COPIA: A New Software for Finding Consensus Patterns in Unaligned Protein Sequences

by

Chengzhi Liang

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Master of Mathematics

in

Computer Science

Waterloo, Ontario, Canada, 2001

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Consensus pattern problem (CPP) aims at finding conserved regions, or motifs, in unaligned sequences. This problem is NP-hard under various scoring schemes [52, 1]. To solve this problem for protein sequences more efficiently, a new scoring scheme and a randomized algorithm based on substitution matrix are proposed here. Any practical solutions to a bioinformatics problem must observe two principles: (1) the problem that it solves accurately describes the real problem; in CPP, this requires the scoring scheme be able to distinguish a real motif from background; (2) it provides an efficient algorithm to solve the mathematical problem. A key question in protein motif-finding is how to determine the motif length. One problem in EM algorithms to solve CPP is how to find good starting points to reach the global optimum. These two questions were both well addressed under this scoring scheme, which made the randomized algorithm both fast and accurate in practice. A software, COPIA (COnsensus Pattern Identification and Analysis), has been developed implementing this algorithm. Experiments using sequences from the von Willebrand factor (vWF) family [66] showed that it worked well on finding multiple motifs and repeats. COPIA's ability to find repeats makes it also useful in illustrating the internal structures of multidomain proteins. Comparative studies using several groups of protein sequences demonstrated that COPIA performed better than the commonly used motif-finding programs.

Acknowledgements

I was previously trained to be a geneticist. Nevertheless, bioinformatics was a quite new area to me when I started my studies here two years ago. I have been working with mathematician/computer scientist exclusively since then. All the discussions that I have made with them have helped me a lot in my understanding many biological problems mathematically. I here would like to thank them all for their help and several of them I wish to mention specifically. First I thank my supervisor Ming Li for his great help in selecting my research topic and for the valuable discussions with him throughout my studies here. I also thank Bin Ma and Guohui Lin for the valuable suggestions on the problems I have met in my research and comments on my thesis. In addition, I thank Jinbo Xu for all the discussions with him on various problems, Jonanthan Badger and Paul Kearney for their help in my research, and all my friends who have helped me during my studies here. Finally I thank the Department of Computer Science that supported me for my studies.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Sequence Alignment: A Fundamental Problem in Computational Biology

Since the nuclear genome of the first free-living organism, *Haremophilus infuenza*, was sequenced in 1995 [26], tens of other genomes have been completed so far[1]. They are mostly bacterial genomes. Among them also is our nearly completed human genome [18, 76]. The complete sequences of those genomes, especially that of our human genome, have provided several order of magnitude of more data than all the sequences obtained in the past decades, which include many small genomes of viruses and organelles (see Table 1.1, 1.2). All the genome sequences are available in Entrez database[2] [72]. These sequence data have provided us an unprecedented chance for our understanding life, improving human living conditions. Most of the information stored in these data, nevertheless, is yet to be decoded through both computational and experimental methods.

Genome Type	No. of Genomes	Genome Size
Viruses	557	1-350kb
Organelles	210	13-350kb
Bacteria	41	0.6-7Mb
Archaea	10	1.5-3Mb
Eukaryotes	5	12-3,000Mb

Table 1.1: The number of sequenced complete genomes

Functional genomics and proteomics are two important disciplines emerged in the so-called post-genome era. For almost all genes (except those do not encode proteins such as rDNA genes), some natural questions are: what protein does it encode, when and under what circumstances it expresses, and what is the function of the protein it encodes? The first question is easy as

[1] Some biological background knowledge can be found in Appendix A.

[2] Website http://www.ncbi.nlm.nih.gov/cgi-bin/Entrez

Organism	Size	Genes
Yeast	12Mb	5800
Nematode	97Mb	19000
Fruit Fly	137Mb	13000
A. thaliana	125Mb	25000
Human	3000MB	30000

Table 1.2: The sizes and gene numbers of the completed eukaryotic genomes

long as we have the gene sequence since protein sequences can be derived from DNA sequences (though it is still hard to find all gene sequences even in the complete genome sequences). The second and third question are addressed by functional genomics and proteomics, respectively.

Functional genomics deals with questions such as the expression patterns of all genes in a genome and their relationship as well as the functions of non-coding regions. Proteomics, on the other hand, which is a counterpart of genomics that studies the collection of genes, studies the structures and functions of complete collection of proteins in an organism. Both functional genomics and proteomics depend heavily on experimental approaches. This is time-consuming and expensive. While the smallest nuclear genome sequence of *M. genitalium* consists only of $580kb$ of DNA with only about 484 protein coding genes[27, 30], our human genome consists of about 3 billion bases of DNA with estimated number of genes being more than $30,000$. With this huge amount of sequence data, it will be too much work to derive all their functions exclusively from experimental results, not mention that some genes are resistant to experimental function analysis because of their unreproducible functional environment.

Fortunately, we have another tool, comparative genomics, to ease the work burden of functional genomics and proteomics. Comparative genomics, by its name, is the study of genomes through sequence comparisons. It deals more theoretically with the raw data provided by sequencing projects. Comparative studies can provide important pointers to the potential functions of genes/proteins. We hope to build various automated sequence analysis tools to reveal the functions of genes and proteins as much as possible from the sequence itself; this will save considerable amount of experimental time and expenses. Our expectation on the high throughput of comparative genomics has been established on a solid empirical ground which can be stated as the following observation:

> *If two sequences are sufficient similar, they are likely to share similar biological functions.*

This observation implies the relationship between sequence and function in two respects:

1. function is encoded into sequence; different sequences generally have different functionalities, and
2. there is a redundancy in the sequence encoding, i.e., a sequence may be changed without perceptible changes in its function.

For example, TATA box [56] is the best known transcription factor-binding site in eukaryotes. It is a conserved DNA element found in the promoters of many protein-encoding genes. It has a consensus sequence TATA(A/T)A(A/T). The usage of A or T in the two A/T pairs has little effect on their binding ability. Other base changes, however, have been shown to alter transcriptional activity both *in vivo* and *in vitro* [79, 16, 82, 40]. Many gene/protein sequences in different species are *orthologous*: they are descendent of a common ancestor and play essentially the same roles in many species. Generally, the closer the two species, the more similar their orthologous gene/proteins (and their functions). A well-known orthologous protein family is globin family [70, 34].

Sequence comparison had been used long before any large genome sequence was completed. A routine work in a molecular biology lab might be to search for similar sequences in databases for a new sequence in order to understand its function. From the evolutionary point of view, similar sequences are more likely to be evolved from the same ancestor rather than occurred by chance. In other words, many sequences have been evolved slowly just because of their functional constraint. The function of TATA-box was first revealed through experimental analysis before enough sequences can be used for comparative studies. If there were some computational tools and enough sequences to find it in the first place, then it should be at least helpful for us to study its function in the right direction.

1.1.1 The Sequence Alignment Problems

One purpose of doing sequence comparison is to find their similarity. We need some formal ways to quantify this. For example, we can use their character frequencies as a measure. Certainly, this is not the best way. While the dinucleotide frequencies provide a genome-level signature in the sequenced eukaryotes [28], much more information is encoded into the linear sequences than the character frequencies themselves. Thus, we want to find their positional similarity both individually and at the whole sequence level. First, we have the following definitions:

Definition 1 *A* global alignment *is an arrangement of two or more sequences such that each character in a sequence corresponds to a character (one to one correspondence) in each of the other sequences. Gaps (blank characters) can be inserted into each sequence. Informally, the sequences can be put one over another, so all the characters in each corresponding position form a column. To do a* local alignment *is to select a subsequence from each sequence and then to align these subsequences (globally). A* pairwise alignment *is the alignment of two sequences. A* multiple alignment *is the alignment of more than two sequences.*

An alignment can be evaluated based on any scoring schemes, with which the sequence similarity can also be computed. The scoring scheme determines which is the best among all the possible alignments for a given set of sequences. The best alignment is one that has the highest or lowest score that depends on the scoring scheme. This leads to the following sequence alignment problems:

GLOBAL ALIGNMENT
Instance: Given a set of sequences over a fixed alphabet and a scoring scheme.

Problem: Find an alignment with its score maximized (or minimized).

LOCAL ALIGNMENT
Instance: Given a set of sequences over a fixed alphabet and a scoring scheme
Problem: Find a subsequence from each sequence such that the alignment of those subsequences give maximum (or minimum) score in all such combinations of subsequences.

The pairwise alignment (either global or local) can be used to find if two sequences are related. It is generally done by dynamic programming. With a scoring scheme such as a substitution matrix and gap penalties (see next section), a global alignment can be done using Needleman-Wunsch algorithm [57, 31] and a local alignment using Smith-Waterman algorithm [68]. The local alignment is to find out if two sequences share any similar regions. It should be noted that the best local alignment is not necessary in the best global alignment. Multiple alignment can capture conserved features in a protein family. The scoring scheme used in a multiple alignment is less well defined. Multiple alignments (either local or global) are generally NP-hard [52, 1] (more on this later). While global alignment is generally used in comparison of closely related homologous sequences, local alignment is more meaningful in the study of distantly related sequences since it aims at finding conserved regions, or motifs. These regions generally each represents a single secondary-structure that contains no gaps in alignment and are crucial to the function of the proteins.

Studying the evolutionary history of a gene or repeated DNA sequences is a global alignment problem since we can treat the gene or each unit of repetition as an independent unit even though they are only part of chromosomes since we can define their boundaries before doing alignment. Finding the transcription factor binding sites is a local alignment problem since these sites are generally very short (several bases) and embeded into long stretches of non-coding regions of a gene. Many methods have been addressed to solve this problem [39, 65].

1.1.2 Protein Sequence Alignment: Local *vs* Global

To study the evolutionary and functional relationship of genes, comparison of DNA sequences has proved to be less useful than that of protein sequences. The variation of DNA sequences is often larger than protein sequences and many changes in DNA level do not lead to changes in protein level (due to degenerate codons).

Functionally related proteins are categorized into (super-)families. From the evolutionary point of view, those proteins are descendents of a common ancestor (common descent). Those proteins are said to be *homologous*. On the other hand, many non-homologous proteins often share similar functions. These proteins share considerable sequence similarity in their functional sites though the other parts of their sequences are very divergent. Examples are various DNA binding proteins, transmembrane proteins. The origin of such similarity has been considered as through the convergent evolution, an example in [15], i.e., different sequences converge to similar sequences. A new explanation of such a similarity, however, might be emerging from the recent results of comparative genomics.

One of the most important contributions of genome comparison is to our understanding how

different species are evolutionary related, and what, at molecular level, makes them different. Sequence comparison of proteins from 21 complete genomes of bacteria, archaea, and eukaryotes has revealed remarkable similarity among these organisms as well as considerable diversity[71]. This study discovered that about $56 - 83\%$, with average of 67%, of the proteins of bacterial and archaeal belong to ancient families conserved across a wide phylogenetic range, and 35% of yeast proteins belong to these families. Although most eukaryotic proteins have been found to have no counterparts in prokaryotes (this might be due to the insensitivity of the analyzing method used as suggested in [47]), those proteins are evolved in many occasions from old ones that lost their identity during evolution.

Most large proteins are composed of multiple domains, which have an average size of about 174 residues in known crystal structures [29] (for comparison, the average human protein length is about 460 [18]). These domains are generally not only structural independent, with weak interaction between them, but also perform distinct functions that may remain intact in isolated domain. Multidomain proteins can be *homomultimeric*, i.e., they contain multiple copies of single type of structural domain, or *heteromultimeric*, i.e., they contain multiple types of domains. Examples of both types can be found in [21]. The homomultimeric proteins are believed to be evolved through the internal duplication of gene segments encoding an entire domain, whereas the heteromultimeric proteins are the outcome of fusion of two or more gene segments that encode different domains [61].

Studies on the multidomain proteins have found that portions of many such proteins (genes) are related by vertical descent, but they also accrete new domains in different lineages and concomitantly acquire new functions. Horizontal gene transfer, transferring of genes from phylogenetically distant species, in contrast to the vertical inheritance from parents to children, is common at least in the evolution of prokaryotes [80, 59]. This phenomena of horizontal gene transfer have made phylogeny tree an incomplete form representing evolutionary relationship of whole genomes [22].

The evolution of eukaryotic proteins, on the other hand, is largely due to domain accretion and shuffling, which is observed in parallel with the increase in complexity of eukaryotic organisms [53, 46]. The novel combinations of existing domains, rather than creation of new domains, is the main force to increase the divergence and complexity of protein families in eukaryotes. The phenomenon of protein domain accretion and shuffling suggested that protein domains rather than whole proteins might be the basic units of evolution. The phylogenetic tree of proteins, therefore, might be best represented in protein domain level.

From these discussions, we can see that local alignment of protein sequences is more general in that it can help find those distantly related domains, and hence the distantly related proteins.

1.2 Statistical Analysis of Sequence Alignment

1.2.1 Model of Evolution and Substitution Matrix

The process of doing sequence alignment is to find the related residues in different sequences. The scoring scheme mentioned in last section is a quantitative measure that determines how

much they are related. If they are highly related (over a certain value), they will be assigned to the same position in an alignment. The aligned sequences can be used to construct a model to predict if other sequences are related to this set of sequences. The relationship of different residues is essentially determined by their physicochemical properties as well as the context in which they appear. Those context-dependent properties, however, are generally too complex to analyze quantatively. Either they must be simplified (but less accurate) or other equivalent measures are used in practice. One successful measure is established based on their evolutionary relationship. One can imagine that the evolutionary process is more likely to select those residues sharing similar physicochemical properties for the same functionality.

In order to set up an evolutionary measure of similarity for each pair of residues, we need a model to describe the evolutionary process of the protein sequences as follows:

> Proteins (or domains, to be more accurately as discussed in last section) evolve through a succession of independent mutations (substitution, insertion and deletion), which are fixed in the present sequence populations.

The evolution of the residues at each corresponding position in all related sequences forms a Markov process. The evolution of a sequence is the results of many independent Markov process. A *Markov process* consists of an initial state and a matrix defining the transition probability from one state to another for all pairs of states. The set of initial states for the set of Markov processes to evolve a protein family is their ancestral sequence, and without considering insertions/deletions, the transition matrix consists of 20 states, the 20 amino acids.

In the model above, we assume for simplicity that: (1) mutation rates, especially the relative mutation rates of different residue pairs, are uniform throughout the evolutionary process; hence, the Markov process is homogeneous. (2) mutations are not position-specific, so all Markov processes have the same transition matrix, but may have different initial states. As will be discussed later, the second assumption is not valid in all cases, but it is included here to make our model simpler.

This process of protein evolution is equivalent to saying that a sequences is generated by this model M with certain probability. For all sequences X, we have

$$\sum_{x \in X} P(x|M) = 1.$$

One purpose of sequence alignment is to find the ancestral sequence and the transition matrix. Then we can used this model to assign a probability to each sequence[3].

To obtain the transition matrix, we must have a set of evolutionary related residues first. This can be obtained from a set of aligned sequences. In this set of sequences, at each position, there are many different residues. These residues are assumed to have been evolved from the same ancestor. Since all sequences that are observable today are the results (rather than intermediate states) of the evolutionary process, we can not distinguish the ancestor and descendents at any

[3]Note that all sequences of length different than the ancestral sequence are assigned probability 0 by this model. It is too restrictive in global alignment but is acceptable in local alignment since we can always find a subsequence of the same length.

positions. We have to treat them equally: each residue can be mutated from another. For a total number of n residues at a position, the maximum likelihood estimator for each transition probability is:

$$q_{i,j} = \frac{n_i n_j}{N},$$

where $q_{i,j}$ is the transition probability of ith to jth residue, n_i and n_j are the number of i,jth residue, respectively, and N is the number of total pairs, which is $n(n-1)/2$. After this probability is averaged for all positions, we can get a more accurate estimation. $q_{i,j}$ is most likely equal to $q_{j,i}$ in reality since otherwise the amino acid composition would be in evolutionary disequilibrium. The transition probability reflects the similarity (to some extent, see below) for each pair of residues. The higher this value, the more likely a pair can mutate to each other, i.e., they are more similar to each other.

The transition matrix obtained above is the maximum likelihood estimator. Obviously, there are other models which can also generate this set of sequences with certain probability. The most frequently referred model is the background model, which states the probability of the aligned residues occurring by chance alone. We hope that the sequences generated by our model with high probability are closely related to each other (and to the ancestral sequence) and the sequences with low probability are not related to the ancestral sequences, i.e., the relatedness defined by our model can not occur purely by chance. A pair of highly abundant residues (so their transition probability is high) is not necessarily more similar than a pair each of which occurs less frequently; the residues occurring with high frequency will have more chance to pair with each other in an alignment.

In order to get a more accurate estimation of residue pairwise similarities, the background model must be taken account for. Using the log odd ratio for all transition probabilities in these two models, we get a *substitution matrix* in which each entry is defined as:

$$m_{i,j} = \log \frac{q_{i,j}}{p_i p_j},$$

where $q_{i,j}$ is the observed transition probability for residue pair a_i, a_j, and $p_i p_j$ are their background probability. $m_{i,j}$ represents the similarity of two residues revealed by evolution rather than by chance.

In above protein evolutionary model, the observed transition probability after different time interval should be different. Otherwise, this Markov process would have entered into its stationary state in which each residue's frequency is its background frequency and the observed transition matrix by maximum likelihood estimation would be the background model. This way, all sequences are like to be generated purely by chance and sequence alignment is meaningless.

The *evolutionary distance* between two sequences can be defined as the number of mutations needed to produce one from another. The number of mutations reflects the similarity between two sequences. Based on the assumption of the uniform mutation rate, this distance represents the time point at which the sequences are in the Markov processes. The actual number of mutations occurred between two sequences, however, is hard to compute due to the following unobservable events:

1. a residue mutated to another can mutate back;
2. several mutations might have occurred between two observed residues.

The evolutionary distance, therefore, is often represented by the minimum number of mutations needed to produce one sequence from another. Using sequences at different distance, different substitution matrices can be obtained to describe their residue similarity at this distance. Two popular set of matrices have been constructed in this way.

PAM Matrices

The PAM matrices were introduced by Dayoff et al [19]. A PAM (percent accepted mutation) is one accepted point mutation per 100 residues occurred on the path between two sequences. For example, PAM250 means that there are 250 mutations occurred per 100 residues. This corresponds to a long evolutionary interval. Not all accepted mutations are observable as stated above. The difference in percentage of residues between two sequences is equivalent to the PAM distance only when this percentage is very small (such as $< 5\%$). Since the long PAM distance is not observable, Dayoff et al used extrapolation from PAM1 to get high PAM matrices. For a homogeneous Markov process,

$$P_n = P_1^n,$$

where P_1, P_n is the transition matrix of PAM1 and PAMn, respectively. They first constructed a phylogeny tree for a set of closely related sequences and aligned them to get the mutation rate at low PAM distance. The transition matrix at 1 PAM distance was extrapolated to obtain all other matrices. The substitution matrices were then obtained using log odd scores to background probability. This way has been considered to introduced some errors since (1) the error at PAM1 is enlarged when go to high PAM distance; (2) the mutation rate might not be the same at all regions in a sequence.

BLOSUM Matrix

Not satisfied with the above extrapolation method, Henikoff and Henikoff [36] provided another one to construct BLOSUM matrices. They used only the conserved regions, aligned blocks, to compute the mutation frequency. The sequences in each block are clustered according to their percentage of identity to obtain each substitution matrix. For example, the clustering of sequences with identity of $> 62\%$ (at observed evolutionary distance $< 38\%$) leads to BLOSUM62.

There are two differences in the meaning of each PAM matrix and BLOSUM matrix. An example can illustrate them easily: (1) PAM250 represents the transition matrix at the evolutionary distance of 250 PAMs per 100 residues; (2) BLOSUM62 represents the (average) transition matrix at observed distance of more than 38 mutations per 100 residues. The *entropy* of a substitution matrix is defined as

$$H(S) = \sum_{i,j} q_{i,j} \log \frac{q_{i,j}}{p_i p_j},$$

```
# This matrix was produced by "pam" Version 1.0.6 [28-Jul-93]
# PAM 160 substitution matrix, scale (\lambda) = ln(2)/2 = 0.346574
# Expected score = -1.14, Entropy = 0.694 bits
# Lowest score = -7, Highest score = 12
```

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	2	-2	0	0	-2	-1	0	1	-2	-1	-2	-2	-1	-3	1	1	1	-5	-3	0	0	0	0	-7
R	-2	6	-1	-2	-3	1	-2	-3	1	-2	-3	3	-1	-4	-1	-1	-1	1	-4	-3	-1	0	-1	-7
N	0	-1	3	2	-4	0	1	0	2	-2	-3	1	-2	-3	-1	1	0	-4	-2	-2	2	1	0	-7
D	0	-2	2	4	-5	1	3	0	0	-3	-4	0	-3	-6	-2	0	-1	-6	-4	-3	3	2	-1	-7
C	-2	-3	-4	-5	9	-5	-5	-3	-3	-2	-6	-5	-5	-5	-3	0	-2	-7	0	-2	-4	-5	-3	-7
Q	-1	1	0	1	-5	5	2	-2	2	-2	-2	0	-1	-5	0	-1	-1	-5	-4	-2	1	3	-1	-7
E	0	-2	1	3	-5	2	4	0	0	-2	-3	-1	-2	-5	-1	0	-1	-7	-4	-2	2	3	-1	-7
G	1	-3	0	0	-3	-2	0	4	-3	-3	-4	-2	-3	-4	-1	1	-1	-7	-5	-2	0	-1	-1	-7
H	-2	1	2	0	-3	2	0	-3	6	-3	-2	-1	-3	-2	-1	-1	-2	-3	0	-2	1	1	-1	-7
I	-1	-2	-2	-3	-2	-2	-2	-3	-3	5	2	-2	2	0	-2	-2	0	-5	-2	3	-2	-2	-1	-7
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	5	-3	3	1	-3	-3	-2	-2	-2	1	-4	-3	-2	-7
K	-2	3	1	0	-5	0	-1	-2	-1	-2	-3	4	0	-5	-2	-1	0	-4	-4	-3	0	0	-1	-7
M	-1	-1	-2	-3	-5	-1	-2	-3	-3	2	3	0	7	0	-2	-2	-1	-4	-3	1	-3	-2	-1	-7
F	-3	-4	-3	-6	-5	-5	-5	-4	-2	0	1	-5	0	7	-4	-3	-3	-1	5	-2	-4	-5	-3	-7
P	1	-1	-1	-2	-3	0	-1	-1	-1	-2	-3	-2	-2	-4	5	1	0	-5	-5	-2	-1	-1	-1	-7
S	1	-1	1	0	0	-1	0	1	-1	-2	-3	-1	-2	-3	1	2	1	-2	-3	-1	0	-1	0	-7
T	1	-1	0	-1	-2	-1	-1	-1	-2	0	-2	0	-1	-3	0	1	3	-5	-3	0	0	-1	0	-7
W	-5	1	-4	-6	-7	-5	-7	-7	-3	-5	-2	-4	-4	-1	-5	-2	-5	12	-1	-6	-5	-6	-4	-7
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-2	-2	-4	-3	5	-5	-3	-3	-1	8	-3	-3	-4	-3	-7
V	0	-3	-2	-3	-2	-2	-2	-2	-2	3	1	-3	1	-2	-2	-1	0	-6	-3	4	-2	-2	-1	-7
B	0	-1	2	3	-4	1	2	0	1	-2	-4	0	-3	-4	-1	0	0	-5	-3	-2	3	2	-1	-7
Z	0	0	1	2	-5	3	3	-1	1	-2	-3	0	-2	-5	-1	-1	-1	-6	-4	-2	2	3	-1	-7
X	0	-1	0	-1	-3	-1	-1	-1	-1	-1	-2	-1	-1	-3	-1	0	0	-4	-3	-1	-1	-1	-1	-7
*	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	-7	1

Table 1.3: PAM160 matrix. The original log-odd scores were multiplied by a constant number and rounded to their nearest integer.

```
#  Matrix made by matblas from blosum62.iij
#  * column uses minimum score
#  BLOSUM Clustered Scoring Matrix in 1/2 Bit Units
#  Blocks Database = /data/blocks_5.0/blocks.dat
#  Cluster Percentage: >= 62
#  Entropy =   0.6979, Expected =  -0.5209

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

Table 1.4: BLOSUM62 matrix. The original log-odd scores were multiplied by a constant number and rounded to their nearest integers.

which is the maximum average information for each pair of aligned residues based on this matrix. Each PAM matrix has a corresponding BLOSUM matrix and vice versa based on their entropy. For example, BLOSUM62 (Table 1.4) is equivalent to PAM160 (Table 1.3), which is at (observed) distance 70% mutation rate.

BLOSUM matrices have overcome the two shortcomings of PAM matrices in that: (1) the matrices were obtained from real data directly; (2) the positions in a sequence are treated differently (only conserved regions are used; these regions should reflect the residue similarity more accurately because it is based on functional constraint). Thus BLOSUM matrices should be more accurate in aligning sequences, especially in local alignment. Their experimental results has proved this.

1.2.2 Statistics of Substitution Matrix and Pairwise Local Alignment

A substitution matrix defined above uses the log odd ratio of their evolutionary relationship to their relationship by chance alone for each pair of residues. A good substitution matrix is thus ideally one that best distinguish such a difference. In the local alignment of two sequences, we want to find which two subsequences are closely related to each other based on the specific substitution matrix and how close they are related to each other.

Definition 2 *A* maximal segment pair *(MSP) is a pair of aligned segments (without gaps) in two sequences that have the greatest aggregate score (which is the sum of the scores of all pairs of aligned residues), i.e., its score can not be increased by extending it on both sides.*

If sequences are considered as being generated from the model above, scoring a pairwise alignment is just like to sample randomly from the substitution matrix. For any two sequences, there must be some stretch of subsequences that give high similarity scores. In order to find a meaningful MSP between two sequences, the substitution matrix must have two conditions satisfied:

1. its expected score is less than 0;
2. at least one entry is positive.

First condition assures that we can not get a good MSP by just extending the alignment, and the second condition assures we can get an alignment of length greater than 0.

These two conditions are guaranteed in the substitution matrices constructed as above, since

$$-\sum_{i,j} p_i p_j \log \frac{q_{i,j}}{p_i p_j}$$

is the relative entropy of background distribution to target model distribution. It is always non-negative and at least one entry is negative unless all entry is 0 (otherwise all $q_{i,j} < p_i p_j$).

For any substitution matrix, we can define λ [44] as a scaling factor in

$$\sum_{i,j} p_i p_j e^{\lambda s_{i,j}} = 1, \tag{1.1}$$

where p_i, p_j are the background probabilities of i, jth residue pair and $s_{i,j}$ is their similarity score. This equation has a unique positive solution for λ if the two conditions above are satisfied.

As stated above, an MSP can be viewed as a sum of random samples generated from a substitution matrix. Among MSPs obtained from random sequences based on any substitution matrix, the residue pair i, j appears with probability ([5, 44], also compare with equation 1.1)

$$q_{i,j} = p_i p_j e^{\lambda s_{i,j}}.$$

Clearly, $q_{i,j}$ is the set of target frequencies (transition probabilities) determined by the substitution matrix and the random background model. From this, we can see that any substitution matrix has an implicit set of target frequencies for aligned residues, and only the alignment with these target frequencies can be best distinguished by the substitution matrix. For example, for a pair of sequences at evolutionary distance 100 mutations per 100 residues, the PAM100 matrix should give a better alignment than PAM250 matrix does. Thus our goal in constructing substitution matrices is to find one with the best target frequencies (as the maximum likelihood estimator used above).

The statistical theory of MSP scores [44] states that with a substitution matrix, the expected number of MSPs with score at least S occur by chance is approximately

$$E = KNe^{-\lambda S}, \tag{1.2}$$

where K is a constant factor depending on the substitution matrix and background frequency, and N is the product of the sequence lengths. The probability P of an MSP with a score of at least S is

$$P(X \geq S) = 1 - e^{-E}. \tag{1.3}$$

1.2.3 Gapped Pairwise Alignment

In our method of constructing a substitution matrix, gaps are not included. The global alignment without gaps is too restrictive to be useful. Insertions and deletions are common in the evolution of protein sequences, especially in the interdomain linker regions [60]. Gaps must be penalized. The mostly frequently used cost function for a gap of length l is given by an affine score:

$$g(l) = -c - (l-1)e, \tag{1.4}$$

where c is called the *gap-open* score and e is the *gap-extension* score. e is usually set to a value less than c. This is often desirable since the chance of insertion/deletion of more than one residue at the same time should be higher than that of insertion/deletion of the same number of residues one by one.

The gapped local alignment is also useful since insertion/deletion can occur occasionally at conserved regions. Although there is no corresponding analytical theory that have been developed for gapped alignment as in ungapped case, considerable empirical evidence has suggested that they are similar [3].

1.2.4 Profile and HMM for Searching Distant Homologies

One main purpose of doing multiple alignment is to find the features that are conserved in a protein family. These features will then be used for searching distantly related members in this family. This way of database search is generally more sensitive than using one member or even all members of a family to do pairwise alignment.

The conserved features are generally built into a model called a profile [33, 32]. A *profile*, or *position-specific score matrix* (PSSM), is a score matrix that assigns a position-specific score to each amino acid at all positions in a sequence, which is different from a substitution matrix in that each position in a sequence corresponds to a column in profile. To construct a substitution matrix, we assume that all positions have the same transition probability. This is generally too simplistic; the mutation rate in functional sites should be much lower than the residues without critical functions.

A PSSM essentially defines a set of target frequencies at different positions in our protein evolutionary model. For example, if a position has $5a$'s, $3b$'s, and $2c$'s, and another position has $9a$'s and one b, clearly, the target frequency (evolutionary distance) at these two positions are different. Different scores should be assigned to the same a's at these two positions. A single substitution matrix, however, can not reflect this difference.

A profile allows gaps. The gap penalty function can be the one described in last section or any others. A profile can be better described by a linear hidden Markov model (HMM), which is also called a *profile HMM*. A profile HMM has 3 states: match, insert and delete at each position. The advantage of a profile HMM over ordinary profile is that in the former case a single probabilistic model can be used to describe the whole extent of alignment (including gaps). A detailed description can be found in [23]. I here just give a brief discussion.

For any sequence x, its probability assigned by a profile HMM M without considering gaps is

$$P(x|M) = \prod_i e_i(x_i),$$

where $e_i(x_i)$ is the emission probability of residue x_i. The log-odd score assigned to residue x_i is

$$S(x_i) = \log \frac{e_i(x_i)}{p_{x_i}},$$

where p_{x_i} is the background frequency of residue x_i. This is equivalent to the score assigned from a substitution matrix (conditioned on a column).

Now let's consider gaps. The insertion and deletion are treated differently. The probability of an insertion includes several parts: the transition probability $a_{M_i I_i}$ from match state M_i to insert state I_i, emission probability e_{i_j} from the insert state I_i to generate residue i_j, the transition probability a_{I_i} from the insert state to itself, and the transition probability $a_{I_i M_{i+1}}$ from the insert state to next match state M_{i+1}. The inserted residues are like to be sampled from background model, so it has emission probability $e_I(x_k) = p_{x_k}$. Since no log-odds contribution from the emission, the score of a gap of length l is

$$g(l) = \log a_{M_i I_i} + \log a_{I_i M_{i+1}} + (l-1) \log a_{I_i I_i},$$

Method	Local	Iterative	Statistical
PIMA	N	N	N
CLUSTALW	N	N	N
ITERALIGN	N	Y	N
DIALIGN	N	N	N
MATCH-BOX	N*	N	N
SAM	N*	Y	Y
Gibbs Sampling	Y	Y	Y
MEME	Y	Y	Y

Table 1.5: Summary of several motif-finding methods

which is equivalent to the affine gap score defined in Equation (1.4). The score for deletion can be obtained similarly. One difference is that the transition probability from a delete state to another can be different, with the formula similar to the affine score as a special case.

A profile or profile HMM can be constructed from a set of aligned sequences. The parameters in a profile HMM can be computed with maximum likelihood estimator for each position. The column IC scores in a set of aligned sequences can also be used to construct a profile: assign each item in the formula of computing IC score as the position-specific score for the residue at each position.

The main drawback of a profile HMM is that it generally needs a large number of sequences for its parameters to be accurately estimated. In above method of constructing a profile or a profile HMM, some kind of pseudocounts must be added to avoid zero probabilities if the number of sequences in the alignment is small. The pseudocount methods include simple pseudocount methods such as a constant for each residue, Dirichlet mixtures, and substitution matrix mixtures (see [23]).

1.3 A Review of Multiple Alignment Methods

Many protein motif-finding methods have been proposed, and computer programs based on them have been developed so far. Those methods can be divided into several categories based on whether they are (1) global or local, (2) progressive or iterative, or (3) statistical or non-statistical. Those programs include PIMA [67], MEME [7], SAM [42], DIALIGN [55], ITERALIGN [12], CONSENSUS [38], CLUSTALW [73], MATCH-BOX [20], BLOCKMAKER [37] and PROBE [58]. The methods using statistical models are finite mixture model [7] (MEME), Gibbs sampling method [50, 58, 77] (BLOCKMAKER and PROBE), and HMM [42, 48, 45] (SAM). The most frequently used methods is listed in Table 1.5 and some details of these methods are given below.

PIMA

PIMA [67] uses a progressive pairwise alignment method to construct a global multiple alignment and extracts the conserved patterns for those sequences. First, the method generates a tree from a set of related sequences by clustering their pairwise similarity scores. Then the tree is reduced from leaves by combining the two leaves with the same parent into one common pattern (a new leaf) until only single root is left. The pairwise alignment is performed through a modified dynamic programming algorithm of Smith and Waterman [68] to generate local optimal patterns. The pattern is formed by compressing the two aligned characters into one new symbol. For example, a pair of D's are represented by D, while D/E is represented by [DE]. X is used to represent two unrelated symbols. Gap is allowed in the alignment. Finally, when only the root is left in the tree, the conserved patterns can be identified and all the sequences are aligned together.

CLUSTALW

CLUSTALW [73] is also a progressive global alignment using a guide tree. The basic steps are: (1) pairwise sequence similarities are calculated (by dynamic programming or heuristics) and then used to construct a (phylogenetic) tree with branch lengths proportional to the estimated distance along each branch; (2) this tree is used to align all the sequences using dynamic programming according to its branching order, each step consisting of aligning two existing alignments or sequences. Different substitution matrices can be selected in the dynamic programming. Sequence weights are calculated from the guide tree and used to downweight closely related sequences. The alignment obtained by the basic step will be further improved by choosing appropriate gap penalties.

ITERALIGN

Iteralign [12] is iterative global method. It uses each sequence to which all other sequences are aligned using a dynamic programming method to find the best set of high-scoring segments pairs (HSPs). An 'ameliorated' sequence is constructed for each sequence after alignment. Each ameliorated sequence is then used to align with all the sequences to construct a new set of ameliorated sequences. This step is iterated until no more new ameliorated sequences are produced (convergence). Those last set of ameliorated sequences are used to replace the original sequences and repeat the above steps. The final set of ameliorated sequences can be used to construct the core blocks, which can be optimized by adding indels, extending blocks or deleting block positions, and aligning the sequences not aligned previously. The final core blocks represent the global alignment, from which the motifs can be retrieved.

MATCH-BOX

This method [20] first finds all the matches using every subsequence (of a specific length) of each sequence to compare with all other sequences. A match means that two subsequences are

similar to each other. A box is formed by a set of matches collinearly overlapping with each other and including all sequences. The set of matches are then screened and optimized (combined or deleted) until no overlapping boxes are left. All the boxes left finally form the global alignment with each box being viewed as a local alignment.

DIALIGN

This method [55] used a local alignment approach to do global alignment. First, each pair was aligned using dynamic programming. All the diagonals in each table were then compared with each other using a greedy algorithm. The diagonals that were consistent with each other and had highest overlapping scores with the diagonals in other sequence pairs were put into the final alignment. Gaps were inserted in a final step to make all diagonals to be matched.

Gibbs Sampling

Gibbs sampling [50] can be viewed as a variation of expectation maximization (EM) methods. It is a supervised learning algorithm in that it assumes one motif in each sequence. It has been implemented in BLOCKMAKER and PROBE. The method is initialized by randomly choosing a subsequence of length L from each sequence and then repeats the following two steps: (1) choose a sequences x, A position specific probability matrix (profile) is constructed using a method similar to the IC method above from the $n-1$ subsequences not in x. The background distribution is computed from the residues not in the $n-1$ subsequences; (2) the probability of each subsequence of length L in x is calculated from the matrix. A subsequence in x is selected according to its probability and replaces the old one of x in the set. The iteration stops when no further improvement can be obtained, i.e., reaching a local optimum. Multiple motifs are found by keeping two or more profiles at the same time.

MEME

MEME [7] uses an MM (mixture model), in which one is the motif model to generate the motif instances, and the other is the background model. MM algorithm is an unsupervised learning algorithm. Each sequence can contain any number of occurrences of the motif. MM treats each subsequence (of the specified motif length) equivalently. In terms of our evolution model above, this model assumes that each subsequence (rather than the whole sequence) is the basic unit of evolution without considering its context. This algorithm is also a special case of EM method. The purpose is to find a set of most probable subsequences generated by the motif model rather than the background model. Multiple motifs are found by probabilistically erasing the motif found and then searching for a new one.

HMM

HMM is implemented in SAM [42]. As discussed in last section, the power of HMM is its flexibility to represent gaps, and its drawback is the requirement of large training datasets. To

construct a profile HMM from a set of unaligned sequences, EM algorithms such as Baum-Welch algorithm [8] are also used for estimating the probability parameters. The first step in constructing a profile HMM is to choose an appropriate length and initialize the parameters. Then the parameters (and the length) can be improved iteratively.

Several comparative studies on some of these methods have been reported in [54, 74, 11, 41]. While these programs have gained considerable achievement in protein sequence alignment, some problems still exist. One of their conclusions was that for sequences of similar length and high similarity ($> 25\%$), global methods are generally performed better than local methods. One of these studies [74], however, also showed that the performance of global alignment programs deteriorated greatly in the presence of large N/C-terminal extensions and internal insertions (as those in multidomain proteins of different length/domain composition). Based on their studies, they [74] suggested that the future work on improving multiple alignment should concentrate on the problems of large insertions, extensions and sequences with low similarities such as non-homologous sequences sharing conserved structural/functional regions (actually, these regions might be homologous; one such example is helix-turn-helix (HTH) proteins [64]), i.e., to improve local alignment methods.

1.4 Consensus Pattern Problem for Protein Sequences

Consensus pattern problem (CPP) is a local alignment problem, which aims at finding motifs in unaligned sequences. CPP has been studied in various cases for both DNA and protein sequences [69, 49, 43, 81].

Definition 3 *Fix an alphabet $\Sigma = \{a_1, a_2, \ldots, a_k\}$, which is of size 20 for protein sequences. A* distance matrix D *defines the distance $D(a_i, a_j)$ between each pair of letters a_i and a_j. The* distance *between two sequences of same length $s = s_1 \ldots s_L$ and $t = t_1 \ldots t_L$, over Σ, is the sum of the distance of each corresponding pair of letters, i.e., $D(s,t) = \sum_{i=1}^{L} D(s_i, t_i)$. We can get their* similarity score *or just* score *if we use a substitution matrix S instead of a distance matrix D. The* consensus letter *of a group of letters, G, is the letter that has the minimum total distance (or maximum similarity) to all these letters, i.e. the letter $a \in \Sigma$ such that $\sum_{b \in G} D(a,b)$ is minimized, or $\sum_{b \in G} S(a,b)$ is maximized. The* consensus sequence *of n sequences of length L is the sequence such that its j-th letter is the consensus letter of the j-th columns of the n given sequences. A* pattern (motif) instance *is a subsequence that is close to the consensus sequence. The* cost/score *of an alignment or a consensus sequence is the sum of the distance/similarity of the consensus sequence to all its pattern instances.*

Consensus Pattern Problem
Given n sequences $s_1, s_2, ..., s_n$ over Σ, with lengths $m_1, m_2, ..., m_n$, respectively, the problem is to find a subsequence t_i of length L from sequence s_i for each i and their consensus sequence, based on some scoring schemes.

Our evolution model assumes that each column of an alignment is independent of each other. Therefore, each column can be treated separately when scoring a multiple alignment. In addition, each sequence is treated separately as they are generated independently from the model. For a (local) alignment A consisting of n (sub)sequences of length L. CPP has been studied mathematically under several scoring schemes [52, 1].

1. minimizing
$$S(A) = \sum_{i=1}^{n} H(t, t_i), \tag{1.5}$$
where $H(t, t_i)$ is the Hamming distance between each (sub)sequence t_i and its consensus sequence t,

2. minimizing
$$S(A) = \sum_{j=1}^{L} \sum_{a \in \Sigma} f_j(a) \log f_j(a), \tag{1.6}$$
where $f_j(a)$ is the frequency of each different residue a in column j.

3. maximizing
$$S(A) = \sum_{j=1}^{L} \sum_{a \in \Sigma} f_j(a) \log \frac{f_j(a)}{p(a)}, \tag{1.7}$$

4. maximizing
$$S(A) = \sum_{j=1}^{L} \sum_{j<i} m_{i,j}, \tag{1.8}$$
where $m_{i,j}$ is the entry for i, jth residue pair in a substitution matrix.

The consensus pattern problems under these scoring schemes are NP-hard. Even worse, the consensus pattern problem under IC score is APX-hard for any alphabet of size of greater than one [1]. This implies that there does not exist a PTAS unless $P = NP$. Although several approximation algorithms were proposed to solve them, these algorithms can hardly be used directly in practice. Generally, any practical solutions to a bioinformatics problem must have the two following conditions satisfied: (1) a good objective function (or scoring scheme) to accurately describe the real problem, and (2) an efficient algorithm to solve the mathematical problem. The first condition is important in that the objective function must be able to distinguish true solutions from false solutions and help to find the true solutions from background. Based on these criteria, new methods are still needed to solve CPP for protein sequences efficiently. Clearly, Hamming distance is not a good scoring scheme for protein sequences since it can not reflect the true relationship of amino acids. Entropy score is not a good one, either, since it is equivalent to ignore the background (null) model in a two model statistical analysis. The problem with SP score is that it is not statistically sound. It can not be explained by our evolutionary model directly; each residue in a column seems to have descended from other $n - 1$ residues but it is not a descendent of itself! IC score is the best one and expected to be able to well reflect

the characteristics of true motifs in the existence of enough sequences. CPP under IC score can be solved by an expectation maximization (EM) algorithm [49] and the Gibbs sampling method [50]. In addition, MEME [7] solves the essential problem in practice. The more general method, HMM [48] can also be used to solve this problem. These methods, however, though have been successful in various cases, either suffer from the difficulty to find a good starting point close to the global optimum or need a large training dataset or are not well suited for studying multidomain proteins. To improve present motif-finding methods in practice, a new scoring scheme based on substitution matrix and a randomized algorithm under it are proposed in the next chapter. The algorithm has been implemented in a software COPIA (COnsensus Pattern Identification and Analysis). The implementation issues and the testing results of COPIA are also discussed.

Chapter 2

A Randomized Algorithm for CPP and Its Implementation

2.1 A New Scoring Scheme for CPP

Using a general distance matrix rather than Hamming distance, the cost of an alignment can be defined as

$$C(A) = \sum_{j=1}^{L} \sum_{i=1}^{n} D(a_{i,j}, c_j), \tag{2.1}$$

where c_j is the consensus letter for column j, and $a_{i,j}$ is the letter in sequence i, column j. For any distance matrix D except the one in which all entries are equal, we can transform it into a substitution matrix S:

$$S_{i,j} = D_{i,j} * (-1) * u + v,$$

where u is a positive number and v is constant.

If we transform the distance matrix D into a substitution matrix S, for the same consensus sequence, the score of A is

$$S(A) = \sum_{j=1}^{L} \sum_{i=1}^{n} S(a_{i,j}, c_j). \tag{2.2}$$

Clearly, minimizing $C(A)$ is equivalent to maximizing $S(A)$ for any given set of protein sequences, i.e., their optimal solutions are the same. The transformation from an arbitrary distance matrix to substitution matrix is not a one to one function. We can get many different substitution matrices from a distance matrix, each with a different set of target frequencies. For fixed length L, this does not affect the optimal solution. To find the best motif, which is often of unknown length, however, we need to use the most suitable matrix reflecting the true target frequencies (more on this in next section).

From the discussion here, we can see that consensus pattern problem under Hamming distance is a special case of the problem under substitution matrix. On the other hand, the latter is a special case of the problem under IC score. From Chapter 1, we know that any substitution

matrix has an implicit set of target frequencies.

$$S(A) == \sum_{j=1}^{L}\sum_{i=1}^{n}\frac{1}{\lambda}\log\frac{q_{i,j}}{p_i p_j} = \frac{1}{\lambda}\sum_{j=1}^{L}\sum_{i=1}^{n}\log\frac{q_i}{p_i} = \frac{1}{\lambda}\sum_{j=1}^{L}\sum_{a\in\Sigma} f_j(a)\log\frac{q_j(a)}{p(a)},$$

where $q_i = q_{i,j}/p_j$ is the conditional probability of letter $a_{i,j}$ appearing in column j. It has been pre-defined in the substitution matrix. The target frequency q_i in each column can only be selected in the set implied by the substitution matrix while under IC the target frequency is estimate with the dataset, which is unrestricted.

As a complement, the scoring scheme in Eq. (2.2) can also be defined as

$$S(A) = \sum_{j=1}^{L}\sum_{a\in\Sigma} f_j(a)s(a,c) - \sum_{j=1}^{L} s(c,c),$$

which can be considered as a natural extension of the scoring scheme used in pairwise alignment. The latter is a better choice in estimating the best pattern length and the significance of a motif (see below) when the sequence number is small.

2.2 A Randomized Algorithm

Under Hamming distance and IC score, an approximation algorithm was proposed to solve CPP in [52]. The basic idea in the algorithm is to select r (a constant $< n$) subsequences (with replacement), but at most one can be from any sequence, and use their consensus sequence to find its closest subsequence in all n sequences to form a pattern. The best pattern is used as an approximation to the optimal solution. To get a good approximation, however, a large r is required [52]. Actually, even for a small r, the number of such combinations are still too large for practical purpose.

In any EM algorithms, the consensus sequence found above will be used further to find its closest subsequence in each sequence and this step will be repeated until convergence, i.e., reaching a local optimum. Therefore, the consensus sequence of the r subsequences above can be considered as a starting point around a local optima (though this iteration step is not pursued further in the algorithms described in [52]). A problem here is that most of the r subsequences are not related (nor even close) to the optimal solution. Under substitution matrix, most of them can be eliminated (to save computation time). For example, for any r sequences, some heuristics such as that used in BLAST [4] can be used to find very quickly a set of best scoring r subsequences, one from each sequence. These discussions lead to a randomized algorithm to solve CPP under substitution matrix:

Algorithm RandomizedConsensusPattern

Input n sequences $\{s_1, s_2, \ldots, s_n\}$ over Σ, with lengths $\{m_1, m_2, \ldots, m_n\}$, respectively, a substitution matrix S, and an integer L.

Output a consensus sequence and all its instances.

1. repeat until no improvement
   Randomly choose r sequences, and obtain a set of best consensus sequence p's of length L for these r sequences, by heuristics.
   For each p, repeat until no improvement (convergence):
   (a) Find a substring p_i of length L from each sequence s_i which is closest to p.
   (b) Compute the new consensus sequence p' of all of p_i's. If the score of p' is greater than that of p, set $p := p'$.
2. Output the p and its $p_1, p_2, \ldots, p_n$ with maximum score in step 1.

It is easy to see that each iteration of 1(a) and 1(b) will always give a better solution (until reaching a local optimum). The success of this algorithm in practice is largely due to the following observations: protein motifs are generally distinct to each other, and the number of instances of any motif in each sequence is small. If every sequence contains at least one motif instance, the instances in a good solution for some subsets (of size r) of sequences will also be in the optimal solution for the whole set. The consensus sequence for the whole set can be viewed as the center of the n points. If we get the center of a subset of those points (the r sequences), these two center points can be expected to be often close to each other. The global optimal solution is thus often reachable from a good solution for a randomly selected subset of sequences (especially with a technique called phase shift [50]). The probability of finding the optimal solution, starting from the good solutions for any subset of r sequences that each contains at least one motif instance, should be high even for weak patterns[1], as long as their instances can be distinguished from background using a substitution matrix. In practice, a few random selections of r sequences are often enough to find the global optimum. Another reason for the algorithm to be practical is that we generally do not "care" whether we have found the best solution or not when there exist multiple instances as long as we can find all of them (see below for finding repeats).

2.3 Implementation

To make the algorithm above more practical in finding protein motifs, several implementation issues need to be addressed here: selecting a substitution matrix, determining the best pattern length, finding multiple motifs and repeats, computing the statistical significance of each motif instance, i.e., how likely it is a real motif instance or just a subsequence occurring by chance, and constructing an explicit position-specific scoring matrix (PSSM) or profile [33, 32] without gaps. A program called COPIA has been developed to implement all these features. Details of

[1] It is hard to define mathematically what a weak or a strong pattern is. Roughly, a strong pattern should consist of a set of highly similar instances.

Sequence Length	5	10	15	20	25	30	35	50	100	200
Score	14	16	16	16	16	16	16	14	8	-9

Table 2.1: The scores above which a score has a probability less than 0.1 for random sequences based on BLOSUM62. These scores were estimated using 100,000 random sequences for each length. See text for the meaning of the scores.

the implementation are discussed below.

2.3.1 Determining Pattern Length

In many cases, the length of a pattern is unknown in advance. Selecting a good pattern length thus becomes a very important issue in these cases. If a pattern is too short, it can occur randomly with a high probability. If it is too long, the signal to noise ratio will be affected severely. The best way to solve this problem is to estimate the pattern length while it is being searched. This is also one of the reasons to use a substitution matrix. With a substitution matrix, a pattern can be extended according its column scores as in constructing MSPs in pairwise alignment. In order to set up the standard for extending a multiple alignment, the possible scores that a column can be assigned needs to be estimated first. In the method defined above to obtain each consensus character, Each column in a multiple alignment can be viewed as random sequence (scores) generated from one of the columns in a substitution matrix. The probability distribution of a random sequence generated from any column in the substitution matrix is a multinomial distribution. The column in the substitution matrix generating each random sample is that assigns the highest score (probability) to it among all columns in the substitution matrix. A *Monta Carlo* method was used to estimate the probability distribution of the highest score of a random sequence assigned by BLOSUM62. The result was shown in Table 2.1.

Clearly, the probability that a random sequence (or a column in an alignment) is assigned a score greater than 0 is too large for it to be the threshold of extending a column in an alignment. To see this, assume that 10 sequences are to be aligned, of which 5 each contains a strong instance (they have high similarity to each other), and the others each contains a weak instance (with great divergence). This can happens if the sequence similarities are between such as $10-40\%$. Generally, the residues flanking strong instances are also similar. This can make the pattern be easily extended beyond its actual length since the similar residues flanking the strong instances can make their column score greater than 0. The over extended pattern, however, increases noise for the weak instances; this may make them insignificant to be recognizable by the consensus sequence. In COPIA, a heuristic based on the ratio of the column score to the average column score of the pattern to be extended. If the ratio is lower than a predefined value, the extension step stops. In local alignment, highly similar sequences are often harmful since the sequence similarity is not from functional constraint. This way of setting pattern length makes COPIA less sensitive to the highly similar sequences included in the set to be aligned. Another possible

method is to down-weight the highly similar sequences using a sequence weighting scheme (see below). This method is yet to be implemented in COPIA.

The initial pattern length is determined by the length of the best MSP of two sequences sharing low similarity or a predefined value. When the pattern can not be extended any further, if this pattern length is significant (such as > 30 which can provide enough information to distinguish it from background), this pattern will be reported. Otherwise, the best pattern at this length is searched again using the basic algorithm. If a better pattern is found, the above extension step will be repeated. Otherwise, this pattern is reported.

2.3.2 Choosing a Substitution Matrix

Two sets of substitution matrices, PAM series [19] and BLOSUM series [36], are often used in sequence alignment. As stated above, the purpose of sequence alignment is to find a set of target frequencies that generate all the pattern instances with high probability and other sequences with low probability. The best substitution matrix with pre-defined target frequencies to make such a distinction is one corresponding to the sequences' evolutionary distance. Therefore, different substitution matrices must be used for sequences at different evolutionary distance. Since BLOSUM matrices generally give better results than PAM matrices in local alignment, they are used as the default in COPIA. Other matrices can be used as an option by users.

2.3.3 Pattern Shift

As described in [50], pattern (phase) shift is a powerful method to escape a local optimum. The idea is that whenever a local optimum is reached, this pattern is compared with the set of subsequences shifted a few positions to both sides of the pattern to try to find a better one. This is especially useful for weak patterns. Generally, the weaker a pattern, the more local optimum around it. Sometimes several shifts could occur to reach the global optimum from a local one in my experiments. Since pattern shift is time-consuming, it is desirable to use it less frequently. Actually, it is generally enough to use it only on the best pattern reported by the basic algorithm (data not shown).

2.3.4 Constructing an Explicit PSSM

The main purpose of solving CP is to construct PSSMs that contain the conserved features belonging to a protein family. The PSSMs can then be used for database searching of related sequences. The consensus sequence constructed with the basic algorithm is essentially a PSSM, in which each column is implied in the underlying substitution matrix. A problem with the consensus sequence is that the evolutionary distance may be different at different positions. Thus the target frequencies defined in a single substitution matrix are often too restricted to be used for all positions in a pattern. In this case, an explicit PSSM is needed to represent the pattern. One way to construct the PSSM is to use the log-odd score of each residue in each column. In COPIA, a set of substitution matrix at different evolutionary distance level was used instead. In this method, the evolutionary distance of a column, which is estimated

by the frequency of the consensus character (see an example below), is used to determine the substitution matrix and the consensus character is used to determine the column to be used in the PSSM. The frequency of the consensus character in each column indicates the evolutionary distance at this column since we assume that all the residues in this column are evolved from it (and here the independency of each residue is important). For example, if we have consensus letter a at a position and its frequency at this column 0.6, we can choose the column for letter a in the substitution matrix BLOSUM60 as the position-specific scores in this position. Specifically, BLOSUM80, BLOSUM62, BLOSUM45, and BLOSUM30 are used in COPIA for the ratio being ≥ 0.8, ≥ 0.6, ≥ 0.4, and < 0.4, respectively. One small problem with this method is that, for different substitution matrix, the scale λ may be different. Bit score is used in COPIA to solve this problem.

A more complicated method is to computed the posterior probability using a set of substitution matrices as prior knowledge is described in [23], which is yet to be implemented in COPIA.

2.3.5 Finding Multiple Patterns in a Dataset

Many protein families studied so far contain more than one pattern. Between those patterns are varying length of unconserved regions which can not be aligned correctly. To search for multiple patterns, one possible way is to maintain more than one pattern at the same time as used in [50]. Here another method was used in COPIA. After the pattern instances are deleted, each sequence is cut into two (sub)sequences; the first part of all sequences becomes a new group, and the second half forms another. New patterns will be searched in these two groups separately. One problem with this method is that if two patterns are not collinear, i.e. not all instances are in the same order,some instances in one of them will not be able to be included in the construction of consensus sequence in the basic algorithm. This is not a serious problem for strong patterns since this is just like searching patterns in a group containing unrelated sequences. The instances in the other half can be found by an extra search using the consensus sequence. The explicit PSSM can then be constructed using all the instances (as above).

2.3.6 Finding Repeats

Finally, multiple instances of a motif in a sequence (repeats) can often be found in protein families. To detect repeats, an extra search step using the consensus sequence is conducted in each sequence to find any subsequences whose E-value is significant. These new instances can also be used in constructing PSSM for this pattern.

2.3.7 Statistical Significance of a Pattern

The basic algorithm can find a pattern within any set of sequences even if it is randomly generated. This is not attractive in practice since a "random" pattern is not interesting. To overcome this limitation, the statistical significance (E-value of occurring by chance alone) of each pattern will be reported. One option in COPIA is that it will stop if no more statistically

significant patterns (i.e. when their E-value is larger than a predefined value) can be found in the dataset. Since the E-value of a consensus pattern is hard to analyze. an estimator is used instead. This estimator is defined as

$$\hat{E} = K\bar{m}e^{-\lambda\bar{S}},$$

where $\bar{m}$ is the average sequence length and $\bar{S} = (S(A) - S(x_c))/(n-1)$ in which $S(x_c)$ is self-pair score of the consensus sequence.

The E-value of each pattern instance in each sequence is also computed by the Equation 1.2 based on its distance to the consensus sequence. This E-vaule is used to determine if the pattern instance is significant. If it is larger than a predefined value, this pattern instance will not be considered as a true occurrence. The threshold value can be input by users.

2.3.8 Setting Sequence Weight

Finally, a feature is yet to be fully implemented in COPIA is to set sequence weights. In the basic algorithm and construction of PSSMs, all sequences are assumed to be independent of each other, i.e., they are all in the similar evolutionary distance. This is generally not true especially in the existence of sequences sharing high similarity (these sequences are not sampled randomly from the sequence pool). Sometimes, two sequence can even have only a few different residues. Closely related sequences are often redundant in a dataset since they provide less information than their distant cousins. Sequence weighting methods will be helpful to reduce this redundancy by assigning more weight to distantly related sequences.

Many weighting schemes using an evolutionary tree for the set of sequences, which is constructed based on their pairwise similarity. The pairwise similarity is computed in COPIA with the following method: for any sequence pairs, compute their MSPs with score larger than a predefined value; their similarity is the ratio of total score of all significant MSPs to the total score of shorter sequence aligned with itself. A similar method has been used in [12]. Each MSP is found by a method similar to the one used in BLAST[2]: each pair of identical residues is used as the anchor point for extension on both sides. This method is much faster than the traditional dynamic programming method. Although the similarity score of sequences that are at high evolutionary distance (when similarity $< 10\%$) are generally underestimated by this method, it is not a problem here since they are simply considered to be independent.

For simplicity, instead of using an explicit weighting scheme, all sequences can be simply clustered according to their similarity. If a sequence has $> 40\%$ similarity to any sequence in a cluster, this sequence is clustered into this group. Only one sequence in each group is used in the alignment. An explicit weighting scheme needs to be used, however, in the construction of PSSM (see below). It should be indicated here that although it is often desirable to use a weighting scheme in multiple alignment, the performance of our program was not observed to be negatively affected even in the existence of many (50%) highly similar sequences (data not shown).

Chapter 3

Results and Discussion

3.1 Testing COPIA

The motif-finding process in COPIA, for simplicity, can be divided into two separate steps: (1) local alignment (2) constructing an explicit PSSM. The first step is to align motif instances correctly and the second step is to build a characteristic function of the motif. These two steps are not totally independent of each other and they are inseparable in practice. The performance and parameters of COPIA were first tested and refined with the 3 sets of test data used in [50]: HTH sequences, lipocalin sequences and isoprenyl-protein transferases (IPPTs).

To do an alignment, the first step is to choose a suitable substitution matrix. As discussed before, the best substitution matrix is one with the implicit target frequencies close to the residue frequencies in the target sequences. The target regions should be much more similar than others especially when only highly divergent sequences are used in the alignment. The good choice of a substitution matrix will greatly affect the pattern length determined by COPIA (see below). The testing on these sequences suggested that BLOSUM62 or BLOSUM40 should be a good choice for general use.

3.1.1 Determining Pattern Length

HTH is a group of DNA binding proteins. Each of them has a helix-turn-helix (HTH) structure of length 20. This is a set of highly divergent sequences. Their 3D structure is very different except for the HTH structure itself. Some statistics of these sequences is listed in Table 3.1.

Several parameters in COPIA especially those involved in determining the best pattern length were tested according to the correct alignment of HTH motifs. In my test, whether the optimal solution[1] gave the correct alignment strongly depended on the pattern length. The correct alignment of all motif instances can only be obtained at length from 20 to 22 using BLOSUM62. The correct alignment at length 22 produced by COPIA was shown in Figure 3.1.

[1] The solutions were considered to be optimal because COPIA always reported the same solutions even the starting points were selected randomly.

```
pattern length 22 average bits per letter  0.9
         consensus:              TQKEVAKMLGISQSTVSRWLKN
  1  Sigma-37      A25944  225  SQKETGDILGISQMHVSRLQRK    24.4  1e-05
  2  SpoIIIC       A28627  198  TQREIAKELGISRSYVSRIEKR    30.4  2e-07
  3  NahR          A32837   22  RVSITAENLGLTQPAVSNALKR    15.7  0.006
  4  Antennapedia A23450  326  RRIEIAHALCLTERQIKIWFQN    16.2  0.005
  5  NtrC (Brady) B26499  449  NQIRAADLLGLNRNTLRKKIRD    17.1  0.003
  6  DicA          B24328   22  TQRSLAKALKISHVSVSQWERG    25.3  3e-06
  7  MerD          C29010    5  TVSRLALDAGVSVHIVRDYLLR    10.2  0.1
  8  Fis           A32142   73  NQTRAALMMGINRGTLRKKLKK    20.3  8e-05
  9  MAT al        A90983   99  EKEEVAKKCGITPLQVRVWFIN    20.3  0.0001
 10  Lamda cII     A03579   25  GTEKTAEAVGVDKSQISRWKRD    21.7  3e-05
 11  Crp(CAP)      A03553  169  TRQEIGQIVGCSRETVGRILKM    24.0  1e-05
 12  Lamda Cro     A03577   15  GQTKTAKDLGVYQSAINKAIHA    17.1  0.0005
 13  P22 Cro       A25867   11  TQRAVAKALGISDAAVSQWKEV    29.4  8e-08
 14  AraC          A03554  196  DIASVAQHVCLSPSRLSHLFRQ     8.9  0.6
 15  Fnr           A03552  196  TRGDIGNYLGLTVETISRLLGR    19.4  0.0004
 16  HtpR          A00700  252  TLQELADRYGVSAERVRQLEKN    17.6  0.001
 17  NtrC (k.a.)   A03564  444  HKQEAARLLGWGRNTLTRKLKE    21.2  0.0002
 18  CytR          A24963   11  TMKDVALKAKVSTATVSRALMN    23.5  3e-05
 19  DeoR          A24076   23  HLKDAAALLGVSEMTIRRDLNN    23.0  3e-05
 20  GalR          A03559    3  TIKDVARLAGVSVATVSRVINN    28.1  1e-06
 21  LacI          A03558    5  TLYDVAEYAGVSYQTVSRVVNQ    20.3  0.0003
 22  TetR          A03576   26  TTRKLAQKLGVEQPTLYWHVKN    21.2  9e-05
 23  TrpR          A03568   67  SQRELKNELGAGIATITRGSNS    16.2  0.001
 24  NifA          S02513  495  VQAKAARLLGMTPRQVAYRIQI    13.0  0.06
 25  SpoIIG        S07337  205  TQKDVADMMGISQSYISRLEKR    32.2  5e-08
 26  Pin           S07958  160  PRQKVAIIYDVGVSTLYKRFPA     8.0  0.7
 27  PurR          S08477    3  TIKDVAKRANVSTTTVSHVINK    20.8  0.0002
 28  EbgR          S09205    3  TLKDIAIEAGVSLATVSRVLND    22.6  5e-05
 29  LexA          S11945   27  TRAEIAQRLGFRSPNAAEEHLK    11.2  0.09
 30  P22 cI        B25867   25  GQRKVADALGINESQISRWKGD    26.7  8e-07
                                 ********************
```

Figure 3.1: An output of COPIA showing the correct alignment of HTH motif. Columns from left to right are: sequence number, sequence name, NBRF/PIR accession number, starting position of the alignment in each sequence, scores (bits) to consensus sequence, E-value of each instance. The alignment was obtained using BLOSUM62. Asterisks (*) indicate the positions of known structural alignment [50].

Maximum length	524
Minimum length	61
Average length	246
Maximum similarity	48%
Minimum similarity	0
Average similarity	5%

Table 3.1: The statistics of 30 HTH sequences used to test COPIA.

In the alignment shown in Figure 3.1, the E-values[2] of several pattern instances are quite large. This suggested that their signals might be two weak to be aligned correctly in the lengths other than $20-22$. On the other hand, the misalignment produced by COPIA actually always had higher score than that of the correct one at a specific length such as 18 (data not shown). This might reflect the limitation of BLOSUM62 and the complexity of this problem.

The column score ratio used to extend a pattern defined in last chapter is dependent on the substitution matrix used. To get the best extension results, the substitution matrix must be selected based on the similarity of the sequences to be aligned. In HTH alignment, when BLOSUM62 is used, the pattern of length $20-22$ can only be obtained with a very small extension ratio starting at a shorter length such as 16. This small ratio may cause problems in other cases since it can increase noise, so it is not good choice. With BLOSUM40, however, the pattern of the good lengths $(20-22)$ can be obtained with a much higher ratio (> 0.4). The experiments on HTH and other sequences suggested that $1/(\log_{10} N + 1)$, where N is the number of sequences to be aligned, is a good ratio with an appropriate substitution matrix. Another question to be addressed is the number of columns to be extended at the same time. As in pairwise alignment a pair with a negative score might be included in an MSP, a column here with a low score should also be extended if the columns following it have a high score. It is thus better to extend more than one column at the same time. Experiments showed that two or three columns could be used for extension based on their average score with the same ratio as that for one column.

3.1.2 Constructing PSSM

The final task of COPIA is to construct an explicit PSSM, which is based on a set of substitution matrices at different evolution distance. The consensus sequence was constructed using all instances throughout the alignment process in the basic algorithm. To construct a PSSM, however, the insignificant instances (whose E-value is greater than a predefined number such as 0.01) were not used unless they have been known to be real instances. The explicit PSSM is then constructed as described in last chapter and used to search each sequence for motif instances.

[2]The E-value here is the expected number of occurrences by chance alone in this sequence. In contrast, the E-value reported by BLAST is the expected number of occurrence by chance in the whole database. The latter should be much larger than the former for the same motif instance.

A comparison of the instances found by a consensus sequence of HTH using BLOSUM40 and those by an explicit PSSM was shown in Figure 3.2. Two instances (in sequence 7 and 24) was misaligned using the consensus sequence but they were aligned correctly using the PSSM. Besides, the bit score of many sequences was increased by using PSSM. This is expected since the PSSM is constructed using BLOSUM80, BLOSUM62 and BLOSUM40, among which BLOSUM40 has the lowest entropy. The exclusion of insignificant motif instances in the construction of PSSM can increase selectivity though it might decrease sensitivity since some distantly related instances might be excluded. The 30 HTH sequences and 5 lipocalin sequences were put together and all the HTH motif instances could be correctly recognized using an explicit PSSM that was constructed with this set of sequences (data not shown).

3.1.3 Searching for Multiple Motifs and Repeats

COPIA was tested with lipocalin group for its ability to find multiple motifs. Lipocalin sequences contain two known motifs A and B. While their actual length is not clearly defined, there are a few conserved residues in each motif [50]. Correct alignment of those conserved residues could be obtained by COPIA with a large range of parameter settings, suggested the signal to noise ratio was large in these cases. (see Figure 3.2).

IPPTs are essential components of the cytoplasmic signal transduction network. Although no direct structural information is available on those sequences, it is likely that they contain 3 different motifs and each motif has several internal repeats [50]. Using the method mentioned in the previous section for finding extra copies, several copies of each motif can be found in each sequences (data not shown).

3.2 Case Study: vWF Family

The performance of COPIA was illustrated using the sequences in vWF family. The von Willebrand factor (vWF) is a multidomain protein [66] that is required for clot formation under conditions of high blood flow/shear. Its type A domain [75] was also found in the complement proteins factor B, C2, CR3, CR4, integrin α subunits, collagens VI, VII, XII and XIV, and other proteins [62]. Proteins that contain vWF domains play various roles in cell adhesion, migration, homing, pattern formation, and signal transduction and interact with a large array of ligands [17, 62, 10, 24, 51, 63].

vWF is defined in PRINTS database[3] [6] as a 3-element fingerprint that provides a signature for the superfamily. Forty-three sequences from vWF family were selected from PRINTS database. Some statistics of these sequences were summarized in Table 3.2

Using the default settings, COPIA reported 4 motifs in this set of sequences. The first three motifs correspond to the three elements in the fingerprint. Several repeats were found for each motif in the long sequences (data not shown), suggesting these proteins consist of multidomains. The repeats found in one of these sequences, CA36_HUMAN (SWISS-PROT accession number

[3]website: http://www.biochem.ucl.ac.uk/bsm/dbbrowser/PRINTS/PRINTS.html

pattern length 22				
1	225	16.6	20.5	24.4
2	198	24.6	28.5	30.4
3	22	9.3	15.9	15.7
4	326	13.4	12.2	16.2
5	449	15.4	17.4	17.1
6	22	22.2	23.9	25.3
7	9	9.0	12.4(5)	10.2
8	73	16.6	19.2	20.3
9	99	15.4	16.8	20.3
10	25	18.0	21.1	21.7
11	169	17.3	19.2	24.0
12	15	13.9	17.2	17.1
13	12	23.9	28.2	29.4
14	196	8.0	9.4	8.9
15	196	19.3	20.5	19.4
16	252	11.9	18.1	17.6
17	444	16.8	18.9	21.2
18	11	17.8	20	23.5
19	23	20.5	21.7	23.0
20	3	23.4	25.6	28.1
21	5	20.2	23	20.3
22	26	14.9	19.8	21.2
23	67	17.1	15.9	16.2
24	456	11.5	13.7(495)	13.0
25	205	24.4	31.1	32.2
26	160	8.5	8.3	8.0
27	3	17.3	17.9	20.8
28	3	20.2	23	22.6
29	27	8.0	13.3	11.2
30	25	24.1	26.7	26.7

Table 3.2: The comparison of HTH motif instances obtained using a consensus sequence and an explicit PSSM. Columns from left to right are: sequence number, starting position of each instance obtained using consensus sequence (based on BLOSUM40), bit score obtained using the consensus sequence, bit score obtained using the PSSM, bit score listed in Figure 3.1 (for comparison). The two numbers in parenthesis are the correct starting position of the instances obtained using the PSSM.

```
Motif A
pattern length 16 average bits per letter  1.8
 All motif instances matched to the consensus sequence:
 consensus          NFDISKFAGTWYEIAK
  1 ICYA_MANSE  17 DFDLSAFAGAWHEIAK    32.6  3e-08
  2 LACB_BOVIN  25 GLDIQKVAGTWYSLAM    26.7  2e-06
  3 BBP_PIEBR   31 NFDWSNYHGKWWEVAK    28.1  7e-07
  4 RETB_BOVIN  14 NFDKARFAGTWYAMAK    33.1  2e-08
  5 MUP2_MOUSE  27 NFNVEKINGEWHTIIL    19.8  0.0002
                           * *
Motif B
pattern length 17 average bits per letter  1.7
 All motif instances matched to the consensus sequence:
 consensus       WVLDTDYKNYLINYMCN
  1          104 WVLATDYKNYAINYNCD    38.1  6e-10
  2          109 LVLDTDYKKYLLFCMEN    25.8  3e-06
  3          115 NVLSTDNKNYIIGYYCK    26.7  2e-06
  4          105 WIIDTDYETFAVQYSCR    30.8  1e-07
  5          109 TIPKTDYDNFLMAHLIN    18.9  0.0004
                      **
```

Figure 3.2: The two motifs in lipocalin sequences reported by COPIA. The meaning of each columns is similar to Figure 3.1. * indicates the structurally conserved residues (see [50].

Average length	1280
Maximum length	3176
Minimum length	328
Average similarity	10%
Maximum similarity	91%
Minimum similarity	0

Table 3.3: The statistics of 43 vWF sequences used to test COPIA.

P12111, length: 3176), were shown in Figure 3.2. This protein has been found to contain a large N-terminal globule that consists of 9 consecutive 200 residue repeats (N-domains) and a small C-terminal globule that contains two C domains similar to the N domains [25]. The repeats reported by COPIA corresponded to this structure (Seven N domain repeats and two C domain repeats in motif 1; nine N domain repeats in motif 2; nine N domain repeats and one C domain repeats in motif 3). The repeats that were not reported by COPIA were due to insertions/deletions in them that make them unrecognizable by the consensus sequence without gaps being allowed.

It is worthy to mention here that vWF family were randomly selected from PRINTs database (no prior knowledge was based on). After running COPIA on these sequences, CA36_HUMAN was selected just because of its large number of repeats in each motif. Literature searching was then performed and it was found that this sequence has been well studied (with the structure mentioned above). Although COPIA did not find all the repeats in each motif since it did not allow gaps, it did give considerable amount of information about the internal structure of this sequence. This study suggested the potential usefulness of COPIA in the study of protein structure based on sequences themselves.

The main purpose of sequence alignment is to construct a consensus sequence (PSSM, or profile) for searching for distant homologies. The performance of a motif alignment program, therefore, can be evaluated based on its ability to construct an accurate profile to detect new family members in databases. The most significant motif in vWF has a length of 35 (see Figure 3.2, which appeared to be long enough for database searching. Since using an explicit PSSM to do database search is yet to be implemented in COPIA, the consensus sequence was used instead in BLAST searching for new vWF members. For comparison, twelve motif instances with various distance to the consensus sequences were also used for BLAST search. These motif instances and the consensus sequence were listed in Figure 3.4. The database search results were shown in Table 3.4.

```
Motif 1
         DIVFIIDGSGSIGPSNFEKVKNFISNIIERLDVGP
      39 DIIFLVDSSWTIGEEHFQLVREFLYDVVKSLAVGE    39.1  6e-09
     242 DIIFLIDGSNNTGSVNFAVILDFLVNLLEKLPIGT    43.6  2e-10
     445 DIVFLVDGSSALGLANFNAIRDFIAKVIQRLEIGQ    50.5  2e-12
     639 DIIFLLDGSANVGKTNFPYVRDFVMNLVNSLDIGN    48.7  7e-12
    1028 KDVVFLLDGSEGVRSGFPLLKEFVQRVVESLDVGQ    20.3  0.002
    1436 DIVFLIDSSEGVRPDGFAHIRDFVSRIVRRLNIGP    47.7  1e-11
    1639 DIVFLLDGSINFRRDSFQEVLRFVSEIVDTVYEDG    29.9  3e-06
    2402 ELAFALDTSEGVNQDTFGRMRDVVLSIVNVLTIAE    20.8  0.002
    2619 DMAFILDSAETTTLFQFNEMKKYIAYLVRQLDMSP    30.4  2e-06

Motif 2
         TQVALVQYSSEVRTEFSLNEYNNKEEVLSAVRNIKYMGGGTRTGSALQH
      76 FHFALVQFNGNPHTEFLLNTYRTKQEVLSHISNMSYIGGTNQTGKGLEY    55.5  6e-14
     279 IRVGVVQFSDEPRTMFSLDTYSTKAQVLGAVKALGFAGGELANIGLALD    38.1  1e-08
     482 IQVAVAQYADTVRPEFYFNTHPTKREVITAVRKMKPLDGSALYTGSALD    38.6  8e-09
     676 IRVGLVQFSDTPVTEFSLNTYQTKSDILGHLRQLQLQGGSGLNTGSALS    38.1  1e-08
     873 TRIAVAQYSDDVKVESRFDEHQSKPEILNLVKRMKIKTGKALNLGYALD    32.6  5e-07
    1065 VRVAVVQYSDRTRPEFYLNSYMNKQDVVNAVRQLTLLGGPTPNTGAALE    51.4  1e-12
    1269 TRVAVIQFSDDPKAEFLLNAHSSKDEVQNAVQRLRPKGGRQINVGNALE    37.7  1e-08
    1473 VRVGVVQFSNDVFPEFYLKTYRSQAPVLDAIRRLRLRGGSPLNTGKALE    34.5  1e-07
    1676 IQVGLVQYNSDPTDEFFLKDFSTKRQIIDAINKVVYKGGRHANTKVGLE    37.2  2e-08
    2443 ARVAVVTYNNEVTTEIRFADSKRKSVLLDKIKNLQVALTSKQQSLETAM    22.1  0.0007

Motif 3
         GARPGVPKVLVVITDGRSQDDV
     137 RAGDGVPQVIVVLTDGHSKDGL    30.8  2e-06
     341 RVEEGVPQVLVLISAGPSSDEI    25.8  5e-05
     544 RAAEGIPKLLVLITGGKSLDEI    30.4  2e-06
     738 RIREHVPQLLLLLTAGQSEDSY    21.7  0.001
     935 RIEDGVLQFLVLLVAGRSSDRV    18.9  0.006
    1127 RITEGVPQLLIVLTADRSGDDV    26.7  3e-05
    1331 RIEEGVPQFLVLISSGKSDDEV    26.2  4e-05
    1535 RIEDGVPQHLVLVLGGKSQDDV    27.6  2e-05
    1738 RLDQRVPQIAFVITGGKSVEDA    18.9  0.006
```

Figure 3.3: The repeats of 3 motifs in CA36_HUMAN found by COPIA. Each column from left to right: starting position, sequence, bit score to the consensus sequence, E-value of each repeat. Note that the consensus sequence is that obtained from all sequences (rather than these repeats only).

```
consensus          DIVFIIDGSGSIGPSNFEKVKNFISNIIERLDVGP
12 tr|O88493   37 DIVFLVDGSSSLGPSNFNAIRDFVTRVIQRLEIGQ    57.3  1e-14
 1 sp|P05099   39 DLVFIIDSSRSVRPQEFEKVKVFLSRVIEGLDVGP    56.4  5e-15
 3 sp|P21941   41 DLVFVVDSSRSVRPVEFEKVKVFLSQVIESLDVGP    55.1  1e-14
 4 sp|P13944 1199 DIVLLVDGSWSIGRPNFKTVRNFISRIVEVFDIGP    54.6  1e-13
 1 sp|P05099  272 DLVFLIDGSKSVRPENFELVKKFINQIVESLEVSE    50.9  2e-13
28 tr|O42401   54 DLVFIIDSSRSVRPEEFEKVKIFLSKMIDTLDVGE    50.9  2e-13
 7 sp|Q60847 1203 DIVLLVDGSWSIGRANFRTVRSFISRIVEVFEIGP    50.9  1e-12
 6 sp|O08746   57 DLVFIIDSSRSVNTYDYAKVKEFILDILQFLDIGP    46.4  1e-11
 7 sp|Q60847  140 DLVFLVDGSWSVGRNNFKYILDFIVALVSAFDIGE    42.3  6e-10
11 sp|P20785  613 DLLFVLDSSESIGLQNFQIAKDFIIKVIDRLSKDE    38.6  2e-09
36 tr|Q21540   47 EVILLLDASGSIGDDTFKKQLSFAMHLASRLNISE    34.0  3e-08
35 sp|Q04857  826 DITILLDSSASVGSHNFETTKVFAKRLAERFLSAG    30.8  5e-07
17 sp|P18614  170 QLDIVIVLDGSNSIYPWESVIAFLNDLLKRMDIGP    24.9  4e-05
19 sp|P15989  841 KDILFLIDGSANLLGSFPAVRDFIHKVISDLNVGP    19.4  0.005
```

Figure 3.4: The motif instances used in BLAST searching for vWF sequences. Each column from left to right is: sequence number in the original alignment (out of 43), NBRF/PIR accession number, starting position of each instance, bit score to the consensus sequence, E-value of the score in this sequence. Note that two repeats from sequence 1 and 7 were used and the low score of the last two instances were probably caused by insertion/deletions.

sequence used for BLAST search		number of sequences with E-value			Number of HSPs (E-value < 100)	
Accession number	Starting position	< 100	< 10	< 1	ungapped HSPs	All HSPs
consensus		345	328	282	310	573
O88493	37	299	270	252	289	525
P05099	39	282	163	91	267	474
P21941	41	252	135	59	242	436
P13944	1199	266	148	98	262	469
P05099	272	307	272	180	293	523
O42401	54	272	159	79	264	439
Q60847	1203	281	172	107	277	483
O08746	57	241	125	69	223	319
Q60847	140	269	126	95	264	480
P20785	613	275	186	92	271	490
Q21540	47	47	10	10	46	85
Q04857	826	146	106	96	146	221
P18614	170	118	79	66	23	143
P15989	841	67	37	30	12	105

Table 3.4: Comparison of BLAST search results.

MATCH-BOX(1.3)	http://www.fundp.ac.be/sciences/biologie/bms/matchbox_submit.shtml
ITERALIGN(1.1)	http://giotto.stanford.edu/~luciano/iteralign.html
MEME(3.0)	http://meme.sdsc.edu/meme/website/meme.html
CLUSTALW	http://www.ebi.ac.uk/clustalw/
BLOCKMAKER	http://bioinformatics.weizmann.ac.il/blocks/blockmkr/www/make_blocks.html
PIMA	http://searchlauncher.bcm.tmc.edu:9331/multi-align/Options/pima.html
DIALIGN(2.1)	http://bioweb.pasteur.fr/seqanal/interfaces/dialign2-simple.html
SAM(T99)	http://www.cse.ucsc.edu/research/compbio/HMM-apps/T99-tuneup.html

Table 3.5: Web address of multiple alignment program servers.

The consensus sequence was derived using BLOSUM62, and the BLAST search is also using BLOSUM62, so the results of the consensus sequence can be compared directly with the results obtained by each instance. From Table 3.4, we can see that the consensus sequence retrieved more sequences than any instances did, as expected. Generally speaking, the more distance to the consensus sequence an motif instance has, the fewer sequences it retrieved from the database. In addition, the E-values of the sequences retrieved by the consensus sequences declined. This is also expected since the consensus sequence contains information general to all instances. One problem is that the E-value of the first false positive retrieved by the consensus sequence also become smaller. An eye-checking of the original BLAST results (not shown) showed that some false positives appeared with E-value between $100 - 1000$ (the so-called twilight-zone [13], where the false positive/negative appears in sequence alignment). The first false positive appears within E-value of $200 - 500$ when the instances were used. The first false positive retrieved by the consensus sequence, however, had an E-value around 100 (another consensus sequence constructed from a set of randomly selected positive sequences retrieved by BLAST using the above consensus sequence was used to do BLAST searching, similar results were obtained, data not shown). These false positive sequences were downloaded from the database and added to a set of true positive sequences for motif searching using COPIA, the other two motifs can not be found in those false positive sequences (with E-value > 0.1). Therefore, it is possible to eliminate those false positives from the family using combination of all these motifs in database search.

In summary, this experiment showed that the significant motifs reported by COPIA were indeed conserved features in a sequence family and they can be used directly for database searching of new members.

3.3 Comparison with Other Motif-finding Programs

To further test the performance of COPIA, it was compared with several other motif-finding programs using several datasets with well defined motifs. The datasets were selected from vWF family, cytochrome P450 and HTHASNC family [78, 9] which is also from PRINTS database. The programs are listed in Table 3.5

A scoring method is required to evaluate the performance of these programs. In [1], the

IC score was used to evaluate the quality of an alignment. In most cases such as [11, 74], the programs were evaluated by their ability to align correctly the sequences with known structures. This latter method is a more practical way since our purpose is to find real motif; any objective function must be evaluated based on its ability to distinguish motif instances and non-instances. Although these two fingerprints are well defined in PRINTS database, their actual structural boundaries are not so clear. It is expected that different length of alignments will be generated by different programs. Hence the IC score is not good objective function for it is hard to compare it at different length. The performance of these programs, therefore, were evaluated here based on its ability to find motifs (and instances) rather than their ability to align all residues (especially those on both ends) correctly. For simplicity, all programs were tested with their default settings on their website. Since many programs limit their input size, small datasets were used in this test.

3.3.1 vWF Family

Fifteen sequences from vWF family were selected to compare COPIA with other programs (see Table 3.6 for some statistics).

Maximum length	2813
Minimum length	341
Average length	1013
Maximum similarity	85
Minimum similarity	0
Average similarity	$< 10\%$

Table 3.6: Statistics of 15 vWF sequences

COPIA reported at least one instance for all the three motifs in each of the 15 sequences (their lengths were different from those when 43 sequences were used, but they both contained the same most conserved regions, not shown). The results of COPIA and other programs were summarized in Table 3.7. Note that except COPIA, MEME was the only program that reported repeats. However, MEME and another motif-finding program, BLOCKMAKER, which implemented Gibbs sampling method, did not perform as well as COPIA for this dataset. Other programs did not give any information reflecting multidomains in some of the sequences, though some aligned one instance in every sequence correctly, suggesting that they were not suitable in aligning this group of sequences.

3.3.2 Cytochrome P5

Cytochrome P450 are a groups of paralogous genes present in all kingdoms. Several hundred of them have been found in plant genomes. These genes are involved in various reactions, including biosynthetic pathways and drug metabolism. Forty-two sequences were chosen from *Arabidopsis*

Program	Results
COPIA	All 3 motifs were correctly aligned
PIMA	All 3 motifs were correctly aligned
BLOCKMAKER	One motif was missed in both methods
MATCH-BOX	All 3 motifs were correctly aligned
MEME	One motif was not reported
ITERALIGN	Motif 3 in sequence 6 was misaligned
SAM	Last sequence was misaligned for all three motifs
CLUSTALW	Three sequences were misaligned for all motifs
DIALIGN	Motif 1 and 2 were not aligned together for all sequences*
WCONSENSUS	Misaligned $3-5$ sequences for each motif

Table 3.7: Comparison of COPIA with other programs using vWF sequences. A motif was considered to be found if it was correctly aligned (though the aligned length might be different for each program) since several programs reported only a global alignment. No repeats were reported by other programs except MEME. * The motif instances in first two sequences were aligned with the first repeats in the last sequences, and the motifs instances in sequence $3-6$ were aligned with the second repeats in the last sequence.

Maximum length	576
Minimum length	457
Average length	506
Maximum similarity	35
Minimum similarity	0
Average similarity	10

Table 3.8: Statistics of 42 Cytochrome P450 sequences

thaliana to compare the performance of COPIA and the other motif-finding programs. Some statistics of these sequences are shown in Table 3.8.

Although these sequences are highly divergent, the structures of the sequences in the P450 family appeared to be conserved [35]. Several highly conserved regions has also been found in P450 sequences, which include the PERF, K-helix, and heme-binding domains. In the 42 sequences used here, several completely conserved residues were included in these regions. The similar length, containing no repeats and conserved residues make these sequences extremely suitable for global alignment [74]. In this case, the comparison of these programs were based on if they aligned the conserved residues correctly. The results of COPIA and the other programs are summarized in Table 3.9.

Several global alignment programs, aligned all these conserved residues correctly. COPIA reported 4 significant motifs covering all the conserved regions. The most significant one is shown in Figure 3.5. This region is the heme-binding domain, which was shown to contain a

Program	Results
COPIA	Misaligned one residue in one sequence
PIMA	Misaligned one residue in two sequences
BLOCKMAKER	Two sequences were not included in its alignment
MATCH-BOX	Completely missed one residue in all sequences
MEME	Misaligned one residue in one sequence
ITERALIGN	Aligned all (conserved) residues correctly
SAM	Aligned all (conserved) residues correctly
CLUSTALW	Aligned all (conserved) residues correctly
DIALIGN	Aligned all (conserved) residues correctly
WCONSENSUS	Misaligned several sequences

Table 3.9: Comparison of COPIA with other programs using Cytochrome P450 sequences

	1	2	3	4	5	6	7	8
1 P19494	-	100	99	100	0	20	0	0
2 P37424		-	99	99	0	20	0	0
3 P37403			-	98	0	21	0	0
4 P37425				-	0	20	0	0
5 S63256					-	0	0	0
6 P03809						-	7	6
7 P15360							-	83
8 P22866								-

Table 3.10: Sequence similarities in HTHASNC group.

signature sequence FxxGxxxCxG in the sequences studied [14].

COPIA missed one of the conserved regions that included a conserved letter in one of the sequences because of an insertion in this region. MEME also missed the same instance. The performance of BLOCKMAKER was a little bit worse in that it missed several instances (not shown).

3.3.3 HTHASNC Family

HTHASNC is a 3-element fingerprint that provides a signature for the HTH motif of the asnC bacterial regulatory proteins [78, 9]. Eight sequences are randomly selected from this family; all of them contain these 3 motifs. Their sequence similarities were shown in Table 3.10.

COPIA reported a motif for 7 sequences (except sequence 5, which is excluded because of its high E-value). This motif contains all the three elements in the fingerprint (Figure 3.6).

It is interesting to mention that sequence 5 was found not to be the sequence that was specified in the database, and actually it did not contain the fingerprint. This was a link error

```
pattern 1 length 19 average bits per letter  2.1
 All motif instances matched to the consensus sequence:
 consensus          FKFLPFGAGRRVCPGKELA
  1 CYP98A3    426 FRLLPFGAGRRVCPGAQLG    49.0  9e-13
  2 CYP97B3    509 FAFLPFGGGPRKCIGDQFA    42.2  1e-10
  3 CYP96A1    448 FKFLSFNAGPRTCLGKEVA    39.0  9e-10
  4 CYP94B1    496 FKFPVFQAGPRVCIGKEMA    36.3  7e-09
  5 CYP93D1    428 EKMMSFGAGRRSCPGEKMV    39.7  6e-10
  6 CYP90A1    406 NVFTPFGGGPRLCPGYELA    43.4  4e-11
  7 CYP89A2    437 IKMMPFGAGRRICPGIGLA    46.6  5e-12
  8 CYP88A3    427 GAFLPFGAGSHLCPGNDLA    41.4  2e-10
  9 CYP86A1    444 YKFVAFNAGPRTCLGKDLA    36.6  5e-09
 10 CYP85A1    404 NSCFVFGGGTRLCPGKELG    37.5  2e-09
 11 CYP84A1    446 FEFIPFGSGRRSCPGMQLG    46.8  4e-12
 12 CYP83A1    430 YEFIPFGSGRRMCPGMRLG    45.3  1e-11
 13 CYP82C2    450 FELMPFGSGRRSCPGSSLA    47.1  4e-12
 14 CYP81D1    428 QKLLAFGLGRRACPGSGLA    40.7  3e-10
 15 CYP79A2    449 LNIISFSAGRRGCMGVDIG    30.0  5e-07
 16 CYP78A5    447 LRLAPFGSGRRVCPGKAMG    44.6  2e-11
 17 CYP77A6    445 VKMMPFGIGRRICPGLAMA    44.9  2e-11
 18 CYP76C1    438 YELTPFGAGRRICPGMPLA    46.8  4e-12
 19 CYP75B1    433 FELIPFGAGRRICAGLSLG    43.4  4e-11
 20 CYP74A     458 PETETPTVGNKQCAGKDFV    14.4  0.02
 21 CYP73A5    435 FRYVPFGVGRRSCPGIILA    45.8  8e-12
 22 CYP72A10   473 VSFFPFAWGPRICIGQNFA    34.9  2e-08
 23 CYP724A1   412 KKTTAFGGGVRVCPGGELG    37.8  2e-09
 24 CYP722A1   411 NSFLAFGMGGRTCLGLALA    35.1  1e-08
 25 CYP71A12   427 LNFIPFGSGRRICPGINLA    46.8  4e-12
 26 CYP718     420 YTYLPFGGGPRLCAGHQLA    41.7  1e-10
 27 CYP716A1   412 YTYVPFGGGPRMCPGKEYA    45.1  1e-11
 28 CYP715A1   455 MGYMPFGFGGRMCIGRNLT    36.6  5e-09
 29 CYP714A1   468 QSFVPFGLGTRLCLGKNFG    37.1  4e-09
 30 CYP712A1   444 FRYLPFGSGRRGCPGASLA    47.8  2e-12
 31 CYP711A1   455 YAFIPFGIGPRACVGQRFA    39.2  8e-10
 32 CYP710A1   422 RNFLAFGWGPHQCVGQRYA    29.0  9e-07
 33 CYP709A1   452 RHFIPFAAGPRNCIGQQFA    35.1  1e-08
 34 CYP708A1   434 KTFMAFGGGARLCAGAEFA    37.1  4e-09
 35 CYP707A1   399 NTFMPFGNGTHSCPGNELA    39.7  5e-10
 36 CYP706A1   449 FKYLPFGSGRRICAAINMA    41.4  2e-10
 37 CYP705A1   437 LNFLPFGSGRRMCPGSNLG    45.8  8e-12
 38 CYP704A1   437 FKFISFHAGPRICIGKDFA    36.6  5e-09
 39 CYP702A1   405 RTYIPFGAGSRQCVGAEFA    39.5  6e-10
 40 CYP701A3   442 HKTMAFGAGKRVCAGALQA    36.1  7e-09
 41 CYP51A2    405 FSYIAFGGGRHGCLGEPFA    33.6  4e-08
 42 CYP51A1    422 CSYISLGAGRHECPGGSFA    30.2  4e-07
                         *   *
```

Figure 3.5: The most significant motif of P450 sequences reported by COPIA. Asterisks(*) indicate the two totally conserved residues in these sequences.

```
pattern length 50 average bits per letter  1.4
       consensus:       LDRIDRNILNELQKDGRISNVELSKRVGLSPTPCHERVRRLERQGFIQGY
1  LRP_ECOLI P19494  11 LDRIDRNILNELQKDGRISNVELSKRVGLSPTPCLERVRRLERQGFIQGY 115.9 2.9e-33
2  LRP_KLEPN P37424  11 LDRIDRNILNELQKDGRISNVELSKRVGLSPTPCLERVRRLERQGFIQGY 115.9 2.9e-33
3  LRP_SALTY P37403  11 LDRIDRNILNELQKDGRISNVELSKRVGLSPTPCLERVRRLERQGFIQGY 115.9 2.9e-33
4  LRP_SERMA P37425  11 LDRIDRNILNELQKDGRISNVELSKRVGLSPTPCLERVRRLERQGFIQGY 115.9 2.9e-33
5            S63256 146 DDRVDKKFVSQIQKNVDLLQFPWLNAIKYRPTSVKLLKTTVPIVSKKRQK 6.6   1.48
6 ASNC_ECOLI P03809   6 IDNLDRGILEALMGNARTAYAELAKQFGVSPGTIHVRVEKMKQAGIITGA 46.8 8.9e-13
7 GYLR_STRCO P15360   8 LERAAAMLRLLAGGERRLGLSDIASSLGLAKGTAHGILRTLQQEGFVEQD 19.8  1.9e-4
8 GYLR_STRGR P22866   8 LERAAAMLRLLAGGERRLGLSDIASTLGLAKGTAHGILRSLQAEGFVEQE 19.4  2.7e-4
                        *****************
                                    ************
                                                ********************
```

Figure 3.6: The motif reported for HTHASNC family by COPIA. Column meaning is the same as before. Asterisks(*) indicate the 3 motifs (in the order of 1, 2 and 3) in the PRINTS fingerprint.

in the database website. The output of each program is summarized in Table 3.11. In this case, Except SAM, the other programs all misaligned some instances.

3.3.4 Summary

As indicated in [74] that multiple alignment methods they tested worked poorly for sequences with low identities, the programs tested here also misaligned some motif instances due to low sequence similarity. Although the last dataset (HTHASNC) might not be considered as a typical case in motif-finding because of the presence of nearly identical sequences in it, it was a randomly selected dataset. In general, COPIA performed very well and comparable to (better in some cases and in illustrating protein internal structures than) the existing best motif-finding programs.

3.4 Running Time of COPIA

It is hard to analyze the running time of COPIA theoretically. Here only a rough analysis is given. For each r (3 in COPIA) sequences, it takes $O(rm^2)$ to find a fixed number of starting points. For each starting position, it takes $O(kLnm)$ to run to convergence, where k is small number (generally < 5). The total number of subsets of r sequences selected is generally much fewer than n. Totally, the running time is not likely larger than $O(nm^2)$. (For comparison, MEME's estimated running time is about $O(n^2m^2)$). It appeared that the running time is affected more by sequence length than sequence number. My experiments confirmed this (data not shown). For small datasets such as HTH, lipocalin and HTHNSHC, a few seconds were needed to report all the motifs on a machine using Pentium-III 700 MHz processor. The running time for the small dataset of vWF (about 15200 characters (average length 1013) and more than six different motifs were reported) was less than 2 minutes on the same machine. The cytochrome P450 dataset used similar amount of time. For the large dataset of vWF (about 55,000 residues, average length 1280), the running time was about 10 minutes.

Program	The result of each program
COPIA	All instances were aligned correctly
PIMA	Element 1 was misaligned
BLOCKMAKER (Gibbs Sampling)	No elements in sequence 6,7,8 were aligned correctly, misaligned part of element 1 and 3
BLOCKMAKER (Motif)	No elements in sequence 7,8 were aligned correctly
MATCH-BOX	Element 1 in sequence 7, 8 was not aligned, element 2, 3 in sequence 6, 7, 8 were misaligned
MEME	No elements in sequence 6,7,8 were aligned due to their large E-value
ITERALIGN	Element 1 were not aligned in sequence 7,8
SAM	All elements were aligned correctly
CLUSTALW	All elements were aligned correctly
DIALIGN	Misaligned element 1 in sequence 7 and 8

Table 3.11: Comparison of multiple alignment programs using HTHASNC sequences

Chapter 4

Conclusion and Future Work

This thesis presents a new scoring method and a randomized algorithm for finding consensus patterns in unaligned protein sequences This scoring scheme is a natural extension of that used in pairwise alignment for finding MSPs. One main advantage of the scoring method is that the motif-searching and determination of the best motif length can be done at the same time. This algorithm has been implemented in a software COPIA. Studies using sequences from vWF family showed that it is useful especially in illustrating the internal structures of multidomain proteins based on sequence comparisons. Comparison with other multiple alignment programs showed that the performance of COPIA is comparable to (or even better than) them in motif-finding. There are several other points worthy of mentioning for this method.

First, it uses simple additive scores so it is easily understandable in the context of evolutionary distance. Meanwhile, it is established on a solid statistical background. The reason behind this is that the model is implied by the substitution matrix. The consensus sequence itself is essentially a profile, which implies the probability of a residue occurring at each position.

Secondly, this method is iterative. Iterative methods generally performs better than progressive ones, especially for the sequences sharing low identity, but at the expense of computation time [74]. This algorithm essentially belongs to the class of EM algorithms. The main problem in motif-finding by EM algorithm is the existence of many local optima in the search space. In the algorithm described here, the number of points in the search space is $O(m^n)$, for n sequences of length m. An exhaustive search is only practical for a few sequences with today's computational power. One solution to the local optima problem is to select a good starting point. Gibbs sampling method selects the starting point randomly. MEME chooses each subsequence as a starting point. Compared with Gibbs sampling method and MEME algorithm, the problem of selecting good starting points is well solved in COPIA. The main reason for this is that the prior knowledge from substitution matrices provides considerable amount of information to help find good starting points. The selection of good starting points makes this algorithm much faster in practice than Gibbs sampling or MEME since it uses much fewer starting points to approximate the optimal solution (it can generally find the optimal solution quickly in my experimental study). Experiments on real data showed that the method used in COPIA gave a much better performance than choosing a starting point randomly or one by one from each

single input sequence.

Thirdly, this method can be easily extended. This method reports a consensus sequence (based on a substitution matrix) and all (real) motif instances. The consensus sequence can be used to do database search directly. The motif instances can be used to construct a PSSM, profile or a profile HMM. Although any other methods can also do this, this simple, fast and accurate method make it a better choice in practice than other methods.

Finally, for future work, COPIA can be improved through the use of dynamic programming that allows gaps when searching for repeats. This will be helpful in finding motif instances with insertions/deletions as those found in vWF sequences. One feature will be added in COPIA is to do database searching using explicit PSSMs for more distantly related sequences.

Appendix A

Biological Background

Cells are the structural units of all free-living life forms. There are two basic cell types: prokaryote and eukaryote; Their difference is that the genetic material is stored in an organelle called nucleus in the latter, while the single circular DNA of prokaryotes floats freely around the cell.

A *gene* is a functional unit that determines a particular biological trait or character. Genes are located on *chromosomes*, which each contains a huge molecule, DNA. Each chromosome (DNA) generally contains many genes. The set of non-homologous[1] chromosomes in an organism constitutes its genome. Almost all information necessary for building and maintaining life is encoded in its genome, i.e., the underlying DNA molecules.

DNA is a double-stranded molecule; each strand consists of a sequence of four different nucleotide bases: A (adenine), C (cytonine), G (guanine), T (thymine). A, C are always paired with T, G in another strand, respectively, i.e., the two strands are complementary. If we have one strand, therefore, we can easily get another. This property of making copy of itself easily makes DNA a very good material for storing and passing information.

To build and maintain life, i.e., to realize the functionalities encoded in DNA, however, another molecule, protein, is needed. A *protein* is a molecule that consists of a sequence of amino acids, which have 20 different basic types (see Table A). Proteins are structural components of life and key active molecule in all biochemical processes. A protein is the product of a gene. The sequence of a protein is determined by its corresponding gene (DNA) sequence. Each amino acid is encoded by a three-letter DNA sequence called codon. Since there are 4^3 different combinations of 3-letter sequences (in which 3 are called stop codons that do not encode any amino acids) to encode 20 amino acids, there is a redundancy in the encoding — different codons can encode the same amino acid. Not all regions of DNA sequences in a genome are genes. Some regions have no coding function (not to be decoded into protein). The functions of those regions in large part remain to be elucidated. On the other hand, not all genes encode proteins; some exceptions are ribosome RNA (rRNA), transfer RNA (tRNA) and small nuclear RNA (snRNA) genes.

The simple central dogma of genetics is modeled as follows:

[1] Many organisms have two similar copies for each chromosome. They are homologous.

Amino Acid	Abbreviation		Codons
	Three Letter	One Letter	
Alanine	Ala	A	GCU, GCC, GCA, GCG
Leucine	Leu	L	UUA, UUG, CUU, CUC, CUA, CUG
Isoleucine	Ile	I	AUU, AUC, AUA
Valine	Val	V	GUU, GUC, GUA, GUG
Proline	Pro	P	CCU, CCC, CCA, CCG
Phenylalanine	Phe	F	UUU, UUC
Tryptophan	Trp	W	UGG
Methionine	Met	M	AUG
Glycine	Gly	G	GGU, GGC, GGA, GGG
Serine	Set	S	UCU, UCC, UCA, UCG, AGU, AGC
Threonine	Thr	T	ACU, ACC, ACA, ACG
Tyrosine	Tyr	Y	UAU, UAC
Cysteine	Cys	C	UGU, UGC
Asparagine	Asn	N	AAU, AAC
Aspartic acid	Asp	D	CAA, CAG
Glutamine	Gln	Q	GAU, GAC
Glutamic acid	Glu	E	GAA, GAG
Lysine	Lys	K	AAA, AAG
Arginine	Arg	R	CGU, CGC, CGA, CGG, AGA, AGG
Histidine	His	H	CAU, CAC

Table A.1: The 20 basic amino acids, their abrievations and codons

where mRNA is an intermediate molecule to pass information from DNA to protein. *Replication* is the process that DNA replicates itself (to make exactly the same copy of itself) for passing information to next generation of cells or organisms. *Transcription* is the process of passing information from DNA to mRNA. *Translation* is the process of passing information from mRNA to protein. The process of translation occurs in ribosome; two other RNA molecules, rRNA and tRNA, are also required to participate this process.

There are two other forms of organelles that also contain genomes: Mitochondria and chloroplasts; the former exists in both animals and plants to convert energy from foodstuffs into forms that can be used by the cell directly through a process called *respiration*, and the latter exists in plants to convert sunlight to foodstuffs through *photosynthesis*. The organelles likely evolved from bacteria that were endocytosed long time ago (billions of years). Although they maintain their own genomes, many genes encoding mitochondrial and chloroplast proteins exist in nuclear genomes.

Both mitochondrial and chloroplast genomes are circular. Animal mitochondrial genomes are 10-20 kB in length, and encode 13 proteins used for energy production, as well as 22 tRNAs and 2 rRNAs. Plant mitochondrial genomes are much larger (up to several hundred kb) and contain additional genes. Many organisms use one genetic code to translate nuclear mRNAs, and a second one for their mitochondrial mRNAs. Chloroplast genomes are 120 - 200 kB in length and contain about 120 genes, which encode ribosomal RNAs and proteins, tRNAs, and proteins involved in photosynthesis. Chloroplast mRNAs are translated with the standard genetic code. Their mRNAs often undergo extensive RNA editing, so it is difficult to predict the protein translations from genomic sequence.

Virus is a non-free living life form. It must use the genetic machinery of free living organisms to produce progenies. The genomes of viruses are generally very small (several kb) and encode only a few genes.

Evolution is the process of fixing mutations in genetic materials in populations; it is the basis for all our study of species, genomes, genes and proteins. The evolution history is generally represented by a tree, called *phylogenetic tree*, in which the root represents the common ancestor of all species or sequences and each internal node represents the common ancestor of some species or sequences. The root and all internal nodes are not observed today (with the exception of fossils).

Bibliography

[1] T. Akutsu, H. Arimura, and S. Shimozono. On approximation algorithms for local multiple alignment. In *RECOMB'2000*, pages 1–7, Tokyo, Japan, April 2000.

[2] S.F. Altschul. Amino acid substitution matrices from an information theoretic perspective. *J. Mol. Biol.*, 219:555–565, 1991.

[3] S.F. Altschul and W. Gish. Local alignment statistics. *Methods in Enzymology*, 266:460–480, 1996.

[4] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403–10, 1990.

[5] R. Arratia, P. Morris, and M.S. Waterman. Stochastic scrabble: large deviations for sequences with scores. *J. Appl. Prob.*, 25:106–119, 1988.

[6] T.K. Attwood, D.R. Flower, A.P. Lewis, J.E. Mabey, S.R. Morgan, P. Scordis, Selley J., and W. Wright. PRINTS prepares for the new millennium. *Nucleic Acids Res.*, 27(1):220–225, 1999.

[7] L.T. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pages 28–36, Menlo Park, California, 1994. AAAI Press.

[8] L.E. Baum. An equality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3:1–8, 1972.

[9] A. Bolotin and S. Biro. Nucleotide sequence of the putative regulatory gene and major promoter region of the streptomyces griseus glycerol operon. *Gene*, 87(1):151–152, 1990.

[10] P. Bork and K. Rohde. More von Willebrand factor type a domains? sequence similarities with malaria thrombospondin-related anonymous protein, dihydropyridine- sensitive calcium channel and inter-alpha-trypsin inhibitor. *Biochem. J.*, 279:908–910, 1991.

[11] P. Briffeuil, G. Baudoux, C. Lambert, X. De Bolle, E. Vinals, C. Feytmans, and E. Depiereux. Comparative analysis of seven multiple protein sequence alignment servers: clues to enhance reliability of predictions. *Bioinformatics*, 14:357–366, 1998.

[12] L. Brocchieri and S. Karlin. A symmetric-iterated multiple alignment of protein sequences. *J. Mol. Biol.*, 276:249–264, 1998.

[13] R. Burkhard. Twilignt zone of protein sequence alignments. *Protein Engineering*, 12:85–94, 1999.

[14] C. Chapple. Molecular-genetic analysis of plant cytochrome P450-dependent monooxygenases. *Annu. Rev. Plant Physiol. Plant Mol. Biol.*, pages 311–343, 1998.

[15] L. Chen, A.L. DeVries, and C.-H.C. Cheng. Convergent evolution of antifreeze glycoproteins in Antarctic notothenioid fish and Arctic cod. *Proc. Natl. Acad. Sci. USA*, 94:3817–3822, 1997.

[16] W. Chen and K. Struhl. Saturation mutagenesis of a yeast his3 tata element: genetic evidence for a specific TATA binding protein. *Proc. Natl Acad. Sci. USA*, 85:2691–2695, 1988.

[17] A. Colombatti, P. Bonaldo, and R. Doliana. Type A modules: interacting domains found in several non-fibrillar collagens and in other extracellular matrix proteins. *Matrix*, 13:297–306, 1993.

[18] International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.

[19] M.O. Dayhoff, R.M. Schwartz, and B.C. Orcutt. A model of evolutionary change in proteins. In M. O. Dayhoff, editor, *Atlas of Protein Sequence and Structure*, volume 5, Suppl. 3, pages 345–352. National Biomedical Research Foundation, Washington D.C., 1978.

[20] E. Depiereux, G. Baudoux, P. Briffeuil, I. Reginster, X. De Bolle, C. Vinals, and E. Feytmans. Match-Box server: a multiple sequence alignment tool placing emphasis on reliability. *Comput. Appl. Biosci.*, 13(3):249–256, 1997.

[21] J.L. Desseyn, J.P. Aubert, N. Porchet, and A. Laine. Evolution of the large secreted gel-forming mucins. *Mol. Biol. Evol.*, 17(8):1175–1184, 2000.

[22] W.F. Doolittle. Lateral genomics. *Trends Cell Biol.*, 9(12):M5–8, 1999.

[23] R. Durbin, S. Eddy, A. krogh, and G. Mitchison. *Biological sequence analysis*. Cambridge University Press, Cambridge, UK, 1998.

[24] Y.J.K. Edwards and S.J. Perkins. The protein fold of the von Willebrand factor type A domain is predicted to be similar to the open twisted beta-sheet flanked by alpha-helices found in human ras-p21. *FEBS Lett.*, 358:283–286, 1995.

[25] M.-L. Chu *et al.* Mosaic structure of globular domains in the human type VI collagen alpha 3 chain: similarity to von Willebrand factor, fibronectin, actin, salivary proteins and aprotinin type protease inhibitors. *EMBO J.*, 9(2):385–93, 1990.

[26] Fleischmann, et al. Whole-genome random sequencing and assembly of Haemophilus influenzae. *Science*, 269:495–511, 1995.

[27] C.M. Fraser, J.D. Gocayne, O. White, M.D. Adams, and R.A. et al. Clayton. The minimal gene complement of Mycoplasma genitalium. *Science*, 270(5235):397–403, 1995.

[28] A.J. Gentles and S. Karlin. Genome-scale compositional comparisons in eukaryotes. *Genome Research*, 11(4):540–546, 2001.

[29] M. Gerstein. A structural census of genomes: Comparing bacterial, eukaryotic, and archaeal genomes in terms of protein structure. *J. Mol. Biol.*, 274:562–576, 1997.

[30] A. Goffeau. Life with 482 genes. *Science*, 270(5235):445–446, 1995.

[31] O. Gotoh. An improved algorithm for matching biological sequences. *J. Mol. Biol.*, 162:705–708, 1982.

[32] M. Gribskov, R. Luthy, and D. Eisenberg. Profile analysis. *Methods in Enzymology*, 183:146–159, 1990.

[33] M. Gribskov, A.D. McLanchlan, and D. Eisenberg. Profile analysis: detection of distantly related proteins. *Proc. Nat. Aca. Sci.*, 84:4355–4358, 1987.

[34] R. Hardison. Hemoglobins from bacteria to man: evolution of different patterns of gene expression. *J. Exp. Biol.*, 201(8):1099–1117, 1998.

[35] C.A. Hasemann, R.G. Kurumbail, S.S. Boddupalli, J.A. Peterson, and J. Deisenhofer. Structure and function of cytochromes P450: a comparative analysis of three crystal structures. *Structure*, 3(1):41–62, 1995.

[36] S. Henikoff and J.G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, 89:10915–10919, 1992.

[37] S. Henikoff, J.G Henikoff, W.J. Alford, and S. Pietrokovski. Automated construction and graphical presentation of protein blocks from unaligned sequences. *Gene*, 163:GC17–26, 1995.

[38] G.Z. Hertz and G.D. Stormo. Identification of consensus patterns in unaligned dna and protein sequences: a large-deviation statistical basis for penalizing gaps. In H.A. Lim and C.R. Cantor, editors, *Proceedings of the Third International Conference on Bioinformatics and Genome Research*, pages 201–216, Singapore, 1995. World Scientific Publishing Co., Ltd.

[39] G.Z. Hertz and G.D. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15:563–577, 1999.

[40] B.C. Hoopes, LeBlanc J.F., and D.K. Hawley. Contributions of the TATA box sequence to rate-limiting steps in transcription initiation by RNA polymerase II. *J. Mol. Biol.*, 277:1015–1031, 1998.

[41] J. Hudak and M.A. McClure. A comparative analysis of computational motif-detection methods. In *Proc. Pacific Symp. Biocomputing '99(PSB'99)*, pages 138–149, 1999.

[42] R. Hughey and A. Krogh. Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *CABIOS*, 12(2):95–107, 1996.

[43] I. Jonassen, J.F. Collins, and D.G. Higgins. Finding flexible patterns in unaligned protein sequences. *Protein Sci.*, 4(8):1587–95, 1995.

[44] S. Karlin and S.F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl. Acad. Sci. USA*, 87:2264–68, 1990.

[45] K. Karplus, C. Barrett, and R. Hughey. Hidden markov models for detecting remote protein homologies. *Bioinformatics*, 14(10):846–856, 1999.

[46] E.V. Koonin, L. Aravind, and A.S. Kondrashov. The impact of comparative genomics on our understanding of evolution. *Cell*, 101:573–576, 2000.

[47] E.V. Koonin, A.R. Mushegian, Galperin M.Y., and Walker D.R. Comparison of archaeal and bacterial genomes: computer analysis of protein sequences predicts novel functions and suggests a chimeric origin for the archaea. *Mol. Microbiol.*, 25(4):619–37, 1997.

[48] A. Krogh, M. Brown, I.S. Mian, K. Sjolander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.*, 235:1501–1531, 1994.

[49] C. Lawrence and A. Reilly. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins*, 7:41–51, 1991.

[50] C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald, and J.C. Wootton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.

[51] J.O. Lee, P. Rieu, M.A. Arnaout, and R. Liddington. Crystal structure of the A domain from the alpha subunit of integrin CR3 (CD11b/CD18). *Cell*, 80:631–638, 1995.

[52] M. Li, B. Ma, and L. Wang. Finding similar regions in many strings. In *Proc. 30th ACM Symp. Theory of Computing (STOC'99)*, pages 473–482, 1999.

[53] E.M. Marcotte, M. Pellegrini, H.L. Ng, D.W. Rice, T.O. Yeates, and D. Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. *Science*, 285(5428):751–753, 1999.

[54] M.A. McClure, T.K. Vasi, and W.M. Fitch. Comparative analysis of multiple protein-sequence alignment methods. *Mol. Biol. Evol.*, 11(4):571–592, 1994.

[55] B. Morgenstern. DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics*, 15:203–210, 1999.

[56] R.M. Myers, K. Tilly, and T. Maniatis. Fine structure genetic analysis of a β-globin promoter. *Science*, 232:613–618, 1986.

[57] S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequences of two proteins. *J. Mol. Biol.*, 48:444–453, 1970.

[58] A.F. Neuwald, J.S. Liu, D.J. Lipman, and C.E. Lawrence. Extracting protein alignment models from the sequence database. *Nucleic Acids Res.*, 25:1665–1677, 1997.

[59] H. Ochman, J.G. Lawrence, and E.A. Groisman. Lateral gene transfer and the nature of bacterial innovation. *Nature*, 405(6784):299–304, 2000.

[60] S. Pascarella and P. Argos. Analysis of insertions / deletions in protein structures. *J. Mol. Biol.*, 224:461–68, 1992.

[61] L. Patthy. *Protein Evolution.* Blackwell Science Ltd., Oxford, 1999.

[62] S.J. Perkins, K.F. Smith, S.C. Williams, P.I. Haris, D. Chapman, and R.B. Sim. The secondary structure of the von Willebrand factor type A domain in factor B of human complement by Fourier transform infrared spectroscopy. its occurrence in collagen types VI, VII, XII and XIV, the integrins and other proteins by averaged structure predictions. *J. Mol. Biol.*, 238:104–119, 1994.

[63] A. Qu and D.J. Leahy. Crystal structure of the I-domain from the CD11a/CD18 (LFA-1, alpha L beta 2) integrin. *Proc. Natl. Acad. Sci. USA*, 92:10277–10281, 1995.

[64] J.A. Rosinski and W.R. Atchley. Molecular evolution of helix-turn-helix proteins. *J. Mol. Evol.*, 49(3):301–9, 1999.

[65] F.P. Roth, J.D. Hughes, P.W. Estep, and G.M. Church. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat. Biotechnol.*, 16(10):939–945, 1998.

[66] Z.M. Ruggeri and J. Ware. von Willebrand factor. *FASEB J.*, 7:308–316, 1993.

[67] R.F. Smith and T.F. Smith. Automatic generation of primary sequence patterns from sets of related protein sequences. *Proc. Natl. Acad. Sci. USA*, 87:118–122, 1990.

[68] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.

[69] G. Stormo and G.W. Hartzell. Identifying protein-binding sites from unaligned DNA fragments. *Proc. Natl. Acad. Sci. USA*, 88:5699–5703, 1991.

[70] T. Suzuki and K. Imai. Evolution of myoglobin. *Cell Mol. Life Sci.*, 54(9):979–1004, 1998.

[71] R.L. Tatusov, M.Y. Galperin, D.A. Natale, and E.V. Koonin. The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Res.*, 28(1):33–36, 2000.

[72] T.A. Tatusova, I. Karsch-Mizrachi, and J.A. Ostell. Complete genomes in WWW Entrez: data representation and analysis. *Bioinformatics*, 15:536–543, 1999.

[73] J.D. Thompson, D.G. Higgins, and T.J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, 22:4673–4680, 1994.

[74] J.D. Thompson, F. Plewnlak, and O. Poch. A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res.*, 27:2682–2690, 1999.

[75] D. Tuckwell. Evolution of von Willebrand factor A (VWA) domains. *Biochemical Society Transactions*, 27(part 6):835–840, 1999.

[76] J.C. Venter et al. The sequence of the human genome. *Science*, 291(5507):1304, 2001.

[77] H.T. Wareham, T. Jiang, C.G. Trendall, and X. Zhang. Stochastic heuristic algorithms for target motif identification. In *Proc. Pacific Symp. Biocomputing (PSB'00) 2000*, 2000.

[78] D.A. Willins, C. Ryan, J.V. Platko, and J.M. Calvo. Characterisation of LRP, an Escherichia coli regulatory protein that mediates a global response to leucine. *J. Biol. Chem.*, 266(17):10768–10774, 1991.

[79] C.R. Wobbe and K. Struhl. Yeast and human TATA-binding proteins have nearly identical DNA sequence requirements for transcription in vitro. *Mol. Cell Biol.*, 10:3859–3867, 1990.

[80] Y.I. Wolf, A.S. Kondrashov, and E.V. Koonin. Interkingdom gene fusions. *Genome Biology*, 1(6):1–13, 2000.

[81] F. Wolfertstetter, K. Frech, G Herrmann, and T. Werner. Identification of functional elements in unaligned nucleic acid sequences by a novel tuple search algorithm. *Comput. Appl. Biosci.*, 12(1):71–80, Feb 1996.

[82] D. Yean and J. Gralla. Transcription reinitiation rate: a special role for the TATA box. *Mol. Cell. Biol.*, 7(17):3809–3816, 1997.