NEUCR
SODC BRARE
SODC HUMAN
SODC MOUSE
SODC CAEEL
SODC YEAST
METK BACSU
METK CAEEL
METK DROME
METK HUMAN
METL RAT S
METK ORYSA
METK NEUCR
METK YEAST
6PGD BACSU
6PGD CANAL
6PG1 YEAST
6PGD DROME
6PGD HUMAN
6PGD MOUSE
DPOA DROME
DPOA HUMAN
DPOA MOUSE
DPOA YEAST
DPOA ORYSA
DPOD CAEEL
DPO1 BACST
CAT1 NEUCR
CATA YEAST
CAT1 CAEEL
CATA BACSU
CATA HUMAN
CATA MOUSE
CATA BRARE
CATA DROME
CATA ORYSA
ADH1 YEAST

Figure 5.5: Phylogenetic Tree generated by random mapping 1 using the Image distance

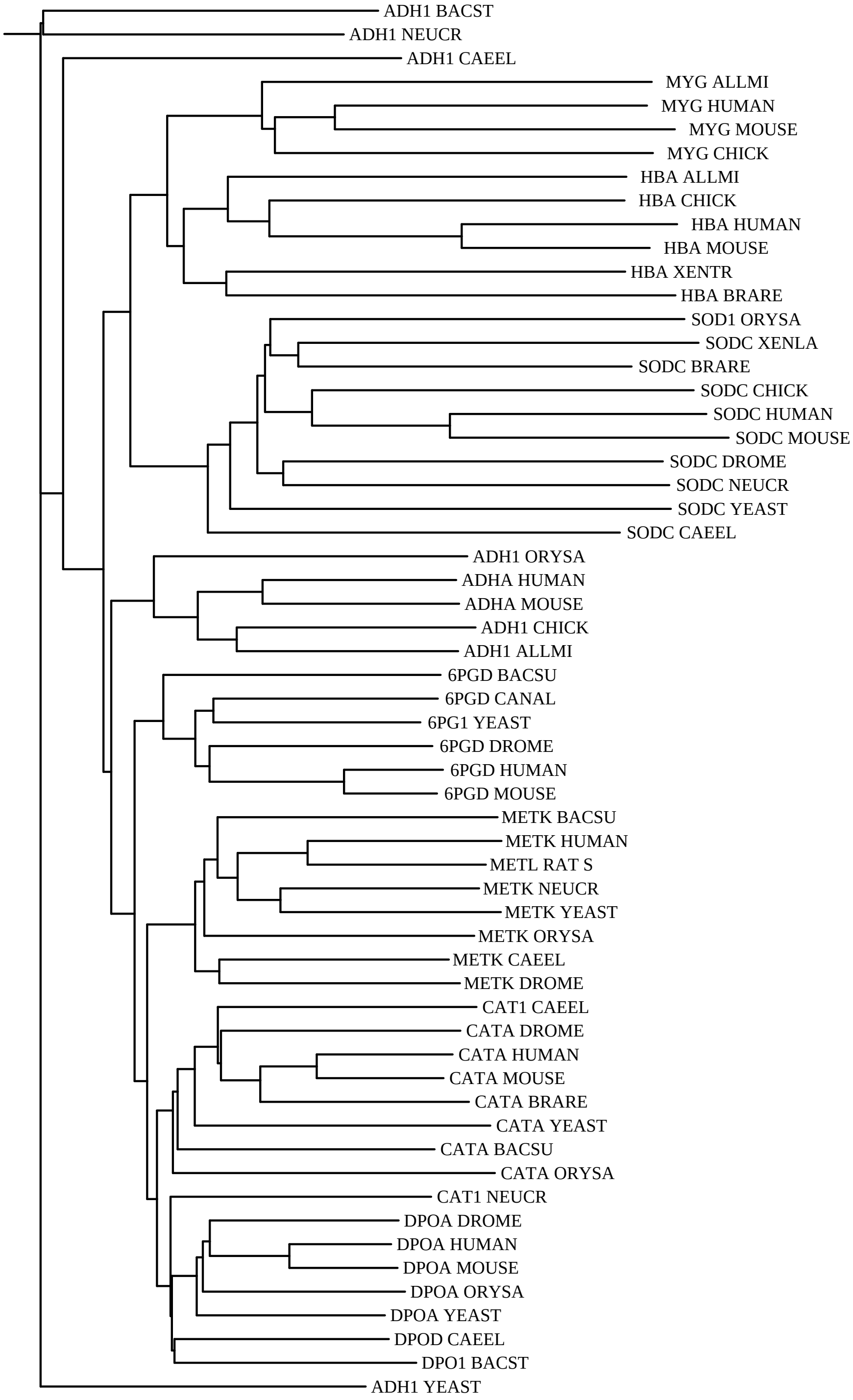

ADH1 BACST
ADH1 NEUCR
ADH1 CAEEL
MYG ALLMI
MYG HUMAN
MYG MOUSE
MYG CHICK
HBA ALLMI
HBA CHICK
HBA HUMAN
HBA MOUSE
HBA XENTR
HBA BRARE
SOD1 ORYSA
SODC XENLA
SODC BRARE
SODC CHICK
SODC HUMAN
SODC MOUSE
SODC DROME
SODC NEUCR
SODC YEAST
SODC CAEEL
ADH1 ORYSA
ADHA HUMAN
ADHA MOUSE
ADH1 CHICK
ADH1 ALLMI
6PGD BACSU
6PGD CANAL
6PG1 YEAST
6PGD DROME
6PGD HUMAN
6PGD MOUSE
METK BACSU
METK HUMAN
METL RAT S
METK NEUCR
METK YEAST
METK ORYSA
METK CAEEL
METK DROME
CAT1 CAEEL
CATA DROME
CATA HUMAN
CATA MOUSE
CATA BRARE
CATA YEAST
CATA BACSU
CATA ORYSA
CAT1 NEUCR
DPOA DROME
DPOA HUMAN
DPOA MOUSE
DPOA ORYSA
DPOA YEAST
DPOD CAEEL
DPO1 BACST
ADH1 YEAST

Figure 5.6: Phylogenetic Tree generated by random mapping 2 using the Image distance

	Random Mapping 1	Random Mapping 2
Dinucleotide Mapping	difference in branching of ADH sequences, difference in the evolution of protein families and difference in branch length between closely related sequences - $(ADHA\_HUMAN, AHDA\_MOUSE)$, (MYG_ALLMI, MYG_CHICK)	difference in branching of ADH sequences, difference in branch length between closely related sequences - $(ADHA\_HUMAN, AHDA\_MOUSE)$, (MYG_ALLMI, MYG_CHICK) and evolution of protein families
Random Mapping 1	identical	difference in branching of ADH sequences

Table 5.3: Pairwise comparisons of the three mapping strategies

5.7.3 3D-CGR and CLUSTALW phylogenetic tree comparison

In order to evaluate the three dimensional CGR method the phylogenetic tree generated by 3D-CGR was compared with the tree generated by the well known multiple sequence alignment program CLUSTALW. CLUSTALW and 3D-CGR follow different approaches to study sequence similarity. In CLUSTALW, the similarity is initially assessed by pairwise alignment, the alignment of any pair of amino acid depends on the alignment score and is irrespective of the pair of amino acid that preceeds it as well as the pair of amino acid that follows it. In contrast, in 3D-CGR a holistic approach is used, every point in the image depends on the preceding sequence of points. Phylogenetic trees were obtained for multiple sequence alignment and 3D-CGR from the dataset. Figure 5.3 and 5.7 represent the phylogenetic trees generated by 3D-CGR and CLUSTALW.

5.7.3.1 Branch length and protein family evolution

The trees when compared showed that they both were able to distinguish protein families and identify species relatedness within the familes. But the following differences were also noted: *a*) In the CLUSTALW tree, when two sequences are closely related (example: ADHA_HUMAN and ADHA_MOUSE), the branch lengths of the two sequences are almost equal, whereas in the 3D-CGR tree significant differences between the branch lengths can be seen; *b*) the evolution of protein families in CLUSTALW tree indicate they had diverged long time back whereas in 3D-CGR tree the evolution of protein families indicate recent divergence and; *c*) the superoxide dismuatse family branches off closely to myoglobin and hemoglobin in the 3D-CGR tree, whereas in CLUSTALW tree, superoxide dismutase does not closely branch off from myoglobin and hemoglobin families.

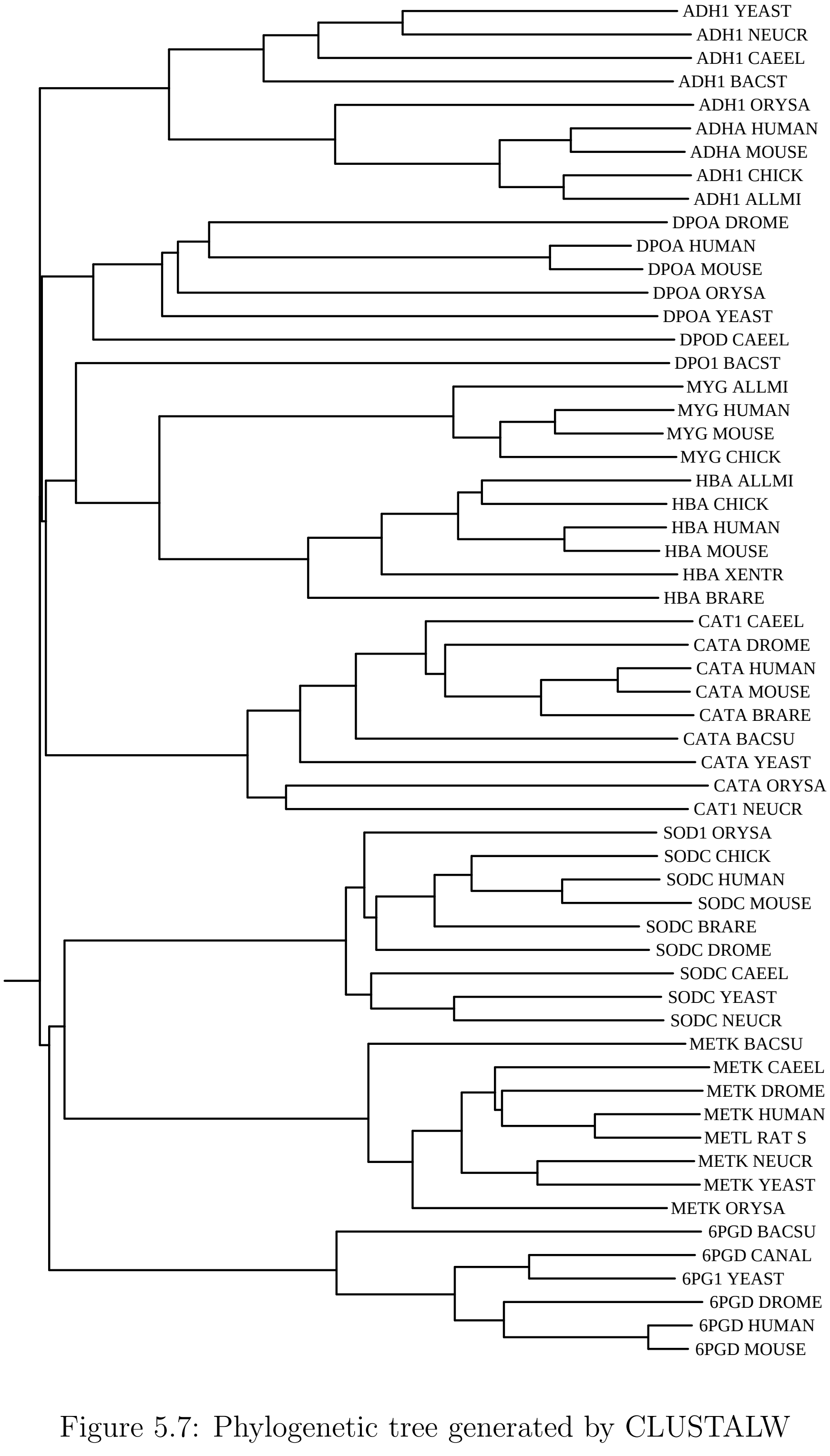
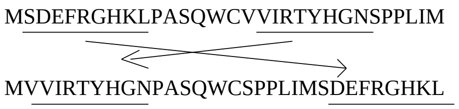

ADH1 YEAST
ADH1 NEUCR
ADH1 CAEEL
ADH1 BACST
ADH1 ORYSA
ADHA HUMAN
ADHA MOUSE
ADH1 CHICK
ADH1 ALLMI
DPOA DROME
DPOA HUMAN
DPOA MOUSE
DPOA ORYSA
DPOA YEAST
DPOD CAEEL
DPO1 BACST
MYG ALLMI
MYG HUMAN
MYG MOUSE
MYG CHICK
HBA ALLMI
HBA CHICK
HBA HUMAN
HBA MOUSE
HBA XENTR
HBA BRARE
CAT1 CAEEL
CATA DROME
CATA HUMAN
CATA MOUSE
CATA BRARE
CATA BACSU
CATA YEAST
CATA ORYSA
CAT1 NEUCR
SOD1 ORYSA
SODC CHICK
SODC HUMAN
SODC MOUSE
SODC BRARE
SODC DROME
SODC CAEEL
SODC YEAST
SODC NEUCR
METK BACSU
METK CAEEL
METK DROME
METK HUMAN
METL RAT S
METK NEUCR
METK YEAST
METK ORYSA
6PGD BACSU
6PGD CANAL
6PG1 YEAST
6PGD DROME
6PGD HUMAN
6PGD MOUSE

Figure 5.7: Phylogenetic tree generated by CLUSTALW

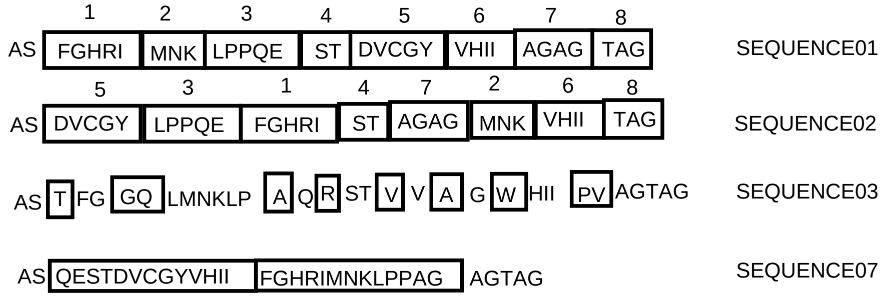
The branch lengths in 3D-CGR indicates 3D-CGR could be used measure the amount of divergence of sequences within protein families. However, we could not identify the biological significance of the protein family evolution obtained by 3D-CGR.

5.7.3.2 Shuffled Motif detection

The power of CGR relies on its holistic approach to biological sequences. Every point on a 3D-CGR depends on its previous point therefore it has a long memory of the preceding amino acids in a protein sequence, whereas in sequence alignment the alignment of any pair of amino acid does not depend on the alignment of its preceding pair of amino acids. Therefore, when two sequences are to be aligned and there are two different motifs present at different positions in both the sequences (fig 5.8a), the sequence alignment would align based on the best alignment and may not detect one of the motifs. In contrast, in the case of 3D-CGR, the comparison of protein sequences is based on the frequency of points in each cubic region, therefore, 3D-CGR is expected to identify motifs better than sequence alignment. In order to test this hypothesis, a protein sequence - SEQUENCE01 of length 300 was selected and a new sequence - SEQUENCE02 was obtained from it by shuffling various regions in the SEQUENCE01. Similarly, another sequence SEQUENCE03 was derived from SEQUENCE01 by performing several insertion/deletions and substitutions, SEQUENCE07 was derived by interchanging two big subsequences, and SEQUENCE06 was made to be identical to SEQUENCE01. Fig 5.8b depicts the above example. Phylogenetic trees for the test sequences together with two other unrelated sequences SEQUENCE04 and SEQUENCE05 were obtained using 3D-CGR (figure 5.9a) and CLUSTALW (figure 5.9b). In the tree obtained from CLUSTALW, SEQUENCE03 was identified more closely to SEQUENCE01 than sequence SEQUENCE02 whereas in 3D-CGR shuffled sequences, SEQUENCE02 and SEQUENCE07 were identified more closely to SEQUENCE01 than SEQUENCE02 with mutations.

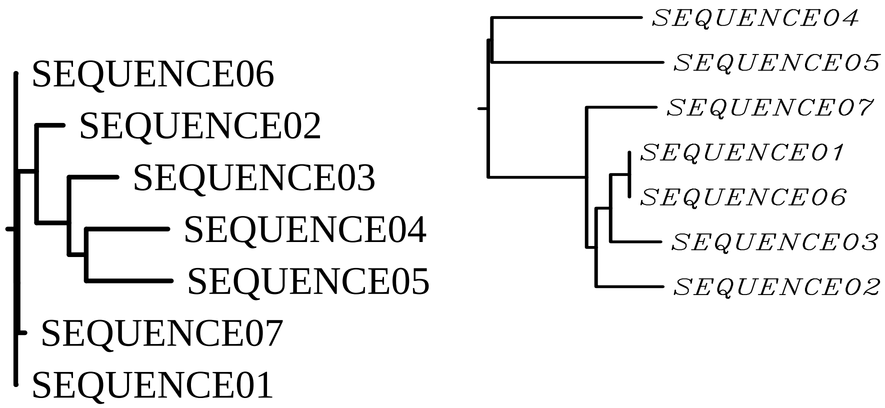

MSDEFRGHKLPASQWCVVIRTYHGNSPPLIM
MVVIRTYHGNPASQWCSPPLIMSDEFRGHKL

(a) Interchanged subsequence


1 2 3 4 5 6 7 8
AS FGHRI MNK LPPQE ST DVCGY VHII AGAG TAG
SEQUENCE01
5 3 1 4 7 2 6 8
AS DVCGY LPPQE FGHRI ST AGAG MNK VHII TAG
SEQUENCE02
AS T FG GQ LMNKLP A Q R ST V V A G W HII PV AGTAG
SEQUENCE03
AS QESTDVCGYVHII FGHRIMNKLPPAG AGTAG
SEQUENCE07

(b) Positions marked 1-8 in SEQUENCE01 are interchanged in SEQUENCE02, positions marked in SEQUENCE03 are mutated from SEQUENCE01, SEQUENCE04 with two big interchanged regions marked

Figure 5.8: Shuffled motif detection


SEQUENCE06
SEQUENCE02
SEQUENCE03
SEQUENCE04
SEQUENCE05
SEQUENCE07
SEQUENCE01
SEQUENCE04
SEQUENCE05
SEQUENCE07
SEQUENCE01
SEQUENCE06
SEQUENCE03
SEQUENCE02

(a) Phylogenetic tree generated by Chaos for Jumbled Sequences

(b) Phylogenetic tree generated by CLUSTALW for Jumbled Sequences

Figure 5.9: Shuffled motif detection by chaos and CLUSTALW

The above result indicates that 3D-CGR can identify shuffled motifs/domains better than CLUSTALW due to its long memory of preceeding sequences. Therefore, 3D-CGR does not require the motifs/domains to be in the same order between sequences in contrast to CLUSTALW which requires the sequences to have same motifs/domains in the same order when assessing sequence similarity. The ability of 3D-CGR to identify shuffled motifs/domians between any pair of sequences indicate it can be used to study protein evolution due to exon shuffling. Exon shuffling is a process by which motifs/domains have been shuffled to form new proteins.

5.7.4 Distance measure comparison

All the results obtained in section 5.7.1, 5.7.2 and 5.7.3 used the image distance. We performed some experiments using another two different distances in order to determine the best distance measure for the output produced by 3D-CGR. The three distance measures used for the comparison were the Image distance, the Euclid Distance and the Pearson distance. The three distance matrices containing the distances between all pairs of protein sequences were calculated using Image, Euclid and Pearson distances and their corresponding phylogenetic trees were generated. Figures 5.3, 5.10 and 5.11 represent the phylogenetic trees generated by the Image distance, Euclid distance and Pearson distance measures.

5.7.4.1 Result and interpretation

The trees generated by the Image distance, Euclid distance and Pearson Distance using 3D-CGR are all able to identify the related species and their families. The only difference was that the tree generated by the Pearson Distance was different from the trees generated by the Image distance and the Euclid distance in branch order and branch length. Therefore, it is not easy to conclude that one distance is better than the other.

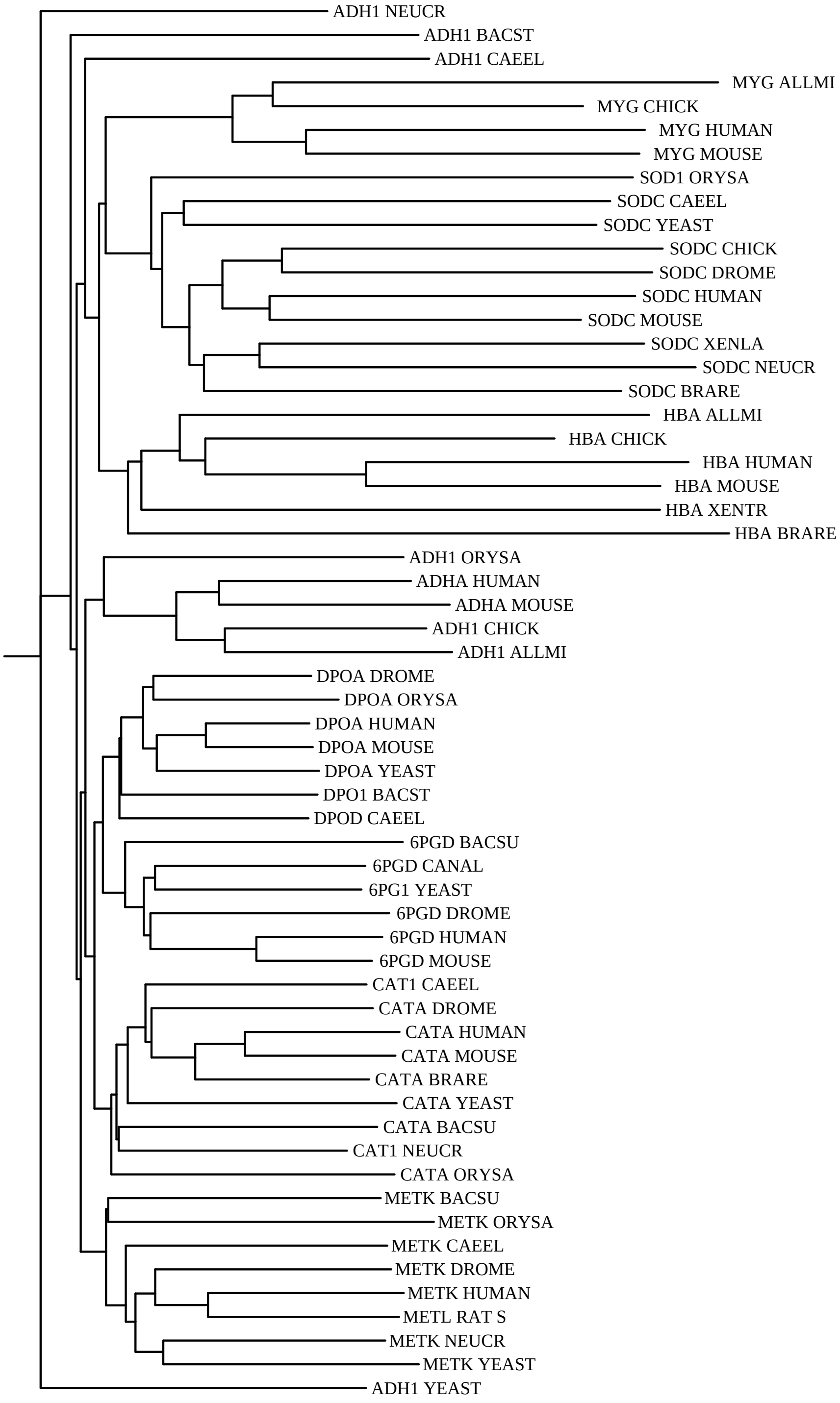

ADH1 NEUCR
ADH1 BACST
ADH1 CAEEL
MYG ALLMI
MYG CHICK
MYG HUMAN
MYG MOUSE
SOD1 ORYSA
SODC CAEEL
SODC YEAST
SODC CHICK
SODC DROME
SODC HUMAN
SODC MOUSE
SODC XENLA
SODC NEUCR
SODC BRARE
HBA ALLMI
HBA CHICK
HBA HUMAN
HBA MOUSE
HBA XENTR
HBA BRARE
ADH1 ORYSA
ADHA HUMAN
ADHA MOUSE
ADH1 CHICK
ADH1 ALLMI
DPOA DROME
DPOA ORYSA
DPOA HUMAN
DPOA MOUSE
DPOA YEAST
DPO1 BACST
DPOD CAEEL
6PGD BACSU
6PGD CANAL
6PG1 YEAST
6PGD DROME
6PGD HUMAN
6PGD MOUSE
CAT1 CAEEL
CATA DROME
CATA HUMAN
CATA MOUSE
CATA BRARE
CATA YEAST
CATA BACSU
CAT1 NEUCR
CATA ORYSA
METK BACSU
METK ORYSA
METK CAEEL
METK DROME
METK HUMAN
METL RAT S
METK NEUCR
METK YEAST
ADH1 YEAST

Figure 5.10: Phylogenetic tree generated by the Euclid distance

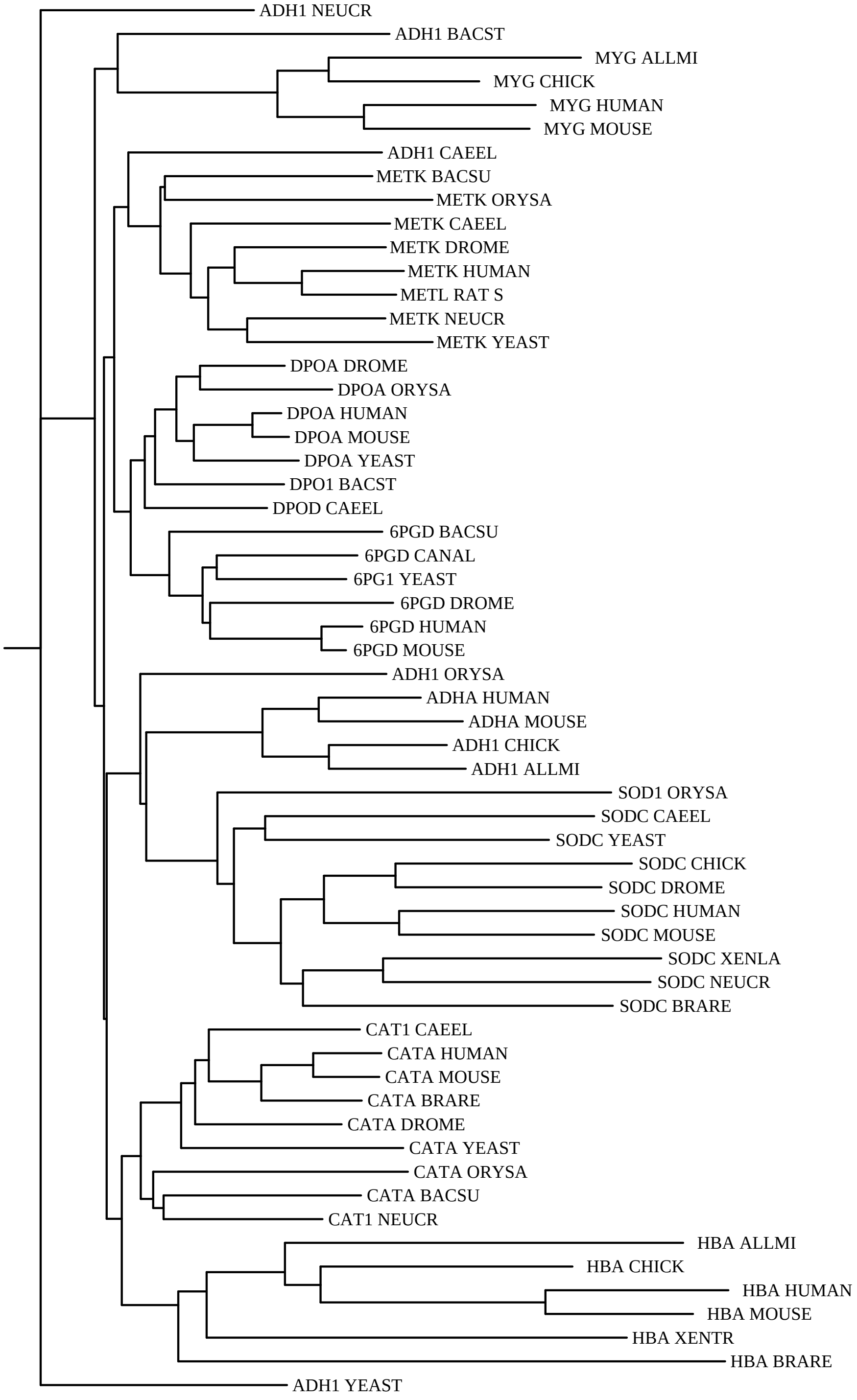

ADH1 NEUCR
ADH1 BACST
MYG ALLMI
MYG CHICK
MYG HUMAN
MYG MOUSE
ADH1 CAEEL
METK BACSU
METK ORYSA
METK CAEEL
METK DROME
METK HUMAN
METL RAT S
METK NEUCR
METK YEAST
DPOA DROME
DPOA ORYSA
DPOA HUMAN
DPOA MOUSE
DPOA YEAST
DPO1 BACST
DPOD CAEEL
6PGD BACSU
6PGD CANAL
6PG1 YEAST
6PGD DROME
6PGD HUMAN
6PGD MOUSE
ADH1 ORYSA
ADHA HUMAN
ADHA MOUSE
ADH1 CHICK
ADH1 ALLMI
SOD1 ORYSA
SODC CAEEL
SODC YEAST
SODC CHICK
SODC DROME
SODC HUMAN
SODC MOUSE
SODC XENLA
SODC NEUCR
SODC BRARE
CAT1 CAEEL
CATA HUMAN
CATA MOUSE
CATA BRARE
CATA DROME
CATA YEAST
CATA ORYSA
CATA BACSU
CAT1 NEUCR
HBA ALLMI
HBA CHICK
HBA HUMAN
HBA MOUSE
HBA XENTR
HBA BRARE
ADH1 YEAST

Figure 5.11: Phylogenetic tree generated by the Pearson distance

5.8 Fractal analysis on protein sequences

Fractal analysis assesses the fractal patterns produced by a cloud of points generated by a sequence. The fractal dimension (Box Counting Dimension) is usually used in assessing the fractal patterns. Since there was no pattern visible to the naked eye in the cloud points produced by the 3D-CGR, the relationship between sequences was assessed by calculating box counting dimensions for the cloud of points generated. The fractal dimensions for various box sizes for the sequences were calculated and the implementation was done using a Spatial Subdivision Algorithm - Octree.

5.8.1 Octree

An Octree is a tree data structure with eight children used to represent spatial subdivision. Each node of the octree holds the physical position of the boxes. Figure 5.12 represents the subdivided cube and its Octree. Initially, a single cube is needed to cover the point cloud, therefore, it is the root node in the Octree. Divide the cube into eight smaller cubes and determine if any of the cube covers a portion of the point cloud. If they are, then subdivide those cubes into further smaller cubes. The process continues for some n times to obtain a better approximation of the fractal dimension.

5.8.2 Fractal dimensions of test sequences

A log-log plot of the number of boxes needed to cover the cloud of points and the box sizes for each sequence in the data set was performed to analyze the fractal nature of the sequences. Fig 5.13 shows the log-log plot of the sequences in the data set.

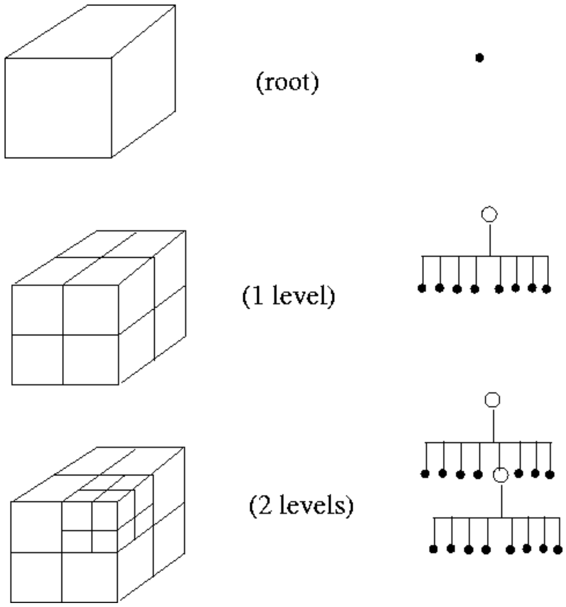


Figure 5.12: Spatial Subdivision using Octree

5.8.2.1 Result and interpretation

Though the fractal curves generated were similar for protein families, the curves were dependent on their sequence lengths. Therefore, fractal analysis is not good method to analyze the relationship between the sequences.

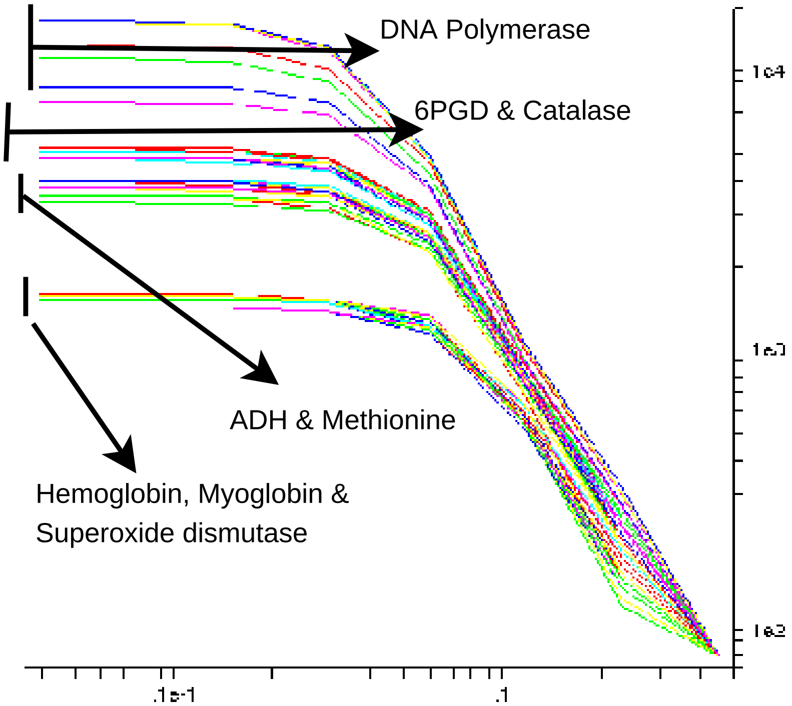


Figure 5.13: Fractal curve for the test data set

Chapter 6

Conclusion

This thesis presented the fundamentals of DNA and protein sequences, mathematical background of chaos and fractals, two dimensional chaos game representation while also exploring a new approach for CGR to study sequence similarity. Chapter 3 and 4 presented the biological and mathematical background for the thesis. Chapter 4 presented the two dimensional CGR of DNA and protein sequences studied in the past. Chapter 5, presented the new three dimensional approach to CGR for protein sequences that provided a holistic approach to sequence analysis

First, in Chapter 2, the biological introduction to the thesis was presented. The meaning, structure and the functionality of DNA and protein sequences and the synthesis of protein sequence from DNA were explained. The representation of sequences/species relatedness was explained through phylogenetic tree and the analysis of sequences/species relatedness was explained through the bioinformatic technique, sequence alignment. Also, the process of multiple sequence alignment was briefly explained through a widely used program CLUSTALW.

Next, in Chapter 3, the mathematics background to understand the chaos game for biological sequences was presented. The mathematical background of chaos game for generating fractals and its ability to reveal the structure present in the non-random sequences were explained. Also, the concepts of fractal and the need for fractal dimension were explained with various examples.

Then, in Chapter 4, the literature on chaos game representation of DNA and protein sequences in two dimension was presented. A detailed description of the methods, novel advances and limitations of chaos game on protein sequence in two dimension was emphasized.

Finally, in Chapter 5, a new approach and results of chaos game representation of protein sequence in three dimension was given. The selection of the geometric solid icosahedron to represent twenty amino acids and its structure were explained. The new three dimensional approach was taken to present a chaos game model by mapping amino acids on to the icosahedron faces based on the dinucleotide relatedness of amino acid from codons. The new approach (3D-CGR) was used to study sequence relatedness, the effect of dinucleotide biases at the amino acid level on the 3D-CGR deduced protein homology, and shuffled motif detection.

The 3D-CGR was evaluated using phylogenetic trees. Trees generated using 3D-CGR were able to distinguish protein families and species relatedness within the families of sequences. Also, the effect of varying the mapping of 20 amino acids on the faces of the icosahedron analysed using phylogenetic tree showed very small difference in the branching of protein families and branch lengths of closely related sequences between the phylogenetic trees generated by random mappings and mapping based on the dinucleotide relatedness of codon. The comparison of phylogenetic trees of 3D-CGR and CLUSTALW revealed the significant difference in branch length between closely related sequences indicated 3D-CGR could be used for measuring the amount of divergence between sequences within a family. Also, 3DCGR can detect sequence relatedness by detecting multiple motifs present in sequences irrespective of the order of the motifs therefore, it could be a useful tool in studying protein evolution due to exon shuffling.

The Image distance measure used for generating phylogenetic trees was compared with Euclid and Pearson distance. All the three distance measures were able to distinguish protein families and relatedness of species within the families. Finally, sequence relatedness explored using fractal curves did not provide much information except the fractal curves generated were similar for protein families.

The patterns produced by 3D-CGR were not visible to the naked eye. Therefore, it is hoped the future research in 3D-CGR on protein sequences could represent the 'cloud of points' in space as a structure for visual comparison. Also, the research could be extended to detect the position of shuffled motifs in 3D-CGR in order to study the protein evolution due to exon shuffling.

Appendix A

Algorithm to determine the subsequence represented by any point in a CGR - Dutta et.al [8]

Input: A point (X,Y) on the CGR.
Output: Sequence that generated the point (X,Y)

Step 1. Let (0,0), (2,0),(2,2) and (0,2) be the vertices of the square and (CX,CY) represent the center of the square. CX = 1 and CY =1

Step 2. Let (X,Y) be the coordinates of the point whose sequence we want to determine (within the resolution limit of the monitor $\pm\delta$ of the monitor).

Step 3. L is the length of the subsequence to be generated.

Let PX and PY be two variables that hold the coordinates of the points as they are generated. Based on the center coordinate and the resolution limit of the monitor, determine in which quadrant the point belongs and change the center accordingly. The process is repeated until the X and Y values are equal to the center +/- the resolution limit of the monitor. These are given by the following steps:

```
Step 4. Repeat step 4 L times
    if $X > CX$ and $Y > CY$ then
        if $X > CX + \delta$ then $PX = 2$
        if $X < CX + \delta$ then $PX = 1$
        if $Y > CY + \delta$ then $PY = 2$
        if $Y < CY + \delta$ then $PY = 1$
        if $X = CX \pm\delta$ or $Y = CY \pm\delta$ then goto step 6
    set $CX = CX + (-1)^{PX}$ (1/2)
    set $CY = CY + (-1)^{PY}$ (1/2)
    if $PX = 2$ and $PY = 2$ then $N = G$
    if $PX = 2$ and $PY = 1$ then $N = T$
    if PX =1 and PY = 2 then N = C
    if PX =1 and PY = 1 then N = A
Step 5. At each step set the value of CX and CY as follows, For the ith step:
    CX = CX + $(-1)^{PX}$ $(1/2)^{i}$
    CY = CY + $(-1)^{PY}$ $(1/2)^{i}$
Step 6. When CX - $\delta < X < CX + \delta$ and CY - $\delta < Y < CY + \delta$ then the length of the sequence is reached or the limit of the resolution of the CGR is reached.
```

Both CX and CY should reach the resolution limit simultaneously.
The above algorithm can also be used to determine missing sequences as well.

Appendix B

Algorithm to generate simulated sequence by predicting the order of nucleotides in a sequence using the probability of dinucleotide frequency - Dutta et.al in 1992 [8]

Input: DNA Sequence
Output: CGR of simulated sequence.

Step 1. Determine the frequency of occurrences of dinucleotides from a given sequence.

Step 2. Set the length L of the hypothetical sequence to be generated.

Step 3. Randomly choose the first base N(1) of the sequence.

Step 4. Set i = N(1), Set j = 2;

Step 5. Repeat 6 and 7 while $j \leq L$

Step 6. Generate a number n between 0 and 1.
if $0 < n < P_{iA}$ then N(j) = A
if $P_{iA} < n < P_{iT}$ then N(j) = T
if $P_{iT} < n < P_{iG}$ then N(j) = G
if $P_{iG} < n < P_{iC}$ then N(j) = C
where P_{iA}, P_{iT}, P_{iG} and P_{iC} represent the probability of A,T,G,C follow the *ith* nucleotide.

Step 7. Set i = N(j) and j = j +1.

Step 8. Generate CGR for the hypothetical sequence and compare with the CGR of the original sequence.

Step 9. If the CGRs do not match then reset the values of the probabilities.

Guidelines for setting the probabilities were also provided based on trial and error method, also said, the algorithm could be modified to use the probability occurrences of tri-, tetra- or *k*-nucleotides. Therefore, [8] concluded, the sparse region in the CGRs of vertebrate gene (Fig 4.2) are due to the rare occurrences of the dinucleotide and not due to any non-random occurrence of single nucleotide.

References

[1] Bioinformatics glossary. URL: http://big.mcw.edu/.

[2] Protein information resource. URL: http://pir.georgetown.edu/.

[3] Fractal geometry, 2005. URL: http://classes.yale.edu/Fractals/.

[4] J.S Almedia, J.A Carrico, A. Maretzek, P.A. Noble, and M. Fletcher. Analysis of genomic sequences by chaos game representation. *Bioinformatics*, 17(5):429–437, 2001.

[5] M. Baranger. Chaos, complexity and entropy.

[6] S. Basu, A. Pan, C. Dutta, and J. Das. Chaos game representation of proteins. *Journal of Molecular Graphics and Modelling*, 15:279–289, 1997.

[7] P. Bourne and H. Weissig, editors. *Structural Bioinformatics.* 2003.

[8] C. Dutta and J. Das. Mathematical characterization of chaos game representation. *Journal of Molecular Biology*, 228(3):715–719, 1992.

[9] I. Eidhammer, I. Jonassen, and W.R. Taylor. *Protein Informatics.* John Wiley & Sons, 2004.

[10] K. Falconer. *Fractal Geometry, Mathematical Foundations and Applications.* John Wiley & Sons, 1990.

[11] A. Fiser, E.G. Tusnady, and I. Simon. Chaos game representation of protein structure. *Journal of Molecular Graphics*, 12, 1994.

[12] N. Goldman. Nucleotide, dinucleotide and trinucleotide frequencies explain patterns observed in chaos game representations of dna sequences. *Nucleic Acids Research*, 21(10):2487–2491, 1993.

[13] D. Harte. *Multifractals - Theory and Application.* 2001.

[14] K.A Hill, N.J Schisler, and S.M Singh. Chaos game representation of coding regions of human globin genes and alcohol dehyrogenase genes of phylogenectically divergent species. *Journal of Molecular Evolution*, 35:261–269, 1992.

[15] K.A. Hill and S.M Singh. The evolution of species-type specificity in the global dna sequence organization of mitochondrial genomes. *Genome*, 40:342–356, 1997.

[16] J.H Jeffrey. Chaos game representation of gene structure. *Nucleic Acids Research*, 18(8), 1990.

[17] D.E Krane and M.L. Raymer. *Fundamental concepts of Bioinformatics.* Benjamin Cummings, 2003.

[18] K. Pleibner, L. Wernisch, H. Oswald, and E. Fleck. Representation of amino acid sequences as two-dimensional point patterns. *Electrophoresis*, 18:2709–2713, 1997.

[19] N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology Evolution*, 4(4):406–425, 1987.

[20] Y. Wang, L. Kari, K.A Hill, and S.M. Singh. The spectrum of genomic signatures: from dinucleotide to chaos game representation. *GENE*, 346:173–185, 2005.

[21] Z. Yu, V. Anh, and K. Lau. Chaos game representation of protein sequences based on the detailed hp model and their multifractal and correlation analyses. *Journal of Theoretical Biology*, 226:341–348, 2004.

[22] K. Zimmerman. *An Introduction to Protein Informatics.* Kluwer Academic, 2003.

Vita

Name	Annie Christiana Vasthi Thomas
Place of Birth	Kancheepuram, Tamilnadu, India
Post-secondary Education and Degrees	Madras Christian College Tambaram, Tamilnadu, India 1992–1995 B.Sc.
	Madras Christian College Tambaram, Tamilnadu, India 1995–1998 M.C.A.
	The University of Western Ontario London, Ontario, Canada 2003–2005 M.Sc.
Honours and Awards	International Student Scholarship, 2003-2004
Related work experience	Lecturer, Madras Christian College Tamilnadu , India 2002-2003
	Lecturer, Bharath Institute of Science and Technology Tamilnadu, India 2001–2002
	Teaching Assistant The University of Western Ontario 2003–2004
	Research Assistant The University of Western Ontario 2003–2005