Waveform Mapping and Time-Frequency Processing of Biological Sequences and Structures

by

Lakshminarayan Ravichandran

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved July 2011 by the
Graduate Supervisory Committee:

Antonia Papandreou-Suppappola, Co-Chair
Andreas Spanias, Co-Chair
Chaitali Chakrabarti
Cihan Tepedelenlioğlu
Zoé Lacroix

ARIZONA STATE UNIVERSITY

August 2011

ABSTRACT

Genomic and proteomic sequences, which are in the form of deoxyribonucleic acid (DNA) and amino acids respectively, play a vital role in the structure, function and diversity of every living cell. As a result, various genomic and proteomic sequence processing methods have been proposed from diverse disciplines, including biology, chemistry, physics, computer science and electrical engineering. In particular, signal processing techniques were applied to the problems of sequence querying and alignment, that compare and classify regions of similarity in the sequences based on their composition. However, although current approaches obtain results that can be attributed to key biological properties, they require pre-processing and lack robustness to sequence repetitions. In addition, these approaches do not provide much support for efficiently querying sub-sequences, a process that is essential for tracking localized database matches.

In this work, a query-based alignment method for biological sequences that maps sequences to time-domain waveforms before processing the waveforms for alignment in the time-frequency plane is first proposed. The mapping uses waveforms, such as time-domain Gaussian functions, with unique sequence representations in the time-frequency plane. The proposed alignment method employs a robust querying algorithm that utilizes a time-frequency signal expansion whose basis function is matched to the basic waveform in the mapped sequences. The resulting WAVEQuery approach is demonstrated for both DNA and protein sequences using the matching pursuit decomposition as the signal basis expansion. The alignment localization of WAVEQuery is specifically evaluated over repetitive database segments, and operable in real-time without pre-processing. It is demonstrated that WAVEQuery significantly outperforms the biological sequence alignment method BLAST for queries with repetitive segments for DNA sequences. A generalized version of the WAVEQuery approach with the metaplectic transform

is also described for protein sequence structure prediction.

For protein alignment, it is often necessary to not only compare the one-dimensional (1-D) primary sequence structure but also the secondary and tertiary three-dimensional (3-D) space structures. This is done after considering the conformations in the 3-D space due to the degrees of freedom of these structures. As a result, a novel directionality based 3-D waveform mapping for the 3-D protein structures is also proposed and it is used to compare protein structures using a matched filter approach. By incorporating a 3-D time axis, a highly-localized Gaussian-windowed chirp waveform is defined, and the amino acid information is mapped to the chirp parameters that are then directly used to obtain directionality in the 3-D space. This mapping is unique in that additional characteristic protein information such as hydrophobicity, that relates the sequence with the structure, can be added as another representation parameter. The additional parameter helps tracking similarities over local segments of the structure, this enabling classification of distantly related proteins which have partial structural similarities. This approach is successfully tested for pairwise alignments over full length structures, alignments over multiple structures to form a phylogenetic trees, and also alignments over local segments. Also, basic classification over protein structural classes using directional descriptors for the protein structure is performed.

To *Amma, Appa*

and

my sister *Nikila*.

ACKNOWLEDGEMENTS

During the course of my graduate studies at ASU, there have been many people that have helped me become what I am today. I certainly would not have been to able to achieve this distinction without the support of these individuals who have supported and influenced me during the good and bad times during these last six years, in some form or another.

To motivate is not an easy task. And to do that for over five and a half years involves a huge amount of patience. I have been really fortunate to be advised by Prof. Antonia Papandreou-Suppappola, who has been a constant pillar of support and inspiration through these years and continues to be one. You have definitely been more than an advisor to me and have been the main reason I have been evolving as a different individual. During the many tough times that I have faced here, you are the first person I would turn to, and you definitely have brought about a change in the way I look at things. Be it the passion in serving your students or the concern you have shown, your selfless nature and dedication never ceases to amaze me and you surely are a role model for me to look up to. I will always be thankful to you for everything you have been to me. I would also like to thank you for funding the numerous conferences and workshops that I attended. The interaction with other researchers in this field that resulted during these trips was surely an enriching experience.

I am greatly indebted to my co-chair Prof. Andreas Spanias, who has provided constant encouragement throughout this study. Your inputs at various stages of the study have been very helpful. Thanks to your useful classes on basic and advanced signal processing techniques, I was able to approach ideas with greater clarity. I would also like to thank you for supporting me financially via the SenSIP center and other grants.

I would like to thank Prof. Chaitali Chakrabarti, Prof. Cihan Tepedelenlioğlu and Prof. Zoé Lacroix for serving on my PhD dissertation committee. Applying the algorithm optimization techniques that I have learnt from Prof. Chakrabarti's classes has been very rewarding in this study. The concepts from advanced linear algebra and convex optimization class by Prof. Tepedelenlioğlu have been immensely helpful in understanding a few of the key papers in literature. Special thanks to Prof. Zoe Lacroix and her Scientific Data Management Laboratory at ASU, as well as Dr. Christophe Legendre for their collaboration and their valuable inputs in relation to the sequence aligment and querying algorithms presented in this study.

I would like to thank Dr. Narayan Kovvali for the discussions involving the Gaussian window design used in the protein structural alignment problem. I would also like to thank Dr. Nathalie Meurice from the Translational Genomics Research Institute (TGen) for providing me with a good insight into the protein structural alignment problem during my internship at TGen during the Summer of 2008. Thanks to Prof. Lina Karam for serving on my PhD Qualifying Exam committee and providing me with valuable inputs. I would also like to thank Prof. Douglas Cochran, Prof. Tolga Duman, and Prof. Jennie Si as the courses I took under them stimulated my research thinking.

The School of Electrical, Computer and Energy Engineering at the Ira A. Fulton Schools of Engineering at ASU provided me with an invaluable opportunity to serve the undergraduate students by awarding me the graduate teaching associateship from August 2006 to May 2011. I would like to thank Prof. Joseph Palais for providing me with this opportunity, and the Undergraduate Laboratory Manager Mr. Clayton Javurek, for his able guidance and support in these five years. It was a fruitful and learning experience, instilling in me the confidence about my communication skills and further helping me develop my time manage-

ment skills. I would also like to express by deepest gratitude to the Graduate and Professional Students Association (GPSA) at ASU for honoring me with the Teaching Excellence Award during the 2010–11 academic year.

I would like to thank Ms. Iona Show from the Division of Graduate Studies at ASU for her motherly affection and constant encouragement during this entire coursework. It was due to her that I was able to adjust to the different way of life in the United States and manage my courses and other activities efficiently. I would also like to thank Ms. Cynthia Moayedpardazi, Ms. Darleen Mandt, Ms. Esther Korner, Ms. Donna Rosenlof, Ms. Cheryl McAfee, Ms. Karen Anderson, Ms. Jenna Marturano, Ms. Ginger Rose, and Ms. Farah Kiaei from the School of Electrical, Computer and Energy Engineering for their assistance in the administrative matters.

In a home away from home, there have been many friends who have helped me cope up through hard times, and I would like to thank them for being there for me. Firstly, I would like to thank Dr. Uma Swamy, under whose able guidance I learnt to work efficiently on research, teaching, mentoring and other activities. I would like to extend my heartfelt thanks to Mahesh, Thripthi, Dilip, Supraja, Sapna, Keerthana, Nivedita, Harish, Prathap, Kumar, Raghavendra, Ashwini, Jyothiswaroop, Teja, Sruthi, Shanta, and Shibani for their help, support and the good times at various stages of this study.

I would also like to thank all the members of the Signal Processing and Adaptive Sensing Laboratory for providing a conducive atmosphere in our lab. The research discussions with the members of Speech and Audio Processing Lab were very helpful, and I would like to thank Jayaraman and Karthikeyan for arranging them.

Finally, I would like to thank my parents and my sister Nikila for everything

that they are to me. Without the three of you, I would not be where I am today. Thank you *Amma, Appa* and *Nikila*!

TABLE OF CONTENTS

Page

TABLE OF CONTENTS viii

LIST OF TABLES xii

LIST OF FIGURES xiii

CHAPTER 1

1 INTRODUCTION 1

1.1 Genomics, Proteomics and Bioinformatics 1

1.2 Genomic Signal Processing 3

Fourier Analysis and Spectrogram 4

Filtering Techniques 5

Time-Frequency Analysis 6

Feature-based Analysis 6

1.3 Genomic Alignment 6

Sequence Alignment 7

Structural Alignment 8

1.4 Motivation for Waveform Mapping and Processing Alignment . . 10

1-D Waveform Mapping and Sequence Alignment . 10

3-D Waveform Mapping and Structural Alignment . 11

1.5 Organization 13

1.6 List of symbols/variables used in the report 14

2 DNA AND PROTEIN BIOLOGY 16

2.1 DNA Biology 16

2.2 Protein Biology 19

2.2.1 Protein Sequence Composition 19

2.2.2 Protein Structures 22

2.3 Reference Databases 23

Chapter Page
3 MAPPING SCHEMES FOR DNA AND PROTEIN SEQUENCES . . 26
3.1 Traditional Numerical Mapping Schemes 26
3.1.1 Indicator sequence mapping 26
3.1.2 Real number mapping 30
3.1.3 Complex number mapping 31
3.2 Waveform Mapping Schemes 32
3.2.1 Sinusoid Waveform Mapping 34
3.2.2 LFM Chirp Waveform Mapping 35
3.2.3 Gaussian Waveform Mapping Scheme 38
4 QUERY-BASED DNA SEQUENCE ALIGNMENT 40
4.1 Types of Sequence Alignment 40
Global sequence alignment 40
Local sequence alignment 41
Multiple sequence alignment 41
4.2 Sequence Alignment Tools 41
4.2.1 Computational Methods 41
4.2.2 Signal Processing-based Approaches 43
4.3 Sequence Alignment Scenarios 46
Case 1: Complete alignment 47
Case 2: Un-gapped local alignment 49
Case 3: Gapped local alignment 51
Case 4: Global alignment 52
4.4 Querying using Cross-correlation based matched filtering 54
4.5 Matching Pursuit Decomposition based Querying Algorithm . . . 55
4.5.1 Matching Pursuit Decomposition Algorithm 56
4.5.2 MPD WAVEQuery Alignment of DNA Sequences 57
4.5.2.1 WAVEQuery for Globalized Querying 58

Chapter Page
4.5.2.2 WAVEQuery for Localized Querying 59
4.5.2.3 WAVEQuery for Localized Querying with Gap Insertions and Deletions 60
4.6 Simulation Results 62
4.7 MPD WAVEQuery Alignment of Protein Sequences 69
4.8 WAVEQuery Using the Metaplectic Transform 71
5 STRUCTURAL WAVEFORM MAPPING FOR PROTEIN ALIGNMENT 74
5.1 Structural Similarities in Proteins 74
5.2 Current Structural Alignment Techniques 75
5.2.1 Computational-based Structural Alignment 75
5.2.2 Signal Processing Based Structural Alignment 77
Gaussian Based Alignment 77
Fourier Transform Based Alignment 79
5.2.3 Other Signal Processing Based Alignment Methods 80
5.3 Need for New Structural Alignment Techniques 80
5.4 Modeling the Protein Superposition Problem 81
5.5 Chirp wAveform Representation for Protein Structures (CARPS) 82
5.5.1 Waveform Representation Model 82
5.5.2 Chirp-based Protein Structure Representation 85
5.5.3 Waveform Parameters relating Sequence to Structure . . . 87
5.6 Chirp-based Alignment for Protein Structures (CAPS) Approach . 89
5.6.1 Pairwise Alignment of Protein Structures 89
Global Structural Alignment 90
Local Structural Alignment 91
5.6.2 Extension to Alignment of Multiple Protein Structures . . 92
5.6.3 Classification among Structural Classes based on Directional Descriptors 94

Chapter Page
5.7 Experimental Setup And Results 95
5.7.1 Global Alignment 95
5.7.2 Special Case of Locally Aligned Segments 97
5.7.3 Classification of Protein Structures 98
6 CONCLUSIONS AND FUTURE WORK 103
6.1 Conclusion 103
6.2 Future Work 106
6.2.1 Sequence Alignment Algorithm 106
6.2.2 Structural Alignment Algorithm 106
REFERENCES 108

LIST OF TABLES

Table Page

4.1 Information on the data sets used for testing the proposed alignment algorithms. 62

4.2 Sample globalized alignment using matched filtering with LFM signals and BLAST; both methods obtained identical results 64

4.3 Comparison of BLAST and WAVEQuery performance for localized querying on dataset DB100 66

4.4 Execution time of WAVEQuery localized sub-sequence querying for different database sets 68

5.1 Pairwise Global Structural Alignment Results. 96

5.2 Confusion Matrix for Protein Structure Classification 102

LIST OF FIGURES

Figure Page

1.1 Proposed globalized and localized query-based alignment scheme for biological sequences. 11

2.1 Double helical strand of DNA [1]. 16

2.2 Formation of building block of DNA [2]. 17

2.3 Single DNA strand. 17

2.4 Double stranded DNA. [2] 18

2.5 Example of an exon-intron structure. [2] 19

2.6 Genetic code [2]. 20

2.7 A general amino acid with a side-chain R. [2] 21

2.8 Formation of a peptide bond from two amino acids. Note the covalent bond formed and and the released water molecule. [2] 21

2.9 PDB file showing the co-ordinates corresponding to the tertiary structure position of an amino acid. 25

3.1 Magnitude of the FT for the coding region of DNA from *S.Cerevisae*. Note that the peak occurs at $f = N/3$ where $N = 1871$ in this example. 28

3.2 Power spectrum related to the FT in Figure 3.1 for the coding stretch of DNA from *S.Cerevisae*. 29

3.3 Spectrogram for the coding stretch of DNA from *S.Cerevisae*. 29

3.4 Real number mapping. 31

3.5 Complex number mapping 32

3.6 Sinusoid signals representing the four nucleobases.The duration of the signal is 0.1 seconds, and the sampling frequency is 1000 Hz. 35

Figure Page
3.7 Instantaneous frequency of LFM chirp waveforms, representing the four nucleobases. The duration of the signal is 0.1 seconds, and the sampling frequency is 1000 Hz. The frequency axis is shown normalized by the sampling frequency 37
3.8 LFM chirp waveforms representing the four nucleobases. When discretizing the chirps, the highest FM rate was chosen to satisfy $c_4 \leq\leq \frac{f_s}{4T_d}$ in order to avoid aliasing. Here f_s is the aliasing frequency and T_d is the duration of the signal. For this example T_d is 0.1 seconds, and f_s is 1000 Hz. The instantaneous frequencies of these waveforms are provided in Figure 3.7. 37
3.9 Gaussian waveforms representing DNA nucleotide bases in the time-frequency plane based on their position in a sequence. 39
3.10 Example of four Gaussian waveforms representing the DNA sequence $\{ATCA\}$. 39
4.1 Query and database sequences that need to be aligned. 46
4.2 Complete alignment. The lines (|) represent a match and the asterisk (*) represents a mismatch. 47
4.3 Sub-sequences $d_1(t)$ and $d_2(t)$ from the database sequence $d(t)$ for $Q =$ 4 characters in each sub-sequence. 48
4.4 Sub-sequence $d_1(t)$ using the sinusoid mapping with $Q = 4$ and $\tau_s = 1$ second. 48
4.5 Un-gapped local alignment. 50
4.6 Gapped local alignment. 52
4.7 Global alignment. The solid portion of the line indicates the query sub-sequence higher measure of similarity, whereas the dotted lines represent the acceptable measures of similarity. 53

4.8 Correlation value (similarity measure) versus position for sequences obtained from *S. Cerevisae*. The maximum correlation value (of 36) occurs at position 51 in the database sequence. 54

4.9 Gapped alignment example using the WAVEQuery approach. (a) No gaps inserted; (b) one gap inserted at iteration 36; and (c) two gaps inserted at iterations 36 and 86-88. Note that the gaps are inserted when the similarity measure reduces. 63

4.10 Alignment report for the localized sub-sequence querying cases with BLAST metrics (score and E-value). Note that we used the name notation as the one used in BLAST for ease of comparison; as a result we use $1.779385\,e^{-31}$ to represent 1.779385×10^{-31}. 67

4.11 Alignment report for the WAVEQuery algorithm for protein alignment compared with BLAST raw score. Note the amino acid mismatches with positive value in the substitution matrix are represented by a '+' and the other mismatches are represented by a '.'. 72

5.1 Example of the CARPS for the NMR structure of lung surfactant peptide SP-B (PDB ID: 1KMR). The axes measurements are all in Angstrom units (10^{-10} m). Note the α-helix in the structure connected by 3-D chirps with a Gaussian window. 87

5.2 Two threshold level similarity matrix plot for the local structural alignment case. The structures of two proteins are aligned locally in the regions specified by the regions of similarity diagonally. The case of 5 and 8 aligned segments is considered as a structural match, while the 2 aligned segments are not considered to be a structural match. . . . 93

5.3 Local structural alignment case where two protein are structures aligned over the entire length of the structure except over a portion. 93

5.4 Pairwise global structural alignment for cyclotide Cter M (2LAM) is shown. Note the superposition of the 29 residues and connecting the segments. 97

5.5 Multiple global structural alignment for (2L24) is shown. We considered over 10 structures with different initial conformations and superposed all of the structures together. Note the superposition of the 13 residues over each of the 10 structures and the segment connections. . 98

5.6 Local structural alignment example case of α-helix in 2L8K. Note the alignment of a helix of length 19 along the structure. The locally aligned structure is connected by blue dots and appears shifted for better view. 99

5.7 Local structural alignment example case of an all β-sheet structure with two sub-structures of lengths 22 and 19 aligned. The locally aligned structure is connected by blue dots and appears shifted for better view. 99

5.8 Local structural alignment case of local alignment in the β-Hairpin Peptidomimetic Inhibitor at the hairpin segment. The locally aligned structure is connected by blue dots and appears shifted for better view. 100

5.9 Local structural alignment example with a short misaligned segment in 1J4M. The locally aligned structure is connected by blue dots and appears shifted for better view. 100

5.10 Local structural alignment example with a short misaligned segment in 1KWE. The locally aligned structure is connected by blue dots and appears shifted for better view. 101

Chapter 1

INTRODUCTION

1.1 Genomics, Proteomics and Bioinformatics

Deoxyribosenucleic acids (DNA) are nucleic acids that are in the form of double helical strands that contain the genetic instructions specifying the biological development of all cellular forms of life. A single strand of DNA is a bio-molecule consisting of many linked, smaller components called nucleotides. Each nucleotide is formed by the nucleobases adenine, thymine, cytosine, and guanine that are represented by the letters A, T, C and G, respectively; each DNA single strand is represented by a character string of these four letters [2].

Proteins are bio-molecules that consist of many linked, smaller components called amino acids. There are twenty types of amino acids in proteins that are linked by strong peptide bonds to form polypeptide chains. The protein functions are actually determined by the DNA character string, since the information in the DNA sequences determines the amino acid sequences.

Genomic engineering is an interdisciplinary field that combines critical biological genome information and knowledge from areas such as bio-science, medicine, computer informatics, and engineering [3]. In particular, genomics deals with the study of large genetic information in order to understand the collective gene function [4]. Genomic information is discrete and represented in sequences of unique elements frame from finite element dictionaries [5]. For example, a DNA sequence consists of the precise ordering of four possible nucleobases (A, T, C, G) from which the DNA is composed; each ordering corresponds to a pattern that influences the formation and development of an organism. Similarly, a protein sequence consists of the precise ordering of twenty amino acids represented by twenty unique letters of the alphabet.

Two technologies, sequence analysis and micro-array analysis, have played significant roles in the extraction and interpretation of genomic information [6, 7]. Sequence analysis helps in the study and understanding of structure-related information. It can be used to reveal some hidden structure, distinguish between coding and non-coding regions, and explore structural similarities between DNA and protein sequences [3]. DNA micro-arrays (also called gene arrays or DNA chips) are useful in simultaneously observing interactions between thousands of genes, in order to determine expression levels of genes, discover genes and drugs, and diagnose diseases.

Proteomics is the study of protein structures and functions, and it includes the prediction of new protein structures, identification of different structural classes and classification of protein structures based on different similarity measures [8]. Bioinformatics, on the other hand, is the application of computer science and information technology methodologies to biology, such as genomics and proteomics. It includes the development of databases and computation and statistic techniques to manage and analyse biological data. These techniques can be applied to gene sequences to search for embedded information in biological systems [9] and they include methods from areas such as data mining, statistics, pattern recognition and visualization. The three main goals in bioinformatics are data organization, analysis and interpretation. Specifically, organization is important as data needs to be stored for easy access in public databases. The development of querying or sequencing tools using computational theory will help analyze the data and gain valuable knowledge. Note, however, that the knowledge will not be meaningful unless the analysis is interpreted in a biologically meaningful manner.

The primary research topics in bioinformatics [9–11] include: sequencing and comparison of genomes of different species; studying relationships between

structures and functions; predicting three-dimensional (3-D) molecular protein structures of amino acid sequences; modeling genetic regulatory network; tracing evolutionary relationships between species and constructing *phylogenetic* trees; and discovering of associations between gene mutations and diseases.

1.2 Genomic Signal Processing

The traditional bioinformatics methods of pattern matching and statistical analysis for processing DNA and protein sequences in their element representation can be very time-consuming as a huge volume of genomic data is currently available [3, 4, 12–14]. One way to apply more efficient processing methodologies to genomic sequences is by converting them into discrete-time signals [4–9, 12–14]. The corresponding recently evolving area of developing approaches to analyze and process genomic signals is called genomic signal processing. Its aim is to integrate the theory and methods of signal processing with the global understanding of genomics, placing special emphasis on genomic regulation [6]. Due to the discrete nature of the DNA and protein data, signal processing techniques to can be used to analyze and understand the characteristics of DNA and proteins as well as their interaction [11].

Signal processing techniques based on the Fourier transform (FT) have been used for gene identification [15, 16]. Statistical analysis methods such as techniques based on the correlation function of DNA sequences have been used to study their inherent functionality and structure as well as their statistical dependencies [17–19]. Time-frequency analysis techniques have also been applied to protein data in [20, 21]. Some additional information on the use of signal processing methodologies to process and analyze biological data is provided next.

Fourier Analysis and Spectrogram One of the primary reasons for using FT analysis techniques with DNA data is to detect periodicities in different DNA sequences. The effect of periodicity in DNA coding sequences on gene evolution is discussed in [15]. Periodicities that are relevant to the DNA structure are those related to telomere region, protein coding in DNA, DNA helical folding, DNA nucleosome binding, and DNA nucleosome superstructure. Patterns of quilts, shafts and bars are significant in describing the structure of the cell surface proteins [22].

The FT has been used to measure periodicity for segments of DNA data with differing in base content [23]. A detailed discussion on the application of FT analysis to DNA sequences is presented in [24–26]. In [16], the FTs of the DNA sequences were computed using a sliding window to identify coding regions present in the sequences. An important property that characterizes and determines the coding regions is the three-base periodicity that is related to the value of the FT at frequency $f = N/3$, where N is the length of the window. The three-base periodicity is also called the $N/3$ periodicity or $2\pi/3$ periodicity. This coding measure is obtained by first mapping the DNA sequence to discrete-time sequences, computing the discrete FT (DFT) of the sequences as in [27], and then looking at the relative strength of the periodicity at $f = N/3$; this should appear as a peak in the average DFT. Note that the origin of the periodicity can be attributed to the codon bias, which is the unequal usage of codons in the coding regions, and the triplet bias, which is the bias in the usage of nucleotide triplets (see Figure 2.6). The advantages of this technique are that it is robust to sequencing errors and its computational complexity is very low. A mathematical explanation for the peak at $N/3$ in a coding region is provided in [28–30], together with a discussion on the use of other windows such as the Bartlett window for the DFT computation and a DFT based splicing algorithm. The use of the warped DFT is presented in [31], where a more pronounced peak is observed at $N/3$.

An extension of the FT approach was proposed in [5, 32, 33], using DNA spectrograms (squared magnitude of the short-time FT) to differentiate between coding and non-coding regions. In order to compute the spectrogram, the DNA sequences are first converted to numbers, and thus discrete time-domain sequences, using binary or complex mapping. Note that weights on the discrete time-domain sequences are optimized such that the observed peaks at $f = N/3$ can be distinguished from the non-coding regions. A color coding scheme to obtain the color maps is also described in [5].

The efficiency and advantages of using digital signal processing based techniques for DNA sequence analysis are presented in [5]. In [22, 34], the DNA spectrogram was used in the study of periodicities from 0 to 300. These periodicities were related to small periodic patterns called tandem repeats (minisatellites, quilts and shafts). The prospect of sequence classification using the relative base content was also discussed. The short-time Fourier transform was used to find latent periodicities in [35] and to study the structure of the exon in [36].

Filtering Techniques The use of discrete infinite-impulse response (IIR) filters to identify coding regions in DNA sequences was presented in [37–39]. As the $N/3$ periodicity is exhibited by the coding region in a DNA sequence spectrum, an anti-notch filter to identify these coding regions was proposed in [38]. The design of the anti-notch filter is based on the fact that there is a sharp peak at $2\pi/3$ in the spectrum. Thus, after observing the mapped DNA sequences through the filter and the output signal spectrum, if there is a pronounced peak at $2\pi/3$, then the sequence is from the coding region of the DNA. A detailed analysis on the design of the anti-notch IIR filter and the trade-offs in the design is also provided in [40].

Time-Frequency Analysis Wavelet transform analysis has been used to characterize the long range correlations in DNA sequences [41–43] and thus to study the structure of the nucleosome. They have also been used to identify the origin and terminating sites for DNA replication and the irregularities in the DNA data.

Time-frequency techniques have also been used to analyze protein data [20]. Specifically, the Wigner-Ville distribution, a quadratic time-frequency representation, was used to observe characteristic patterns in protein sequences, to obtain spatial information about the secondary structure of proteins, and determine biologically active sites in the protein molecule. A novel real number mapping rule was proposed in this case based on the hydrophobicity value of the amino acid and the codons corresponding to it.

Feature-based Analysis Autoregressive (AR) models have been used with DNA sequences to perform linear prediction analysis in [19]. The AR model parameters were used as features of the DNA segments, and it was found that the AR model was very specific to the fitting coding sequence and the specificity increased with the order of the AR model. These feature-based analysis were used to perform DNA string searches and study the characteristics of coding versus non-coding regions.

1.3 Genomic Alignment

Two important problems in the analysis of genomic and proteomic data are the problems of sequence alignment and structural alignment. Alignments of sequences and structures are useful for studying evolutionary relationships between organisms and for determining DNA or protein functionality [2].

Sequence Alignment The sequencing and comparison of genomic and proteomic data is referred to as sequence alignment. The goal of sequence alignment is to determine if there are any sequences in the public databases that are similar to a query sequence [44]. If two sequences are similar, then the sequences have *related structures or functions*. Specifically, a sequence structure relates to the 3-D protein structure and its function refers to the gene function. As information about the sequence structure and function is known for most sequences, it can be shared between the similar sequences. In addition, similar sequences may have a *common ancestor sequence*. If two sequences are similar up to a few elements, it is likely that both sequences evolved from a common ancestor, and an evolutionary relationship may exist between the source of each sequence. If the query sequence is a partial sequence (portion of a larger sequence), it may be possible to *gain information about the sequence's position and about the role of the original sequence*.

A simple approach towards aligning two sequences would be to search for an exact match within the two sequences. However, there are a few problems with using exact matches to align two sequences. Most of the sequences studied are derived from cloned copies. During cloning, mutations may occur and errors may be introduced while sequencing. As a result, the alignment method must be tolerant of these mutations and errors. The DNA strand may contain many nucleobases that do not create amino acids. Hence, it is desirable not to discard alignments with those nucleobases. Sequences within species and between species have variations. If the similarity is greater between the DNA strands, then it can be concluded that the species are closely related. During evolution, a sequence might have dropped a few properties or gained a few properties. In sequence alignment, this is seen as insertions and deletions in the sequences. The alignment tool must insert gaps in the sequences, if required.

Structural Alignment In sequence alignment, the primary DNA and protein sequences are arranged based on the similarity in their composition. In proteins, this sequence composition of amino acids is also called the primary 1-D structure. In addition to this primary structure, there also exist secondary, tertiary and quaternary structures which contain crucial information about the protein function. These structures are related to the primary structure based on the composition of the protein. However, the structures take different shapes due to the hydrogen bonds, ionic bonds, and *Van der Waals* attractions between the molecules that make up the amino acids. The aim of structural genomics is to determine the protein structure and classify them based on their functions.

There is a lot of structural information generated for proteins and stored in reference databases, especially after the sequencing of the human genome [45]. Some known protein reference databases, that contain information about proteins and their structures include Protein DataBank (PDB) [46], International Protein Sequence Database Collaboration, Swiss-Prot, TrEMBL, Protein Information Resource Protein Sequence Database (PIR-PSD). The Protein Structural Initiative (PSI) [47] is a federal research funding effort for universities and industry for cost-effective protein structure determination. Note, however, that the reference databases lack complete protein structural annotation, i.e., identification of gene elements such as coding regions, gene structure, or regulatory motifs. This happens even though the protein's 3-D folds, such as α helices and β sheets, are known. This is important because a similarity in the 3-D folds structure of two proteins, characterized by the secondary and tertiary structures, implies similarity in the function of the two proteins that possess this structure. In particular, if two proteins have similarities, then in their structure, they have similarities in their function.

The genome sequencing project that started in 1990, aimed at relating

a protein sequence to its structure. Two sequences with a different amino acid composition could result in a similar structure, i.e., structural similarity need not imply sequence similarity [48]. There was a large difference between sequence similarity and structural similarity in distantly related proteins. This is because protein structures have a much higher degree of property conservation compared to sequences; the proteins share common structures, even though the sequence composition may be different. Hence, a need for aligning two proteins beyond aligning their sequences became necessary. This resulted in the use of secondary and tertiary protein structures to determine the similarities in their functions. This resulted in the birth of a new field called structural genomics in the late 1990s, which aimed at assigning one structure per functional class [47].

Protein structure comparison or structural alignment is important in classifying the proteins into different structural classes, and in comparing the theoretically predicted structures with the experimentally determined structures. Also the conserved regions are local in distantly related proteins, i.e., the similarity need not occur over the entire structure. Hence, there is a need to find the structural similarity locally.

The protein secondary and tertiary structures are in the form of 3-D structures or 3-D shapes. The structural comparison and detection of similarities in these 3-D structures requires optimal comparison or alignment of structures in the 3-D plane. This is also known as protein structural superposition, and it allows for classification of structures and identification of relationships among the structures. This is very helpful in establishing hierarchical relationships among protein structures and provides a evolutionary view of known structures. Once a newly obtained protein structure is classified, its function can be better understood.

1-D Waveform Mapping and Sequence Alignment Most state-of-the-art approaches for sequence query-based alignment have difficulties when sequences have repetitive segments. As more promising results were shown by the signal processing approaches with sequences mapped to numerical values, we propose a new querying scheme that addresses the repeats problem by mapping sequences to actual time-domain waveforms. Although these waveforms can be correlated in the time-domain to obtain a measure of similarity for sequence alignment, the correlation process can quickly become computationally intensive for localized searches over query sub-sequences.

Our proposed query-based alignment approach, called WAVEQuery, is based instead on effectively processing the waveforms in the time-frequency plane using signal basis expansion techniques whose basis function is the basic waveform used in the sequence mapping. An example of a signal basis expansion is the matching pursuit decomposition (MPD) algorithm whose basis function is the Gaussian waveform [49]. Using the MPD algorithm with a Gaussian dictionary, we assign the MPD frequency-shift parameter to the type of DNA element in a sequence and the time shift parameter to the position of the element in the sequence. Thus, when a query sequence is represented using the WAVEQuery approach, the MPD decomposes the corresponding waveform into a sum of Gaussian functions and as a result, each element of the query sequence has a unique time-frequency shift associated with it. Thus, highly-correlated regions in the database and query sequences are considered to be aligned. The steps involved in the WAVEQuery alignment algorithm are depicted in Figure 1.1.

The WAVEQuery approach can also be used for protein sequence alignment, with partial rewards assigned to amino acids that are similar in composition.

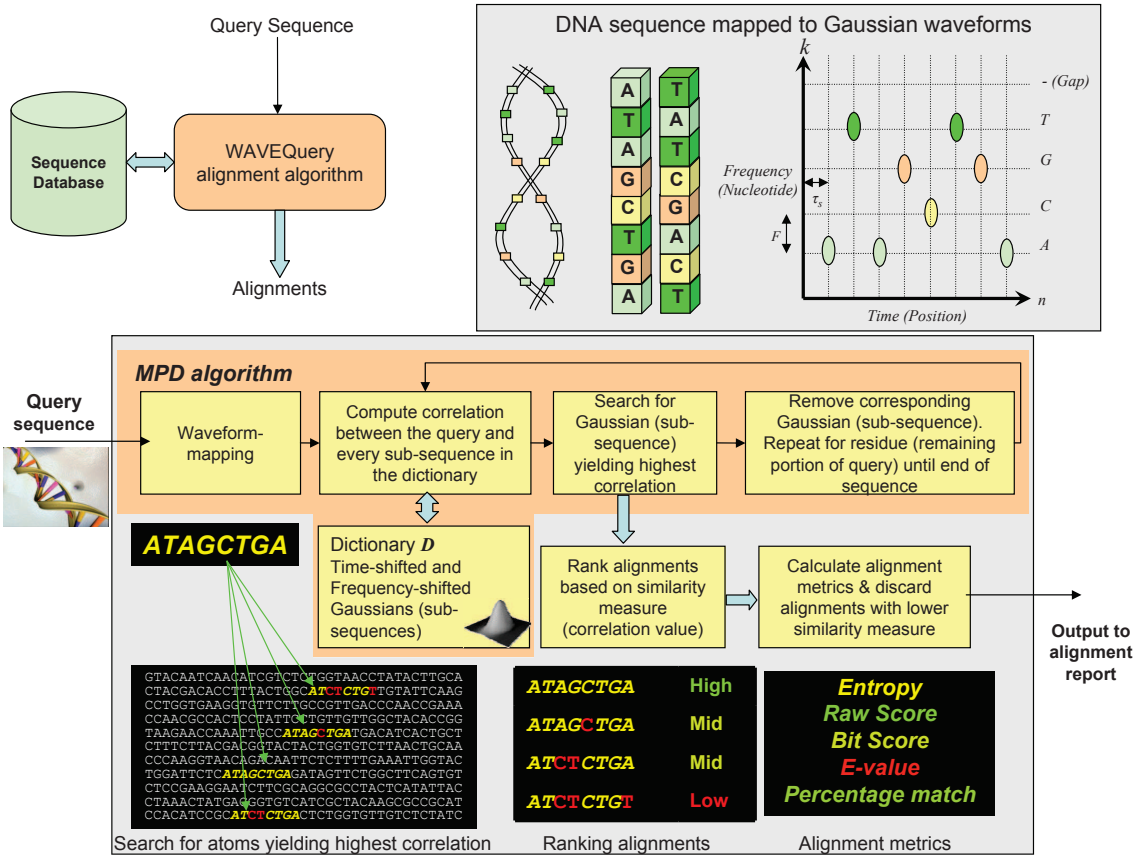


Figure 1.1: Proposed globalized and localized query-based alignment scheme for biological sequences.

Specifically, amino acids that do not correspond to the same letter may still have a similarity due to the DNA transcription process. This similarity is taken into account by assigning the scale change parameter of the Gaussian waveform, so that there is considerable overlap in the Gaussian waveform to denote the partial match in the sequence composition.

3-D Waveform Mapping and Structural Alignment One of the key components in the success of 3-D protein structural alignment methods for identifying folds in structures is directionality. However, current state-of-the-art methods do not consider this important aspect. Methods using waveforms to represent protein structures [50, 51], mostly do not consider all available degrees of freedom

in structure, which can to undergo multiple global and local conformations. As proteins structures have six degrees of freedom, models representing the structure must be translated and rotated in the 3-D plane. As a result, there is a need for a parametric waveform representation that provides a unique shape in the 3-D space and whose parameters can be used to identify changes in 3-D conformations. An important motivation for a 3-D basis representation of to protein structures is the fact that distantly related proteins need not have similarity over the entire structure. Similarities can be localized, and if the representation is linearly separable, it can be used to analyze similar segments over shorter lengths of the structures.

Our proposed chirp-based alignment for protein structures (CAPS) approach is based on a representation that consists of a linear combination of 3-D Gaussian-windowed linear frequency-modulated (LFM) chirp waveforms. These chirp waveforms embed critical information about the structure including the 3-D coordinates ad structure directionality. The use of Gaussian windows makes the representation highly concentration in the time-frequency plane, and by definition, the representation is also linearly separable. By introducing a hydrophobicity parameter in the representation, we also relate the sequence information to the structure information. This is important in predicting and classifying protein structural classes. Unlike other waveform-based representations that require pre-processing of the structural data to achieve rotational invariance, this representation does not require involve any pre-processing.

For the proposed CAPS algorithm, following the representation of the protein structure, we first map the structure to a 3-D time-domain chirp waveform and then perform matching to examine similarity over segments of the structure (locally) and over the entire length of the structure (globally) using 3-D inner product based correlation metric between two or more structures. Furthermore, using the directionality metrics, we can also perform classification of protein struc-

tures among various protein structural classes.

1.5 Organization

This dissertation is organized as follows. In Chapter 2, we provide a brief outline of the biology of DNA and proteins. In Chapter 3, we describe the numerical mapping techniques for DNA and protein sequences and present the proposed sinusoid, LFM chirp and Gaussian waveform mapping techniques for DNA and protein sequences. In Chapter 4, we first discuss current methods of sequence alignment and their drawbacks. We then propose the use of the matching pursuit decomposition algorithm for DNA and protein sequence alignment problems and provide comparative results to demonstrate its performance. We also extend the proposed algorithm to protein sequence alignment by generalizing the waveform basis representation. In Chapter 5 , we propose the 3-D waveform representation and protein structural alignment algorithm, also we provide notable results for structural alignment and classification.

1.6 List of symbols/variables used in the report

Symbols/Variable	Description
A, T, C, G	DNA nucleotide bases
f	Frequency
N	Length of sequence/Window size
$\mathbf{u}_k, (k = A, C, G, T)$	Indicator sequence
$b_k[n](k = A, C, G, T)$	Binary indicator sequence
T_d	Duration of the sine/chirp waveform
f_s	Sampling frequency
$s_l(t)$	Sinusoid signal
$h_l(t)$	LFM chirp signal
c_l	FM rate for LFM chirp signal
$g(t)$	Elementary Gaussian signal
$g_{n,k,l}(t)$	Gaussian waveform function, with time shift m, frequency shift k and time scale l
$O(.)$	Order of computational complexity
$d[n]$	Data sequence
$q[n]$	Query sequence
$d(t)$	Time-domain mapped data signal
$q(t)$	Time-domain mapped query signal
τ_s	Time distance between consecutive nucleotide bases
$d_i(t)$	ith data sub-signal
$q_i(t)$	ith query sub-signal
Q	Number of characters in query sequence
γ	Threshold for alignment
$o_i(t)$	Time-domain signal corresponding

	to gaps at the ith position
$\mathcal{D}$	Dictionary for matching pursuit decomposition
α_i	Expansion coefficient of the MPD for the ith iteration
$r_i(t)$	Residue function after the ith iteration of the MPD
$\mathbf{c}$	FM rate of 3-D chirp waveform
$h_{\mathbf{c}}(\mathbf{t})$	Non-windowed version of 3-D chirp waveform for CARPS
$\boldsymbol{\tau}$	Center of Gaussian window for CARPS
$\boldsymbol{\Sigma}$	Variance of Gaussian window for CARPS
$g(\mathbf{t}; \boldsymbol{\tau}, \boldsymbol{\Sigma})$	Gaussian window for CARPS
$h^g(\mathbf{t}; \mathbf{c}, \boldsymbol{\tau}, \boldsymbol{\Sigma})$	CARPS representation
$\theta_x, \theta_y, \theta_z$	Angle of the Gaussian window w.r.t. (x, y, z) axes
$R_x(\theta_x), R_y(\theta_y), R_z(\theta_z)$	Orientation/Rotation matrices w.r.t. (x, y, z) axes
ρ_i	Hydrophobicity value of the ith amino acid

Chapter 2

DNA AND PROTEIN BIOLOGY

2.1 DNA Biology

Deoxyribonucleic acids (DNA) are nucleic acids that are usually in the form of double helical strands as shown in Figure 2.1. They are very important as they contain the genetic instructions specifying the biological development of all cellular forms of life, as well as most viruses [2]. A single strand of DNA is a bio-molecule

Figure 2.1: Double helical strand of DNA [1].

consisting of many linked, smaller components called nucleotides. The formation of a nucleotide is depicted in Figure 2.2. The DNA polymer is formed by the strong bonding of the sugar molecule of one nucleotide with the phosphate molecule of the next nucleotide, thus creating a sugar phosphate backbone. Each molecule of the DNA consists of two strands around each other to form a double helix, and each rung of the helix consists of a pair of chemical groups called nucleobases or bases. Each nucleotide consists of four bases that are represented by the letters A, T, C and G. The sugars in the DNA are joined together by phosphate groups that form bonds between the third and the fifth carbon atoms of the adjacent sugar rings. Because of these asymmetric bonds, the DNA has an associated direction. In particular, the bases combine in such a way that the sequence on one strand of the double helix is complementary to that on the other. Also, the

nucleotide's direction in one strand is opposite to their direction in the other strand, which means that the strands are antiparallel. The DNA strand ends are thus symmetric and are called five prime (5′) and three prime (3′)

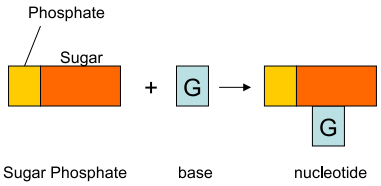


Figure 2.2: Formation of building block of DNA [2].

The (5′) end of a nucleotide is linked to the (3′) end of another nucleotide by a strong chemical bond which forms a long one-dimensional chain with a specific direction. Thus, each DNA single strand is represented by a character string which specifies the (5′) to (3′) direction when read from left to right. This is demonstrated in Figure 2.3.

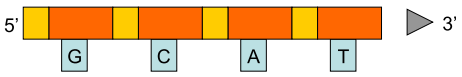


Figure 2.3: Single DNA strand.

Single DNA strands form double helices with other strands in a complementary fashion: A and T are linked together, and C and G are linked together. Even though each single bond is weak, all the bonds together form a stable, double helical structure. The two strands are linked by weak hydrogen bonds, and a simplified straightened out depiction of the linked strands is shown in Figure 2.4. Thus, the three-dimensional (3-D) structure of the DNA, also called the double

helix comes from the chemical and structural features of the two poly-nucleotide chains of the DNA sequences.

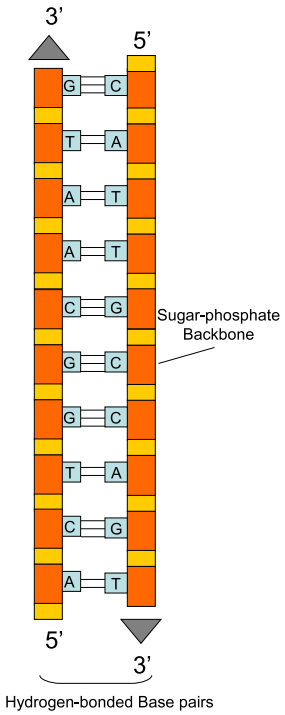


Figure 2.4: Double stranded DNA. [2]

The discrete information that constitutes the genetic blueprint of an organism is stored in the DNA. This information was created and stored during the years of evolution, and a few vital regions of the DNA sequences have been preserved. Note that upon observation, scientists have concluded that the related DNA sections of whales and humans share some common information.

Hence, the genetic information is contained in the sequence of the nucleotide bases of the DNA. The genetic information stored in the organism's DNA contains the instructions for all the proteins the organism will ever synthesize. Specifically, DNA has information about protein coding regions, regions that serve regulatory functions, and regions that serve unknown functions. Protein coding regions in DNA are separated into several isolated sub-regions called

exons. The region between successive exons is called intron. This is shown in Figure 2.5. These introns are eliminated before protein coding by a process called *splicing*.

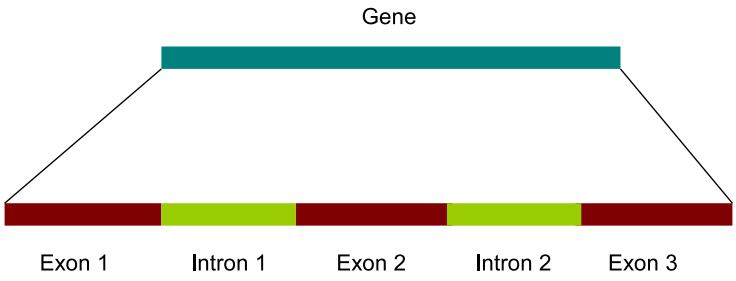


Figure 2.5: Example of an exon-intron structure. [2]

2.2 Protein Biology

2.2.1 Protein Sequence Composition

Proteins are a bio-molecules that consist of many linked, smaller components called amino acids. There are twenty types of amino acids in proteins that are connected by strong bonds as in the DNA case. The bonding process forms a long 1-D chain, known as a polypeptide, with a specific ordered amino acid sequence that determines the protein function. Specifically, each gene or distinct DNA segment contains instructions for making a specific protein. Firstly, a messenger ribonucleic acid (mRNA) is synthesized from the DNA gene segment and the gene information $\{A, C, G, T\}$ is transferred into the alphabet $\{A, C, G, U\}$ in the mRNA; this is called the transcription process. Secondly, the mRNA carries the code for the protein synthesis (translation process). Thus, the flow of genetic information is from DNA to RNA to proteins. The synthesis of proteins is governed by the genetic code that maps all possible triplets (also called codons) of DNA characters into one of twenty possible amino acids. The genetic code is shown in Figure 2.6.

AAA:	K	(Lys)	GAA:	E	(Glu)	TAA:	STOP		CAA:	Q	(Gln)
AAG:	K	(Lys)	GAG:	E	(Glu)	TAG:	STOP		CAG:	Q	(Gln)
AAT:	N	(Asn)	GAT:	D	(Asp)	TAT:	Y	(Tyr)	CAT:	H	(His)
AAC:	N	(Asn)	GAC:	D	(Asp)	TAC:	Y	(Tyr)	CAC:	H	(His)
AGA:	R	(Arg)	GGA:	G	(Gly)	TGA:	STOP		CGA:	R	(Arg)
AGG:	R	(Arg)	GGG:	G	(Gly)	TGG:	W	(Trp)	CGG:	R	(Arg)
AGT:	S	(Ser)	GGT:	G	(Gly)	TGT:	C	(Cys)	CGT:	R	(Arg)
AGC:	S	(Ser)	GGC:	G	(Gly)	TGC:	C	(Cys)	CGC:	R	(Arg)
ATA:	I	(Ile)	GTA:	V	(Val)	TTA:	L	(Leu)	CTA:	L	(Leu)
ATG:	M (Met)/START		GTG:	V	(Val)	TTG:	L	(Leu)	CTG:	L	(Leu)
ATT:	I	(Ile)	GTT:	V	(Val)	TTT:	F	(Phe)	CTT:	L	(Leu)
ATC:	I	(Ile)	GTC:	V	(Val)	TTC:	F	(Phe)	CTC:	L	(Leu)
ACA:	T	(Thr)	GCA:	A	(Ala)	TCA:	S	(Ser)	CCA:	P	(Pro)
ACG:	T	(Thr)	GCG:	A	(Ala)	TCG:	S	(Ser)	CCG:	P	(Pro)
ACT:	T	(Thr)	GCT:	A	(Ala)	TCT:	S	(Ser)	CCT:	P	(Pro)
ACC:	T	(Thr)	GCC:	A	(Ala)	TCC:	S	(Ser)	CCC:	P	(Pro)

Figure 2.6: Genetic code [2].

Proteins are composed of polypeptide chain molecules as amino acids are linked by covalent peptide bonds. Note that covalent chemical bonds cause the release of a water molecule and the resulting process is called dehydration synthesis of condensation reaction. A general amino acid with a side-chain R is shown in Figure 2.7 and the formation of the peptide bond is shown in Figure 2.8. This repeating sequence along the core of the polypeptide chain is also known as the polypeptide backbone. The sequence of chemically different side-chains of each of the amino acids is what makes one protein different from another.

The folding of a protein chain is due to the difference in the side-chains for each of the amino acids, and three types of weak interactions between molecules: electrostatic attractions, *Van der Waals* forces and hydrogen bonding. These weak interactions act in parallel to hold the two regions of a polypeptide chain tightly together. Each protein type has a specific 3-D structure which is determined by the order of the amino acids in its chain. It also depends on the hydrophobic interactions based on the type of the side-chain [52].

Proteins have a variety of shapes and can be composed of 50 to 2000 amino

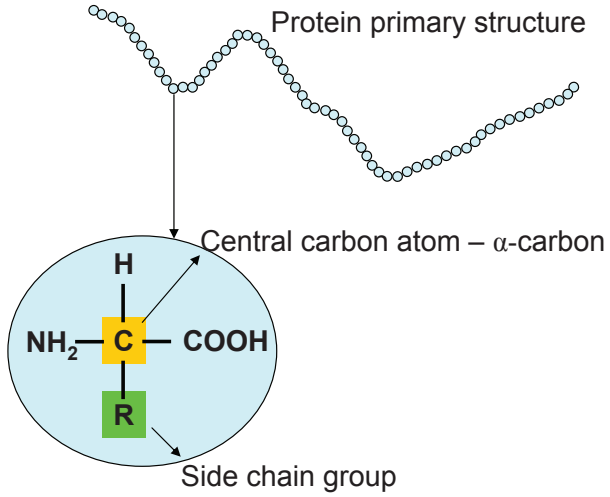


Figure 2.7: A general amino acid with a side-chain R. [2]

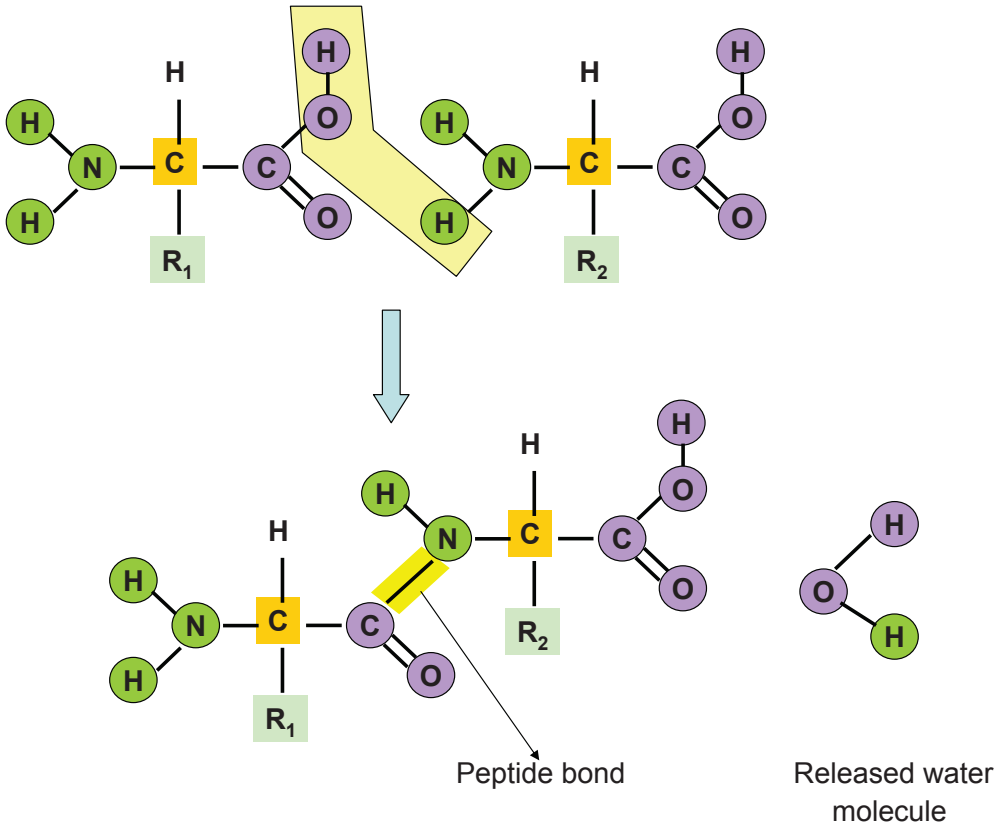


Figure 2.8: Formation of a peptide bond from two amino acids. Note the covalent bond formed and and the released water molecule. [2]

acids. Large proteins consist of a structural unit or protein domain that folds independently of one another. This complex protein structure can be depicted in many different ways, each of which emphasizes different features of the protein. The two most common folding patterns are the α helices and the β sheets, and they are obtained as a result of the hydrogen bonding between the nitrogen-hydrogen (N-H) and the carbon-oxygen (C=O) groups in the polypeptide backbone. They are particularly useful in the representation of the protein structure as a ribbon model.

2.2.2 Protein Structures

There are types of protein structure: primary, secondary, tertiary and quaternary structures. The primary protein structure is based on the 1-D amino acid composition of the protein sequence. It is the simplest structural type with the residues linked together via peptide bonds. The secondary structure is composed of polypeptide chain segments that form α helices and β sheets, which are highly regular and structured. The amino acid residues are stabilized by hydrogen bonds between the main-chain atoms of the C=O group and the N-H group of different residues. The secondary structure is also defined as the local conformation of a peptide chain. The tertiary structure consists of the complete 3-D structure of all the amino acids in a single polypeptide chain. It is defined based on the protein's atomic co-ordinates, which are determined using nuclear magnetic resonance (NMR) or X-ray crystallography techniques. The quaternary structure refers to the association of the several polypeptide chains into a protein complex and these are maintained by non-covalent interactions. The individual polypeptide chains are called monomers or subunits. A level of super-secondary structures exists between the secondary and the tertiary structures and is defined as two or three secondary structural elements forming a unique functional domain. It is observed

as a recurring structural pattern, which is usually conserved in evolution.

2.3 Reference Databases

Nucleotide datasets are publicly accessible, and they are maintained in public databases called genome databases. One important databases is *GenBank* (Genetic Databank) that is maintained by the National Center of Biotechnology Information (NCBI) at the National Library of Medicine of the National Institutes of Health [53]. Two other public databases are *EMBL* (European Molecular Biology Laboratory) [54] and *DDBJ* (DNA Data Bank of Japan) [55]; these databases share the information with each other as well as with the *GenBank*. *Entrez* [56] is a search and retrieval system of the *GenBank* database, wherein queries for sequences can be placed. Secondary nucleotide sequence databases include UniGene, STACK, Ribosomal database project, HIV Sequence database, Eukaryotic promoter database, and REBASE [57].

The protein datasets are also publicly accessible from many public databases. An important protein database is the Protein DataBank (PDB), which is maintained by the Research Collaboratory for Structural Bioinformatics (RCSB). The PDB [46] contains information about experimentally-determined structures of proteins, nucleic acids, and complex assemblies. In addition to this data, the PDB also contains tools for the visualization of the protein structures, and searches based on the sequence annotation, structure and function. Other databases include, the International Protein Sequence Database Collaboration [58], SWISS-PROT + TrEMBL (UniProt) [59], Protein Information Resource Protein Sequence Database (PIR-PSD) [60]. The secondary and specialized protein sequence databases include Gene Ontology Annotation (GOA), MEROPS, GRCRDb, yeast protein database (YPD), ENZYME, CATH (class, architecture, topology and homologous superfamily), PROSITE, PRINTs, InterPro etc. [57]. As with DNA

sequences, *Entrez* is primarily used as a protein sequence information search and retrieval system.

As an example of how information is provided in reference databases, we consider the file format in the PDB database [46]. The available information consists fo 3-D atomic co-ordinates corresponding to amino acids for different types of structures. All the proteins that have been sequenced have a PDB file associated with them. This is the most common type of file-format associated with amino acid composition in proteins, and it is similar to the FASTA format for the DNA nucleobase sequence composition.

The PDB file format specifications are provided in [46]. We describe in brief the portions of the file that we will use in our study. The file consists of segments that provide information about the primary structure, secondary structure, the atomic co-ordinates, and connectivity. The primary structure section contains the sequence of residues in each chain of the macromolecules. The secondary structure section describes helices, sheets, and turns found in protein and polypeptide structures. The coordinate Section contains the collection of atomic coordinates which describe the tertiary structure of the sequence. The connectivity section provides information on the number of interactions.

A screenshot of a PDB file with tertiary structure information is shown in Figure 2.9. In particular, the 3-D co-ordinates of the locations of the atoms in an amino acid are provided; these locations will be used to superpose tertiary structures.

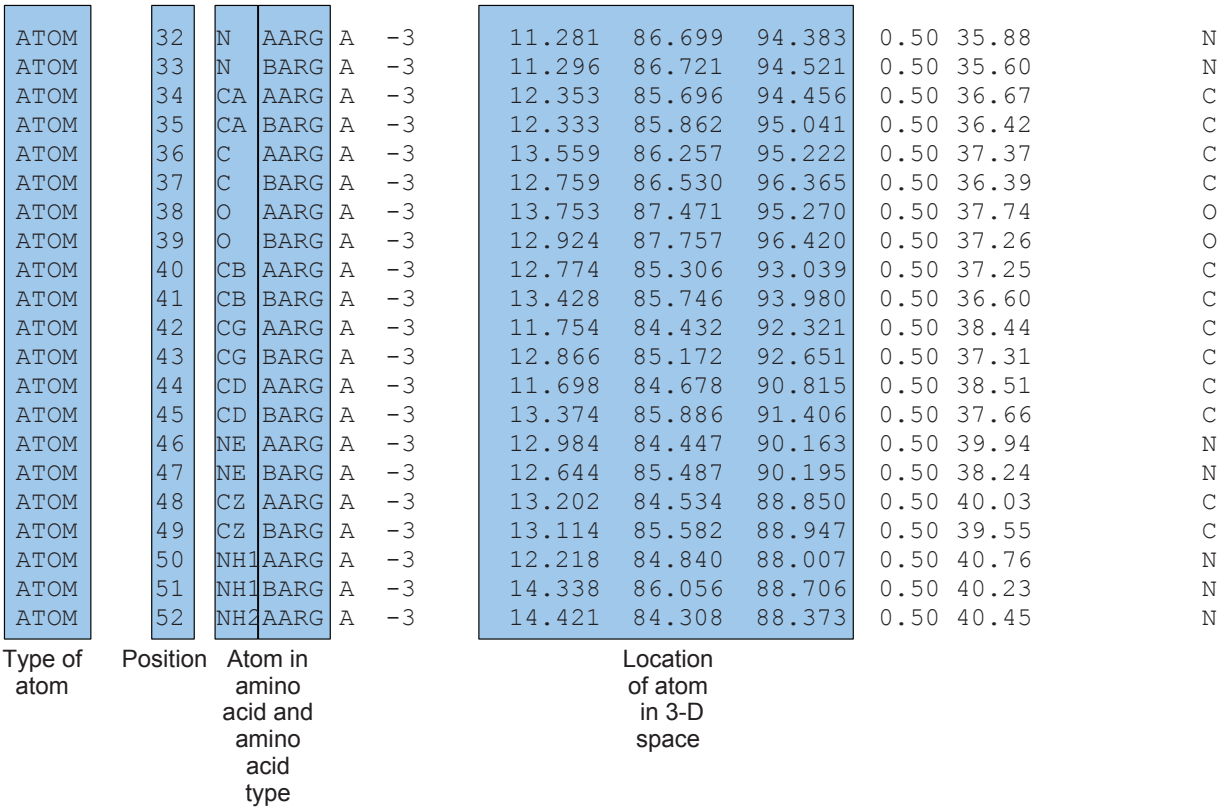


Figure 2.9: PDB file showing the co-ordinates corresponding to the tertiary structure position of an amino acid.

Chapter 3

MAPPING SCHEMES FOR DNA AND PROTEIN SEQUENCES

DNA and protein data are in the form of discrete sequences. This is because DNA data are composed of four nucleobases A, T, C and G whereas protein data are composed of twenty possible amino acids. As discussed in Chapter 1, by mapping nucleobases or amino acids to numerical sequences or discrete time-domain signals, the existing signal processing techniques can be used to successfully analyze biological data without much modification.

3.1 Traditional Numerical Mapping Schemes

In literature, various nucleotide bases signal mapping schemes have been presented that can be broadly classified into three categories, as we describe next.

3.1.1 Indicator sequence mapping

Numerical domain mapping for DNA sequences has been performed using indicator vectors [61]. Specifically, the kth nucleobase, $k = \{A, T, C, G\}$ is represented by the 4×1 indicator vector $\mathbf{u_k}$, that is defined such that the value of the kth vector position is set to the number one and all other vector values are set to zero. In other words, the presence of a nucleobase is represented by the number one and its absence by the number zero. The resulting four nucleobase vectors are given by [61]. As a result the nucleotide bases are represented as:

$$\mathbf{u_A} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{u_C} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{u_G} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \mathbf{u_T} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.1}$$

As indicator vectors cannot be used directly to perform any numerical operations, weights w_a, w_c, w_g and w_t are assigned to the nucleobases. For example, the

sequence $\{AGCAT\}$ can be represented by the indicator vector

$$\mathbf{S} = \begin{bmatrix} w_a & w_g & w_c & w_a & w_t \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

Binary indicator discrete time signals have been used to obtain Fourier transform (FT) information for DNA sequences [5,16,27,38]. The kth indicator sequence ($k = \{A, C, G, T\}$) is the kth discrete-time sequence $b_k[n]$, $n = 1, 2, \ldots, N$, whose value at time n is set to one of the corresponding DNA sequence has the element k at discrete time n; otherwise, it takes the value zero. Specifically, $b_k[n] = 1$, if the nth element of the DNA sequence is k and $b_k[n] = 0$, if the nth element of the DNA sequence is not k. Here, N is the length of the DNA sequence. As an example, we consider the $N = 17$ DNA sequence $\{ATTCAGGCTAGTCTAAC\}$. For this sequence, the four binary indicator discrete time signals are given by:

$$\begin{aligned}
b_A[n] &= \begin{cases} 1, & n = 1, 5, 10, 15, 16, \\ 0, & n = 2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 17, \end{cases} \\
b_T[n] &= \begin{cases} 1, & n = 2, 3, 9, 12, 14, \\ 0, & n = 1, 4, 5, 6, 7, 8, 10, 11, 13, 15, 16, 17, \end{cases} \\
b_C[n] &= \begin{cases} 1, & n = 4, 8, 13, 17, \\ 0, & n = 1, 2, 3, 5, 6, 7, 9, 10, 11, 12, 14, 15, 16 \end{cases} \\
b_G[n] &= \begin{cases} 1, & n = 6, 7, 11, \\ 0, & n = 1, 2, 3, 4, 5, 8, 9, 10, 12, 13, 14, 15, 16, 17 \end{cases}
\end{aligned} \tag{3.3}$$

For example, $b_A[n]$ has one at positions $n = 1,\ 5,\ 10,\ 15,\ 16$ since the DNA sequence has A in the corresponding positions.

The discrete Fourier transform (FT) of each of the four binary indicator sequences corresponding to each of the nucleobases is computed and summed up to obtain the overall FT of the whole sequence [27]. Also, integer weights are used to study spectral characteristics for gene evolution [15], whereas complex weights were used to finding DNA complements [5].

The FT magnitude of an indicator discrete time-domain sequence for the DNA coding region of *Saccharomyces Cerevisae* or *S. Cerevisiae* (a species of budding yeast) is shown in Figure 3.1. Its corresponding power spectrum is shown in Figure 3.2.

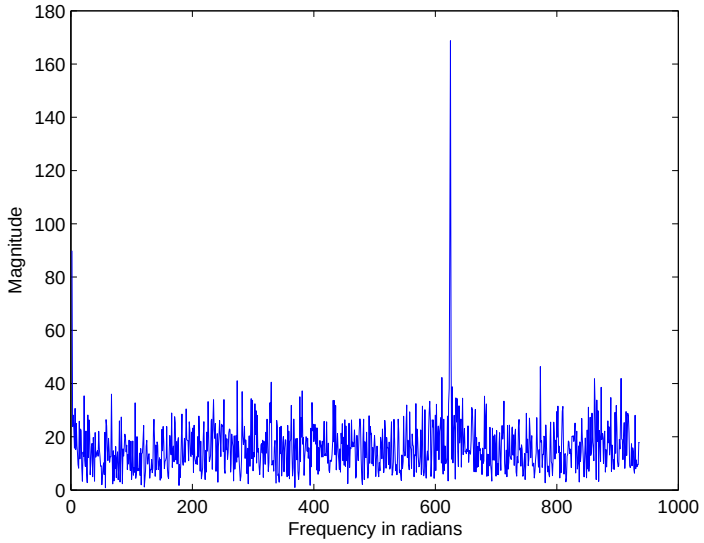


Figure 3.1: Magnitude of the FT for the coding region of DNA from *S.Cerevisae*. Note that the peak occurs at $f = N/3$ where $N = 1871$ in this example.

The spectrogram of the DNA sequence of *S. Cerevisiae* for a window size of $N = 60$ is shown in Figure 3.3. The example taken demonstrates periodicity in the spectrogram. The horizontal axis indicates the location in the DNA sequence measured in base pairs from the origin, and the vertical axis indicates the discrete frequency of the DFT measured in cycles per window size. Although traditional spectrogams use pseudo-color to achieve greater contrast, the spectrograms in this

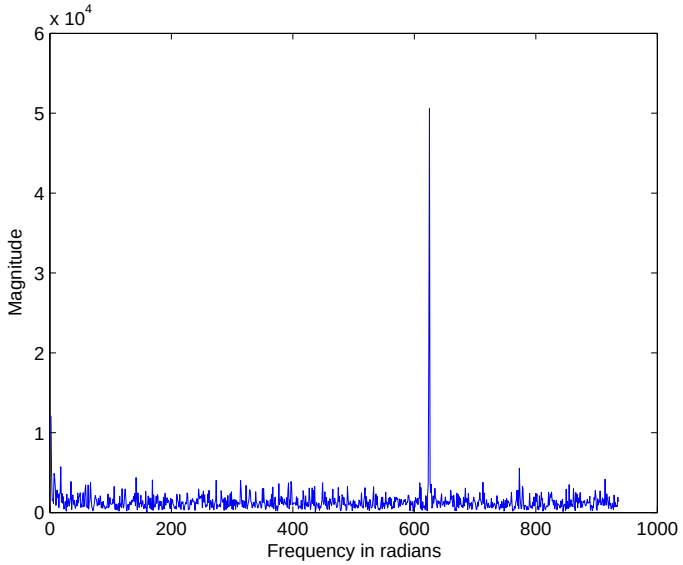


Figure 3.2: Power spectrum related to the FT in Figure 3.1 for the coding stretch of DNA from *S.Cerevisae*.

case contain useful information encoded in color.

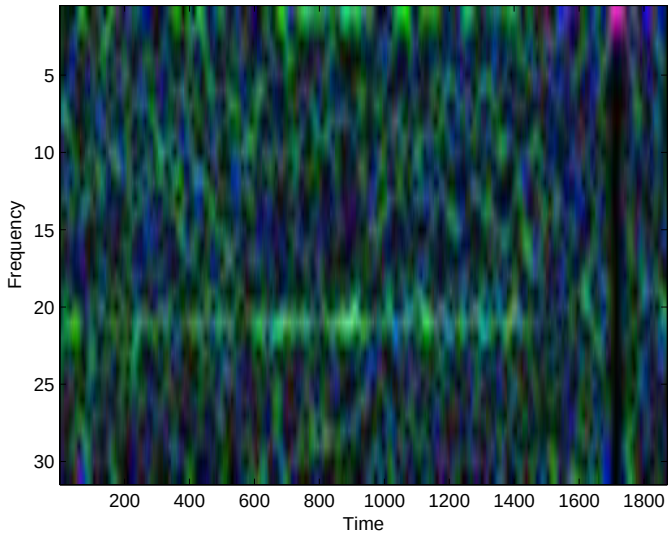


Figure 3.3: Spectrogram for the coding stretch of DNA from *S.Cerevisae*.

Each DNA sequence is represented by four indicator discrete-time signals, as in (3.3). In order to reduce the computational cost when using this representa-

tion for DNA sequencing, the four signals can be reduced to three by using a technique that is symmetric to all the sequences [23]. This technique assigns a vertex of a regular tetrahedron in 3-D space to each of the four DNA nucleobases. Specifically, the three numerical sequences are first defined as $x_1 = \{A_1, T_1, C_1, G_1\}$, $x_2 = \{A_2, T_2, C_2, G_2\}$, $x_3 = \{A_3, T_3, C_3, G_3\}$, by considering four 3-D vectors with unit magnitude, pointing to the four directions from the center to the vertices of the tetrahedron. In the example cited in [5, 23], the values chosen are:

$$\begin{aligned}
(A_1, A_2, A_3) &= (0, 0, 1), \\
(T_1, T_2, T_3) &= \left(\frac{2\sqrt{2}}{3}, 0, -\frac{1}{3}\right), \\
(C_1, C_2, C_3) &= \left(-\frac{\sqrt{2}}{3}, \frac{\sqrt{6}}{3}, -\frac{1}{3}\right), \\
(G_1, G_2, G_3) &= \left(-\frac{\sqrt{2}}{3}, -\frac{\sqrt{6}}{3}, -\frac{1}{3}\right).
\end{aligned}$$

Using these values, we obtain the three new indicator discrete time signals as

$$\begin{aligned}
x_1[n] &= \frac{\sqrt{2}}{3}(2b_T[n] - b_C[n] - b_G[n]), \\
x_2[n] &= \frac{\sqrt{6}}{3}(b_C[n] - b_G[n]), \\
x_3[n] &= \frac{1}{3}(3b_A[n] - b_T[n] - b_C[n] - b_G[n])
\end{aligned}$$

where $b_A[n]$, $b_A[n]$, $b_A[n]$, and $b_A[n]$ are defined as earlier. This mapping is particularly useful in computing the spectrogram of the DNA data [5], since the three colors in the spectrogram (red, green and blue) can be attributed to the three indicator discrete-time signals.

3.1.2 Real number mapping

The real number mapping technique is an efficient technique for finding complements in DNA sequences [19]. Using the real number mapping, complementary nucleobases are mapped using the same magnitude but opposite signs.

$$A \to -1.5, T \to -0.5, C \to 0.5, \text{and } G \to 1.5. \tag{3.4}$$

In Equation (3.4), the notation $A \to -1.5$ reads: the nucleobase A is mapped to real number -1.5. The mapping is demonstrated in Figure 3.4. Although this mapping is also suited for computing correlation values, it should not be used to draw conclusions on the correlation structure of DNA sequences, as the correlations are biased. This approach has been used for autoregressive (AR) modeling and feature distribution analysis in [19]. Note that since only four real numbers are used for DNA data, this approach can be considered as a special case of a four signal pulse amplitude modulation (4-PAM) scheme.

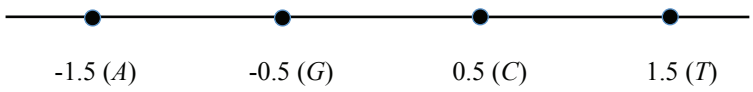


Figure 3.4: Real number mapping.

Another method that is discussed in [61] assigns an increasing sequence of positive integers to the alphabetically sorted nucleobases after obtaining the indicator sequences. The assignment is performed as $A \to 1$, $C \to 2$, $G \to 3$, and $T \to 4$.

3.1.3 Complex number mapping

The complex number mapping approach as presented in [5]. The complex numbers n_A, n_T, n_C, and n_G are assigned to the characters A, T, C, and G respectively. The complex conjugate pairs for the mapping are chosen as $n_T = n_A^*$ and $n_G = n_C^*$ are chosen. The complementary DNA strand is represented by

$$\tilde{x}[n] = x^*[-n + N - 1], \text{ for } n = 0, 1, \ldots, N - 1,$$

where N is the length of the DNA sequence. A specific complex number mapping is given by (3.5).

$$\begin{aligned} A &\rightarrow n_A = 1 + j \\ T &\rightarrow n_T = 1 - j \\ C &\rightarrow n_C = -1 - j \\ G &\rightarrow n_G = -1 + j \end{aligned} \tag{3.5}$$

The mapping is demonstrated in Figure 3.5 and can be considered as a special case of quadrature phase shift keying (QPSK) [62].

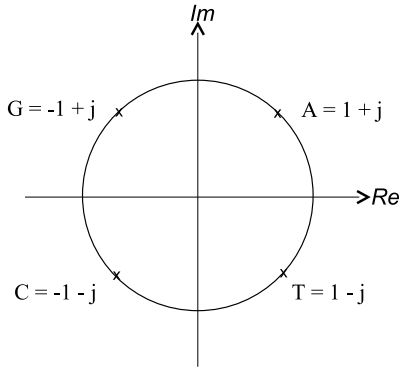


Figure 3.5: Complex number mapping

Another example of complex number mapping that has been used in the literature [63,64] is the assignment of roots of unity to the sequences, i.e., $n_A = 1$, $n_T = j$, $n_C = -1$, $n_G = j$. This type of mapping has also been extended to the case of protein sequences in [65] with the twenty roots of unity mapped to the twenty amino acids.

3.2 Waveform Mapping Schemes

So far, although DNA and protein sequences were mapped to time-domain signals, the signals considered were only discrete-time sequences over small finite

sets. For example, only four real numbers were for DNA sequences when real number mapping was used. This type of mapping inherently puts a limit on the type of signal processing algorithms that can be used to process the biological data. We propose to use continuous time-domain waveform as our mapping mechanism instead. In addition to using a waveform to represent, for example, a DNA nucleotide base, we will also embed useful biological properties onto the waveform parameters in order to increase the amount of distinct data features as well as have more available signal process methodologies to use for processing.

We consider mapping DNA nucleotide base sequences to time-domain waveforms. We choose the type of waveform used in the mapping based on the waveform properties and on the signal processing method adopted for the sequence alignment algorithm. For a correlation-based matched filtering sequence alignment approach, the waveform used for the mapping must be orthogonal in order to achieve maximum correlation values [66]. For an alignment approach based on an orthogonal signal basis expansion, the mapping waveform will again need to be orthogonal. However, if the alignment approach uses a signal expansion, then the mapping waveforms do not have to be orthogonal; they only need to be functions with time-varying, highly-localized spectra in the time-frequency plane.

We first consider two types of waveforms, sinusoids and linear frequency-modulated (LFM) chirps, that can be made orthogonal by their choice of parameters. We then consider Gaussian waveforms that are highly localized in the time-frequency plane.

3.2.1 Sinusoid Waveform Mapping

When we use sinusoids to map DNA nucleobases, we consider $L = 4$ orthogonal sinusoids given by

$$s_l(t) = e^{j2\pi(\frac{l}{T_d})t}, \quad l = 1, \ldots, L, \quad 0 \leq t < T_d\,, \tag{3.6}$$

where T_d is the duration of the waveform. The frequency of the lth sinusoid is $f_l = l/T_d$, corresponding to the frequency of the lth multiple of the harmonic frequency $1/T_d$. The L harmonics ensure that the sinusoids are orthogonal so that the inner product between any two sinusoids is given by

$$\begin{aligned} \langle s_k, s_l \rangle &\triangleq \int_0^{T_d} s_k(t)\, s_l^*(t)\, dt \\ &= \int_0^{T_d} e^{j2\pi(\frac{k}{T_d})t}\, e^{-j2\pi(\frac{l}{T_d})t}\, dt \\ &= \begin{cases} T_d, & \text{for } \; k = l \\ 0, & \text{for } \; k \neq l \end{cases} . \end{aligned} \tag{3.7}$$

Here, the inner product is effectively the FT of windowed sinusoids, thus the computation is fast and efficient. Note that the mapping of the nucleobases to sinusoids is similar to the orthogonal frequency division multiplexing (OFDM) scheme [67]. Also, the sinusoid mapping scheme can be shown to be a more general case of the complex number mapping discussed in Section 3.1. This follows from the fact that the roots of unity in the complex mapping scheme correspond to specific fixed values of orthogonal sinusoids. As a result, the complex mapping scheme uses four numbers whereas the sinusoid mapping uses four waveforms to represent the four DNA nucleobases.

For implementation purposes, the continuous-time waveform is discretized using a sampling frequency f_s. For example, for the sinusoid waveform in (3.6), the discrete-time waveform, $s_l[n]$=$s_l(n/f_s)$, is used instead of the continuous-time signal $s_l(t)$. An example of four normalized sinusoids, corresponding to the

four DNA nucleobases, are shown in Figure 3.6. For this example, the waveform duration was chosen as $T_d = 0.1$ second in Equation (3.6) and the sampling frequency was $f_s = 1000$ Hz.

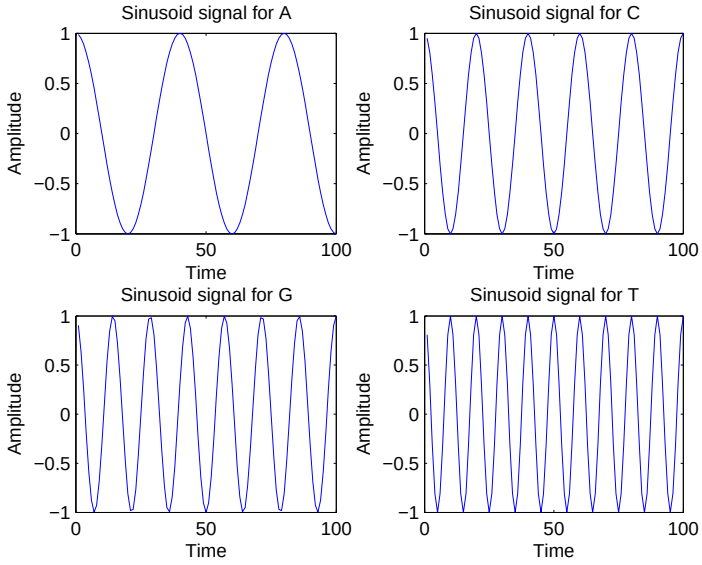


Figure 3.6: Sinusoid signals representing the four nucleobases.The duration of the signal is 0.1 seconds, and the sampling frequency is 1000 Hz.

3.2.2 LFM Chirp Waveform Mapping

The LFM chirp is time-varying since its spectrum varies linearly with time. It is defined as [68]

$$h_l(t) = \sqrt{2\,t}\; e^{j2\pi c_l\, t^2}, \quad 0 < t < T_d \tag{3.8}$$

where c_l in $(\text{Hz})^2$ is the frequency-modulation (FM) rate and T_d is the waveform duration in seconds. The instantaneous frequency (IF) of the LFM chirp, given by $2\,c_l\,t$, represents the linear frequency variation of the waveform with respect to time. Ideally, the time-frequency representation of this waveform is a line, going through the origin of the time-frequency plane, with slope $2\,c_l$. Note that the amplitude modulation $\sqrt{2\,t}$ in (3.8) ensures that the LFM chirp is an orthogonal signal. This can be shown by taking the inner product between two LFM

chirp signals with different FM rates and infinite duration. With finite duration, and using $L = 4$ LFM chirps to map the four DNA nucleobases, we can show orthogonality by computing the inner product

$$\begin{aligned}\langle h_k, h_l \rangle &\triangleq \int_0^{T_d} h_k(t)\, h_l^*(t)\, dt \\ &= \int_0^{T_d} 2\,t\; e^{j2\pi c_k\, t^2}\; e^{-j2\pi c_l\, t^2}\, dt \\ &= \frac{1}{T_d^2} \int_0^{T_d^2} e^{j2\pi c_k\, \tau}\; e^{-j2\pi c_l\, \tau}\, d\tau\end{aligned} \tag{3.9}$$

If we compare equations (3.9) and (3.7) and let the difference between the FM rates be given by $\Delta c = c_k - c_l = K/T_d^2$, for some integer number K, then

$$\langle h_k, h_l \rangle == \begin{cases} 1, & \text{for } \; k = l \\ 0, & \text{for } \; k \neq l \end{cases} . \tag{3.10}$$

As the minimum possible value for K is 1, the minimum FM rate difference is given by $\Delta c_{\min}=1/T_d^2$, and the FM rate can be chosen as $c_l = l/T_d^2$, $l = 1, \ldots, 4$.

As a result, the LFM chirp that we can use for the mapping is given by $h_l(t) = \sqrt{2\,t}\; e^{\frac{j2\pi\, t^2}{T_d^2}}$, $\quad 0 < t < T_d$.

Using these FM rates, an example of the corresponding IFs of four LFM chirps that can be used to represent the four nucleobases is demonstrated in Figure 3.7. The chirp signals corresponding to the four nucleobases are shown in Figure 3.8. Note that the FM rate can be made negative to represent complementary strands or complementary nucleotides; this is also possible in the sinusoid mapping by using the negative of the chosen frequencies. The LFM scheme, however, is preferred over the sinusoid scheme when bandwidth requirements are limited since it is possible to place many orthogonal LFM chirps in a given bandwidth.

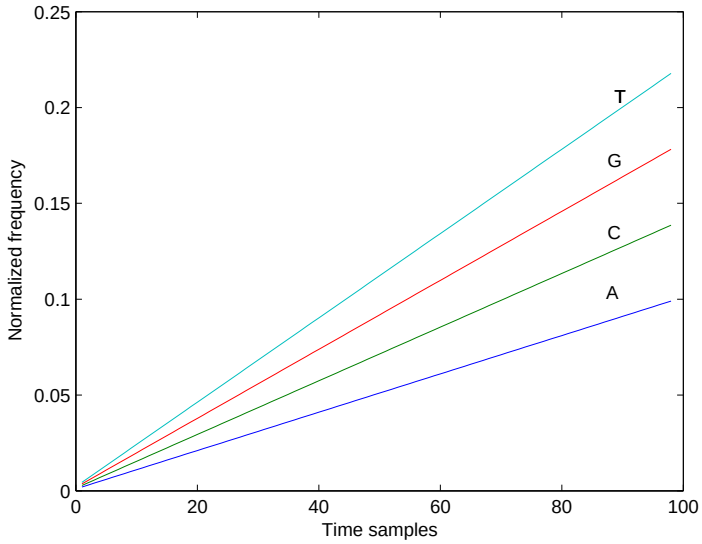


Figure 3.7: Instantaneous frequency of LFM chirp waveforms, representing the four nucleobases. The duration of the signal is 0.1 seconds, and the sampling frequency is 1000 Hz. The frequency axis is shown normalized by the sampling frequency

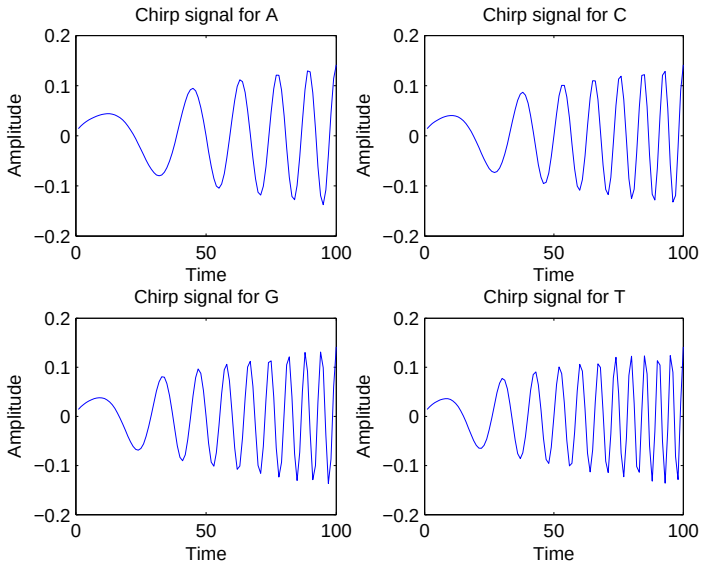


Figure 3.8: LFM chirp waveforms representing the four nucleobases. When discretizing the chirps, the highest FM rate was chosen to satisfy $c_4 \leq\leq \frac{f_s}{4T_d}$ in order to avoid aliasing. Here f_s is the aliasing frequency and T_d is the duration of the signal. For this example T_d is 0.1 seconds, and f_s is 1000 Hz. The instantaneous frequencies of these waveforms are provided in Figure 3.7.

3.2.3 Gaussian Waveform Mapping Scheme

Gaussian waveforms can be shown to be the most localized waveforms in both time and frequency as they satisfy the uncertainty principle [69]. This high time-frequency localization property of the Gaussian waveforms makes them good candidates for representing DNA sequences in the time-frequency plane. Specifically, we map the kth nucleotide base using the frequency shift kF, k=1, . . . , 4, of a basic Gaussian waveform $g(t)$=$e^{-\pi t^2}$. This mapping is such that the k=1 frequency shift represents character A, k=2 represents C, k= 3 represents G, and k=4 represents T. We use an additional frequency shift with k=5 to represent the gap for insertions and deletions. By also time-shifting the Gaussian waveform,

$$g_{m,k}(t) = g(t - m\tau_s)\, e^{j2\pi kFt} = e^{-\pi(t-m\tau_s)^2}\, e^{j2\pi kFt}\,, \tag{3.11}$$

we provide the time-shift parameter $m\tau_s$ that can be used to represent the position of a nucleotide base in a sequence. For example, if a DNA sequence has 16 elements, and we are considering the 9th element in the sequence, then the Gaussian waveform will be at $m\tau_s$=$9\tau_s$. In summary, by sampling the time-frequency plane, the discrete point (m, k), which is the center location of the Gaussian waveform $g_{m,k}(t)$ provides the following information: nucleotide base k is in position m of the DNA sequence. The time-frequency sampling is demonstrated in Figure 3.9, and an example is demonstrated in Figure 3.10, where the sequence $\{ATCA\}$ is represented in terms of the Gaussian waveforms $g_{1,1}(t)$, $g_{2,4}(t)$, $g_{3,2}(t)$, and $g_{4,1}(t)$. The waveform representation for the protein sequence is discussed in Section 4.7.

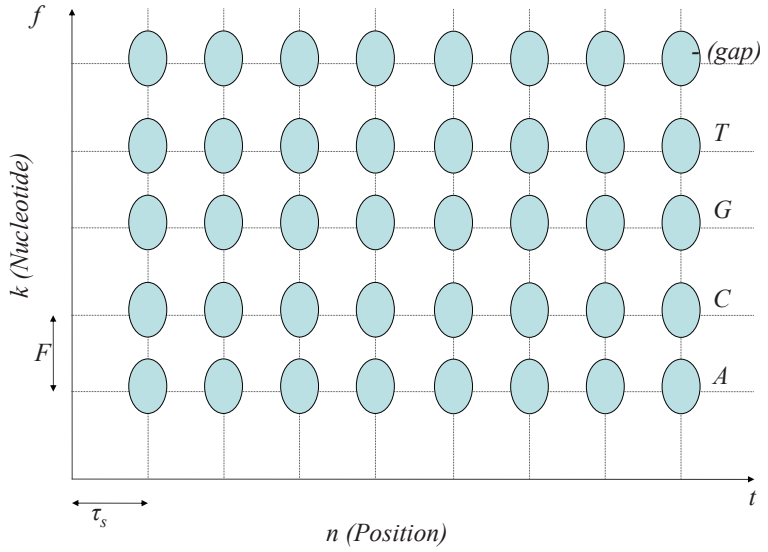


Figure 3.9: Gaussian waveforms representing DNA nucleotide bases in the time-frequency plane based on their position in a sequence.

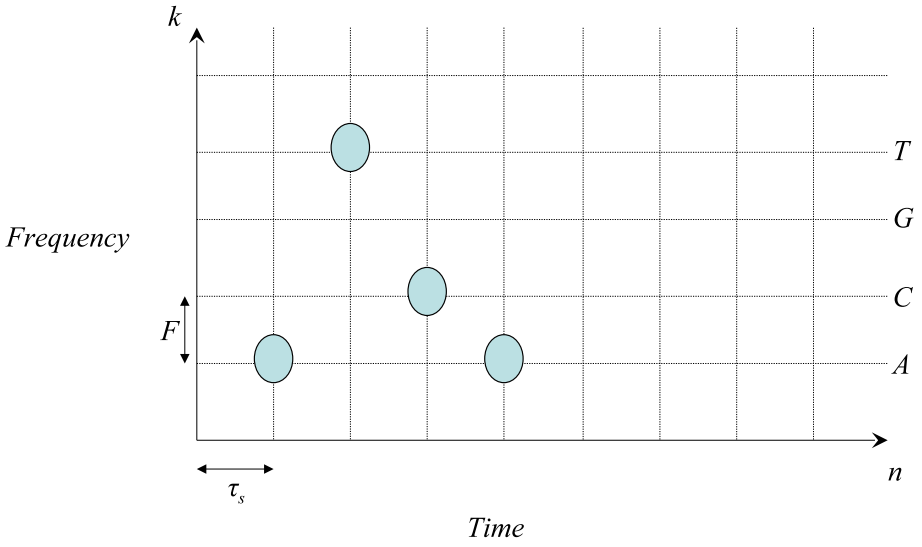


Figure 3.10: Example of four Gaussian waveforms representing the DNA sequence $\{ATCA\}$.

Chapter 4

QUERY-BASED DNA SEQUENCE ALIGNMENT

In general, sequence alignment is an arrangement of primary sequences of the DNA, RNA (ribonucleic acid) or proteins, in order to identify regions of similarities among them. This identification of similarity can be attributed to functional relationships between the sequences. By studying the sequence similarity between a new gene sequence and sequences of known structure or function, we can infer the functionality of the newly sequenced gene [70].

A sequence alignment tool must take into account the mutations due to cloning, sequencing errors, and the variations in the nucleotides, when comparing a given sequence with the sequences in the database. A variety of alignment tools have been developed using dynamic-programming techniques in bioinformatics [71–73]. A few alignment tools have also been developed using signal processing techniques [63–65, 74–78].

4.1 Types of Sequence Alignment

There are three main types of sequence alignment: global, local and multiple alignments.

Global sequence alignment refers to aligning each and every residue (character) in every sequence, i.e., alignment over the whole length. It occurs when the two sequences are roughly of the same size. Global alignment may fail to find the best local region of similarity, and will return only the best matching segment for a given pair of sequences. The Needleman-Wunsch algorithm [71] is an efficient dynamic algorithm for global sequence alignment.

Local sequence alignment refers to finding regions of similarity between a large sequence and a query sequence. This does not have to occur over the entire length of the sequence. The regions in the large sequence with a high degree of similarity are found; there can be multiple such regions for a given pair of sequences. The Smith-Waterman algorithm provides a dynamic programming algorithm for local sequence alignment [72].

Multiple sequence alignment refers to finding regions of similarity between a larger set of sequences. It is an extension of the above two pairwise alignments, and aims at the alignment of more than two sequences simultaneously. Multiple sequence alignment tools try to align all the sequences of a given query set. This is particularly helpful in identifying conserved sequence regions across of a group of sequences that are related evolutionally.

4.2 Sequence Alignment Tools

A plethora of alignment tools are available for local, global and multiple sequence alignment algorithms. There have been many computational approaches developed that are best suited to identify select alignments that are of interest to the developer. Hence, an alignment tool may be suitable for capturing a few alignments and fail to capture other alignments. A comprehensive list of available alignment tools can be found at `http://pbil.univ-lyon1.fr/alignment.html`. We discuss two of the most popular alignment tools below.

4.2.1 Computational Methods

The basic local alignment search tool (BLAST) [73] is a powerful local alignment tool which can be accessed through the Internet at `http://www.ncbi.nlm.nih.gov/BLAST/`. The input to the BLAST tool is a query sequence and a database of sequences, and the output is a pair of sequences with maximum similarity. This

has been a benchmark tool in the area of local sequence alignment. There are a variety of searches in BLAST that are broadly classified into basic searches and specialized searches. Each of these alignments are processed using various different programs such as *blastn, blastp, blastx, tblastn, tblastx* for nucleotide and protein sequence alignments [73]. The database sequences can be protein or nucleotide databases depending on the type of alignment performed, i.e., protein sequence alignment or DNA sequence alignment. Although this is a widely used tool, its major drawback is accuracy. At the cost of efficiency in terms of time, there is a compromise on the accuracy of the alignment. Especially for queries with repetitive segments, BLAST does not provide satisfactory results. In addition, the sequences in the database of BLAST are pre-processed and indexed for faster retrieval during the query process. Thus, if newer sequences need to be queried, the indexing process must be performed before the query, thus delaying the query process.

Another powerful alignment tool that was used before BLAST was FASTA (`http://www.ebi.ac.uk/fasta/`). This tool was derived from the logic of the dot plot. It was the first widely used program for database similarity searching. The program is better suited for nucleotide alignments than proteins. However, after BLAST came into use, FASTA became more popular as a format for the nucleotide and protein sequences than as an alignment tool. In terms of performance, the following is stated in [79] "FASTA is slower compared to the BLAST, however the results produced are equivalent for highly similar sequences. BLAST is faster than FASTA without significant loss of ability to find the similar database sequences. FASTA may be better for less similar sequences."

Other computational approaches include BLAT [80], OASIS [81], BWT-SW [82], and SST [83]. There have also been many q-gram based querying approaches such as QUASAR [84] and VGRAM [85]. These methods were shown

to be efficient for shorter query lengths. As in the case with BLAST, a few of these methods also perform indexing on the database before the querying takes place. As a result, there is an underline need for a tool that performs efficient query processing over larger databases in real-time.

There are a few drawbacks in using dynamic programming for sequence alignment. In particular, the data to be queried must be indexed and pre-processed prior to the query process. Also, the method is insensitive to alignments over repetitive or periodic data segments, and the method is not always capable of handling large query lengths.

4.2.2 Signal Processing-based Approaches

While there have been many dynamic computational-based approaches to solve the sequence alignment problem, a few algorithms based on signal processing have also been developed. These algorithms consider sequence alignment as a sequence-matching problem. The common premise of the algorithms is to use cross-correlation of the sequences as a measure of similarity. Often, the cross-correlation is obtained using the fast Fourier transform (FFT) that can reduce the computational complexity from $O(N^2)$ to $O(\log_2 N)$, where N is the length of the sequences to be aligned.

In [74] , an FFT approach was considered for a very general case of sequence matching. Specifically, the DNA sequence is first mapped to four binary indicator sequences and then the overall number of matches at a shift is found using convolution; this can be computed as the product of FFTs of the two sequences in the frequency domain. This method, however, is very limited in terms of computations and number of insertions and deletions. This basic algorithm was improved in [64], where complex number mapping was used instead of binary indicator sequence mapping. The peaks observed in the correlation domain de-

termined that the sequences compared were similar. Note that the binary and complex indicator sequence mapping do not clearly distinguish between global and local alignment.

After nearly more than a decade, an efficient algorithm using the FFT was proposed in [63] to capture local similarities, i.e., to perform local alignment. This algorithm forms sub-sequences from the original query and tries to find the best match for the sub-sequence; the best match is then extended until the threshold is reached. This technique has provided insight into the benefits of the FFT approach for the sequence alignment problem. A performance metric for this method, called position specific match score, was presented in [75]. This scheme used a variant of the complex coding scheme, which used two indicator sequences instead of one. Another version of the FFT-based correlation method was described in [76], which used complex mapping for the sequence. The method obtained the similarity scores by plotting the time shift with respect to cross-correlation values. However, the algorithm was developed for a very general case of global alignment, and the position of the similarities was not clearly described. A base by base comparison had to be performed on the best similarity scores to find the exact alignment, and that was an overload on the algorithm.

In an algorithm called MAFFT [77], multiple sequence alignment was performed for protein sequences using the FFT. The correlations between two amino acid sequences were first computed, and based on the correlation values obtained, the homologous regions in the sequences were found. The optimal arrangement of the homologous segments results in an alignment. The above process was repeated with other sequences, on a group-to-group basis, to perform multiple alignment. However, this algorithm failed to provide information about the exact position of the match and only provided information about the relative shifts in position between sequences. The technique proposed by [65], called sequence-wide investi-

gation using Fourier transform (SWIFT), describes a pattern search algorithm for protein sequences. This method used an FFT-based cross-correlation approach with complex mapping for the amino acids.

A wavelet transform based cross-correlation method was used in [86] for protein sequence alignment based on spatial resolution. Wavelet transform analysis was also used to characterize long range correlations in DNA sequences [41–43] and thus to study the structure of the nucleosome and infer similarities in the sequences.

The performance of the signal processing methods is comparable to the performance of the dynamic programming based approaches, and at times better in identifying alignment. However, the regular cross-correlation approach has not been shown to provide good alignment reports with respect to the position or for sequences with repetitive patterns. Also the problem of local alignment has not been handled very well. The cross-correlation approach does not consider partial sequence mismatches that occur due to mutations or errors during data entry and it may also provide incorrect alignments when applied to periodic sequences [78]. The symmetric phase-only matched filter approach proposed in [78,87] performed better than the regular cross-correlation approach for sequences with repetitive patterns.

In protein sequence alignment, the similarity in amino acid sequence composition is dependent not only on complete amino acid matches, but also on partial matches (due to the mRNA transcription from the DNA sequence). The partial matches have not been well represented by other signal processing approaches because the mapping schemes either provide a complete match or declare a mismatch.

4.3 Sequence Alignment Scenarios

The essence of sequence alignment is to find regions of similarity between two or more sequences. If the similarity is captured over the entire length of the two sequences, it is called global alignment. If the similarity is to be captured over smaller portions of the two sequences locally, it is called local alignment. In other words, we wish to find the regions (sub-sequences) in the pair of sequences that are considered, that will provide a good alignment in terms of various performance parameters. The aligned regions may occur anywhere in the sequences. Also, there can be more than one region of alignment for a given query sub-sequence.

In the alignment problem, a short sequence is to be aligned with a long sequence. That is, we must find the position in the long sequence, where the short sequence appears. The short sequence (usually in the order of a few hundreds or thousands of characters) is called the query sequence and the long sequence (usually in the order of hundreds of thousands of characters) is a sequence in the database; we will refer to the long sequence as the database or simply data sequence. An illustration of this is shown in Figure 4.1.

Figure 4.1: Query and database sequences that need to be aligned.

Consider the data sequence $d[n]$ and the query sequence $q[n]$. These sequences are first mapped to the time-domain using one of the mapping techniques described in Section 3.2. The time-domain signals are referred to as $d(t)$ and $q(t)$,

respectively. The signals that map each character are chosen to be orthogonal. The proposed algorithm considers four different cases of alignment between the two sequences, and they are described next.

Case 1: Complete alignment Complete alignment occurs when the query sequence is similar in its entirety or up to a small number of mismatches to within a portion of the long database sequence. The aligned region can occur anywhere in the data sequence. This is illustrated in Figure 4.2, where the query sequence has a complete match in the database sequence with one nucleotide base mismatch.

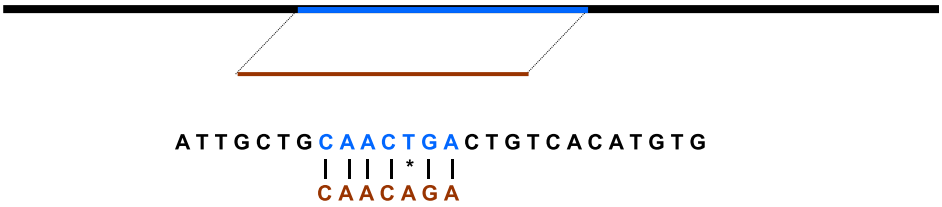


Figure 4.2: Complete alignment. The lines (|) represent a match and the asterisk (*) represents a mismatch.

Let the database sequence be composed of p sub-sequences, i.e. $d(t) = \{d_1(t), ...d_p(t)\}$. Each of these sub-sequences are different by one character, that is the subsequences have overlapping regions. Also, let each subsequence be composed of Q characters and let the time between the consecutive characters be τ_s. This is demonstrated for Q=4 in Figure 4.3; Figure 4.4 depicts $d_1(t)$ using sinusoid mapping with Q=4 and τ_s=1 s. The time distance between consecutive nucleotide bases is τ_s. The duration of $d_i(t)$ is $Q\tau_s$.

Then, the best match for $q(t)$ from the sub-sequence $d_i(t)$, where $i = 1, \ldots,$ p needs to be found. The similarity statistic normally used is the inner product

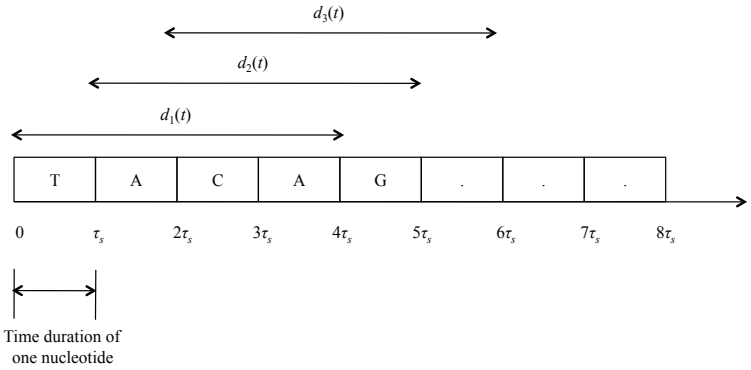


Figure 4.3: Sub-sequences $d_1(t)$ and $d_2(t)$ from the database sequence $d(t)$ for $Q = 4$ characters in each sub-sequence.

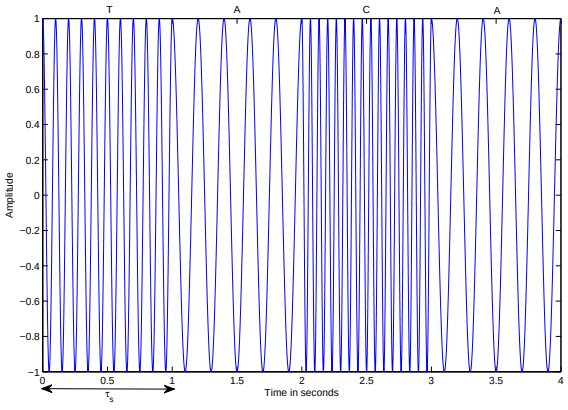


Figure 4.4: Sub-sequence $d_1(t)$ using the sinusoid mapping with $Q = 4$ and $\tau_s = 1$ second.

or cross-correlation between each $d_i(t)$ and $q(t)$, which is $\langle d_i, q \rangle$.

$$\langle d_i, q \rangle = \int_{(i-i)\tau_s)}^{(Q+i-1)\tau_s} d_i(t) q^*(t) dt, \tag{4.1}$$

where Q is the number of characters in the sub-sequence.

If this correlation is greater than a threshold γ, then we can say that the sequences are similar. Note that this correlation value corresponds to the number of matches between the two sequences. This is obtained by mapping the sequences to orthogonal chirps or sinusoids. As a result of that, when two characters are matched, the correlation is one and it is zero for a mismatch.

$$\text{If } \langle d_i, q \rangle > \gamma \Rightarrow \text{possible complete alignment.} \tag{4.2}$$

The position in the database sequence can be identified using the index i of $d_i(t)$ in $d(t)$. A plot of the correlation value versus the shift value provides with the measure of similarity and the corresponding position of the ith sub-sequence in the data sequence. The maximum correlation value is the best fit. The mismatch count in the alignment can also be obtained by subtracting the correlation value from the length of the aligned sequences.

Case 2: Un-gapped local alignment In the local alignment case, portions of the query sequence are aligned with portions of the database sequence as shown in Figure 4.5.

The query sequence in the example in the figure is $\{TGCTAACTCACA\}$. A best match is not found for the entire sequence, however portions of the sequence have matches in the database sequence at different positions. The subsequences $\{TGCT\}$, $\{AACT\}$, $\{CACA\}$ found exact matches in different positions in the database sequence.

However, if the sub-sequences are of small length, a large number of alignments will occur. Thus, there is a need for a parameter which defines the minimum length of acceptable alignment and this is defined by a threshold γ. The sequences to be aligned might not be completely similar and may have a few mismatches occurring in between alignments. For example, if two sub-sequences have 30 matches followed by a mismatch, followed by 70 consecutive matches, this is not considered as two cases of alignment. This is just one case of alignment with a mismatch occurring. The algorithm should allow for a small number of mismatches, if the best local alignments are to be obtained.

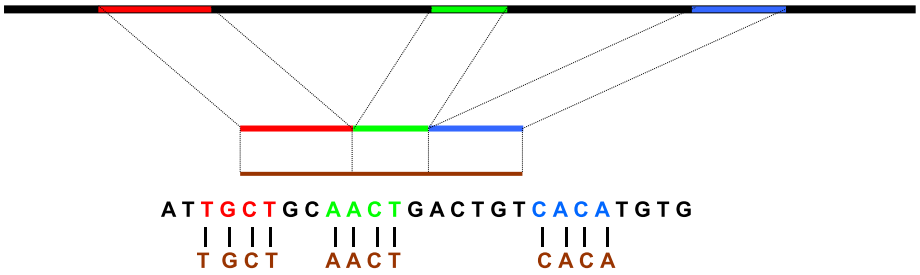


Figure 4.5: Un-gapped local alignment.

Consider sub-sequences for both the data sequence $d(t)$ and the query sequences $q(t)$:

$$d(t) = \{d_1(t), \ldots, d_p(t)\}$$

$$q(t) = \{q_1(t), \ldots, q_r(t)\}, r \leq p$$

The test $\langle d_i, q_j \rangle > \gamma$, $i = 1, ..., p$ and $j = 1, ..., r$, is considered. If the test holds, then $d_i(t)$ and $q_j(t)$ are considered as possible local alignment pairs. All combinations of $d_i(t)$ and $q_j(t)$ that satisfy the threshold are considered as cases of local alignment. The permissible mismatch count is absorbed within the value of the threshold γ itself.

The possible cases of alignment are categorized based on the similarity

measure, and more importantly, the length of each alignment. For example, an alignment of length 40 with 2 mismatches is considered as important as or even more important than an alignment of length 20 with zero mismatches. The similarity measures are obtained from the correlation value, and we can determine the length of the alignment. The goodness of the alignment is well-defined by these two metrics, however on a large scale, it is important to have a single performance metric. This will be addressed in the proposed work.

This case has been implemented using the chirp mapping and the sinusoid mapping, however the performance was not satisfactory. Even though the algorithm is fairly simple, the computational intensity and the number of variables used make the direct cross-correlation method an unsuitable candidate for this alignment case.

Case 3: Gapped local alignment In certain cases, in order to obtain the best alignment, it may be necessary to insert gaps in the query sequence or in the data sequence. These are referred to as *insertions* and *deletions*. These gaps may be attributed to the fact that, during evolution, the nucleotide sequence may have lost or gained a few properties. Thus, it is important to identify the alignment, even with gaps incorporated within the aligned sequence, since this might lead to a better aligned sequence when compared to other alignments. The case of the gapped alignment is shown in Figure 4.6. In the example given in the figure, the query sub-sequence $AATG$ does not have an exact match in the database sequence. There is a possible alignment with the data sequence $AACTG$, if a gap is inserted in the query sub-sequence, as $AA-TG$. Similarly, query sub-sequence $CCCA$ does not have an exact match with the database sequence, however it aligns with CCA with the insertion of a gap in the data sequence or deletion of C in the query sequence. Most correlation-based approaches do not handle the

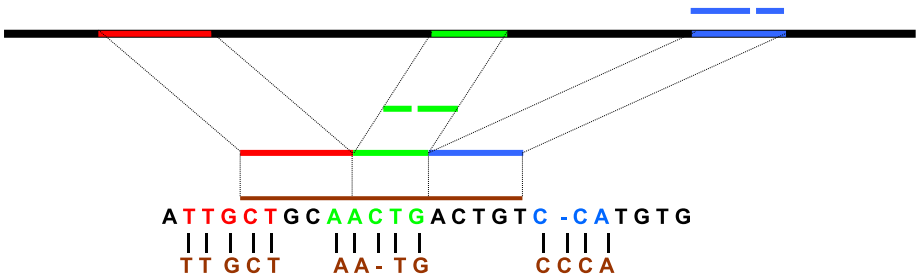


Figure 4.6: Gapped local alignment.

problem of the gapped alignment well.

Consider sub-sequences of the data sequence $d(t)$ and the query sequences $q(t)$:

$$d(t) = \{d_1(t), \ldots, d_p(t)\}$$

$$q(t) = \{q_1(t), \ldots, q_r(t)\}, r \leq p$$

The test $\langle d_i + o_{i'}, q_j + o_{j'} \rangle > \gamma$, $i = 1, ..., p$ and $j = 1, ..., r$ is considered.

If the test holds, then $d_i(t)$ and $q_j(t)$ are considered as possible local alignment pairs, where $o_{i'}(t)$ and $o_{j'}(t)$ are the signals corresponding to the gaps inserted in the data sequence and the query sequence respectively. Here, i' and j' provide the position of the gaps that are inserted in the data and the query sequence, respectively.

As described in Case 2, all combinations of $d_i(t)$ and $q_j(t)$ that satisfy the threshold are considered as cases of local alignment.

Case 4: Global alignment Global alignment refers to the alignment of two sequences over the entire length of the sequences. This is similar to the alignment presented in Case 1. However, the alignment in Case 1 compared the sequence over the entire length of the query, and failed to identify the region of local similarity. Global alignment is the case when a portion of the query sequence (query subsequence) has high similarity to the database sequence at a particular position;

the rest of the sub-sequences do not align well with the database sequence at the same position, but still have acceptable measures of similarity. Thus, the local similarities are well captured for the sub-sequences, as opposed to Case 1. This is illustrated in Figure 4.7. This case of global alignment occurs when the sequences are not of comparable length, and it can be viewed as a case of combining different local alignments.

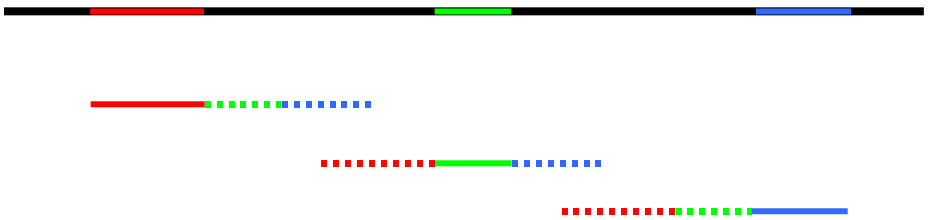


Figure 4.7: Global alignment. The solid portion of the line indicates the query sub-sequence higher measure of similarity, whereas the dotted lines represent the acceptable measures of similarity.

A more general case of global alignment has the alignment performed over the entire length of the two sequences, if they are of comparable length. It is a fairly simple and straightforward case, provided we allow insertions, deletions, and mismatches. For example, consider the two sequences:

Sequence 1 : AATCGTCGATGCATGTCACATGCGTA,

Sequence 2 : AATCTCGAGGCCATGGTCACTGCGA.

The two sequences can be globally aligned as shown below:

$$\begin{array}{l} \text{AATCGTCGA}\mathbf{T}\text{GC–ATG–TCACATGCGTA} \\ \text{AATC–TCGA}\mathbf{G}\text{GCCATGGTCAC–TGCG–A} \end{array} \tag{4.3}$$

4.4 Querying using Cross-correlation based matched filtering

We illustrate an example of global querying with sequences from *S. Cerevisiae* using a simple query with $Q = 36$ nucleotide bases. The database sequence and the query sequence were mapped to LFM chirp waveforms, and correlations between the database and query sequences were computed for every position in the database sequence. A plot of correlation values (or similarity measure) versus the position in the database sequence, where an exact match with the query was obtained, is shown in Figure 4.8.

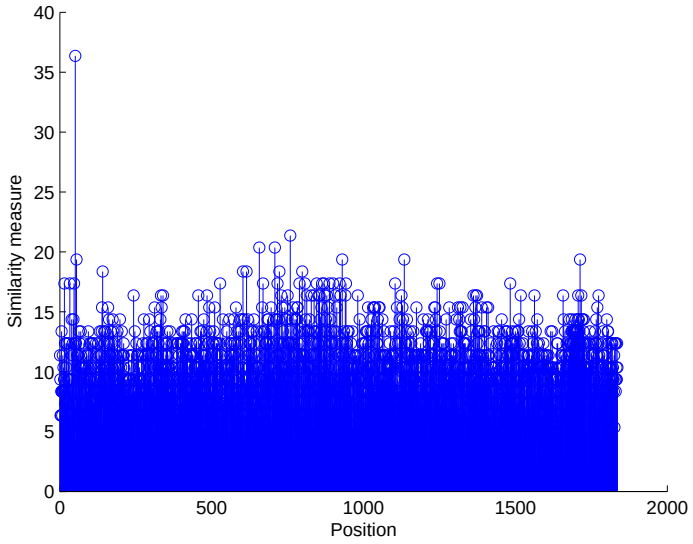


Figure 4.8: Correlation value (similarity measure) versus position for sequences obtained from *S. Cerevisae*. The maximum correlation value (of 36) occurs at position 51 in the database sequence.

The maximum correlation value occurs at position 51 in the database sequence with 36 nucleotide bases captured in the alignment. As the length of the query sequence is 36, the best match for the query signal is at position 51. Therefore, the query is found in the database sequence between positions 51 and 86. By defining the threshold γ, other possible alignments can also be obtained by comparing the correlation value with the threshold. The sinusoid mapping

scheme gave an identical performance but it was much faster as the use of the FFT reduced the number of computations. The complexity of the LFM chirp mapping scheme is in the order of $O(Q^2)$ whereas the complexity of the sinusoid mapping scheme is in the order of $O(Q \log Q)$.

The cross-correlation based matched filter approach was used for localized querying. However, the length of the alignment as well as the start and stop positions of the alignments are not known. The algorithm needs to adaptively find the local alignments by beginning with a small alignment length and extend the length of the database sequence and the query sequence until the threshold condition is not satisfied. This results in a large number of alignments, and, more importantly, in a large number of variables in order to store the position, length, start and stop points of every alignment. The result of every alignment is compiled and arranged at the end of the analysis, and the best alignment results are obtained. Even though it is fairly simple, the matched filtering query-based alignment algorithm is highly computationally intense and has a large memory requirement.

4.5 Matching Pursuit Decomposition based Querying Algorithm

The use of signal correlations with orthogonal waveform mapping for localized querying is not efficient due to the intensity of the computations and the number of variables used in storing the positions of the alignments. Thus, we propose an algorithm that performs querying and alignment based on the matching pursuit decomposition (MPD) algorithm. The new WAVEQuery algorithm provides an additional mapping parameter to control the position of an element in the sequence. The details of the MPD algorithm are outlined next, followed by its application to DNA globalized and localized querying. For the globalized querying case, we expect the algorithm to perform equally well as the chirp and sinusoid

mapping alignment using correlations. We expect a better performance for the local alignment case.

4.5.1 Matching Pursuit Decomposition Algorithm

The MPD algorithm expands a time-domain signal $x(t)$ into a linear combination of basis functions called atoms, which are selected form a dictionary $\mathcal{D}$. The atoms in the dictionary are defined as:

$$g_{n,k,l}(t) = g\left(\frac{t-\tau_n}{a_l}\right) e^{-j2\pi f_k t}, \tag{4.4}$$

where τ_n is the nth time shift, f_k is the kth frequency shift and a_l is the lth scale change on the basic Gaussian atom $g(t) = e^{-\pi t^2}$. The range of values of n, k, and l depend on how finely we sample the time-frequency plane. The advantage of using Gaussian atoms is that they are the most concentrated signals in both time and frequency [49]. Note that the MPD does not require orthogonal waveforms in its dictionary.

The decomposed signal is given by

$$x(t) = \sum_{i=0}^{M-1} \alpha_i g_i(t) + r_M(t) \tag{4.5}$$

where M is the number of iterations, α_i are the expansion coefficients,

$$\alpha_i = \int_{\tau_s} r_i(t) g_i^*(t) dt, \quad i = 0, \ldots, M-1 \tag{4.6}$$

and $r_i(t)$ denotes the residue function after the ith iteration, with the initial residue taken as the signal itself. At the ith iteration, the selected atom $g_i(t)$ is chosen as the atom that resulted in the maximum correlation between any dictionary atom and the ith residue signal,

$$g_i(t) = \arg\max_{n,k,l} \int_{\tau_s} r_i(t) g_{n,k,l}^*(t)\, dt. \tag{4.7}$$

The MPD is an iterative process that yields a sparse decomposition; if the waveform to be decomposed matches the basis functions, the MPD requires only the first few atoms to obtain a good approximation of the waveform [49]. The procedure steps are summarized as follows:

1. Initialize the residue vector: $r_0(t) = x(t)$

2. a) For iterations $i = 0, \ldots, N-1$, compute the correlation (inner product) between $r_i(t)$ and every atom $g_{n,k,l}(t)$ in the dictionary D:

$$\forall g \in D : \Lambda_{n,k,l} = |\langle r_i, g_{n,k,l} \rangle|$$

where $\langle r_i, g_{n,k,l} \rangle = \int_{\tau_s} r_i(t) g^*_{n,k,l}(t) dt$

b) Search for the atom that resulted in the highest correlation value:

$$g_i(t) = \arg \max_{g(t) \in D} \Lambda_{n,k,l}$$

c) Subtract the weighted atom from the residue:

$$r_{i+1}(t) = r_i(t) - \alpha_i g_i(t)$$

where α_i is computed as in (4.6).

3. The iterations are terminated when the desired level of accuracy is reached in terms of the extracted number of atoms or in terms of the energy ratio between the original signal and the current residue $r_i(t)$.

4.5.2 MPD WAVEQuery Alignment of DNA Sequences

Our proposed WAVEQuery method first maps DNA sequences onto Gaussian waveforms using a mapping that is matched to the MPD dictionary, and then it uses the MPD algorithm to perform querying and alignment. Specifically, we choose a basic Gaussian waveform $g(t)$=$e^{-\pi t^2}$ and only two of the three MPD

transformation parameters in (4.4). We use the time shift parameter $\tau_m{=}m\tau_s$ and the frequency shift parameter $f_k{=}kF$, where τ_s and F are the sampling periods in time and frequency, respectively. Thus, we form a dictionary of Gaussian atoms, $g_{m,k}(t)$, with $k{=}1,2,3,4$ representing the 4 nucleotide bases and $m{=}1,\ldots,Q$, representing the position of the nucleotide base in a DNA sequence of length Q [88].

It is important to note that our WAVEQuery approach makes use of the MPD algorithm in a unique and efficient way. Specifically, we pre-determine the time-frequency grid spacing of the dictionary atoms since we generate the waveforms using the mapping scheme, thus ensuring that the decomposed atoms are guaranteed to either be present or not be presented on this fixed grid. By choosing Gaussian waveforms, we ensure high localization in the time-frequency plane. We also need to run as many MPD iterations as the number of elements in the data sequences; this implies that we do not have to worry about stopping criteria for the iterative MPD algorithm. Since we perform the mapping, the query and data mapped waveforms are not noisy, and thus correlations between residues and dictionary atoms result in either very high or very low values. As a result, the resulting querying algorithms do not suffer from accumulated errors due to the iterative nature of the MPD algorithm.

4.5.2.1 WAVEQuery for Globalized Querying

For complete alignment in DNA sequences, we consider the database waveform $d(t){=}\{d_1(t),\ldots,d_{P_d}(t)\}$, $P_d \in \mathbb{N}$, and the query waveform $q(t){=}\sum_{m=1}^{Q-1} g_{m,k}(t)$. The query waveform consists of Q Gaussian waveforms, $g_{m,k}(t)$, from the MPD dictionary $\mathcal{D}$. The WAVEQuery algorithm for globalized querying is outlined in Algorithm 1.

In Algorithm 1, the outer loop is iterated P_d times, where P_d is the number

Algorithm 1 Globalized Querying - **global-align**$(d(t), q(t))$

for $p = 1$ to P_d **do**
  let $r_1(t) = d_p(t)$ {*Initialize residue*}
  $\xi_1^p = 0$ {*Initialize variable to store correlation value*}
  for $i = 1$ to Q **do**
    $\Lambda_{m,k} = |\langle r_i, g_{m,k} \rangle|$
      $= |\int_{\tau_s} r_i(t)\, g^*_{m,k}(t)\, dt|$ {*Compute correlation between residue and all dictionary elements*}
    $g^{(i)}(t) = \arg \max_{g_{m,k}(t) \in \mathcal{D}} \Lambda_{m,k}$ {*Search for atom that corresponds to the maximum correlation value*}
    $r_{i+1}(t) = r_i(t) - \alpha_i\, g^{(i)}(t)$ {*Subtract weighted atom from residue*}
    $\xi_i^p = \xi_i^p + \alpha_i$ {*Update correlation value*}
  end for
end for
$\hat{d}(t) = \arg \max_{p=1,\ldots,P_d} \xi_Q^p$ {*Sub-sequence in $d(t)$ that resulted in the best fit with the query sequence*}

of sub-sequences in the database sequence; the inner loop is iterated Q times, where Q is the length of the query sequence. Note that $\hat{d}(t)$ is the sub-sequence in $d(t)$ that resulted in the best fit for $q(t)$ [88, 89].

4.5.2.2 WAVEQuery for Localized Querying

The globalized query algorithm is modified for localized querying as follows (and as also outlined in Algorithm 2).

1. We consider the database waveform $d(t)$ consisting of sub-sequences $d_p(t)$, i=$1, \ldots, P_d$, $P_d \in \mathbb{N}$, and the query waveform $q(t)$. We consider the sub-sequence $q_j(t)$ (whose minimum length Q_j is specified by the user), j=$1, \ldots, P_q$, of the query sequence and the MPD decomposition of the sub-sequence $q_j(t)$=$\sum_{m_j=1}^{Q_j-1} g_{m_j,k_j}(t)$. The dictionary $\mathcal{D}$ is formed by all Gaussian atoms needed to map all sub-sequences of $q(t)$.

2. The dictionary length increases as the length of the query sub-sequence increases. Based on the required accuracy, the user can define the increment

Algorithm 2 Localized Querying - **local-align**$(d(t), q(t))$

$Q_j = Q_u$ {*Initialize user-defined minimum length Q_u for query sub-sequence $q_j(t)$*}
while $\xi^p_{Q_j} \geq$ threshold **do**
  $\xi^p_{Q_j}$ = **global-align**$(d_p(t), q_j(t))$ {*Perform alignment and obtain the maximum correlation value*}
  $Q_j = Q_j$+increment {*Extend dictionary elements based on user-defined increment in query length*}
end while
$\hat{d}_j(t) = \arg\max_p \xi^p_{Q_j}$ {*Best possible alignment of $q_j(t)$*}
$q_{j+1}(t)$ is the unaligned portion of the query, from Q_j to Q

value, taking into account that the increment can also increase the computational expense of the algorithm. This is continued until the best possible alignment is obtained and the minimum threshold condition is satisfied.

3. The unaligned portion of the query then becomes the new query.

The alignment steps are repeated until the end of the query sequence. Once the entire query is aligned with the database waveform, the aligned sequences are stored in the order of their similarity scores, together with information about the position of the aligned portions in the query and database sequences.

4.5.2.3 WAVEQuery for Localized Querying with Gap Insertions and Deletions

By inserting gaps in the database or query sequences, we may be able to obtain better and longer alignments. This is one feature that is not provided by most signal processing approaches. A gap is represented by a fifth frequency shift in the time-frequency plane. The algorithm adaptively inserts gaps in the query and database sequences to find better alignment results. The details of the algorithm are outlined in Algorithm 3.

From the localized querying algorithm in Section 4.5.2.2, we obtain alignments of minimum length, as specified by the user. The choice of the position

Algorithm 3 Gapped localized Querying - **glocal-align**$(d(t), q(t))$

$Q_j = Q_u$ {*Initialize user-defined minimum length Q_u for query sub-sequence $q_j(t)$*}
while $\xi^p_{Q_j} \geq$ threshold **do**
  $\xi^p_{Q_j}$ = **global-align**$(d_p(t), q_j(t))$ {*Perform alignment and obtain the maximum correlation value*}
  if mismatch-count $> \xi^p_{Q_j}$ **then**
    global-align$(d_p(t) + o_m(t), q_j(t))$ {*Perform alignment with gap $o_m(t)$ at position m in database*}
    global-align$(d_p(t), q_j(t) + o_m(t))$ {*Perform alignment with gap $o_m(t)$ at position m in query*}
  end if
  $Q_j = Q_j$+increment {*Extend dictionary elements based on user-defined increment in query length*}
end while
$\hat{d}_j(t) = \arg\max_p \xi^p_{Q_j}$ {*Best possible alignment of $q_j(t)$*}
$q_{j+1}(t)$ is the unaligned portion of the query, from Q_j to Q

of the gap depends on this minimum length of alignment. If a mismatch is encountered at a position, instead of stopping the alignment at that position, the algorithm inserts a gap in the query or the data sequence and continues with the alignment. This is done using frequency element k=5 in the dictionary. If the insertion of one gap does not provide better alignments, additional gaps, up to a user-defined limit, may be inserted. However, greater penalty is incurred while scoring. A limit on the length of the gaps is also specified by the user (usually atmost 5 gaps at a stretch), and the gaps are added if a mismatch is encountered, as long as the threshold condition is satisfied.

An example of a gapped alignment is illustrated in Figure 4.9. The similarity measure when no gaps are inserted is shown in Figure 4.9(a), where after iteration 36, the similarity measure is reduced and thus the algorithm assumes that there is a mismatch. When a gap is inserted at iteration 36, the similarity measure is high until iteration 86, as shown in Figure 4.9(b). Note that the insertion of gaps at iterations 36 and 86-88 (when the similarity measure decreases)

leads to an increased length of alignment, as shown in Figure 4.9(c). Also note that, without the insertion of these gaps, the alignment would have been shorter, and this single alignment could have been considered as three different alignments.

4.6 Simulation Results

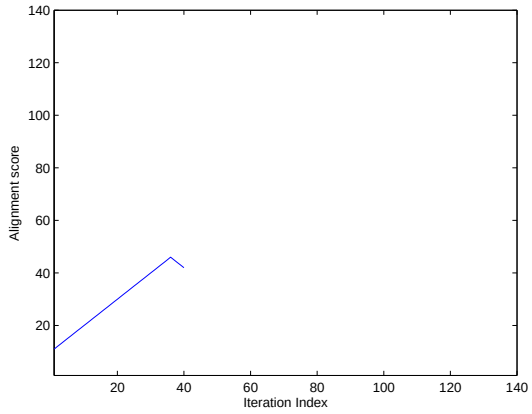
We simulated different globalized and localized querying algorithms for comparison using Matlab on a 2-core system with a 3 GHz processor, 2 GB RAM Intel Pentium D computer. We tested various alignment scenarios using sequences from the NCBI database, and in particular sequences from the *Escherichia coli* (E. Coli) genome, the chromosome 9 of homo sapiens genome, and the *Saccharomyces cerevisiae* (yeast) genome. The length of the database sequences ranged from 500–20,000 base pairs (bp) and the length of the outliers ranged from 200–362,040 bp, as summarized in Table 4.1.

Data set	Number of Sequences	Minimum Length	Maximum Length	Average Length
DB50	50	389	11,640	3,671
DB100	100	404	362,040	50,670
DB500	500	387	22,549	3,974
DB1000	1,000	387	25,674	1,847
DB5000	5,000	404	362,040	22,789

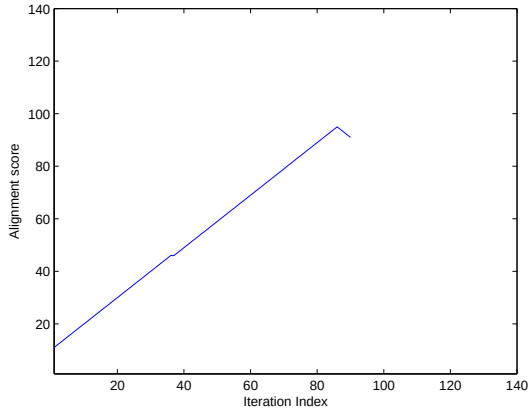
Table 4.1: Information on the data sets used for testing the proposed alignment algorithms.

The length of the query sequences varied from 200–20,000 bp. Our proposed methods supported queries on a large database sequence and performed pairwise alignment with every database sequence. Note that if we had combined the entire database into a single sequence (instance), the algorithms would have still returned the same alignments for a given query.

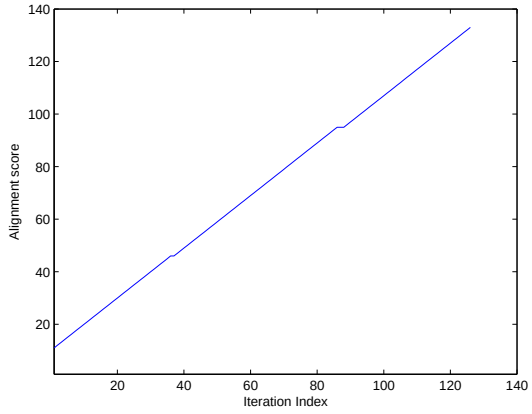
We first simulated the matched filtering algorithm (MFA) presented in Section 4.4. We implemented globalized querying using the MFA with LFM chirp and also sinusoid waveform mapping, using a 10 kHz sampling frequency. To illustrate



(a)



(b)



(c)

Figure 4.9: Gapped alignment example using the WAVEQuery approach. (a) No gaps inserted; (b) one gap inserted at iteration 36; and (c) two gaps inserted at iterations 36 and 86-88. Note that the gaps are inserted when the similarity measure reduces.

the MFA, we considered five query scenarios and run it using the database set DB100. The MFA returned completely aligned portions in the database sets in the order of their similarity measure. For comparison, we used the BLAST algorithm on a user-defined database. The performance of the MFA using the LFM chirp mapping was identical to that of the BLAST in all query cases. The algorithm validation was based on the quality of the alignments in terms of the E-value and raw scores of identical alignments from BLAST [90]. The MFA parameters, such as the threshold γ, were found to be the same as the default values in the BLAST algorithm. A comparison of the two algorithms in terms of alignment length and start points is provided in Table 4.2 for the query cases.

Query case (genome)	Matched filtering algorithm and BLAST	
	Length of alignment	Start point of alignment
MSH6	279	661
PARP1	433	1,293
MUTYH	318	1,221
MUTYH	262	429
MUTYH	558	804

Table 4.2: Sample globalized alignment using matched filtering with LFM signals and BLAST; both methods obtained identical results

We observed that the BLAST and the MFA have captured identical alignments. In the MFA, the alignment length is the correlation value, and the alignment start point is the index at which the maximum correlation value is achieved. The MFA using Gaussian waveform mapping was also simulated for the globalized querying case and identical results were obtained. Note that the time taken by the MFA was in the order of a few seconds. This included the time taken for the sequence mapping to time-domain waveforms, which was less than 20% of the total execution time. This run-time can be reduced by performing the database sequence mapping assignment and storing the resulting waveforms prior to the querying process.

We also simulated the MPD WAVEQuery algorithm for localized subsequence querying using the database set DB100 for different query sequences. The simulation results were compared to those of the BLAST algorithm *bl2seq* using the BLAST performance metrics raw score, bit score, and expect or E-value [90]. We used a query set of 100 sequences of length 1,000–10,000 bp. We observed that the localized query matches obtained using the WAVEQuery approach were identical to the query matches obtained using BLAST in 90% of the cases. We incorporated penalties for gap insertions and extensions in the computation of the raw score in the same way that the BLAST algorithm does. The quality of the alignments was identical in terms of the raw score and E-value metrics. Therefore, the performance of the WAVEQuery was identical to that of the BLAST in these query cases. For ten query cases, however, the WAVEQuery algorithm performed better than the BLAST in terms of the number of alignments captured or the length of the alignments captured in the localized query. The details of the performance improvement for these ten cases are provided in Table 4.3.

Considering query Q4 in Table 4.3, we can see that the WAVEQuery approach detected two more alignments than the BLAST approach for the query subsequence in the *Saccharomyces cerevisiae* database set (chromosome I right arm sequence). These alignments were significant ones, with E-value scores 4×10^{-13} and 2×10^{-34}. Note that we want the E-value score to be as low as possible since that indicates that the query alignment provides a good match in the database. For query Q8, the WAVEQuery approach provided a longer alignment, in addition to the two other alignments also detected by the BLAST approach in the *E. Coli* database set (Homo sapiens mutY homolog (E. coli) (MUTYH), transcript variant beta3, mRNA). Upon inspecting the database sets, we observed that these additional alignments or longer alignments of the WAVEQuery over the BLAST

Query (Length)	Number of Alignments		WAVEQuery Localized Querying Performance Improvements Over BLAST
	BLAST	WAVEQuery	
Q1 (214)	3	**5**	Two more alignments with E-values: 10^{-20}, 7×10^{-18}
Q2 (177)	3	**4**	One more alignment with E-value: 5×10^{-24}
Q3 (306)	2	**4**	Two more alignments with E-values: 3×10^{-15}, 2×10^{-31}
Q4 (333)	2	**4**	Two more alignments with E-values: 4×10^{-13}, 2×10^{-34}
Q5 (758)	2	**4**	Two more alignments with E-values: 3×10^{-13}, 2×10^{-11}
Q6 (338)	2	**4**	Two more alignments with E-values: 8×10^{-18}, 6×10^{-15}
Q7 (1,104)	3	3	One longer alignment with E-value 3×10^{-39} (4×10^{-37} in BLAST)
Q8 (740)	3	3	One longer alignment with E-value 3×10^{-69} (3×10^{-54} in BLAST)
Q9 (622)	3	3	One longer alignment with E-value 3×10^{-72} (3×10^{-65} in BLAST)
Q10 (1,136)	3	**4**	One more alignment with E-value: 8×10^{-16}

Table 4.3: Comparison of BLAST and WAVEQuery performance for localized querying on dataset DB100

were captured over the repetitive regions in the sequence in each of the queries. Thus, the WAVEQuery approach is not affected by repeats in the sequences when compared to BLAST. Mainly, this is based on our use of Fourier-based techniques (used for fast computation of the correlations in the MPD algorithm) that are well-matched to periodic segments and also because BLAST considers these repetitive regions as low complexity regions.

We designed the WAVEQuery approach to provide an alignment report with the metrics used by BLAST together with the positions of the alignments in the database and the query sequences (similar to that of the BLAST). A sample alignment report with one of the additional alignments detected by the WAVE-Query approach (but not by BLAST) is shown in Figure 4.10. It is important to note the repeats in the nucleotide composition in the sequence.

```
WAVEQuery DNA Alignment

Query Match 34 to 93 found in database sequence at position 429 to 488

Score = 116.0 bits (60), Expect = 1.779385e-31, Identities = 60/60 (100%)
Gaps = 0/60 (0%)

Query 34 to 93

GTGCAAAAGACAAAAGGGAAAAACAAAAAAAAAAATGAAGAAAAACGGACAAACGGATAAC

in Subject 429 to 488

GTGCAAAAGACAAAAGGGAAAAACAAAAAAAAAAATGAAGAAAAACGGACAAACGGATAAC
```

Figure 4.10: Alignment report for the localized sub-sequence querying cases with BLAST metrics (score and E-value). Note that we used the name notation as the one used in BLAST for ease of comparison; as a result we use $1.779385\,e^{-31}$ to represent 1.779385×10^{-31}.

The computational complexity of the MFA globalized querying scheme with LFM waveform mapping is in the order of $O(Q \log Q)$, where Q is the length of the database sequence. Using the sinusoid mapping scheme, the complexity

$O(Q \log Q)$ too. The globalized querying in the MPD WAVEQuery approach is of the order of $O(Q \log Q)$, since the correlation between the signal and the Gaussian atoms in the dictionary is computed using the fast Fourier transform. For the localized sub-sequence querying case, the complexity varies based on the number of sub-sequences used in the query process. If q sub-sequence based alignments are captured using the algorithm, the complexity is in the order $O(q\,Q \log(qQ))$. The execution time (time to perform alignments over the entire database set) for the implementation of the WAVEQuery localized sub-sequence querying algorithm on a database set of 100 sequences (DB100) is approximately 20 s. The processing can be performed in real-time, and no indexing or pre-processing are needed on the database sequence. This was also tested for the database sets DB50, DB500, DB1000 and DB5000, and the corresponding times taken to perform the querying are shown in Table 4.4. This result demonstrates that the algorithm is scalable in terms of the length of the database sets without affecting the quality of the captured alignments.

Dataset	Execution time in seconds
DB50	10.87
DB100	18.66
DB500	45.20
DB1000	115.93
DB5000	495.20

Table 4.4: Execution time of WAVEQuery localized sub-sequence querying for different database sets

As in the case of the MFA, the mapping accounts for less than 20% of the execution time in the algorithm. The mapping can be performed in real-time, and it can be improved by mapping the database sequences ahead of time and storing them for future querying.

The computational complexity of the matched-filter based globalized query-

ing scheme with LFM mapping is of the order $O(Q^2)$, where Q is the length of the data sequence. Using the sinusoid mapping scheme, this complexity is reduced to $O(Q \log Q)$. The globalized querying in the WAVEQuery scheme is of the same order, since the inner product between the signal and the atoms in the dictionary is computed using the FFT. For the localized sub-sequence querying case, the complexity varies based on the number of sub-sequences used in the query process. If q sub-sequence based alignments are captured using the algorithm, the complexity is of the order $O(qQ \log qQ)$.

4.7 MPD WAVEQuery Alignment of Protein Sequences

The primary structure of the protein is formed by a sequence of twenty amino acids. The amino acids are derived as a result of the DNA transcription process. In particular, the synthesis of proteins is governed by the genetic code that maps all possible triplets or codons of DNA characters into one of twenty possible amino acids. In a similar method as with the DNA sequence where we mapped four characters, we can now use the Gaussian atom mapping with the amino acid sequence but map twenty characters. This will require the use of additional frequency shift parameters to represent the additional characters for the different amino acids, and the time shift parameter of the MPD decomposition can still be used, as before, to control the position of the amino acid in the sequence.

In the DNA sequence alignment, the mismatch between the nucleotide bases results in a very small correlation value between the different Gaussian atoms. This is essential because the correlation values represent the measure of similarity between two sequences, i.e.,

$$\langle g_{m,l}, g_{m,k} \rangle = \int_{\tau_s} g_{m,l}(t)\, g^*_{m,l}(t)\, dt \begin{cases} = 1, & l = k \\ \approx 0, & l \neq k \end{cases} . \tag{4.8}$$

Equation (4.8) corresponds to the inner product between any pair of nucleotide bases at position m, and it defines the correlation matrix between each of the

four nucleotide bases as the identity matrix. Note that a negative penalty may be applied to mismatches to obtain the alignment score. The correlation matrix for proteins is called the substitution matrix, and it is not an identity matrix. It is a matrix that contains match rewards, partial mismatch penalties, and complete mismatch penalties. This is because two amino acids that are not identical also have some similarity measure or non-zero correlation value. If we use the MPD decomposition as for the DNA sequence mapping, the Gaussian atoms will have an almost zero correlation for different frequency shifts. As a result, we need to modify the MPD mapping in order to take into consideration the BLOSUM-62 substitution matrix information [91, 92].

For the protein sequence mapping, we use all three MPD transformation parameters, time-shift, frequency-shift and scale change, of the Gaussian atom in (4.4). The time-shift and frequency-shift parameters again map the position and type of amino acid. The scale change parameter is used to assign a specific non-zero correlation value from the BLOSUM-62 substitution matrix between two non-identical Gaussian atoms (corresponding to two different amino acids). A look-up table based approach was adopted with the scale parameter pairs to realize a unique inner product corresponding to the penalties or rewards in the substitution matrix. The Gaussian signal for the kth amino acid that is mapped to frequency kF is defined in (4.4) as $g_{m,k,k}(t)=g((t-m\tau_s)/a_k)\, e^{-j2\pi kFt}$, where the time shift $m\tau_s$ maps the mth position of the amino acid in the sequence. The scale change parameter a_k may be sampled dyatically for fast implementation. It is given the same subscript as the frequency shift parameter to ensure its uniqueness to the kth amino acid type; it is a parameter that is used to ensure that the correlation value between two non-identical Gaussian atoms is not zero. Specifically,

$$\langle g_{m,l,l}, g_{m,k,k} \rangle = \begin{cases} 1, & l = k \\ \eta_{l,k}\,, & l \neq k \end{cases} . \tag{4.9}$$

The value $\eta_{l,k}$ in (4.9) is the (l,k)th element of the subsitution matrix that is directly related to the two scale parameters, a_l and a_k. Thus, the number of scale parameters that are assigned in the mapping is related to the number of different values in the substitution matrix that correspond to correlation values.

The WAVEQuery protein sequence alignment algorithm is very similar to the WAVEQuery DNA sequence alignment algorithm. The main differences are: (a) the atoms chosen from the dictionary also have a scale parameter in addition to the time-shift and frequency shift parameters; and (b) the threshold value, that the correlation values are compared to, is different as it has to take into consideration the elements of the substitution matrix.

For the protein sequence alignment case, the WAVEQuery algorithm was compared with the BLASTP algorithm [90] and the alignment results were identical for the two algorithms. Note that the sequences used in this testing did not have inherent repeats to check for better performance, as in the case of the BLAST. This can be attributed to the fact that, during the transcription process, these repetitive regions in the DNA were not transcribed to from amino acids. A sample alignment for the WAVEQuery algorithm compared with the BLASTP alignment is shown in Figure 4.11. Note that the raw scores of the two algorithms are close in value, indicating that the quality of the WAVEQuery alignment is comparable to that of the BLAST.

4.8 WAVEQuery Using the Metaplectic Transform

The Gaussian mapping provided three waveform transformation parameters that we exploited in the WAVEQuery mapping for use in the DNA and protein sequence alignment. The DNA sequence mapping used only two parameters whereas the

```
WAVEQuery Protein Alignment
Sequence #1
DIHSAGYFSAINQGVQSVMASFNSWNGKRVHGDKHLLTDVLKNQLGFDGFVVSDWNAHKFVEGCDLEQCAQAINAGVDVI

Seq Aligned
DI.SAG+FSAINQGVQSV.ASFNSWNGKRVHGDKHLLTDVLKNQLGFDGFVVSDWNAHKFV.G.DLEQCAQAINAGVDVI

Sequence #2
DISSAGFFSAINQGVQSVSASFNSWNGKRVHGDKHLLTDVLKNQLGFDGFVVSDWNAHKFVFGFDLEQCAQAINAGVDVI

Raw Score = 390

BLASTP
Score =  378

Query  1   DIHSAGYFSAINQGVQSVMASFNSWNGKRVHGDKHLLTDVLKNQLGFDGFVVSDWNAHKF  60
           DIHSAG+FSAINQGVQSV ASFNSWNGKRVHGDKHLLTDVLKNQLGFDGFVVSDWNAHKF
Sbjct  1   DISSAGFFSAINQGVQSVSASFNSWNGKRVHGDKHLLTDVLKNQLGFDGFVVSDWNAHKF  60

Query  61  VEGCDLEQCAQAINAGVDVI  80
           V G DLEQCAQAINAGVDVI
Sbjct  61  VFGFDLEQCAQAINAGVDVI  80
```

Figure 4.11: Alignment report for the WAVEQuery algorithm for protein alignment compared with BLAST raw score. Note the amino acid mismatches with positive value in the substitution matrix are represented by a '+' and the other mismatches are represented by a '.'.

protein sequence mapping required all three parameters in order to achieve high alignment performance. If more parameters are necessary for use in the WAVE-Query mapping, then a different generalized waveform transform needs to be exploited.

The metaplectic representation is an example of such a waveform representation [93,94]. It is a five-dimensional (5-D) waveform expansion into five different discrete transformations of an orthonormal basis function in the time-frequency plane. The metaplectic transform of a signal $x(t)$, using a generalized wavelet function $w(t)$, is defined as [93]

$$\Gamma_x(\tau, \nu, a, p, q; w) = \langle x, (\mathcal{F}_\nu \, \mathcal{T}_\tau \, \mathcal{A}_a \, \mathcal{Q}_q \, \mathcal{P}_p \, w) \rangle \tag{4.10}$$

where,

$(\mathcal{F}_\nu w)(t) = w(t) \, e^{j2\pi\nu t}$ causes a frequency shift ν,

$(\mathcal{T}_\tau w)(t) = w(t - \tau)$ causes a time shift τ,

$(\mathcal{A}_a w)(t) = |a|^{-1/2} w(t/a)$ results in a time scale change a,

$(\mathcal{Q}_q w)(t) = w(t)\, e^{j\pi q t^2}$ causes a shearing along the IF qt (multiplication in the time-domain by an LFM chirp),
and $(\mathcal{P}_p w)(t) = (-jp)^{-1/2} w(t) * e^{j\pi(1/p)t^2}$ causes a shearing along the group-delay pf (multiplication with an LFM chirp in the frequency domain), where $*$ denotes convolution.

When the metaplectic transform in (4.10) is used for WAVEQuery mapping, the time-shift and frequency-shift parameters can be used to represent the position of the character and the type of character in a sequence, just as before. The time scale parameter can be used in the protein sequence alignment problem to represent the non-zero correlation values between non-identical amino acids. The new time-shearing parameter q, which is essentially the modulation rate of an LFM chirp in the time domain, can be used to represent the prediction value of a character being in the next position in a DNA or protein sequence. This prediction values are based on a probability matrix which describes the probability of the character (nucleobase or amino acid) occurring in the next position. The probability matrix can be either a matrix with equi-probable values (probability value of 1/4 in the case of nucleotides or probability value of 1/20 in the case of amino acids), or it can have probability values derived using the composition of a given set of sequences in a database. The fifth parameter, frequency shearing parameter p, of the metaplectic transform can be used to represent gaps, instead of using an additional frequency, as in the case of the Gaussian mapping. The frequency shearing parameter represents a modulation along a line, and it can be extended to the next position in a sequence to represent the gaps. The choice of the wavelet function $w(t)$ is crucial in this scenario, and most of the current wavelet basis functions, while efficient in time localization, are not simultaneously very efficient in frequency localization.

Chapter 5

STRUCTURAL WAVEFORM MAPPING FOR PROTEIN ALIGNMENT

5.1 Structural Similarities in Proteins

As proteins that are similar in structure with unrelated sequences have been discovered, sequence alignment techniques as discussed in Chapter 4, are not sufficient for finding similarities in those proteins. It becomes necessary to search for similarities and establish homology between proteins based on their shape and three-dimensional (3-D) conformation. The secondary structure of a protein is in the form of α helices and β sheets, which are collectively called secondary structure elements and are connected by loops. The tertiary structure is based on the 3-D representation of the proteins as defined by their atomic co-ordinates [95]. Protein structural superposition deals with the alignment of two or more protein secondary and tertiary structures in this 3-D co-ordinate space. In particular, structural alignment finds and compares multiple protein structural conformations based on either global similarity measures or local features [96]. The metric commonly used in finding the similarity is the root-mean-square distance (RMSD) metric. Similarity measures based on local features may include packing size or interaction patterns.

There are two main methods for comparing protein structures: the intermolecular method and intramolecular method. The intermolecular method compares and superposes two or more protein structures in order to achieve maximum overlap in the 3-D space. This is achieved by geometric fitting of the two structures on a residue-residue pair basis. The intramolecular method compares protein structures based on the structural internal statistics by providing a quantitative similarity between the corresponding residue pairs. It is achieved by reducing the 3-D information into 2-D information.

A hybrid scheme using both the intermolecular and intramolecular methods is also used. In our work, we consider an intermolecular structural alignment in the 3-D space.

5.2 Current Structural Alignment Techniques

5.2.1 Computational-based Structural Alignment

Various techniques exist in literature for protein structural alignment. When using the intermolecular method, the basic principle behind superposing two protein structures is to minimize the RMSD between the two structures. This can be achieved by obtaining a residue-residue correspondence: fixing one of the structures and moving the other structure laterally and vertically towards the other structure. This process is called translation, and it results in the two structures having the same coordinate frame. The structures are also rotated relative to each other along the 3-D coordinate axis system, and the RMSD is measured at each orientation. The orientation that yields the lowest RMSD measure results in the best fit for the alignment of the two structures. Note that the amino acid atoms have six degrees of freedom in the 3-D coordinate space: translations in the x, y and z axes, and rotations along the (x, y), (y, z) and (z, x) planes. The 3-D coordinates of the atoms that constitute an amino acid, and thus a particular protein in the structure, can be found at the Protein Data Bank (PDB) as discussed in Chapter 2. An in-depth review on RMSD measures and the comparison algorithms is provided in [3, 97–100].

There are a few structure alignment software tools available on the World Wide Web, and we discuss some of them next. DALI [101] is a structure comparison method that is hosted at [102]. This is an intramolecular distance measure based approach, which maximizes the similarity between two distance graphs. For each of the proteins, the distance between all α-carbon Cα atoms of each indi-

vidual protein is calculated and the matrices are compared to identify the regions with the highest similarity. These become the algorithm seeds which are later clustered together using an average score measure derived from the probability distribution in the database. This algorithm, first introduced in 1993, has seen improvements in performance and the latest version of the algorithm is presented in [103].

VAST [104] performs structural alignment using both intermolecular and intramolecular approaches [105]. This superposition is based on the directionality of the secondary structural elements, which are represented as vectors. Depending on the number of vector matches, the similarity level between two structures is determined, and the optimal alignment is obtained.

The combinatorial extension (CE) is a method for calculating pairwise structure alignments [106, 107]. It is an intramolecular distance approach that considers eight (or octemeric) residues as one single residue and the distance matrices are constructed at that level. Using combinatorial extensions, the aligned fragment pairs that result in continuous alignment pairs are extended and the optimal alignment is obtained. Since this method considers eight residues at once, the computational time is reduced. This is, however, at the cost of the alignment accuracy.

Other computational tools include the Rapid Alignment of Protein In Terms of DOmains (RAPIDO) [108] that is based on genetic algorithm , MAtching Molecular Models Obtained from THeory (MAMMOTH) [109], and the Sequential Structural Alignment Program (SSAP) [110] that uses double dynamic programming that are in use for the protein structure alignment. The 3D-COFFEE approach [111] uses both protein sequences and structures and combines them to obtain multiple alignments.

The structural alignment problem has found solutions in many areas, including computational techniques, data mining, signal processing and media engineering. Some techniques that have been developed include dynamic programming algorithms [72], [71], [112], [113], hashing techniques for the RMSD measure in [114, 115], reduced dimensionality representations [116], genetic algorithms [117, 118], n-gram based language modeling techniques [119], spectral kernel methods [120], hidden Markov models [121], vector representation based methods [98, 122], and regression analysis methods [123].

5.2.2 Signal Processing Based Structural Alignment

Some of the alignment methods proposed in the literature are based on the use of signal processing approaches and waveform basis functions. We will discuss some of these approaches next.

Gaussian Based Alignment The Gaussian-based alignment for protein structures (GAPS) algorithm [1] was first used for the superposition of small molecules [124], and then extended to the superposition of protein structures [51, 125]. In the GAPS algorithm, the kth atom of A_ith amino acid is represented by the spherically symmetric Gaussian waveform

$$g_k^{A_i}(\mathbf{r}) = c_k \ \exp(-d_k|\mathbf{r} - \mathbf{R}_k|^2) \tag{5.1}$$

that is defined using the 3-D atomic co-ordinate axis $\mathbf{r} = (x, y, z)$. In (5.1), $\mathbf{R}_k$ is the nuclear coordinate position of the kth atom, and the coefficient c_k and exponent parameter d_k determine the value of its maximum height at the origin and its decay, respectively.

[1] We would like to acknowledge our discussion on the signal processing interpretation of protein superposition with one of the authors of [51, 124].

The A_ith amino acid residue is expressed as a linear combination of the Gaussians placed at each of the atoms in the amino acid.

$$G_{A_i}(\mathbf{r}) = \sum_{k \in A_i} g_k^{A_i}(\mathbf{r}) \tag{5.2}$$

The Gaussians are either placed along the main chain atoms or along the α-carbon atoms. It is to be noted that placing the atoms along the α-carbon atom approximates the performance obtained by placing the Gaussians along the main chain atoms.

Finally, protein A is represented as a linear combination of the amino acid representations as:

$$G_A(\mathbf{r}) = \sum_{A_i \in A} G_{A_i}(\mathbf{r}) \tag{5.3}$$

Using this representation for proteins, the similarity between two proteins A and B is given by the following similarity measure, which provides a measure of the structural overlap:

$$\Omega_{AB} = \int G_A(\mathbf{r})\ G_B(\mathbf{r}).d\mathbf{r} \tag{5.4}$$

The normalized measure, also called a similarity index, is provided by:

$$\mathrm{Sim}(A, B) = \frac{\Omega_{AB}}{\sqrt{\Omega_{BB}}\sqrt{\Omega_{BB}}} \tag{5.5}$$

and this value is bound between 0 and 1.

The similarity measure is maximized by rotating and translating one structure with respect to the other until the superposition of the two structures is optimized. The rotations and the translations in the optimization procedure are carried out directions that span the 3-D coordinate system axis. The transformations are performed in 45 degree increments and the similarity is evaluated at a fixed number of points. Based on a rank order list of the similarities, the positions

that correspond to the best fit are used in a standard gradient-descent technique. Finally. there is a post-alignment analysis step that performs a structure-based sequence matching; this step enables the alignment of two structurally aligned sequences.

The GAPS algorithm was used for pairwise and multiple structure alignment, where the structures were classified based on their pairwise sequence and structural similarities. The main drawback of this method is its computational intensity when used with a large number of amino acids, since it is applied at the small molecule level. Also, this method cannot perform local alignment, i.e. alignment over smaller segments of the structure.

Fourier Transform Based Alignment In [126], the fast Fourier transform (FFT) was used to compute correlations for determining the geometric fit between two protein structures. This algorithm assigned the protein location by representing them using discrete binary functions.

A crystallographic Fourier transform approach was presented in [127] for molecule superposition, based on optimizing the overlap of electron density as a function of molecule translation. RigFit, a rigid body molecular ligand superimposition algorithm was presented in [128]. This algorithm also used Gaussian assignments to molecules as in [124], but it performed the translation and rotation in the Fourier space based on convolution properties. A similar algorithm using a Laplacian filter was presented in [129], and an algorithm with FFT based convolution and Gaussians was presented in [130]. By reducing the degrees of freedom from six to five in [131], where there were five angular degrees of freedom and just one linear degree of freedom, the structural alignment algorithm was made faster using the matching algorithm in [132].

Spherical polar Fourier correlations were used for protein superposition

in [133]. This algorithm was capable of fitting protein structures in the 3-D space taking into consideration all six degrees of freedom, and it was further improved and extended in [134]. In [50, 135], the polar FFT and Radon bases were used for shape matching and the algorithm was extended for protein structures using the spherical trace transform [135, 136]. The algorithm uses the 3-D Radon transform to examine descriptors and then applies a set of functionals to the transform coefficients. Similarity measures are created for the descriptors and introduced into a 3-D model matching algorithm. Note,however, that since Radon bases are not translation and rotation invariant, a pre-processing step is necessary to achieve rotational invariance. This is performed using the center of masses and principal component analysis.

5.2.3 Other Signal Processing Based Alignment Methods

A Gaussian weighted RMSD measure algorithm for protein superposition of proteins was presented in [137]. An algorithm based on the use of cepstral feature components of the primary amino acid sequence that was mapped to the electron ion interaction potential (EIIP) was presented in [138, 139]. In [140], an approach using curve moment invariants and iterative closest points, similar to the DALI algorithm were discussed. A survey on local shape similarity alignment methods for protein structures is provided in [141], where curved surfaces are represented by circular curvature patches and pairwise overlays over the entire structure are evaluated. In [142], 3-D shape based signatures were used in the retrieval of protein structures from databases. A maximum likelihood estimation algorithm was also proposed in [143, 144].

5.3 Need for New Structural Alignment Techniques

The current state-of-art signal processing based approaches for structural alignment use representation for protein structures that are largely based on the posi-

tion of the atomic coordinates in 3-D space. As a result they are not successful in modeling the shape of a protein structure , information of which is either predicted from analysis or measured fro experiments. These representations do not provide good models for important features such as protein folds in the α helices and β sheets, and they do not preserve directionality information, especially for multiple folds in compact spaces.

5.4 Modeling the Protein Superposition Problem

Given two protein structures, the protein superposition problem matches the structures by having them undergo transformations such as translations and rotations, in order to find their best possible structural overlap. Protein structures have six degrees of freedom: translations along the x, y and z axes, and rotations along the (x, y), (y, z) and (z, x) planes. Hence, there is a need for a 3-D structural representation model whose information content remains unchanged when translated or rotated in the 3-D space. The representation model needs to be linearly separable so that it can be able to store information on the structures' 3-D atomic co-ordinates as wells as on the order of the individual amino acids in the protein sequence. Also, we must be able to detect localized similarity (or motifs) in the structure in addition to the similarity over the entire structure.

In order to further illustrate the need for a linearly separable representation, we consider two proteins, P_A and P_B, whose structures need to be aligned. The two proteins are separable into small substructures, and alignments in the substructures are to be detected. The substructure representations are given by

$$P_A = \sum_{m=1}^{M} P_{A_m}, \quad P_B = \sum_{n=1}^{N} P_{B_n} \tag{5.6}$$

where M and N denote the number of small substructures. A sub-structure is defined as a segment of a structure with a minimum of three amino acids so it can contribute to the shape of the structure. Note that, in practice, the length

of a sub-structure is dependent on the similarity measure between two aligned segments.

If the substructures P_{A_m}, $m = 1, \ldots, M$ and P_{B_n}, $n = 1, \ldots, N$ have similar structures, they can be found using local structural alignment. If the similarity occurs over the entire structure length, then the proteins are said to be globally structurally similar. Note that P_{A_m} can be a small substructure or a cluster of amino acid,s depending on the desired level of alignment performance.

If the structures of P_A and P_B are similar over their entire lengths, i.e., $P_A \equiv P_B$, then

$$\sum_{m=1}^{M} P_{A_m}\ T_{A_m}(x, y, z) \equiv \sum_{n=1}^{N} P_{B_n}\ T_{B_n}(x, y, z) \tag{5.7}$$

where $T_{A_m}(x, y, z)$ and $T_{B_n}(x, y, z)$ are the transformations on the structures along the 3-D coordinate space that can result in the similarity of the structures of P_{A_m} and P_{B_n}.

5.5 Chirp wAveform Representation for Protein Structures (CARPS)

5.5.1 Waveform Representation Model

We propose a waveform-based representation for depicting the secondary and tertiary structures in proteins. Our aim is to use this representation for protein structural alignment. This Chirp wAveform Representation for Protein Structures (CARPS) used linear frequency-modulated (LFM) chirp waveforms that are defined as multi-time domain higher order functions. We first describe the CARPS for a one-dimensional (1-D) case and then extend it to the 3-D case to represent protein structures. As we will demonstrate, the CARPS is capable of depicting protein folds and by embedding a unique parameter for directionality, sufficient computational time is saved in analyzing of protein structures.

As discussed in Chapter 3, an LFM chirp signal is a time-varying waveform

defined as:

$$h_l(t) = \sqrt{2\,t}\; e^{j2\pi c_l\, t^2}, \quad 0 < t < T_d \tag{5.8}$$

where c_l in $(\text{Hz})^2$ is the frequency-modulation (FM) rate and T_d is the waveform duration in seconds. The instantaneous frequency (IF) of the LFM chirp, given by $2\,c_l\,t$, represents the linear frequency variation of the waveform with respect to time. Ideally, the time-frequency representation of this waveform is a line with slope $2\,c_l$. The amplitude modulation in (5.8) ensures that an infinite-duration LFM chirp is orthogonal. This can be shown by taking the inner product between two LFM chirp signals with different FM rates and infinite duration. For finite duration signals, we can show orthogonality by fixing the difference between the FM rates as $\Delta c = K/T_d^2$, for some integer number K [145].

For a highly localized waveform representation, the chirp is windowed with a Gaussian signal. This is because Gaussian signals are the most concentrated signals in both time and frequency due to Heisenberg's uncertainty principle

A 1-D time-frequency shifted and scale transformed Gaussian signal is given by

$$g(\tau, \nu, a) = g\left(\frac{t-\tau}{a}\right) e^{-j2\pi\nu t} \tag{5.9}$$

where $g(t)$ is a basic Gaussian waveform given by $g(t) = e^{-\pi t^2}$, τ is the time shift, a is the time scale, and ν is the frequency shift.

In order to represent the shape of a protein structure, we consider an extension of the windowed chirp signal in a 3-D time-domain, (t_x, t_y, t_z). Time-shift (or translation) parameters along each of the time axes and rotations characterized by the 3-D FM rate parameters will be used to provide key information about the spatial co-ordinates, folds, and directionality of the protein structure.

The non-windowed version of the 3-D chirp waveform is given by:

$$h_{\mathbf{c}}(\mathbf{t}) = 2\sqrt{2t_x t_y t_z}\; e^{j2\pi\mathbf{t}(\text{diag}(\mathbf{c}))\mathbf{t}^T}. \tag{5.10}$$

where the 1×3 row vector $\mathbf{t} = [t_x \ t_y \ t_z]$ represents the three coordinate axis (x, y, z), $\mathbf{c} = [c_x \ c_y \ c_z]$ provides the FM rates along each axes, and T denotes the vector transpose. The amplitude modulation is needed for orthogonality, similar to the 1-D chirp waveform case. The 3×3 matrix diag($\mathbf{c}$) is a diagonal matric whose off diagonal elements are given by the row vector $\mathbf{c}$.

The Gaussian window can be represented in the form of a multivariate Gaussian waveform as

$$g(\mathbf{t}; \boldsymbol{\tau}, \boldsymbol{\Sigma}) = \frac{1}{2(\pi)^{\frac{n}{2}} \ |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \ \exp\left(-\frac{1}{2}(\mathbf{t} - \boldsymbol{\tau})\boldsymbol{\Sigma}^{-1}(\mathbf{t} - \boldsymbol{\tau})^T\right) \tag{5.11}$$

that is centered at $\boldsymbol{\tau} = [\tau_x \ \tau_y \ \tau_z] \in \mathbb{R}^3$, and has covariance matrix $\boldsymbol{\Sigma} \in S^3_{++}$. The term $1/2(\pi)^{\frac{n}{2}} \ |\boldsymbol{\Sigma}|^{\frac{1}{2}}$ which provides the normalization factor is independent of $\mathbf{t}$. While the Gaussian window can also be represented as the product of three independent Gaussian signals using Equation (5.9), the representation in Equation (5.11) is preferred since the cross terms in the covariance matrix $\boldsymbol{\Sigma}$ provide some measure of control over the spread of the Gaussian in the three planes.

Using the 3-D chirp signal from (5.10) and the Gaussian window from (5.11), we represent the windowed chirp signal as

$$h^g(\mathbf{t}; \mathbf{c}, \boldsymbol{\tau}, \boldsymbol{\Sigma}) = h_{\mathbf{c}}(\mathbf{t} - \boldsymbol{\tau}) \ g(\mathbf{t}; \boldsymbol{\tau}, \boldsymbol{\Sigma}) \tag{5.12}$$

Equation (5.12) provides the CARPS with time shift vector parameter $\boldsymbol{\tau}$, FM rate vector parameter $\mathbf{c}$ and covariance matrix parameter $\boldsymbol{\Sigma}$; each of these parameters can be appropriately chosen to represent a unique property of the protein structure.

Note that the CARPS satisfies the properties that were desired in a representation for protein structures. Specifically, the Gaussian chirp is sampled compactly such that the correlation between two CARPS with different parameters is almost zero. Due to the use of the Gaussian window that is highly concentrated in

both time and frequency, the CARPS can provide a good model the density of the protein atoms very well. The rotation transformation is inherent to the CARPS model since changing the FM rate vector causes changes in directionality. By using the most concentrated window and a linear representation, we ensure that translations do not result in overlaps. Furthermore, the linear separability of the CARPS enables local similarity searches.

5.5.2 Chirp-based Protein Structure Representation

We consider the 3-D protein structure whose co-ordinates are specified in the PDB file from [46]. Let $A_i = (x_i, y_i, z_i)$ and $A_{i+1} = (x_{i+1}, y_{i+1}, z_{i+1})$ be two consecutive points in a protein structure. The points correspond to the coordinates of two neighbor amino acids. We want to use CARPS in (5.12) such that these points appear as two outer-most points in the mapped $[t_x, t_y, t_z]$ plane. In order to achieve this, we first place the Gaussian window at the center of the two points. The covariance matrix $\mathbf{\Sigma}$ of the Gaussian window plays an important role in its 3-D orientation. Note that the eigen decomposition of the covariance matrix provides the eigen vector matrix, which is the orientation or rotation matrix of the Gaussian signal in 3-D space. The design of the rotation matrix is based on the pair-wise angle between the two points and it can be obtained from geometry. The angles with respect to each of the axes are given by $(\theta_x, \theta_y, \theta_z)$, and they are calculated for each segment of the structure using the co-ordinates that connect the segment. For the points A_i and A_{i+1}, the angles are given by,

$$\begin{aligned}
\theta_x &= \arccos\left(\frac{(x_{i+1} - x_i)}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}}\right) \\
\theta_y &= \arccos\left(\frac{(y_{i+1} - y_i)}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}}\right) \\
\theta_z &= \arccos\left(\frac{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}}\right)
\end{aligned} \tag{5.13}$$

Using these angle values, the rotation matrices are obtained using:

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix}$$
$$R_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \tag{5.14}$$
$$R_z(\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The covariance matrix is then calculated by considering the orientation along a particular plane. The rotation matrix can also be obtained using a Gram-Schmidt procedure to find the orthonormal plane for a given set of vectors. The two methods provide identical results. The value of the variances for each of the three axes of the Gaussian window are set such that the window has the widest region of support in the plane that links the two points A_i and A_{i+1} and is very narrow in the other planes.

Following the design of the Gaussian window, we modulate the 3-D chirp signal using this window. This process of modulating the 3-D chirp signal with the Gaussian signal significantly reduces the number of cross terms in the time-frequency (TF) plane when multiple segments of the structure are being considered. Also, since the ideal TF representation of a chirp signal is as a line whose slope is related to the FM rate $\mathbf{c}$, information about the directionality within the structure (angles) is embedded in the higher dimensions as well.

Hence, the covariance matrix $\mathbf{\Sigma}$ and the chirp rate $\mathbf{c}$ of the Gaussian window and the chirp signal, respectively, have the information on the orientation and the directionality of the protein structure embedded in them.

Thus for a protein structure A with $N+1$ coordinates connected by N segments, the CARPS is given by

$$H_A = \sum_{i=1}^{N} h^g(\mathbf{t}; \mathbf{c}_{A_i}, \boldsymbol{\tau}_{A_i}, \boldsymbol{\Sigma}_{A_i}) \tag{5.15}$$

where $\mathbf{c}_{A_i}, \boldsymbol{\tau}_{A_i}$, and $\boldsymbol{\Sigma}_{A_i}$ are the windowed chirp parameters for the ith segment in the structure A.

We mapped the protein 3-D structure using the CARPS, and an example is shown in Figure 5.1 for the NMR structure of the lung surfactant peptide SP-B (PDB ID: 1KMR) is shown in . Note that the windowed chirp signal replicates the 3-D shape of the structure exactly.

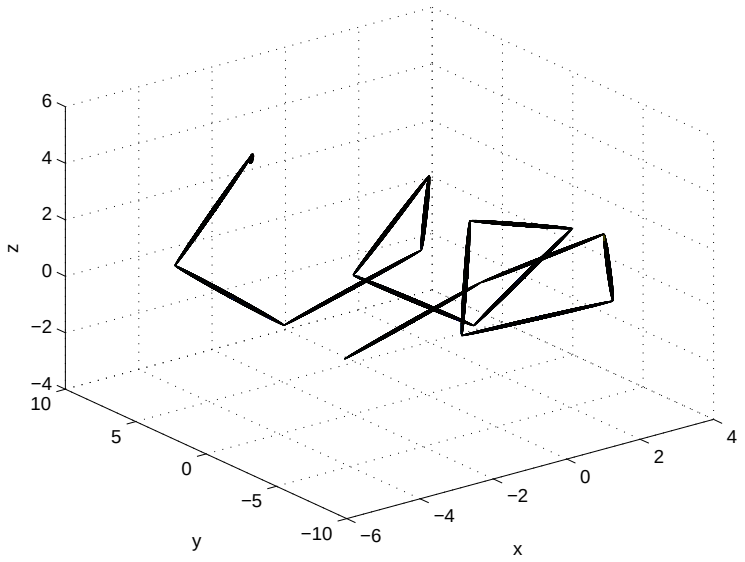


Figure 5.1: Example of the CARPS for the NMR structure of lung surfactant peptide SP-B (PDB ID: 1KMR). The axes measurements are all in Angstrom units (10^{-10} m). Note the α-helix in the structure connected by 3-D chirps with a Gaussian window.

5.5.3 Waveform Parameters relating Sequence to Structure

While the shape and folds of the protein structure obtained from NMR or X-Ray crystallography experiments convey important information, it is also possible to

derive or predict the protein structural information based on the primary amino acid sequence. The structure, as mentioned earlier, has six degrees of freedom, and the bond between the amino acids is stable due to various factors including hydrogen bonds, hydrophobic interactions and the conformational entropy. While the hydrogen bonds are known to have an effect on the shape of the structure, hydrophobic moments of the entire molecule and that of the segments of the secondary structure can help analyze the structure of a protein [52]. The sequence of amino acids determines the 3-D shape of the protein, and this is due to the free energy resulting from the hydrophobic effect [146]. As a result, hydrophobicity is an important parameter that can be used to control the stability of a protein structure.

For every amino acid in a protein sequence, there is a value of hydrophobicity that can be assigned, and this is a representation of how stable the structure is. The parameter can be viewed in a signal representation scenario as the energy or the amplitude of the signal. For the CARPS system, we will introduce an amplitude parameter ρ_i for an amino acid A_i, where ρ_i is the hydrophobicity value of the amino acid A_i. By embedding this parameter in the structural representation of a protein, we not only represent the folds and shape of the protein structure, but also the stability of the structure and the ability of the structure to undergo conformations based on the stability value. Note that we are also indirectly embedding the amino acid composition of the protein in the structural representation in the form of a numerical map. This is particularly helpful in the problem of protein structure prediction, when the amino acid composition is known and the structural information is unknown or needs to be verified.

The resulting overall CARPS is now given by

$$H(\mathbf{t}) = \sum_{i=1}^{N} \rho_i \; h^g(\mathbf{t}; \mathbf{c_i}, \boldsymbol{\tau}_i, \boldsymbol{\Sigma}_i) \tag{5.16}$$

where N is the number of segments connecting a pair of amino acids in the structure, and ρ_i is the hydrophobicity value of the ith amino acid in the structure, with the windowed chirp parameters for the ith segment in the structure as described in Equation (5.15).

5.6 Chirp-based Alignment for Protein Structures (CAPS) Approach

The new chirp-based alignment for protein structures (CAPS) approach is based on the use of the CARPS proposed in Section 5.5.2 and a correlation measure based matched filter approach. Note that the use of the hydrophobicity parameter of the protein structure presented in Section 5.5.3 is optional in this case, because the alignment is based on the directional descriptors of the representation, i.e., IF of the LFM chirps and the covariance matrix of the Gaussian window.

5.6.1 Pairwise Alignment of Protein Structures

We consider two protein structures that are to be aligned in after applying the CARPS in Equation (5.15). For protein structures A and B with M and N segments, respectively, the representation is given by:

$$H_A(\mathbf{t}) = \sum_{i=1}^{M} h^g(\mathbf{t}; \mathbf{c}_{A_i}, \boldsymbol{\tau}_{A_i}, \boldsymbol{\Sigma}_{A_i})$$
$$H_B(\mathbf{t}) = \sum_{j=1}^{N} h^g(\mathbf{t}; \mathbf{c}_{B_j}, \boldsymbol{\tau}_{B_j}, \boldsymbol{\Sigma}_{B_j})$$

This can be perceived as a signal expansion representation, with protein structure features embedded in the parameters such as the LFM chirp rate and the mean and covariance of the Gaussian window. If the signals $H_A(\mathbf{t})$ and $H_B(\mathbf{t})$ have similar signal parameters over the entire length, the structures are said to be completely aligned. If the signal parameters are similar over a portion of the

length of the structure, the structures are said to be partially aligned. This partial structural alignment is not considered by most state-of-art techniques.

The proposed CARPS algorithm first performs transformations to one of the structures based on the orientation of the other structure, in order to be able to align the first few segments of the two structures. This transformation is usually a shift (translation) in the center of the Gaussian or a change in the structural orientation (rotation) in order to align the first segments. The rotation is performed by using the angles from (5.13) such that the first segments align. In order to obtain the similarity measure between the two structures, we consider the inner product between the signals representing the structures. The cross-correlation provides a similarity measure between the two structures. The inner product α_{pq} between the segments $H_{A_p}(\mathbf{t})$ and $H_{B_q}(\mathbf{t})$ is given by:

$$\alpha_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H_{A_p}(t_x, t_y, t_z)\; H_{B_q}(t_x, t_y, t_z)\; dt_x\; dt_y\; dt_z \tag{5.17}$$

Due to the orthogonally designed chirps and the nature of the Gaussian window, this similarity measure is maximized when the windowed chirp parameters of the two signals are almost identical. Due to the highly concentrated nature of the windowed chirp signal in the TF plane, the inner product is a very sensitive measure, and tracks the similarity in every parameter of the signal. Due to this, the key parameter of directionality of the structure is preserved in the 3-D plane. Note that we have normalized the amplitudes of the LFM chirp signal and the Gaussian window during the process of modulation, hence eliminating the need for normalization of the similarity measure at this stage.

Global Structural Alignment In an ideal scenario of global alignment with two protein structures of identical lengths, and almost similar shapes, the cross-

correlation over the entire length of the two protein structures will provide a maximal inner-product measure. However, in practice, the similarity measure will not be maximum, since the structure undergoes multiple conformations due to the degrees of the freedom in the structure. Hence, in order to account for the conformations, we consider a threshold measure ξ for the inner product to consider the similarity between the two structures. Note that this threshold is applied to the inner products between each of the segments in the two structures. This ensures that the structures are compared on a piecewise basis rather than over the entire length of the structure. For $M \simeq N$, if $\alpha_{pq} \geq \xi$, $\forall\ p \in M,\ q \in N$, the two structures H_A and H_B are said to be aligned over the entire length.

Local Structural Alignment For the local alignment of protein structures, two structures with different lengths and similarity over local segments of the structure (sub-structure) are considered, and this local structural similarity is attributed to distantly related proteins. Since the length of the similarity and the start and stop positions of the sub-structures are usually not known, we adopt a similarity search method that searches for the similarity of a given segment over the entire length of the other structure. This is accomplished using the inner product between two segments as shown in Equation (5.17). This is obtained for all segments in one of the structures with segments in the other structure, and a correlation matrix representation for the two structures is obtained. This matrix is of the form,

$$\Xi(A,B) = \begin{bmatrix} \xi_{11} & \cdots & \xi_{1N} \\ \xi_{21} & \cdots & \xi_{2N} \\ \vdots & \ddots & \vdots \\ \xi_{M1} & \cdots & \xi_{MN} \end{bmatrix} \tag{5.18}$$

With this similarity measure for all segments of the two structures, similarity over two sub-structures is found by observing the correlation values diagonally.

Similarity over the entire structure would be represented by the primary diagonal elements having values greater than the threshold ξ. However, since we are identifying locally aligned sub-structures, we look for correlations in the entire matrix, diagonally observing segments with similarity measures greater than the threshold. In order to simplify the identification of similar regions, we apply a thresholding on the matrix to represent values below the threshold and above the threshold. Note that multi-level thresholding (three or four levels) will provide better results in the case of local alignments, since just one segment of the sub-structure may end up undergoing more conformations when compared to the rest of the segments. Hence, by observing the similarity measures diagonally, we are able to identify locally aligned sub-structures. Note that, a minimum length of the segments maybe incorporated in order to be able to classify two sub-structures as similar. An illustration for the similarity matrix of locally aligned segments in two structures is shown in Figure 5.2.

In structural alignment, it maybe possible that two structures maybe completely aligned except for a portion of the structure, as shown in Figure 5.3. Even though this alignment occurs over the entire length of one sequence, it is considered to be local structural alignment.

Note that the number of inner product computations in this case may cause an overload on the algorithm. In order to improve on the computational efficiency, the inner products can be computed using fast Fourier transforms.

5.6.2 Extension to Alignment of Multiple Protein Structures

Multiple protein structure alignment is an extension of the pairwise structural alignment as multiple protein structures can be simultaneously aligned. The aim of multiple structure alignment is to build a phylogenetic tree depicting evolutionary relationships among species. Firstly, all the structures are iteratively

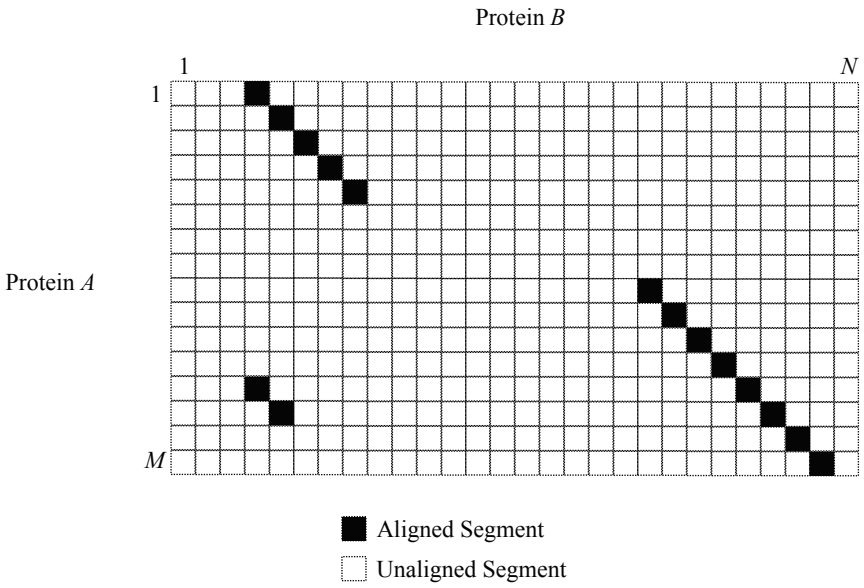


Figure 5.2: Two threshold level similarity matrix plot for the local structural alignment case. The structures of two proteins are aligned locally in the regions specified by the regions of similarity diagonally. The case of 5 and 8 aligned segments is considered as a structural match, while the 2 aligned segments are not considered to be a structural match.

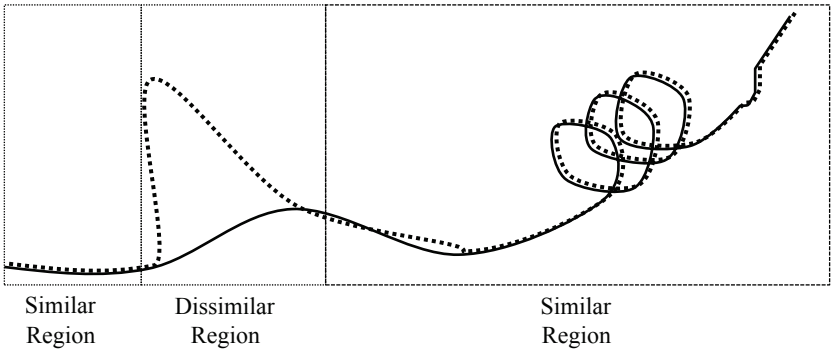


Figure 5.3: Local structural alignment case where two protein are structures aligned over the entire length of the structure except over a portion.

compared in a pairwise manner. Following this, the two structures with the largest similarity measure are re-aligned. The similarity measure between these re-aligned structures is used as the basis for comparison to perform pairwise alignment for the other structures. This is performed such that the similarity measure between all structures is maximized, hence forming the phylogenetic tree.

5.6.3 Classification among Structural Classes based on Directional Descriptors

We will consider classification based on α-helices and β-sheets structural classes using the directionality feature in the CARPS representation and the CAPS algorithm described in Sections 5.5 and 5.6.1.

Structural classes are first defined based on a generalized representation of the α-helices, β-sheets and their shape descriptors. In order to achieve this, we take the tested structures of these classes and use them as a reference for classification. We next consider a new structure that is to be classified. We first determine the shape descriptors $(\mathbf{c}, \mathbf{v})$ of the structure by mapping them to windowed LFM chirps, where $\mathbf{c} = [c_x, c_y, c_z]$ represent the chirp rates of the windowed chirp signal and $\mathbf{v} = [v_p, v_q, v_r]$ represent the eigen vectors of the covariance matrix $\mathbf{\Sigma}$ described in Equation (5.11). The shape descriptors of the structure to be classified are then compared with the shape descriptors of the reference classes. Usually, the comparison is performed as a binary operation matching process over a short segment of the structure. This is specifically done in order to stop the process of classification if an α-helix is compared with a β-sheet or vice-versa, before proceeding to check classification along the entire length of the structure. Following this, we look into further classification by performing a pairwise alignment with the reference structures in the class. Note that while performing this pairwise alignment, the hydrophobicity value of the amino acid is also considered, since the hydrophobicity plays an important role in determining the folds in

the protein structure. The class of the reference structure providing the highest similarity measure is the structural class of the new structure.

5.7 Experimental Setup And Results

We simulated the global and local alignment for protein structures for both pairwise and multiple alignment scenarios. This was done using MATLAB on a 2-core system with a 2.4 GHz processor, 4 GB RAM Intel Core 2 Duo computer. We tested various alignment scenarios using structures from the PDB [46] and the number of residues in the structures ranged from 10 to 200. In a few cases the model of the structures were used, since the actual structure was not available. Note that the model usually gives a close approximation of the structure. The results from DALI [102] were used as the ground truth in the analysis, and the metric used to determine the closeness to structure was the root mean-squared distance (RMSD) metric. Note that we represent the protein structure using the α-carbon coordinates from the PDB file, and this is also referred to as the backbone structure.

5.7.1 Global Alignment

In the pairwise global alignment, we consider structures that have identical or almost identical lengths, since we wish to find alignment over the entire length of the structure. We obtain different structures for the same protein including different models in a few cases. These structures undergo multiple conformations along the entire length, and we want to find matches between them. In our experimental setup, we have the proteins with the PDB ID as mentioned in Table 5.1 and perform pairwise alignment using the algorithm outlined in Section 5.6.1. We tabulate the number of aligned residues and compare it with the total number of residues, and we also obtain the mean RMSD measure between the alignment

structures. Our results are compared with the structural alignment obtained using DALI.

PDB ID number	Number of Aligned Residues (Number of Residues aligned by DALI)	Mean RMSD in angstroms (Å)
2L24	13 (13)	0.1079
2KIB	56 (56)	0.7557
2KWY	41 (43)	2.097
2LAT	35 (37)	0.7462
2L07	15 (18)	1.7017
2L10	33 (37)	4.7716
2KXK	54 (54)	1.4181
2KYK	38 (39)	1.5504
2KY8	68 (70)	0.8580
2LAM	29 (29)	0.1340
2L2L (Coil complex)	75 (79)	2.1904
2KXW	98 (100)	1.0979
7ZNF	26 (30)	1.4550
1AMC	28 (28)	0.6884
1KTX	36 (37)	0.4326

Table 5.1: Pairwise Global Structural Alignment Results.

The total number of aligned residues is provided and compared with the total residues in the protein structure in parenthesis. In the case of pairwise global alignment, we noticed that for each of the structure pairs to be aligned, more than 90% of the residues were aligned. In other words, the majority of the segments which underwent conformations were superposed efficiently. This was validated using the DALI tool, and also the RMSD distance measure was obtained. Note that the RMSD measure does not exceed 5Å in any of these cases, thus ensuring that the two structures are efficiently aligned. A sample alignment for the structure cyclotide Cter M (2LAM) is shown in Figure 5.4. Note that all 29 residues in the two structures are superposed efficiently.

We next studied the performance of the algorithm by extending it to the

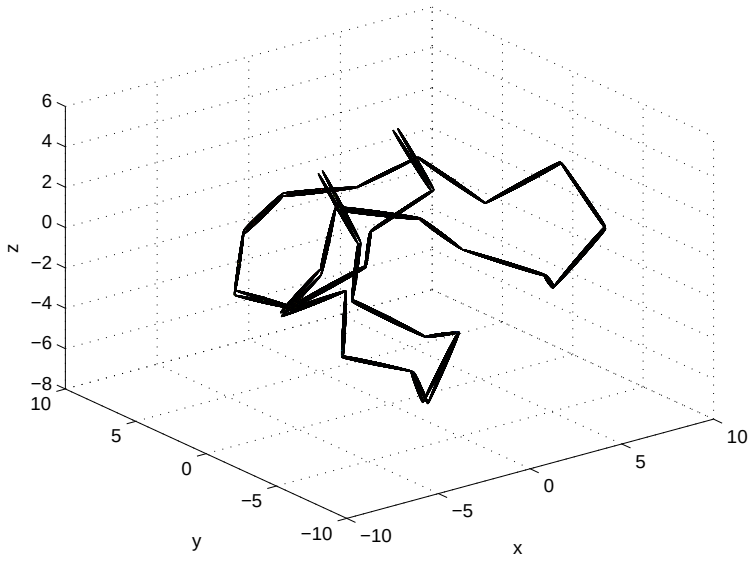


Figure 5.4: Pairwise global structural alignment for cyclotide Cter M (2LAM) is shown. Note the superposition of the 29 residues and connecting the segments.

global alignment of multiple structures. We used DALI as the tool to verify our results and it was observed that the alignment over multiple structures was similar to that of the global alignment over pairwise structures. A sample structural alignment result for 10 multiple structures of 2L24 with different initial conformations is shown in Figure 5.5. Note that the segments and the residues of all the 10 structures seem aligned as in the case of the pairwise alignment. It is also observed that the RMSD measure between each of the structures is minimal as in the case of the pairwise alignment. However, the last segment is misaligned. This is due to the lack of binding forces at that end of the structure, which gives it more freedom to undergo conformations.

5.7.2 Special Case of Locally Aligned Segments

In order to simulate the case of distantly related proteins with similarities over local segments, we considered similar protein structures and added multiple con-

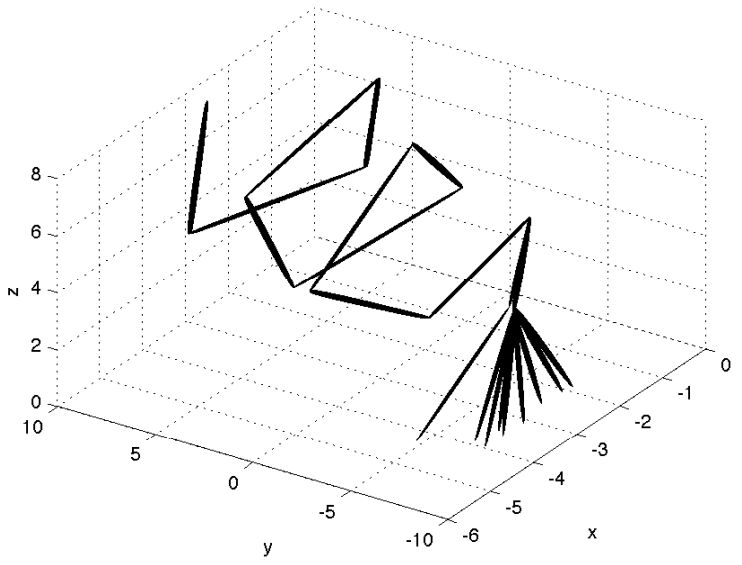


Figure 5.5: Multiple global structural alignment for (2L24) is shown. We considered over 10 structures with different initial conformations and superposed all of the structures together. Note the superposition of the 13 residues over each of the 10 structures and the segment connections.

formations over substructures in the protein while ensuring that there were locally similar segments in the protein. This case was simulated because in practice, there are not too many known instances of distantly related proteins which possess similar substructures for us to test our algorithm on. We present in Figures 5.7.2 and 5.7.2, five cases of local structural alignment including the cases of structures with an α-helix and an all-beta sheet.

5.7.3 Classification of Protein Structures

To perform classification we built a database with 50 structures that belonged to five different classes: two types of α helices (which we will refer to as α_1 and α_2 helices), π helix (an evolutionary variant of an α helix), β bridges and β strands. There were 10 structures in each of these classes. The ground truth for this classification is established while building the database by extracting structures from

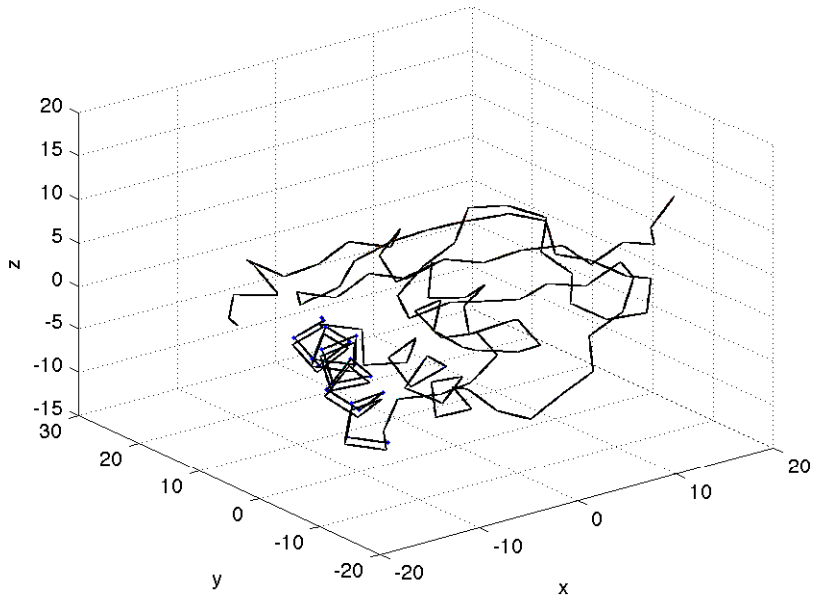


Figure 5.6: Local structural alignment example case of α-helix in 2L8K. Note the alignment of a helix of length 19 along the structure. The locally aligned structure is connected by blue dots and appears shifted for better view.

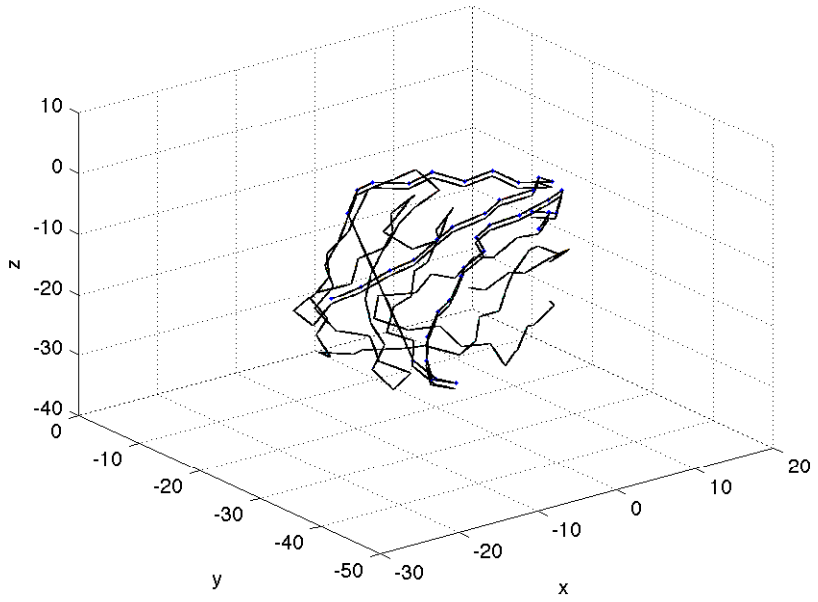


Figure 5.7: Local structural alignment example case of an all β-sheet structure with two sub-structures of lengths 22 and 19 aligned. The locally aligned structure is connected by blue dots and appears shifted for better view.

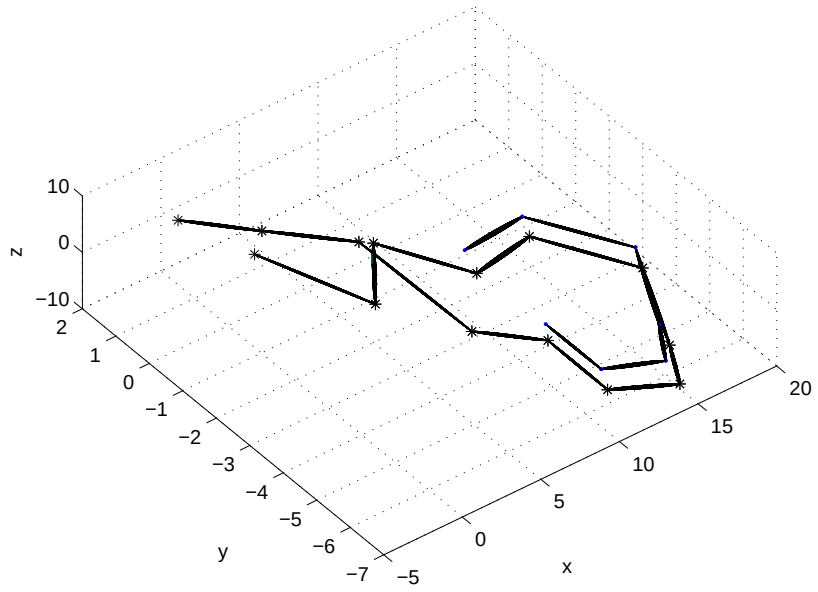


Figure 5.8: Local structural alignment case of local alignment in the β-Hairpin Peptidomimetic Inhibitor at the hairpin segment. The locally aligned structure is connected by blue dots and appears shifted for better view.

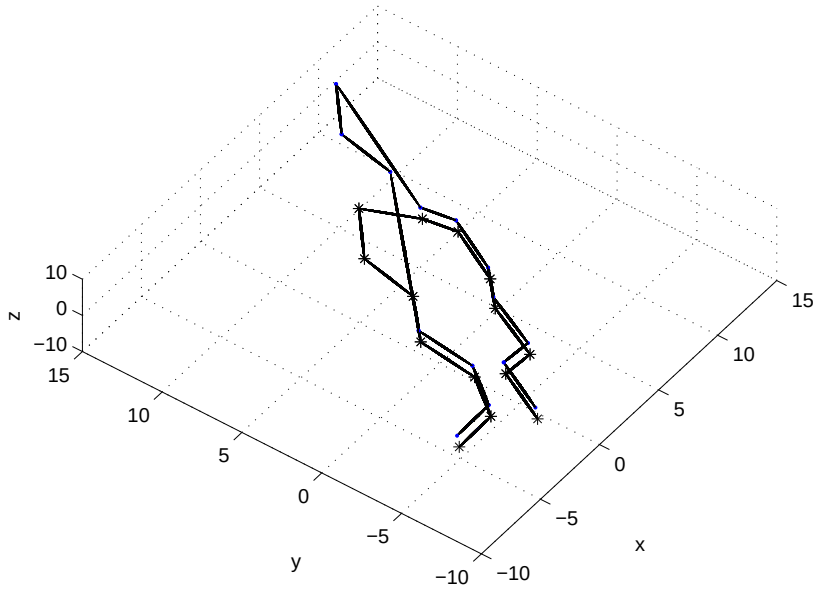
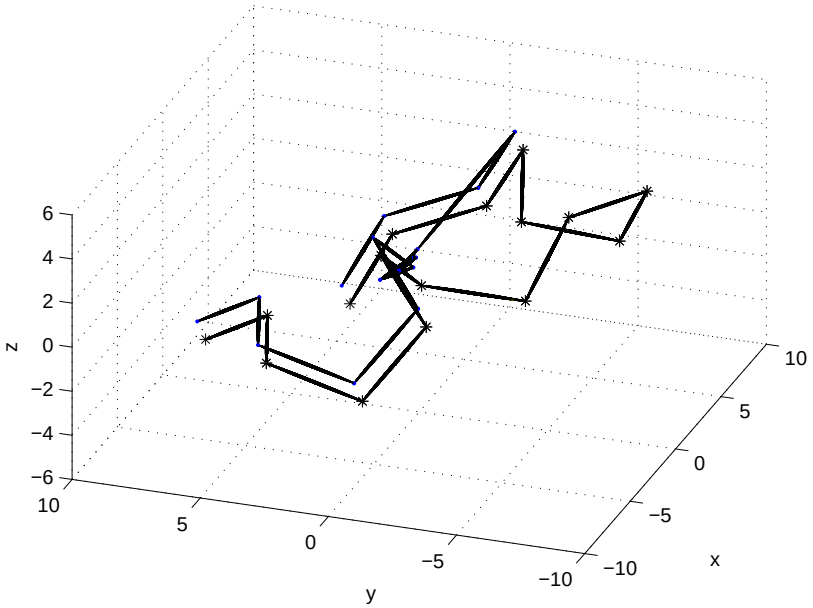


Figure 5.9: Local structural alignment example with a short misaligned segment in 1J4M. The locally aligned structure is connected by blue dots and appears shifted for better view.



W

Figure 5.10: Local structural alignment example with a short misaligned segment in 1KWE. The locally aligned structure is connected by blue dots and appears shifted for better view.

the PDB. We first extracted the directional descriptors for the reference structures and stored them for comparison with the directional descriptors of the unclassified structures. We then considered each structure for classification, and extracted the descriptors and compared them with those of the reference structures. This was performed over all five different classes. Following the classification of the structure into α class or β class, further classification was performed using the pairwise alignment algorithm. The distance measure used in the classification included both the directional descriptors at the first stage, and then the RMSD metric for pairwise alignment. Based on the classification results and the ground truth, a confusion matrix was constructed and is shown in Table 5.2.

Upon observing the classification performance, we noticed that the classification among the helices and the β classes is accurate and there is no misclassification between the two of them. For the helix subclasses, we observed that the

	Classification Result				
	α_1 helix	α_2 helix	π helix	β bridge	β strand
α_1 helix	**0.9**	0.2	0.1	0	0
α_2 helix	0.2	**0.95**	0.1	0	0
π helix	0.1	0.1	**0.9**	0	0
β bridge	0	0	0	**0.8**	0.2
β strand	0	0	0	0.1	**0.8**

Table 5.2: Confusion Matrix for Protein Structure Classification

classification achieved is greater than 80% for each of the three classes. For the β classes, we observed that there is greater than 80% classification achieved for both bridges and sheets. However there is misclassification amongst the helices and the β classes. For instance, we noticed that 20% of the structures from α_1 were misclassified in α_2 and 10% in π class. This can be attributed to the fact that there is a certain amount of similarity in helix subclasses and the β subclasses. Also, increasing the size of the training set will help enhance the classification performance and reduce the number of misclassifications in other classes. Note that there has been very few cases of misclassification in the actual class (ground truth), i.e., the structures have been classified under their respective classes with an accuracy of 80%.

Note that the CARPS process is performed in real-time and the computational speed of cross-correlations between two signals was increased by using FFTs.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusion

In this dissertation, we proposed two major types of algorithms, sequence alignments and structural alignment, based on the use of time-varying parametric waveforms to uniquely represent biological molecules and their properties [66, 88, 89, 145, 147]. Sequence alignment corresponds to querying primary biological sequences to find regions of similarity.Looking beyond the sequence based similarity for protein sequences, structural alignment tries to find similarities between two or more atoms based on the shape and three-dimensional (3-D) conformation of their secondary and tertiary structures.

The proposed sequence alignment technique for DNA is based on mapping DNA nucleobases to unique Gaussian waveforms, and then using the matching pursuit decomposition (MPD) algorithm to perform the query based on the mapped waveform's parameters in the time-frequency plane [88, 89]. For protein sequence alignment, we modified the representation to include a scale parameter. When sequence alignment does not yield good results but sequences still share common properties, we proposed to improve waveform-query based DNA and protein sequence alignment using the metaplectic transform; the five parameters of this transform allow additional sequence properties to be mapped and this to be used to increase alignment performance.

In particular, we proposed a robust WAVEQuery sequence alignment algorithm that is based on waveform mapping and on exploiting waveform transform parameters that propagate the waveform throughout the time-frequency plane. We investigated the matching pursuit decomposition that transforms the highly-localized Gaussian atoms in the time-frequency plane using time shifts, frequency

shifts and scale changes. These three transformation parameters are used to represent properties of the biological sequences, such as character type, position in the sequence, and correlation between two different characters. The robustness of the WAVEQuery algorithm can be increased by mapping more biological sequence properties; this will require transforms with higher dimensionality, such as the metaplectic transform that uses five transformation parameters to uniquely propagate a basic atom in the time-frequency plane.

We considered two important cases of querying: globalized querying and localized querying for DNA sequences. For localized sub-sequence querying, the WAVEQuery algorithm significantly outperformed the well-known BLAST algorithm when the queries had unknown lengths and repetitive database segments. The WAVEQuery algorithm also outperformed matched filtering type methods that also use waveform mapping, for these querying cases. The WAVEQuery query processing can be performed in real-time and no sequence indexing or preprocessing is required beforehand.

We also simulated the WAVEQuery algorithm for protein sequences by extending the MPD to include the scale change parameter. Note that the amino acid sequence constitutes the primary structure of the protein sequence, and similarity in this primary structure implies that the two protein structures have the same functional properties. However, the proteins also have secondary, tertiary and quaternary structures, which are not directly related to the primary structure. The secondary and tertiary structures are represented by 3-D shapes, and similarity in these shapes will imply that the two proteins have similar functions. Hence, it is important to measure the similarity in two proteins by obtaining the extent of functional similarity between the two proteins using their secondary structures. If two proteins are dissimilar in their primary structures, they may still be similar in their secondary and tertiary structures.

We first formulated the protein superposition as a shape matching problem, where the structures to be matched have six degrees of freedom. We then proposed a protein structural representation in 3-D space based on a 3-D time-domain Gaussian windowed chirp waveform. Our proposed chirp-based representation for protein structures (CARPS) uses Gaussian and chirp parameters to map structure translations and directional rotations, and we demonstrated that the α-Carbon based backbone of the protein structure is well depicted in the 3-D plane. As the CARPS is a linearly separable representation, it has the option of finding locally similar segments in two structures.

For the structural alignment approach, the CARPS was used to form the chirp-based algorithm for protein structures (CAPS). The new algorithm uses a cross-correlation based matched filter approach to identify similarities in two or more protein structures both globally and locally. The matched filter approach takes into account all structural conformations due to the six degrees of freedom in the structure, so that the CAPS bases its matching only on the structural shape and directionality. We first applied the CAPS to pairwise global structure alignments and compared its performance with the results obtained using the DALI pairwise alignment algorithm. Our results showed that the CAPS was capable of aligning structures with a great precision (with higher than 90% of residues being aligned) and the root mean-squared distance (RMSD) distance measure between the structures was less than 5Å in all the cases. We extended the CAPS to perform multiple protein structure alignment in order to construct a phylogenetic tree between the protein structures, and we demonstrated an example where multiple structures were aligned. We also extended the CAPS algorithm to local structural alignment, an important case most existing techniques do not consider. We applied our new technique to another important problem, the identification of structural classes, and showed that by using the directional descriptors for the

different structural classes and the pairwise alignment algorithm, we can successfully perform structural classification. We also mapped hydrophobicity, a physical atomic property, into the structural representation and showed that it can be used to improve protein structural classification.

6.2 Future Work

6.2.1 Sequence Alignment Algorithm

The alignment results from the WAVEQuery algorithm are currently represented and ranked in terms of the BLAST performance metrics such as Raw-Score, Bit-Score and E-value. However, the quality of the alignment can be studied using a cross-correlation based measure or a measure based on the number of residues obtained from the MPD alignment algorithm. Hence, it would be beneficial to have a metric for performance accuracy that better matches the alignment method in a particular database.

This algorithm can be modified as a querying algorithm and used beyond the sequence alignment application. Specifically, a possible representation from which we maybe able to extract sequence features is desired, and is based on the given sequence annotation.

6.2.2 Structural Alignment Algorithm

In the current structural alignment algorithm, we consider windowed chirp representations to map protein structures to waveforms. In the CARPS, we used a Gaussian window as it is highly concentrated in both time and frequency. A possible future modification is to investigate the use of other windows that have additional parameters to represent properties of protein structures.

We consider the parameter of hydrophobicity in order to be able to classify structures. This parameter can possibly be used when aligning sequences

and structures simultaneously; this implies an integration of the sequence and structural alignments. If we were to expand the classification set to all protein structural classes (> 250), then other properties could be incorporated and our algorithm could be used for classifying a much larger set of structural classes.

The computational complexity of the CAPS structural alignment technique is high as it involves 3-D correlation computations and a search for correlated residues all over the sequence. An optimized local-search technique will be best suited to reduce the number of computations involved in detecting locally aligned structures.

REFERENCES

[1] http://www.biotechnologyonline.gov.au/images/contentpages/helix.jpg.

[2] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson, *Molecular Biology of the Cell*. New York: Garland Publishing, 1994.

[3] X. Y. Zhang, F. Chen, Y. T. Zhang, S. C. Agner, M. Akay, Z. H. Lu, M. M. Y. Waye, and S. K.-W. Tsui, "Signal processing techniques in genomic engineering," *Proceedings of the IEEE*, vol. 90, no. 12, pp. 1822–1833, December 2002.

[4] E. R. Dougherty and A. Datta, "Genomic signal processing: Diagnosis and therapy," *IEEE Signal Processing Magazine*, pp. 107–112, January 2005.

[5] D. Anastassiou, "Genomic signal processing," *IEEE Signal Processing Magazine*, pp. 8–20, July 2001.

[6] E. R. Dougherty, A. Datta, and C. Sima, "Research issues in Genomic Signal Processing," *IEEE Signal Processing Magazine*, pp. 46–68, November 2005.

[7] A. A. Hanzel, "Signal processing challenges in the post-genomic era," in *Proceedings of IEEE Conference on Acoustic, Speech, and Signal Processing*, vol. 5, 2005, pp. 761–764.

[8] D. Schonfeld, J. Goutsias, I. Shmulevich, I. Tabus, and A. H. Tewfik, "Introduction to the Issue on Genomic and Proteomic Signal Processing," *IEEE Journal Selected Topics Signal Processing*, vol. 2, pp. 257–260, 2008.

[9] J. Chen, H. Li, K. Sun, and B. Kim, "How will Bioinformatics impact signal processing research?" *IEEE Signal Processing Magazine*, pp. 16–26, November 2003.

[10] R. M. Karp, "Mathematical challenges from genomics and molecular biology," *Notices of the American Mathematical Society*, vol. 49, no. 5, pp. 544–553, May 2002.

[11] Z. Aydin and Y. Altunbasak, "A signal processing application in genomic research: Protein secondary structure prediction," *IEEE Signal Processing Magazine*, pp. 128–131, July 2006.

[12] P. Cristea, "Conversion of nucleotide sequences into genomic signals," *J.Cell.Mol.Med*, vol. 6, no. 2, pp. 279–303, 2002.

[13] *Genomic Signal Processing and Statistics*. Hindawi Publishing Corporation, 2005,, ch. Representation and analysis of DNA sequences.

[14] I. Shmulevich and E. R. Dougherty, *Genomic Signal Processing (Princeton Series in Applied Mathematics)*. Princeton, NJ, USA: Princeton University Press, 2007.

[15] A. A. Tsonis, J. B. Elsner, and P. A. Tsonis, "Periodicity in DNA coding sequences: Implications in gene evolution," *Journal of Theoretical Biology*, vol. 151, pp. 323–331, 1991.

[16] S. Tiwari, S. Ramachandran, A. Bhattacharya, S. Bhattacharya, and R. Ramaswamy, "Prediction of probable genes and by Fourier analysis of genomic sequences," *Computer Applications in the Biosciences*, vol. 13, no. 3, pp. 263–270, 1997.

[17] H. Herzel and I. Grobe, "Correlations in DNA sequences: The role of protein coding segments," *The Americal Physical Society: Physical Review E*, vol. 55, no. 1, pp. 800–810, January 1997.

[18] L. Luo, W. Lee, L. Jia, F. Ji, and L. Tsai, "Statistical correlation of nucleotides in a DNA sequence," *The Americal Physical Society: Physical Review E*, vol. 58, no. 1, pp. 861–871, July 1998.

[19] N. Chakravarthy, A. Spanias, L. Iasemidis, and K. Tsakalis, "Autoregressive modeling and feature analysis of DNA sequences," *EURASIP Journal on Applied Signal Processing*, vol. 1, pp. 13–28, 2004.

[20] K. M. Bloch and G. R. Arce, "Time-frequency analysis of protein sequence data," in *Proceedings of the IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, 2001.

[21] C. J. Langmead and B. R. Donald, "Extracting structural information using time-frequency analysis of protein NMR data," in *Proceedings of the Fifth Annual International Conference on Computational Molecular Biology (RECOMB)*. ACM Press, 2001, pp. 164–175.

[22] D. Sussillo, A. Kundaje, and D. Anastassiou, "Spectrogram analysis of genomes," *EURASIP Journal on Applied Signal Processing*, vol. 1, pp. 29–42, 2004.

[23] B. D. Silverman and R. Linkser, "A measure of DNA periodicity," *Journal of Theoretical Biology*, vol. 118, pp. 295–300, 1986.

[24] V. Afreixo, P. J. S. G. Ferreira, and D. Santos, "Fourier analysis of symbolic data: A brief review," *Digital Signal Processing*, vol. 14, pp. 523–530, 2004.

[25] A. Fukushimaa, T. Ikemura, M. Kinouchie, T. Oshimae, Y. Kudod, H. Morig, and S. Kanaya, "Periodicity in Prokaryotic and Eukaryotic genomes identified by power spectrum analysis," *Gene*, vol. 300, pp. 203–211, 2002.

[26] G. Dodin, P. Vandergheynst, P. Levoir, C. Cordier, and L. Marcourt, "Fourier and wavelet transform analysis, a tool for visualizing regular patterns in DNA sequences," *Journal on Theoretical Biology*, pp. 323–326, 2000.

[27] J. A. Berger, S. K. Mitra, and J. Astola, "Power spectrum analysis for DNA sequences," *IEEE Transactions on Signal Processing*, vol. 2, pp. 29–32, 2003.

[28] S. Datta, A. Asif, and H. Wang, "Prediction of protein coding regions in DNA Sequences using Fourier spectral characteristics," in *Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering (ISMSE)*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 160–163.

[29] S. Datta and A. Asif, "DFT based DNA splicing algorithms for prediction of protein coding regions," in *Proceedings of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, vol. 1, November 2004, pp. 45–49.

[30] ——, "A fast DFT based gene prediction algorithm for identification of protein coding regions," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, March 2005, pp. 653–656.

[31] S. Bagchi and S. K. Mitra, *The Nonuniform discrete Fourier Transform and its Applications in Signal Processing*. Boston, MA: Kluwer Academic Publishers, 1999.

[32] D. Anastassiou, "Frequency-domain analysis of biomolecular sequences," *Bioinformatics*, vol. 16, no. 12, pp. 1073–1081, 2000.

[33] ——, "DSP in genomics: Processing and frequency-domain analysis of Character strings," in *Proceedings of the IEEE Conference on Acoustic, Speech, and Signal Processing*, 2001, pp. 1053–1056.

[34] D. Sussillo, A. Kundaje, and D. Anastassiou, "Spectral analysis of genome," *Eurasip Journal of Applied Signal Processing*, no. 4, December 2003.

[35] Q. Fang and I. Cosic, "Can short time Fourier transform detect the localized latent periodicity of a protein sequences?" in *IEEE Engineering in Medicine and Biology Society Asian-Pacific Conference on Biomedical Engineering*, October 2003, pp. 66–67.

[36] C. Hwang and I. Sohn, "Analyzing exon structure with PCA and ICA of short-time Fourier transform," *Proceedings of the Autumn Conference, Korean Statistical Society*, 2004.

[37] P. P. Vaidyanathan, "Genomics and proteomics: A signal processors tour," *IEEE Circuits and Systems Magazine*, pp. 6–29, 2004.

[38] P. P. Vaidyanathan and B.-J. Yoon, "The role of signal-processing concepts in genomics and proteomics," *Journal of the Franklin Institute, special issue on Genomics*, vol. 341, pp. 111–135.

[39] P. P. Vaidyanathan, "Signal processing problems in genomics," International Symposium on Circuits and Systems Plenary, May 2004.

[40] P. P. Vaidyanathan and B.-J. Yoon, "Digital filters for gene prediction applications," in *Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, vol. 1, November 2002, pp. 306–310.

[41] A. A. Tsonis, P. Kumar, J. B. Elsner, and P. A. Tsonis, "Wavelet analysis of DNA sequences," *The American Physical Society: Physical Review E*, vol. 53, no. 2, pp. 1828–1838, February 1996.

[42] M. Altaiski, O. Mornev, and R. Polozov, "Wavelet analysis of DNA sequences," *Genetic Analysis: Biomolecular Engineering*, pp. 165–168, December 1996.

[43] A. Arneodo, Y. D'Aubenton-Carafa, B. Audit, E. Bacry, J.F.Muzy, and C. Thermes, "What can we learn with wavelets about DNA sequences?" *Physica A 249*, pp. 439–448, 1998.

[44] M. Dipperstein, "DNA sequence databases." [Online]. Available: http://michael.dipperstein.com/dna/DNApaper.html

[45] L. Rowen, G. Mahairas, and L. Hood, "Sequencing the human genome," *Science*, vol. 278, no. 5338, pp. 605–607, 1997.

[46] http://www.wwpdb.org/documentation/format32/v3.2.html.

[47] http://www.structuralgenomics.org/.

[48] S. Lorenzen, C. Gille, R. Preissner, and C. Frmmel, "Inverse sequence similarity of proteins does not imply structural similarity," *Federation of European Biochemical Societies Letters*, vol. 545, pp. 105–109, 2003.

[49] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3397–3415, December 1993.

[50] P. Daras, D. Zarpalas, D. Tzovaras, and M. G. Strintzis, "Shape Matching using the 3D Radon Transform," in *International Symposium on 3D Data Processing Visualization and Transmission,*, 2004, pp. 953–960.

[51] G. M. Maggiora, D. C. Rohrer, and J. Mestres, "Comparing protein structures: A Gaussian-based approach to the three-dimensional structural similarity of proteins," *Journal of Molecular Graphics and Modelling*, vol. 19, no. 1, pp. 168–178, 2001.

[52] D. Eisenberg, R. M. Weiss, T. C. Terwilliger, and W. Wilcox, "Hydrophobic moments and protein structure," *Faraday Symposia of the Chemical Society*, vol. 17, pp. 109–120, 1982.

[53] http://www.ncbi.nlm.nih.gov/genbank/.

[54] http://www.ebi.ac.uk/embl/.

[55] http://www.ddbj.nig.ac.jp/.

[56] http://www.ncbi.nlm.nih.gov/sites/gquery.

[57] *Bioinformatics: Databases, Tools and Algorithms.* Oxford University Press, 2007, ch. 1–3.

[58] D. G. George, W. C. Barker, H.-W. Mewes, F. Pfeiffer, and A. Tsugita, "The pir-international protein sequence database," *Nucleic Acids Research*, vol. 24, no. 1, pp. 17–20, 1996.

[59] http://www.ebi.ac.uk/uniprot/.

[60] http://pir.georgetown.edu/.

[61] W. Wang and D. H. Johnson, "Computing linear transforms of symbolic signals," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 628–634, March 2002.

[62] J. G. Proakis, *Digital Communications*, 4th ed. McGraw-Hill International Edition, 2001.

[63] S. Rajasekaran, H. Nick, P. M. Pardalos, S. Sahni, and G. Shaw, "Efficient algorithms for local alignment search," *Journal of Combinatorial Optimization*, vol. 5, pp. 117–124, 2001.

[64] E. Cheever, D. Searls, W. Karunaratne, and G. Overton, "Using signal processing techniques for DNA sequence comparison," in *Northeast Bioengineering Conference*, 27-28 March 1989, pp. 173–174.

[65] G. D. Avenio, M. Grigioni, G. Orefici, and R. Creti, "SWIFT (sequence-wide investigation with Fourier transform): A software tool for identifying proteins of a given class from the unannotated genome sequence," *Bioinformatics*, vol. 21, no. 13, pp. 2943–2949, 2005.

[66] L. Ravichandran, A. Papandreou-Suppappola, A. Spanias, Z. Lacroix, and C. Legendre, "Waveform Mapping based Alignment methods for DNA Sequences," in *Proceedings of the SenSIP Workshop*, Sedona, AZ, 2008.

[67] F. Harris, "Orthogonal frequency division multiplexing (OFDM)," in *Vehicular Technology Conference*, 2004.

[68] A. Papandreou-Suppappola, "Time-varying processing: Tutorial on principles and practice," in *Applications in Time-Frequency Signal Processing*, A. Papandreou-Suppappola, Ed. CRC Press, 2002, pp. 1–84.

[69] L. Cohen, *Time-frequency analysis*. Prentice-Hall, 1995.

[70] S. Ben-Dor and I. Orr, "Sequence comparison: Pairwise alignment," Weizmann Institute of Science, Tech. Rep., 2005. [Online]. Available: http://bioportal.weizmann.ac.il/course/introbioinfo/lecture5/pairwise09.pdf

[71] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, pp. 443–453, 1970.

[72] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *Journal of Biomolecular Techniques*, vol. 147, pp. 195–197, 1981.

[73] S. F. Altschul, W. .Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic Local Alignment Search Tool," *Journal of Molecular Biology*, vol. 215, pp. 403–410, October 1990.

[74] J. Felsenstein and R. K. S. Sawyer, "An efficient method for matching nucleic acid sequences," *Nucleic Acids Research*, vol. 19, 1982.

[75] S. Rajasekaran, X. Jin, and J. L. Spouge, "The efficient computation of position-specific match scores with the fast Fourier transform," *Journal Of Computational Biology*, vol. 9, pp. 23–33, 2002.

[76] A. L. Rockwood, D. K. Crockett, J. R. Oliphant, and K. S. J. Elenitoba-Johnson, "Sequence alignment by cross-correlation," *Journal of Biomolecular Techniques*, vol. 16, pp. 453–458, 2005.

[77] K. Katoh, K. Misawa, K. chi Kuma, and T. Miyata, "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform," *Nucleic Acids Research - Oxford University Press*, vol. 30, no. 14, pp. 3059–3066, 2002.

[78] A. K. Brodzik, "A comparative study of cross-correlation methods for alignment of DNA sequences containing repetitive patterns," in *Proceedings of the 13th European Signal Processing Conference*, vol. 2, 2005, pp. 2–5.

[79] D. Gilbert, "Sequence comparison." [Online]. Available: http://www.brc.dcs.gla.ac.uk/~drg/courses/bioinformatics_mscIT/slides/slides3/sld001.htm

[80] W. J. Kent, "BLAT–the BLAST-like alignment tool," *Genome Research*, vol. 12, no. 4, pp. 656–664, April 2002.

[81] C. Meek, J. M. Patel, and S. Kasetty, "OASIS: An Online and Accurate Technique for Local-alignment Searches on biological sequences," in *Proceedings of the 29th International Conference on Very Large Data Bases (VLDB)*, 2003, pp. 910–921.

[82] T. W. Lam, W. K. Sung, S. L. Tam, C. K. Wong, and S. M. Yiu, "Compressed indexing and local alignment of DNA," *Bioinformatics*, vol. 24, no. 6, pp. 791–797, 2008.

[83] E. Giladi, M. Walker, J. Wang, and W. Volkmuth, "SST: An algorithm for searching sequence databases in time proportional to the logarithm of the database size," Stanford InfoLab, Technical Report 2000-3, 2000. [Online]. Available: http://ilpubs.stanford.edu:8090/460/

[84] S. Burkhardt, A. Crauser, P. Ferragina, H. P. Lenhof, E. Rivals, and M. Vingron, "q-gram based database searching using a suffix array (QUASAR)," in *International Conference on Computational Molecular Biology*, 1999, pp. 77–83.

[85] C. Li, B. Wang, and X. Yang, "VGRAM: Improving performance of approximate queries on string collections using variable-length grams," in *International Conference on Very Large Data Bases*, Vienna, Austria, 2007, pp. 303–314.

[86] J. Fang, I. Cosic, and C. de Trad, "Protein sequence comparison based on the wavelet transform approach," *Protein Engineering*, vol. 15, pp. 193–203, 2002.

[87] A. Brodzik, "Phase-only filtering for the masses (of DNA data): a new approach to sequence alignment," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2456 – 2466, June 2006.

[88] L. Ravichandran, A. Papandreou-Suppappola, A. Spanias, Z. Lacroix, and C. Legendre, "DNA sequence alignment using matching pursuit decompositions," in *IEEE International Workshop on Genomic Signal Processing and Statistics*, June 2008, pp. 1–7.

[89] ——, "Time-frequency based biological sequence querying," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Dallas, TX, March 2010.

[90] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, pp. 3389–3402, 1997.

[91] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks," *Proceedings of the National Academy of Sciences*, vol. 89, pp. 10 915–10 919, November 1992.

[92] S. R. Eddy, "Where did the BLOSUM62 alignment score matrix come from?" *Nature Biotechnology*, vol. 22, no. 8, pp. 1035–1036, August 2004.

[93] R. G. Baraniuk and D. L. Jones, "New signal-space orthonormal bases via the metaplectic transform," in *IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*, October 1992, pp. 339–342.

[94] M. García-Bullé, W. Lassner, and K. B. Wolf, "The metaplectic group within the Heisenberg-Weyl ring," *Journal of Mathematical Physics*, vol. 27, no. 1, pp. 29–36, 1986.

[95] "Internation union of pure and applied chemistry compendium of terminology," http://goldbook.iupac.org/index.html, 2005.

[96] A. Godzik, "The structural alignment between two proteins: is there a unique answer?" *Protein Science*, vol. 5, no. 7, pp. 1325–38, 1996.

[97] I. Eidhammer, I. Jonassen, and W. Taylor, "Structure comparison and structure patterns," *Journal of Computational Biology*, vol. 7, pp. 685–716, 1999.

[98] C. C. Huang, W. R. Novak, P. C. Babbitt, A. I. Jewett, T. E. Ferrin, and T. E. Klein, "Integrated tools," in *Pacific Symposium on Biocomputing*, 2000, pp. 227–238.

[99] C. Lemmen, M. Zimmerman, and T. Lengauer, "Multiple molecular superpositioning as an effectivetool for virtual database screening," *Perspectives in Drug Discovery and Design*, vol. 20, pp. 43–62, 2000.

[100] C. Lemmen and T. Lengauer, "Computational methods for the structural alignment of molecules," *Journal of Computer-Aided Molecular Design*, vol. 14, pp. 215–232, 2000.

[101] L. Holm and C. Sander, "Protein structure comparison by alignment of distance matrices," *Journal of Molecular Biology*, vol. 233, no. 1, pp. 123–138, 1993.

[102] http://www2.ebi.ac.uk/dali/.

[103] L. Holm, S. Kaariainen, P. Rosenstrom, and A. Schenkel, "Searching protein structure databases with Dalilite v.3," *Bioinformatics*, vol. 24, no. 23, pp. 2780–2781, December 2008.

[104] http://www.ncbi.nlm.nih.gov/Structure/VAST/vastsearch.html.

[105] J. F. Gibrat, T. Madej, and S. H. Bryant, "Surprising similarities in structure comparison," *Current Opinion in Structural Biology*, vol. 6, no. 3, pp. 377 – 385, 1996.

[106] http://cl.sdsc.edu/ce.html.

[107] I. N. Shindyalov and P. E. Bourne, "Protein structure alignment by incremental combinatorial extension (CE) of the optimal path." *Protein Engineering*, vol. 11, no. 9, pp. 739–747, September 1998.

[108] T. R. Schneider, "A genetic algorithm for the identification of conformationally invariant regions in protein molecules." *Acta Crystallogr D Biol Crystallogr*, 2002.

[109] A. R. Ortiz, C. E. Strauss, and O. Olmea, "MAMMOTH (MAtching Molecular models Obtained from THeory): An automated method for model comparison," *Protein Science*, 2002.

[110] W. R. Taylor, T. P. Flores, and C. A. Orengo, "Multiple protein structure alignment," *Protein Science*, 1994.

[111] O. O'Sullivan, K. Suhre, C. Abergel, D. G. Higgins, and C. Notredame, "3DCoffee: Combining protein sequences and structures within multiple sequence alignments," *Journal of Molecular Biology*, vol. 340, pp. 385–295, 2004.

[112] A. P. Singh and D. L. Brutlag, "Hierarchical protein structure superposition using both secondary structure and atomic representations," in *Proceedings of the International Conference on Intelligent System Molecular Biology*, 1997.

[113] S. A. Aghili, D. Agrawal, and A. E. Abbadi, "PADS: Protein structure alignment using directional shape signatures," in *Proceedings of the 9th International Conference on Database Systems for Advanced Applications (DASFAA)*, 2004.

[114] T. Akutsu, K. Onizuka, and M. Ishikawa, "New hashing techniques for three-dimensional protein structures," in *Proceedings of the Genome Informatics Workshop*, Yokohama, Japan, 1994.

[115] ——, "New hashing techniques and their application to a protein structure database system," in *Proceedings of the 28th Hawaii International Conference on System Sciences*, 1995, p. 197.

[116] B. Albrecht, G. H. Grant, and W. G. Richards, "Evaluation of structural similarity based on reduced dimensionality representations of protein structure," vol. 17, no. 5, 2004.

[117] J. D. Szustakowski and Z. Weng, "Protein structure alignment using a genetic algorithm," *Proteins: Structure, Function, and Genetics*, vol. 38, no. 4, 2000.

[118] S. Park and M. Yamamura, "FROG (fitted rotation and orientation of protein structure by means of real-coded genetic algorithm) : Asynchronous parallelizing for protein structure-based comparison on the basis of geometrical similarity," *Genome Informatics*, vol. 13, pp. 344–345, 2002.

[119] A. Bogan-Marta, "A new statistical measure of protein similarity based on language modeling," in *Proceedings of the Genomic Signal Processing and Statistics*, Newport, Rhode Island.

[120] S. Bhattacharya, C. Bhattacharyya, and N. Chandra, "Structural alignment based kernels for protein structure classification," in *Proceedings of the 24th International Conference on Machine Learning*, 2007, pp. 73–80.

[121] M. Fujita, H. Toh, and M. Kanehisa, "Protein sequence-structure alignment using 3D-HMM," in *Proceedings of the Fourth International Workshop on Bioinformatics and Systems Biology*, Kyoto, Japan, 2004, pp. 7–8.

[122] Z. Huang and X. Zhou, "High dimensional indexing for protein structure matching using bowties," in *APBC*, 2005, pp. 21–30.

[123] T. D. Wu, T. Hastie, S. C. Schmidler, and D. L. Brutlag, "Regression analysis of multiple protein structures," in *Proceedings of the Second Annual International Conference on Computational Molecular Biology*. New York, NY, USA: ACM, 1998, pp. 276–284.

[124] J. Mestres, D. Rohrer, and G. Maggiora, "MIMIC: A molecular-field matching program. exploiting the applicability of molecular similarity approaches," *Journal of Computational Chemistry*, vol. 18, pp. 934–954, 1997.

[125] J. Mestres, "Gaussian-based Alignment of Protein Structures: Deriving a consensus superposition when alternative solutions exist," *Journal of Molecular Modeling*, vol. 6, no. 7–8, pp. 539–549, August 2000.

[126] E. Katchalski-Katzir, I. Shariv, M. Eisenstein, A. A. Friesem, C. Aflalo, and I. A. Vakser, "Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques." *Proceedings of the National Academy of Science*, vol. 89, no. 6, pp. 2195–2199, March 1992.

[127] J. W. M. Nissink, M. L. Verdonk, J. Kroon, T. Mietzner, and G. Klebe, "Superposition of molecules: Electron density fitting by application of Fourier transforms," *Journal of Computational Chemistry*, pp. 638–645, 1997.

[128] C. Lemmen, C. Hiller, and T. Lengauer, "Rigfit: A new approach to superimposing ligand molecules," *Journal of Computer-Aided Molecular Design*, vol. 12, no. 5, pp. 491–502, 1998.

[129] P. Chacn and W. Wriggers, "Multi-resolution contour-based fitting of macromolecular structures," *Journal of Molecular Biology*, vol. 317, no. 3, pp. 375–384, March 2002.

[130] W. Wriggers and P. Chacn, "Modeling tricks and fitting techniques for multiresolution structures," *Structure*, vol. 9, no. 9, pp. 779–788, September 2001.

[131] J. Kovacs, P. Chacon, Y. Cong, E. Metwally, and W. Wriggers, "Fast rotational matching of rigid bodies by fast Fourier transform acceleration of five degrees of freedom," *Acta crystallographica. Section D, Biological crystallography*, vol. 59, pp. 1371–1376, 2003.

[132] J. Kovacs and W. Wriggers, "Fast rotational matching," *Acta crystallographica. Section D, Biological crystallography*, vol. 58, pp. 1282–1286, 2002.

[133] D. W. Ritchie and G. J. L. Kemp, "Protein docking using spherical polar fourier correlations," *Proteins*, vol. 39, pp. 178–194, 1999.

[134] D. W. Ritchie, D. Kozakov, and S. Vajda, "Accelerating and focusing proteinprotein docking correlations using multi-dimensional rotational FFT generating functions," *Bioinformatics: Structural Bioinformatics*, vol. 24, no. 17, pp. 1865–1873, 2008.

[135] D. Zarpalas, P. Daras, A. Axenopoulos, D. Tzovaras, and M. G. Strintzis, "3D model search and retrieval using the spherical trace transform," *EURASIP Journal of Applied Signal Processing*, 2007.

[136] P. Daras, D. Zarpalas, A. Axenopoulos, D. Tzovaras, and M. G. Strintzis, "Three-dimensional shape-structure comparison method for protein classification," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 3, no. 3, pp. 193–207, 2006.

[137] K. L. Damm and H. A. Carlson, "Gaussian-weighted RMSD superposition of proteins: A structural comparison for flexible proteins and predicted protein structures," *Biophysical Journal*, vol. 90, pp. 4558–4573, 2006.

[138] T. Pham and B. S. Shim, "A cepstral distortion measure for protein comparison and identification," in *Proceedings of International Conference on Machine Learning and Cybernetics*, vol. 9, August 2005, pp. 5609–5614.

[139] T. Pham, "LPC cepstral distortion measure for protein sequence comparison," *IEEE Transactions on NanoBioscience*, vol. 5, no. 2, pp. 83–88, June 2006.

[140] D. Xu, H. Li, and T. Gu, "Protein structure superposition by curve moment invariants and iterative closest point," in *The 1st International Conference on Bioinformatics and Biomedical Engineering*, July 2007, pp. 25–28.

[141] D. A. Cosgrovea, D. M. Bayadab, and A. P. Johnson, "A novel method of aligning molecules by local surface shape similarity," *Journal of Computer-Aided Molecular Design*, vol. 14, no. 6, August 2000.

[142] E. Paquet and H. L. Viktor, "Exploring protein architecture using 3D shape-based signatures," in *Proceedings of the 29th Annual International Confer-*

ence of the IEEE Engineering in Medicine and Biology Society (EMBS), Aug. 22–26, 2007, pp. 1204–1208.

[143] D. L. Theobald and D. S. Wuttke, "THESEUS: Maximum likelihood superpositioning and analysis of macromolecular structures," *Bioinformatics*, vol. 22, no. 17, pp. 2171–2172, 2006.

[144] ——, "Accurate structural correlations from maximum likelihood superpositions," *PLoS Computational Biology*, vol. 4, no. 2, 2008.

[145] L. Ravichandran, A. Papandreou-Suppappola, A. Spanias, Z. Lacroix, and C. Legendre, "Waveform Mapping and Time-Frequency Processing of DNA and Protein sequences," *IEEE Transactions of Signal Processing*, 2011.

[146] B. W. Matthews, *Hydrophobic Interactions in Proteins.* John Wiley & Sons, Ltd, 2001.

[147] L. Ravichandran, A. Papandreou-Suppappola, A. Spanias, and Z. Lacroix, "Multiple protein structure alignment using time-frequency processing techniques," in *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, November 2010, pp. 94–97.