FILTERED DISTANCE MATRIX FOR CONSTRUCTING HIGH-THROUGHPUT MULTIPLE SEQUENCE ALIGNMENT ON PROTEIN DATA

by

MUHANNAD ABDUL-QADER MOHAMMAD ABU-HASHEM

Thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

February 2015

ACKNOWLEDGEMENT

IN THE NAME OF
ALLAH
THE ALL-COMPASSIONATE, ALL-MERCIFUL

“All the praises and thanks be to Allah, the Lord of the worlds, the most Beneficent, the most Merciful”

(Al Fatiha: 1-3)

I would like to express my gratitude and appreciation to my main supervisor, Assoc. Prof. Dr. Nur’Aini Abdul Rashid, for her constant encouragement and guidance. Her support, motivation and magical comments have given me the strength that enables me to move through this challenge. I am also grateful to my co-supervisor, Prof. Dr. Rosni Abdullah, for her comments and guidance.

Also, I would like to express my gratitude to Parallel and Distributed Computer Center, School of Computer Sciences, Institute of Postgraduate Studies, and Universiti sians Malaysia for providing me with extensive resources and facilities.

My special thanks to my parents, without them I won’t be here at this stage of my life. Their guidance, moral support, encouragement and prayers always enlighten my way. Huge credit goes to my brothers, sisters, wife, nephews, nieces and friends for their love, support, and patience.

TABLE OF CONTENTS

LISTS OF TABLES

LISTS OF FIGURES

LISTS OF ALGORITHMS

LIST OF ABBREVIATIONS

ABC	Application for Browsing Constraints
AC	Ant Colony
ALN3NN	Three-Way Neighbor-Net Based Alignment
ARMiCoRe	Alignment Refinement by Mining Coupled Residues
BaliBASE	Benchmark Alignment DataBASE
BLOSUM	BLOck SUbstitution Matrix
BPSO	Binary Particle Swarm Optimization
CPSO	Chaotic PSO
CS	Column Score
D&C	Divide & Conquer
DIALIGN	DIagonal-ALIGNment
DNA	DeoxyriboNucleic Acid
DP	Dynamic Programming
EA	Evolutionary Computing
FDM-MSA	Filtered Distance Matrix for MSA
FSA	Fast Statistical Alignment
GA	Genetic Algorithm
HMM	Hidden Markov Model
H-NGH	Hashing-N-Gram-Hirschberg
HT-NGH	HashTable-N-Gram-Hirschberg

IRMBASE	Implanted Rose Motifs Base
KB-MSA	Knowledge-Based Multiple-Sequence Alignment
LE	Log-Expectation
MAUSA	Multiple Alignment Using Simulated Annealing
MB	Mega Byte
MISHIMA	Method for Inferring Sequence History In terms of Multiple Alignment
MSA	Multiple Sequence Alignment
MSASA	Multiple Sequence Alignment using Simulated Annealing
MUSCLE	MUltiple Sequence Comparison by Log-Expectation
NGH	N-Gram-Hirschberg
NGSW	N-Gram-Smith-Waterman
NJ	Neighbor Joining
NMR	Nuclear Magnetic Resonance
NP complete	Non-deterministic Polynomial-time complete
NP hard	Non-deterministic Polynomial-time hard
NSGA-II	Non Dominated Sorting Genetic Algorithm
OTUs	Operational Taxonomic Units
PAM	Point Accepted Mutation
PCMA	Profile Consistency Multiple Sequence Alignment
POA	Partial Order Alignment
ProbCons	Probabilistic Consistency

PSO	Particle Swarm Optimization
PSP	Protein Structure Predicting
RDPSO	Random Drift Particle Swarm Optimization
RNA	RiboNucleic Acid
SA	Simulated Annealing
SABmark	Sequence Alignment Benchmark
SCOP	Structural Classification of Proteins
SKA	Sliding window and Keyword tree based Alignment algorithm
SNN	Shared Nearest Neighbor
SP	Sum of Pairs
SPS	Sum of Pairs Score
SSAHA	Sequence Search and Alignment by Hashing Algorithm
TC	Total Column
T-coffee	Tree-based Consistency Objective Function for alignment Evaluation
TS	Tabu Search
UPGMA	Unweighted Pair Group Method with Arithmatic Mean

MATRIK JARAK YANG DITAPIS UNTUK PEMBINAAN URUTAN PENJAJARAN BERGANDA PENGHASILAN TINGGI KE ATAS DATA PROTEIN

ABSTRAK

Urutan Penjajaran Berganda (MSA) adalah satu proses yang penting dalam biologi pengkomputeran dan bioinformatik. MSA optima adalah masalah NP-keras sementara membina penjajaran optimum menggunakan pengaturcaraan dinamik merupakan masalah NP lengkap. Walaupun sebilangan besar algoritma telah dicadangkan untuk MSA, namun menghasilkan MSA yang cekap dengan ketepatan yang tinggi masih merupakan cabaran yang besar. Pengkomputeran MSA untuk set data yang besar boleh mengambil masa selama berjam-jam. Kaedah penjajaran progresif telah digunakan secara meluas untuk membina MSA. Ia menggunakan pepohon panduan sebagai input untuk memandu proses penjajaran. Penjajaran pasangan memainkan peranan yang penting dalam membina matrik-matrik jarak yang perlu untuk membina pepohon panduan. Matrik jarak yang teguh membawa kepada MSA yang lebih baik. Dalam kajian ini, kaedah HashTable-N-Gram-Hirschberg (HT-NGH) dan Jarak Matrik yang Ditapis untuk MSA (FDM-MSA) untuk membina MSA dibentangkan. Kaedah HT-NGH adalah lanjutan kepada dua kaedah penjajaran pasangan iaitu N-Gram-Hirschberg (NGH) dan hashing-N-Gram-Hirschberg (H-NGH); ia menggunakan keupayaan jadual cincangan untuk meningkatkan fasa transformasi. Kaedah FDM-MSA dibahagikan kepada empat fasa: membina matrik jarak, membina sistem penapisan, membina pepohon panduan, dan membina MSA. Kaedah HT-NGH digunakan untuk membina matrik jarak. Dua pengesan jujukan terlibat dalam membina sistem penapisan: pengesan Multi-domain dan pengesan titik terpencil. Selepas menapis matrik jarak, Penyertaan Jiran (NJ) dan kaedah penjajaran progresif telah digunakan untuk membina MSA. Eksperimen menunjukkan bahawa algoritma HT-NGH melebihi performa kaedah NGH dan kaedah H-NGH kaedah sebanyak 60% dan 30% dari segi masa dan prestasinya melebihi daripada prestasi algoritma H-NGH dari segi ketepatan dengan menghasilkan keputusan yang sama seperti algoritma NGH. Tambahan pula, eksperimen menggunakan algoritma FDM-MSA menunjukkan prestasi yang lebih baik dalam kedua-dua segi; masa dan ketepatan. Algoritma FDM-MSA mendapat prestasi masa yang terbaik berbanding semua kaedah yang setara dalam kebanyakan dataset (jumlah masa dalam saat ke atas Balibase =1087, IRMbase=163, SABMARK=81, dan Oxbench=83), berserta mendapat jumlah tertinggi dalam Jumlah Skor Pasangan (SPS) dengan menggunakan dataset RV2 (skor SP = 0.9437) Balibase dan kedua terbaik secara purata bagi perincian Jumlah Lajur (TC).

FILTERED DISTANCE MATRIX FOR CONSTRUCTING HIGH-THROUGHPUT MULTIPLE SEQUENCE ALIGNMENT ON PROTEIN DATA

ABSTRACT

Multiple sequence alignment (MSA) is a significant process in computational biology and bioinformatics. Optimal MSA is an NP-hard problem, while building optimal alignment using dynamic programming is an NP complete problem. Although numerous algorithms have been proposed for MSA, producing an efficient MSA with high accuracy remains a huge challenge. Computing the MSA of a large dataset may take hours. Progressive alignment method is broadly used for constructing MSA. It uses guide trees as an input to guide the alignment process. Pair-wise alignment plays a significant role in building the distance matrices where distance matrices are necessary for building the guide trees. Robust distance matrix leads to better MSA. In this research, HashTable-N-Gram-Hirschberg (HT-NGH) and Filtered Distance Matrix for building MSA (FDM-MSA) to construct MSA methods are presented. HT-NGH is an extension to the N-Gram-Hirschberg (NGH) and Hashing-N-Gram-Hirschberg (H-NGH) pair-wise alignment methods; it uses the hash table capabilities to enhance the transformation phase. FDM-MSA is divided into four phases: constructing the distance matrix, building the filtering system, building the guide tree, and constructing the MSA. HT-NGH is used to build the distance matrix. Two sequence detectors are involved in building the filtering system: multi-domain detector and outlier detector. After filtering the distance matrix, Neighbor Joining and progressive alignment methods are employed to construct the MSA. The experiments show that the HT-NGH algorithm outperforms the NGH and H-NGH methods by 60% and 30% respectively in terms of time, while it outperforms the H-NGH algorithm in terms of accuracy by producing the same results as the NGH algorithm. On the other hand, experiments on the FDM-MSA algorithm show improved performance in both terms; time and accuracy. FDM-MSA algorithm obtains the best time performance over all competitive methods in most datasets (total time in seconds on Balibase =1087, IRMbase=163, SABMARK=81, and Oxbench=83), as well as obtains the highest Sum-of-Pairs Score on RV2 (SP score = 0.9437) dataset of BAlibase dataset and the second best Total Column score on average.

CHAPTER 1
INTRODUCTION

1.1 Background

Bioinformatics is an interdisciplinary research field that combines computer sciences and biology to produce helpful biological knowledge. Although the term bioinformatics was used as early as 1970, it was only in 1978 that Paulien Hogeweg formally coined the term (Dayhoff and Schwartz, 1978; Hogeweg, 1978; Paulien, 2011) and defined it as *"the study of informatic processes in biotic systems"* (Paulien, 2011).

Bioinformatics research is increasingly attracting the interest of researchers. The size and number of protein, DeoxyriboNucleic Acid (DNA), and RiboNucleic Acid (RNA) databases are growing rapidly and, as such, require faster and efficient methods to manage and retrieve data (Fernández-Suárez and Galperin, 2013). UniProt, RCSB, and EXPASY (UniProt, 2012; RCSB, 2013; EXPASY, 2014) are examples of protein database websites that show the growth of database size. Figure 1.1 shows the exponential growth of the Swiss-prot database, "*Release 2014_04 of 16-Apr-14 of UniProtKB/Swiss-Prot contains 544996 sequence entries, comprising 193815432 amino acids abstracted from 227703 references"* (EXPASY, 2014). The rapid growth of molecular databases drives the need for efficient and fast sequence comparison algorithms to manage and control large data sizes. The main goal of bioinformatics is to help biologists collect, manage, process, store, analyze, predict, and retrieve genomic information (Jacques, 2004). This goal challenges researchers in computer sciences to produce faster and efficient algorithms for analyzing and organizing large data.

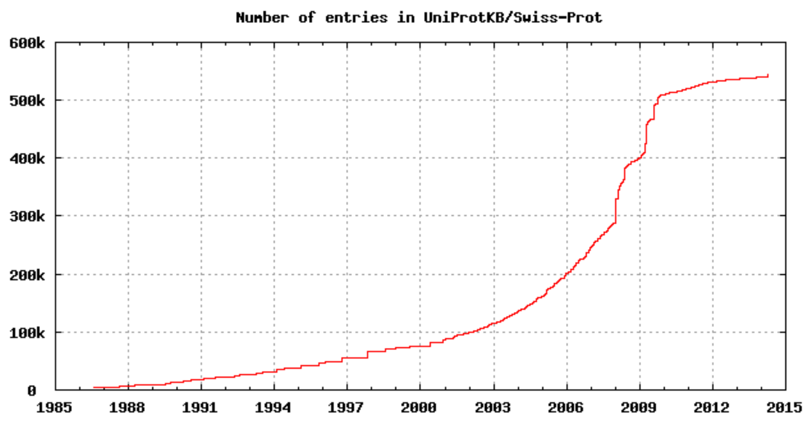


Figure 1.1 Swiss-Prot database growths

Proteins are both the primary and largest components in living organisms. They help determine the function and structure of cells. Proteins have many structural levels, including primary, secondary, tertiary, and quaternary. Each structural level helps in clarifying the functions and chemical properties of living cells. With the increasing interest in protein sequences, many studies are focusing on protein analysis, prediction, comparison and similarity, retrieval, representation, and much more (Sharma, 2009).

Current technologies produce enormous biological data that require fast and scalable data analysis and data management techniques. This demand challenges researchers to produce fast and efficient algorithms in order to analyze and organize large data (Fernández-Suárez and Galperin, 2013). Sequence Alignment (SA) is the basic operation in sequence comparison. It rates the similarity between two or more sequences by aligning the primary sequences of protein to identify similarity regions.

SA can be defined as the process of lining up a group of sequences (two or more sequences) to find the matched regions among or between them. If the number of sequences to align is equal to two, then it is called pair-wise alignment. If the number of sequences is greater than two, then it is called multiple sequence alignment (MSA).

Pair-wise sequence alignment is a method used to measure the similarity/distance between a sequence couples by aligning them initially and then finding the regions of matches and mismatches. This similarity may refer to structural, functional, or evolutionary relationships between a pair of sequences. In most methods of building MSA, pair-wise alignment represents the first stage.

MSA plays an essential role in identifying sequences and collecting information about them (Liu et al., 2010). It is used for several purposes, with varying degrees of importance and motives. Representing and identifying sequence families is the most significant role of MSA. Indirectly, MSA helps in predicting the structure and function of sequences by relating them to their closest similar families. It also helps build the phylogenetic tree to represent the evolutionary history of species and study the evolution of molecules (Pirovano and Heringa, 2008).

Optimal MSA is considered as an NP-hard problem because the size of the problem increases radically when the number and length of sequences increase (Richer et al., 2007; Wang and Jiang, 1994; Rausch, 2010; Ebedes and Datta, 2004; Thomsen et al., 2002; Corel et al., 2010; Wohlers et al., 2011). On the other hand, to reach optimal results using dynamic programming (DP) is an NP-complete problem (Sharma, 2009). The length and number of sequences are important factors to consider in MSA methods.

In this research, the possibilities of enhancing MSA performance in terms of execution time and accuracy have explored. To enhance execution time, the pair-wise alignment research field to look for fast and accurate algorithms has investigated. The investigation has led to a fast method called N-Gram-Hirschberg (NGH) (AbdulRashid, 2008), which is an extension to the Hirschberg algorithm (Hirschberg, 1975). An enhancement to the NGH algorithm is proposed in this research to further improve the execution time. The main reason for adopting and enhancing the NGH algorithm is to speed up the process of building the distance matrix, which contributes directly toward enhancing the time performance of MSA.

To improve the accuracy of MSA, two levels of enhancements have proposed: constructing a robust distance matrix by using the enhanced NGH algorithm and a filtering system. The main purpose of the filtering system is to detect the sequences that do not belong to any family in a given dataset. Two clustering methods are involved in building the filtering system. These methods are introduced in (Ali, 2008; Enright and Ouzounis, 2000; Ertöz et al., 2003) and they are used to build clusters and identify the families out of a group of sequences. The algorithm has improved by introducing outlier detection. Detecting the outlier sequences assists enhancing the accuracy of MSA because giving high attention to those sequences during the alignment process can bring the alignment score down.

1.2 Motivations and Research Problems

MSA plays an important role in identifying sequences and collecting information about them such as similarity, distances, and matched regions between them. It is used for several purposes, with varying degrees of importance and motives. Representing and identifying sequence families are the most significant tasks of MSA. Indirectly, MSA helps predict the structure and function of sequences by relating them to their closest similar families. It also builds the phylogenetic tree, which helps in constructing the evolutionary history of species and in conducting evolution studies on molecules.

Pair-wise alignment plays an important role in building MSA especially with the process of constructing the guide tree, because it rates the similarity among the sequences by aligning them to find similar regions. Accurate pair-wise alignment leads to better and faster MSA because it contributes directly to the construction of a guide tree that guides the alignment process. Since building the distance matrix requires $\frac{n^2-n}{2}$ pair-wise alignments, a faster pair-wise alignment algorithm has a significant effect on the speed of building a distance matrix, constructing a guide tree, and constructing an MSA construction.

On the other hand, a guide tree has a significant role in constructing MSA by guiding the progressive alignment in the process of building the MSA. An accurate and precise guide tree leads to better and faster MSAs because it directs the progressive alignment in terms of which sequences to focus on and which to ignore. Detecting the sequences that do not belong to any family in the dataset before building the guide tree reduces the workload and contributes directly toward enhancing the accuracy performance of MSA.

Although pair-wise alignment methods reach optimal accuracy (AbdulRashid, 2008; Hirschberg, 1975; Smith and Waterman, 1981), they are still consuming processes. NGH algorithm (AbdulRashid, 2008) reduces the time and space required to build the alignment. Time comparison for two sequences is $\boldsymbol{O(mn/k)}$, where $\boldsymbol{k}$ is the size of the word. The comparison results are best when k is 2 (AbdulRashid, 2008). With the very fast increase in database information, (e.g., Swiss-Prot database increased from about 100,000 sequences in 2003 to 542,782 sequences in March

2014), a faster algorithm is becoming an urgent requirement.

MSA is necessary for almost all aspects of computational sequence analysis, but it is a difficult task to compute (Edgar and Batzoglou, 2006; Morrison, 2006; Kemena and Notredame, 2009; Jeevitesh et al., 2010). Despite the diversity of methods and the large number of algorithms that have been proposed to solve MSA, producing efficient MSAs with high accuracy remains a huge challenge (Jeevitesh et al., 2010; Kryukov and Saitou, 2010; Liu et al., 2010). Applying multiple protein sequence alignments on large datasets through progressive alignment also requires hours (Yongchao et al., 2009). Faster algorithms are necessary because the biological sequence databases are growing rapidly (Liu et al., 2006a; Liu et al., 2007b). Some methods have reached high accuracy, such as MSAprobs algorithm (Liu et al., 2010) with high execution time, while other methods have low execution time, such as Multiple Sequence Comparison by Log-Expectation (MUSCLE) (Edgar, 2004b) and Clustal-W (Thompson et al., 1994), but suffer from low accuracy. Guide tree construction has an impact on MSA performance since construction requires the time-consuming process of building the distance matrix. Moreover, guide tree has a significant role in guiding progressive alignment in the process of building MSA. In short, the problems highlight the need for faster and more efficient SA algorithms.

1.3 Research Questions

This research will answer and address the following questions:

a. How can the optimal pair-wise alignment method be improved further?

- What are the pair-wise alignment methods that reach optimal accuracy without sacrificing computational time?
- Will the table inspired from hash table technique improve the time

performance of pair-wise alignment algorithms without sacrificing the accuracy?

b. Can pair-wise alignment contribute toward enhancing the time and accuracy performance of MSA?
   - Does the time consumed to build the distance matrix affect the time performance of MSA?
   - How will robust distance matrix affect the accuracy performance of MSA?
c. How will filtering the distance matrix from outlier sequences affect the MSA accuracy?
   - How can outlier sequences that hide behind multi-domain sequences be detected?
   - Will detecting multi-domain sequences affect the process of detecting outlier sequences?
   - Will a robust guide tree affect the accuracy performance of MSA?

1.4 Research Objectives

This research aims to improve MSA in terms of speed and accuracy by improving the guide tree. Improving the guide tree involves many factors, such as: (a) pair-wise alignment to construct the distance matrix and (b) filtering system to filter the distance matrix. Therefore the research objectives can be concluded as follows:

a. To enhance the time performance and accuracy of the adopted methods [NGH and Hashing-N-Gram-Hirschberg (H-NGH)] by integrating and enhancing the hash table technique to the sequence transformation phase.
b. To enhance MSA performance by reducing the execution time of distance matrix

construction as well as increase the accuracy performance by using the enhanced version of the NGH and HNGH algorithms.

c. To further enhance the accuracy performance of MSA by filtering the distance matrix from outlier sequences (the sequences that do not belong to any family in the dataset) using two clustering methods: multi-domain detection and outlier detection.

1.5 Research Scope

This research focuses on pair-wise alignment, distance matrix construction, guide tree construction, and progressive alignment to build MSA with high throughput. Pair-wise alignment is used to build the distance matrix used to construct the guide tree. Three clustering methods are involved in constructing the guide tree out of a given distance matrix: two methods are used to filter the distance matrix, while the other one is used to construct the guide tree out of the filtered distance matrix. Finally, progressive alignment is used to construct MSA out of the produced guide tree. This research has focused on building MSA for protein sequences by using protein data.

1.6 Research Contributions

This research explores guide trees as a possible way to improve MSA. Improving the execution time and accuracy performance of MSA is the main contribution of this research. Research contributions can be concluded as follows:

a. Enhancing the performance of the NGH and HNGH methods by using a table technique inspired from the hash table method. The adopted technique will be used to enhance the transformation phase of the two methods (NGH and

HNGH).

b. Enhancing the performance of MSA construction by adopting the enhanced version of NGH and HNGH. The adopted method will be used to build the distance matrix which would contribute to:
    - Enhancing the time performance of MSA construction process by speeding up the process of building the distance matrix.
    - Enhancing the accuracy performance of MSA, since the robust distance matrix leads to better guide trees, which, in turn, lead to better alignment.
c. Building a filtering system to filter the distance matrix by detecting the outlier sequences. The filtering system includes two clustering methods: multi-domain sequence detection and outlier sequence detection.

1.7 Thesis Organization

This thesis consists of six chapters organized as follows:

Chapter 2 gives a background on the basic biological data types and investigates the current state of the pair-wise alignment and MSA research fields. Pair-wise alignment methods are grouped (according to the approach used in the method) into two groups: Dynamic Programming (DP) and heuristics. The second part of the literature investigates MSA methods that are classified according to the approach used to solve the problem and a special sub-section to discuss the leading methods in the MSA research field.

Chapter 3 gives an overview and discusses the general methodology of this research. It also discusses the evaluation methods for the proposed algorithms in this research. Chapter 4 discusses the methodology, results, and experimental analysis of the

HashTable-N-Gram-Hirschberg (HT-NGH) algorithm. HT-NGH algorithm is an extension to the NGH and H-NGH pair-wise alignment algorithms. Chapter 5 discusses the methodology, results, and experimental analysis of the Filtered Distance Matrix for building MSA (FDM-MSA) algorithm. FDM-MSA algorithm is an enhancement to MSA proposed to enhance the time and accuracy performance of MSA. Finally, Chapter 6 summarizes and concludes the thesis starting with the problems passing through methods, results, and drawbacks, and then finishing with some recommended future works.

CHAPTER 2
RELATED WORK

2.1 Introduction

This chapter covers the background of some basic materials and the related works of two research fields: pair-wise alignment and MSA. The background defines and explains some basic terms related to this research. Pair-wise alignment methods are categorized into two main groups according to the approach they used. These approaches are heuristic and DP. MSA methods are classified under two main categories: the approach taken and the leading methods. Finally, a summary for the chapter is given.

The background is discussed in Section 2.2. Then, the pair-wise alignment methods are discussed in Section 2.3. After that, Section 2.4 provides details of existing MSA approaches and methods. Finally, a chapter summary is provided in Section 2.5.

2.2 Background

This section discusses and explains some basic terms related to this research, including, biological data categories, substitution matrices, distance matrices, and guide trees.

2.2.1 Basic Biological Data Types

Protein sequences and nucleic acid sequences (RNA and DNA) are the main types of biological data in the databases.

The main difference between the two types of nucleic acids is the sugar that presents in the molecule (ribose in RNA or deoxyribose in DNA). Each nucleotide has three parts: ribose or deoxyribose, phosphoric acid, and a nitrogenous base. Adenine (A), guanine (G), cytosine (C), and thymine (T) are the base nucleotides that represent the DNA nucleotide alphabet. RNA shares the same alphabet of nucleotides with DNA except for thymine (T), which is replaced by uracil (U). Furthermore, nucleic acids have many structural levels, starting with the primary structure, followed by the secondary structure, and ending with the tertiary structure.

Protein comes from the Gree word "prota," which means "of primary importance" (Sharma, 2009) or "protos," meaning "primary" (Lau, 2005). Sir Frederick Sanger in 1958 solved the first protein structure, for which he won a Nobel Prize (Sanger, 1960; Sanger et al., 1977; Sanger, 1981). The first crystal 3D structure of protein myoglobin was determined and solved in 1958 using x-ray diffraction analysis (Bluhm et al., 1958).

Proteins are the biggest chemical component of a cell, where it is required to determine the structure, function, and regulation of the body’s cells. Each protein has a specific function in the cells. Proteins are important because they are the basic components of human physiology (AbdulRashid, 2008; Sharma, 2009). They consist of a chain of amino acids (20 amino acids) connected to one an other by peptide bonds (Lau, 2005). Figure 2.1 shows the molecular structure of the 20 essential amino acids.

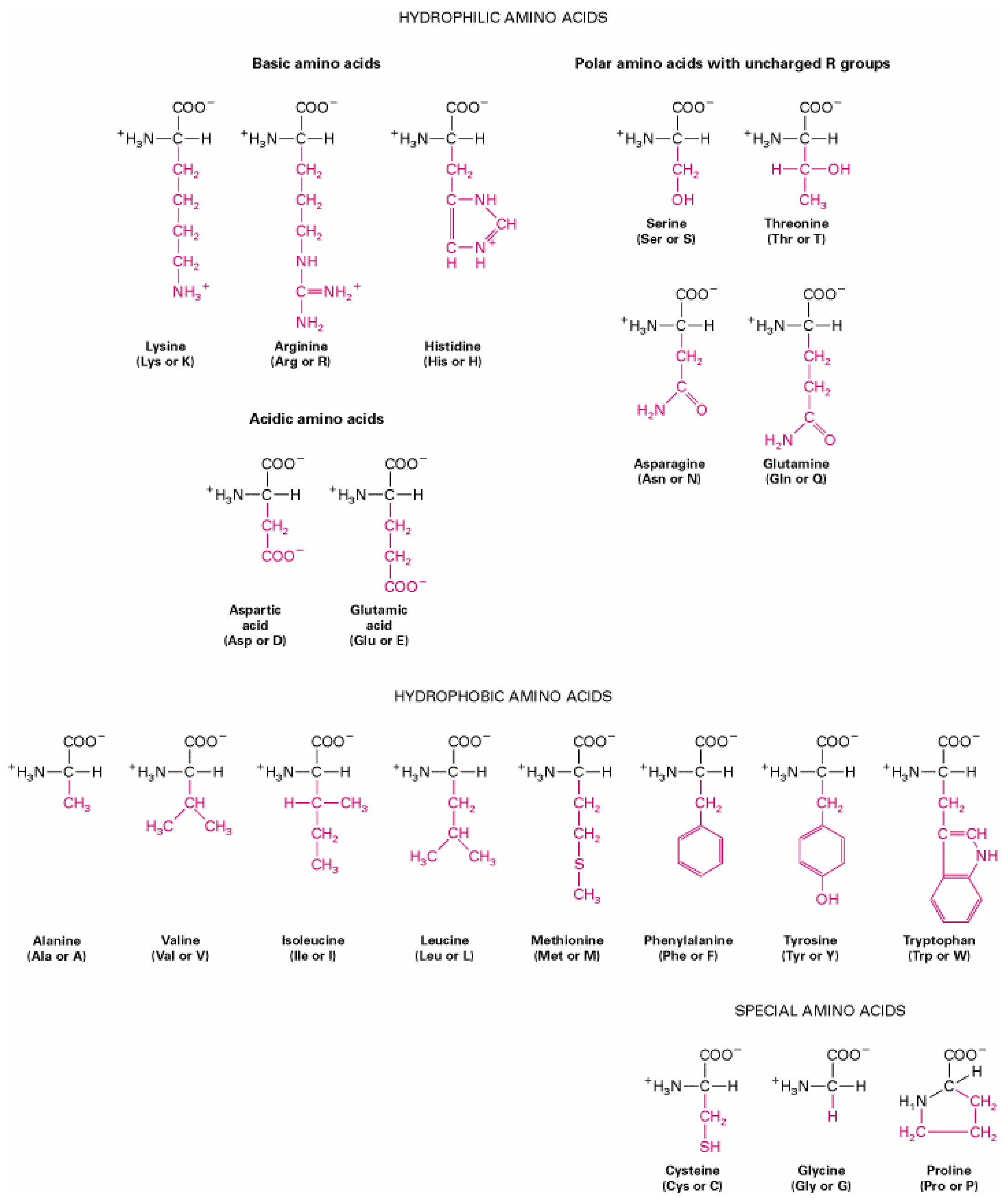


Figure 2.1: Molecular structure of the 20 essential amino acids that are present in all living organisms (Lodish et al., 2000)

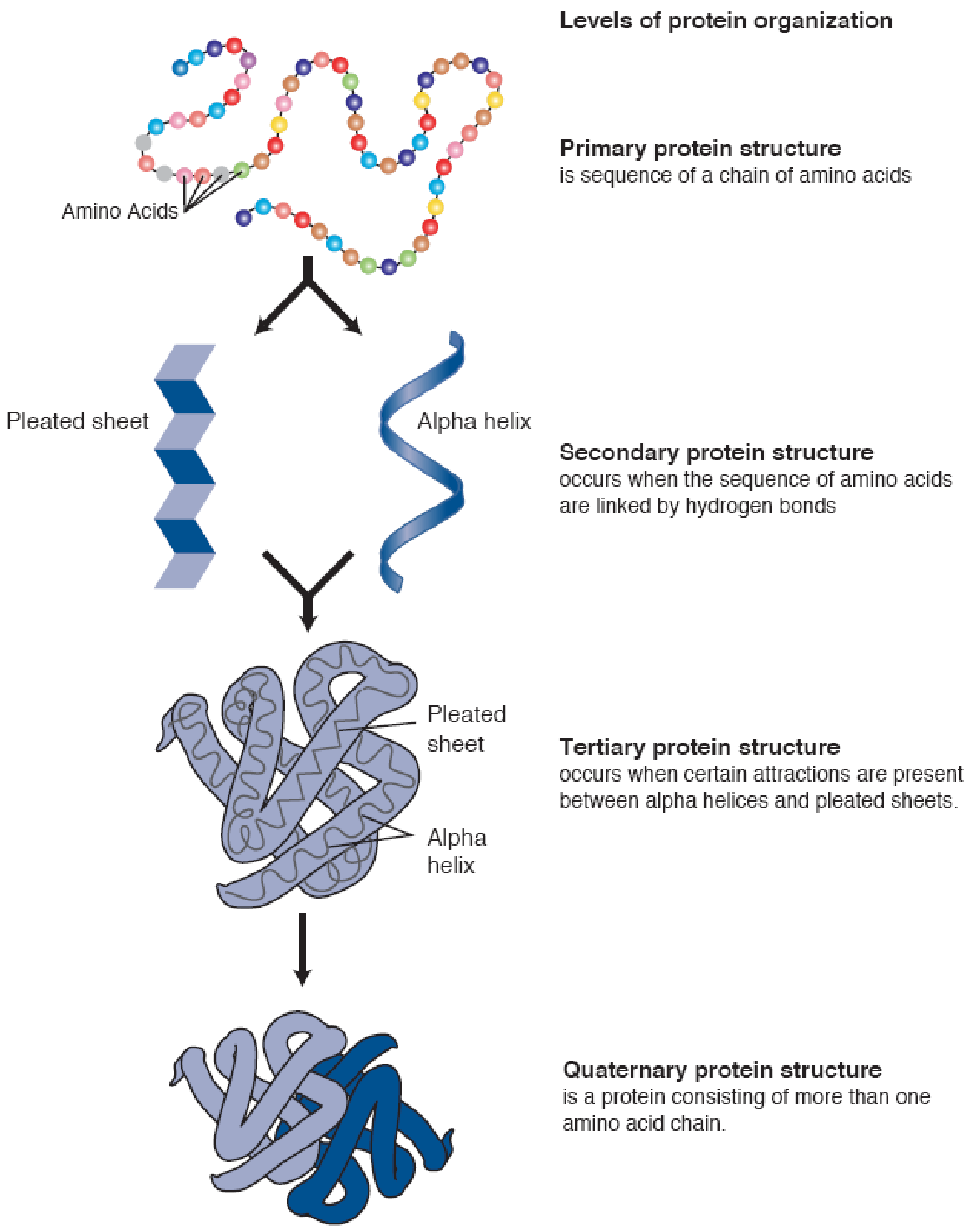
Proteins have four structural levels (i.e., primary, secondary, tertiary, and quaternary). Each level of protein structure is represented by a different shape.

Primary structure: Usually known as protein sequence, this is the first level of protein structure. It is a linear sequence of amino acids arranged randomly and connected by peptide bonds. Each amino acid has two main parts, the backbone and a unique side chain (Sharma, 2009).

Secondary Structure: It is formed by the folding of a peptide chain into one of the three common shapes of protein secondary structures: alpha helix (α-helix), beta sheet (β-pleated sheet), or random coil. Protein secondary structure is a combination of the two main types of structures, namely, α-helix and β-pleated sheet. This structure helps to predict the protein tertiary structure.

Tertiary structure: Formed by the folding of the secondary structure where the amino acids can rotate in a way that makes it possible for far located amino acids to come in contact with one another. The importance of the protein tertiary structure draws researchers' attention because of the valuable information that protein 3D structure can provide. The most important roles related to the level of protein tertiary structure are identifying and understanding the function and chemical properties of proteins (Branden and Tooze, 1999). The most accurate methods for determining the tertiary structure of a protein are X-Ray crystallography methods and Nuclear Magnetic Resonance (NMR), but they are too slow and expensive.

Quaternary Structure: Formed by folding two or more tertiary structure proteins, it is the tertiary structure of a multi-subunit protein. Figure 2.2 shows the four levels of protein structures.

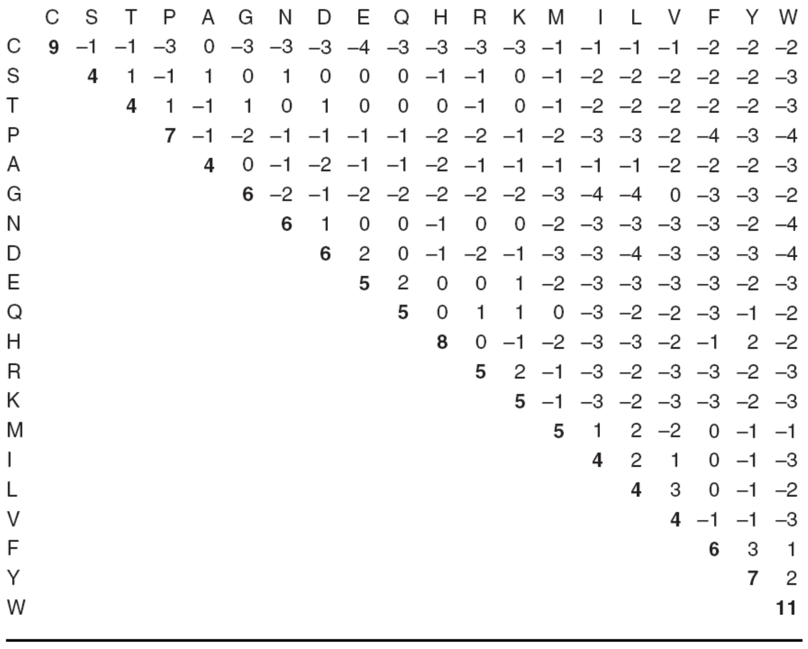


Levels of protein organization
Amino Acids
Primary protein structure
is sequence of a chain of amino acids
Pleated sheet
Alpha helix
Secondary protein structure
occurs when the sequence of amino acids are linked by hydrogen bonds
Pleated sheet
Alpha helix
Tertiary protein structure
occurs when certain attractions are present between alpha helices and pleated sheets.
Quaternary protein structure
is a protein consisting of more than one amino acid chain.

Figure 2.2: Protein structures (NHGRI, 2013)

2.2.2 Substitution Matrix

In bioinformatics, substitution matrix is a matrix that expresses the rate/cost of naturally replacing one residue with another in sequences over time. Substitution matrices are important to SA for calculating the score of an alignment since it grades the alignment for an aligned pair of amino acids. Furthermore, DP methods use substitution matrices in the process of filling the similarity matrix. Substitution matrices or scoring matrices play a significant role in increasing SA performance. Point Accepted Mutation (PAM) and BLOck SUbstitution Matrix (BLOSUM) matrices are the most used matrices in sequence alignment and comparison. PAM matrix is the first created substitution matrix (Dayhoff and Schwartz, 1978). Later in 1992 BLOSUM was proposed for protein sequence alignment (Henikoff and Henikoff, 1992). Figure 2.3 shows BLOSUM62 matrix, an example of the substitution matrix.



	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	**9**	–1	–1	–3	0	–3	–3	–3	–4	–3	–3	–3	–3	–1	–1	–1	–1	–2	–2	–2
S		**4**	1	–1	1	0	1	0	0	0	–1	–1	0	–1	–2	–2	–2	–2	–2	–3
T			**4**	1	–1	1	0	1	0	0	0	–1	0	–1	–2	–2	–2	–2	–2	–3
P				**7**	–1	–2	–1	–1	–1	–1	–2	–2	–1	–2	–3	–3	–2	–4	–3	–4
A					**4**	0	–1	–2	–1	–1	–2	–1	–1	–1	–1	–1	–2	–2	–2	–3
G						**6**	–2	–1	–2	–2	–2	–2	–2	–3	–4	–4	0	–3	–3	–2
N							**6**	1	0	0	–1	0	0	–2	–3	–3	–3	–3	–2	–4
D								**6**	2	0	–1	–2	–1	–3	–3	–4	–3	–3	–3	–4
E									**5**	2	0	0	1	–2	–3	–3	–3	–3	–2	–3
Q										**5**	0	1	1	0	–3	–2	–2	–3	–1	–2
H											**8**	0	–1	–2	–3	–3	–2	–1	2	–2
R												**5**	2	–1	–3	–2	–3	–3	–2	–3
K													**5**	–1	–3	–2	–3	–3	–2	–3
M														**5**	1	2	–2	0	–1	–1
I															**4**	2	1	0	–1	–3
L																**4**	3	0	–1	–2
V																	**4**	–1	–1	–3
F																		**6**	3	1
Y																			**7**	2
W																				**11**

Figure 2.3: BLOSUM62 matrix

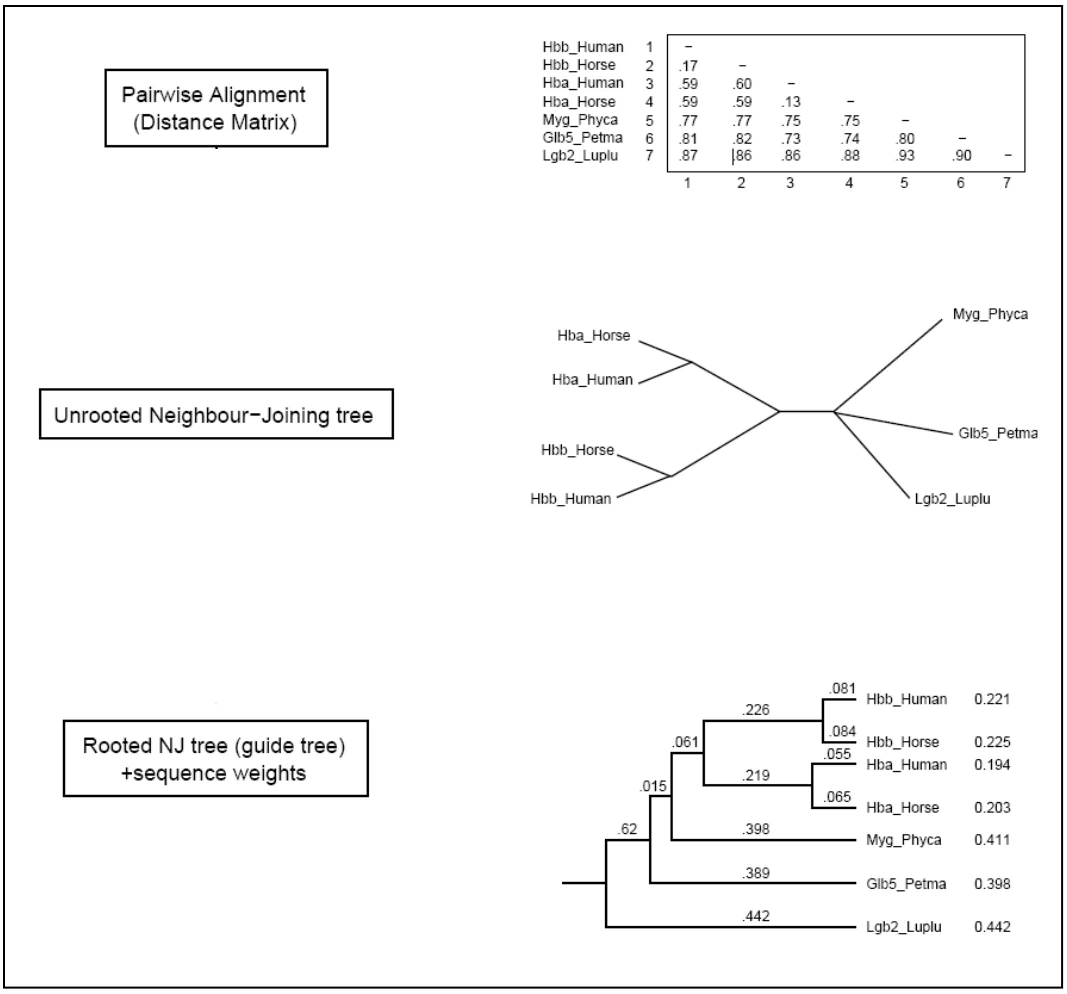
2.2.3 Distance Matrix

Distance matrix is a basic representation in mathematics, computer science, graph theory, and bioinformatics. In bioinformatics, distance matrix is a matrix that carries the distances between sequences taken by applying a pair-wise alignment. It is used for various aims, such as building phylogenetic trees (Blackshields et al., 2010) and constructing MSAs.

Distance matrix is a set of values that tells the distance between a group of sequences where each cell in the matrix represents a distance between pair of sequences. Mostly, these values are calculated using pair-wise SA. Pair-wise alignment calculates the similarities and distances between the sequences by finding the matched regions between them.

2.2.4 Guide Tree

Guide tree is a binary tree that represents the relations among a group of sequences based on distance matrix scores. Sequences are branched in the tree according to the amount of similarity they share, that is, the most similar are the closer to each other. Each leaf in the tree represents a different sequence, while nodes show how far these sequences are from each other. Building a guide tree requires (1) distance matrix, which can be built by applying all-to-all pair-wise alignment, and (2) a clustering method to arrange the sequences in a tree according to the distances in the distance matrix. Figure 2.4 shows an example of guide tree construction using the neighbor-joining (NJ) clustering method.



Hbb_Human	1	−						
Hbb_Horse	2	.17	−					
Hba_Human	3	.59	.60	−				
Hba_Horse	4	.59	.59	.13	−			
Myg_Phyca	5	.77	.77	.75	.75	−		
Glb5_Petma	6	.81	.82	.73	.74	.80	−	
Lgb2_Luplu	7	.87	|86	.86	.88	.93	.90	−
		1	2	3	4	5	6	7

Figure 2.4: Example of guide tree construction using NJ algorithm (Thompson et al., 1994)

Guide trees are used broadly in progressive alignment methods to guide the alignment process because the sequences are organized in the tree according to their resemblance score. Guide trees offer a pre-computed map that can be used as a compass to direct the progressive alignment. Progressive alignment starts constructing the alignment by aligning two sequences, and then continues aligning sequences to the previous aligned sequences. This process makes guide trees a great support to enhance the alignment score and reduce the running time.

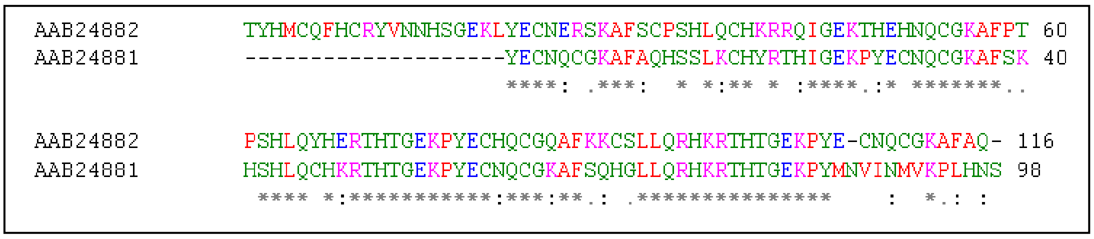
Distance matrix clustering methods are broadly used to construct guide trees. Unweighted Pair Group Method with Arithmetic Mean (UPGMA) (Sneath and Sokal, 1973) and NJ (Saitou and Nei, 1987) clustering algorithms are the most used methods to convert the distance matrix into a guide tree.

UPGMA: is a bottom-up distance-based clustering method used to construct phylogenetic/guide trees (Sokal and Sneath, 1963). It is a simple guide tree construction method. UPGMA algorithm uses a pre-computed distance matrix as input in order to measure the distances between sequences. First, it starts tree branching by branching out to the closest pair of sequences; the two sequences are then considered as one sequence in the matrix. Afterward, the algorithm recalculates the distances in the matrix based on the new combination. A new branch gets added to the tree by taking the closest sequence to the first chosen pair. The algorithm continues repeating these steps until all the sequences in the matrix get branched-in in the tree.

NJ: is a bottom-up distance matrix clustering method that was proposed to build phylogenetic/guide trees (Saitou and Nei, 1987). To build the guide tree, the NJ method uses a pre-computed distance matrix which is by performing all-to-all pair-wise SA. It starts building the tree by constructing a star-like tree depending on the pre-computed distance matrix. Then, it reconstructs the tree by bringing the neighbor sequences together and focusing on decreasing the overall branch length. NJ is recommended over UPGMA when the evolutionary rate of the sequences is not equal, which happens when a sequence has more mutations than the other sequences (Felsenstein, 2004).

2.3 Pair-wise Sequence Alignment

Pair-wise SA is the process of comparing two sequences by looking for similar regions (individual or series of characters) between them. In pair-wise alignment, the two sequences are lined up in two rows that create a matrix of two rows and n columns (n is the length of the longest sequence), where each sequence is represented by one row in the matrix. Gaps are then inserted into the sequences in the matrix in order to place the matched characters in the same columns, while the mismatched characters either face a gap or are placed in the same column (Mount, 2004). Figure 2.5 shows an example of pair-wise alignment for two protein sequences.



```
AAB24882        TYHMCQFHCRYVNNHSGEKLYECNERSKAFSCPSHLQCHKRRQIGEKTHEHNQCGKAFPT 60
AAB24881        --------------------YECNQCGKAFAQHSSLKCHYRTHIGEKPYECNQCGKAFSK 40
                                    ****: .***:  * *:** * :****.:* *******..

AAB24882        PSHLQYHERTHTGEKPYECHQCGQAFKKCSLLQRHKRTHTGEKPYE-CNQCGKAFAQ- 116
AAB24881        HSHLQCHKRTHTGEKPYECNQCGKAFSQHGLLQRHKRTHTGEKPYMNVINMVKPLHNS 98
                 **** *:***********:***:**.: .***************    :  *.: :
```

Figure 2.5: Example of a protein sequence pair-wise alignment

Researchers have been developing and proposing algorithms for more than three decades to reach faster and accurate alignment. Different methods using different approaches and techniques have been proposed to solve and improve pair-wise SA. This section focuses on the leading methods of pair-wise alignment. These methods can be categorized under two main approaches: heuristic-based algorithms and DP-based algorithms.

2.3.1 Heuristic-based Algorithms

The main property shared by heuristic approaches is the scanning of all possible solutions of a certain problem to find or select the most optimal one. The selected solution is considered a near-optimal solution of the problem because there is no guarantee of finding the optimal solution. However, algorithms that follow the heuristic approach are quite fast, and they produce results in a reasonable time compared to DP-based methods. The rapid increase in size of protein sequence databases gives rise to the need for faster methods of constructing pair-wise alignment. Researchers have applied this approach to the pair-wise alignment problem to reach the optimal solution with less run time (AbdulRashid, 2008).

FASTA is a heuristic-based pair-wise alignment method used for sequence comparison and alignment (Lipman and Pearson, 1985). It divides the sequence into patterns, and then looks for the matches between patterns. Each pattern called K-tuple indicates the number of matches within the two sequences to measure the similarity and build the alignment (Wilbur and Lipman, 1983; Lipman and Pearson, 1985; Pearson and Lipman, 1988). Later on, a method similar to FASTA, called BLAST, was designed to improve the sensitivity (Altschul et al., 1990; Altschul et al., 1997). BLAST algorithm is also designed to improve the time performance of the overall search and to place the database search on a firm statistical foundation. In 2001, another word's method called Sequence Search and Alignment by Hashing Algorithm (SSAHA) was proposed. It uses the hash table technique to store the K-tuple occurrence positions (Ning et al., 2001). An improved particle swarm optimization (PSO) method is also applied to the pair-wise alignment problem in (Wenjie et al., 2009). Later on, a DNA pair-wise alignment method called gpALIGNER was proposed (Hadian Dehkordi et al., 2011). It is a segment-based method that uses the similar score schema proposed in DIalign-T algorithm to build the alignment. Recently, a space-sufficient local pair-wise alignment was proposed in (Stojanov et al., 2012) for optimal ungapped alignment.

2.3.1 (a) FASTA

Lipman and Pearson (1985) developed a rapid pair-wise alignment method for protein sequences called FASTA (Lipman and Pearson, 1985; Pearson and Lipman, 1988; Wilbur and Lipman, 1983). This method divides the sequence into patterns, after which it looks for the matches between patterns. Each pattern called K-tuple indicates the number of matches within the two sequences compared. Algorithm 2.1 shows the FASTA algorithm.

Algorithm 2.1: FASTA algorithm (Pearson and Lipman, 1988)

```
Input: A query (DNA or protein sequence), related Database.
Output: The match sequences in descending order of matching and local
alignment.
Process:
    1.  The 10 best-matching in each pair of sequences are located by rapid
screening.
     (a)    All sets of k-consecutive matches are found by
            (i)Find match for short word 4-6 of DNA, 1-2 of protein.
            (ii)Those matches with distance less than certain threshold are
                combined.
            (iii)The regions of highest density of matches are being identified.
2.  The highest density region identified in A are evaluated using a suitable
            scoring matrix.  The highest scoring regions are identified (INIT1).
3.  Longer regions of identity are generated by joining the shorter region with
            scores greater than certain threshold.
4.  This step is the optimisation steps where Smith Waterman Algorithm is
            being used.
```

FASTA uses the K-tuples procedure, which is equivalent in nature to the N-Gram method. The location of all K-tuples that comes from the query sequence is stored in a hash table. Putting these K-tuples in a hash table takes a long *O (N)* time because as many as N-K+1 tuples can be generated from a query of length N. Using the K-tuple information derived from the query sequence marks all K-tuple matches with the protein sequences in the database. Then positive values will be given to the marked cells, while negative values will be assigned for unmarked cells to facilitate the calculation of the number of matches and mismatches. However, FASTA's speed cannot deal with the exponentially growing size of a database, and thus the need for faster algorithms is justified (Chan, 2004).

2.3.1 (b) BLAST

BLAST method is a heuristic pair-wise alignment method (Altschul et al., 1990). It is similar to FASTA and uses the concept of words, word expansion, and high segment pairs. BLAST algorithm is proposed to enhance the accuracy and sensitivity of FASTA algorithm. It is also designed to improve the speed of overall search and to place database searching on firm statistical foundations. In addition, the generated local alignment must not contain gaps.

Although, the time complexity of BLAST *(O (MN))* is theoretically the same as that of Smith-Waterman, in reality it is much faster than the Smith-Waterman algorithm because of the pre-processing stage where the filtering and finding of pairs is performed. However, the procedure still suffers from the heuristic weakness because of the reduced sensitivity, where reducing execution time reduces the quality of the results (Pearson, 1995; Hara et al., 2010).

2.3.2 Dynamic Programming-based Algorithms

Dynamic programming (DP), a term coined by Richard Bellman (Bellman, 1952; Dreyfus, 2002), is a technique of solving a problem by dividing it into smaller sub-problems recursively until the sub-problems become indestructible. Then, it builds the solution by combining the solutions of all the sub-problems (Pearson, 1995; Berman and Paul, 2005). DP guarantees to find the optimal solution of a given problem, which is an advantage over the heuristic approach, but it usually requires more time compared to the heuristic-based methods.

DP approach is used by numerous methods to improve the performance of pair-wise SA. Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) is the pioneering method for all DP-based methods in sequence alignment and comparison. Needleman-Wunsch algorithm measures the similarity between sequences by building a similarity matrix guided by a substitution matrix (Dayhoff, 1965; Henikoff and Henikoff, 1992). Building the similarity matrix requires *O (MN)* space, which is quite high if the length of protein sequences has considered.

Hirschberg algorithm (Hirschberg, 1975) proposes a solution to solve the space problem of the Needleman-Wunsch algorithm, which reduces the space requirement to *O (min (m,n))*. Unlike the Needleman-Wunsch and Smith-Waterman algorithms, Hirschberg algorithm divides the similarity matrix into two smaller matrices depending on the fact that similarity matrix can be filled from both directions; top down and bottom up. Since Hirschberg algorithm has one row (array) and two variables, the space requirement is reduced to linear with respect to the shorter length of the two sequences (Driga et al., 2006).