Motif extraction from complex data : case of protein classification

Rabie Saidi

▶ To cite this version:

Rabie Saidi. Motif extraction from complex data : case of protein classification. Bioinformatics [q-bio.QM]. Université Blaise Pascal - Clermont-Ferrand II, 2012. English. NNT : 2012CLF22272 . tel-02063250

HAL Id: tel-02063250
https://theses.hal.science/tel-02063250

Submitted on 11 Mar 2019

University of Clermont-Ferrand II

LIMOS - CNRS UMR 6158

Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes

PHD THESIS

To obtain the title of

PhD of Science

Specialty : Computer Science

Defended by

Rabie Saidi

Motif Extraction from Complex Data: Case of Protein Classification

Prepared at LIMOS

Defended on October 3^{rd}, 2012

Jury :

Reviewers :	
Pr. Mohammed Javeed Zaki	Rensselaer Polytechnic Institute, USA
Pr. Florence d'Alché-Buc	University of Evry, France
Dr. Henry Soldano	University of Paris-Nord, France
Advisor :	
Pr. Engelbert Mephu Nguifo	University of Clermont-Ferrand II, France
Co-Advisor :	
Pr. Mondher Maddouri	University of Gafsa, Tunisia
Examiners :	
Pr. Rumen Andonov	University of Rennes I, France
Pr. Abdoulaye Baniré Diallo	University of Québec, Canada
Pr. David Hill	University of Clermont-Ferrand II, France

Acknowledgment

This thesis would not have been possible without the help, support and patience of my advisors, Prof. Mondher Maddouri and Prof. Engelbert Mephu Nguifo, not to mention their advice and unsurpassed knowledge in data mining and machine learning.

Special thanks are directed to Prof. Florence d'Alché-Buc, Dr. Henry Soldano and Prof. Mohammed Zaki for having accepted to review my thesis manuscript and for their kind efforts to comply with all the administrative constraints. My thanks go as well to Prof. Rumen Andonov, Prof. Abdoulaye Baniré Diallo and Prof. David Hill for having accepted to act as examiners of my thesis.

Thanks to Prof. Alain Quillot for having hosted me at the LIMOS Lab and allowed me working in comfortable conditions, with financial, academic and technical support.

I would also like to thank my beloved mother, brothers, sisters and all my family. They were always supporting me and encouraging me with their best wishes. Their faith in me allowed me to be as ambitious as I wanted and helped me a lot through the past years. It was under their watchful eye that I gained so much drive and ability to tackle challenges head on.

Special thanks go to my beloved late father, wishing he was able to see this important step of my life, and feel his pride of his son.

On a more personal note, I thank my brave Tunisian people and especially the martyrs and the woundeds of the Tunisian Revolution for having offered me a novel clean air of liberty after decades of dictatorship. Thank you for having given me my homeland back, thank you for having lighted the first candle in the Arab obscurity, thank you for being the first building block towards a democratic, united and modern Arab world, a world where Arabic is a language of science, innovation and progress.

At last but certainly not least, I would like to thank my friends, they were always there cheering me up and standing by me through both good and bad times.

Clermont-Ferrand, France, 2012
Rabie Saidi
رَبيع سَعِيدِي

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Contents

Goals

This chapter summarizes the contents and describes the plan of the thesis. First, we highlight the emergence of bioinformatics, and we state the issue of data preprocessing in the scope of protein classification using data mining. Then, we present some useful information that may help with the reading of the manuscript, such as the main assumptions, the interchangeably used terms, and succinct information about the appendices and the glossary.

1.1 Context and motivation

1.1.1 Bioinformatics emergence

The emergence of the bioinformatics that we have witnessed during the last years finds its origin in the technological progress which has helped to conduct large scale research projects. The most remarkable one was the human genome project (HGP) [Baetu 2012] accomplished in 13 years since 1990; a period that seems to be very short compared with the quantity of the collected data on the human genome: 3 billion bases which constitute the human DNA. Thus, several problems are open:

- How does the gene express its protein?
- Where does the gene start and where does it end?
- How do the protein families evolve and how to classify them?
- How to predict the three-dimensional structure of proteins?
- etc...

The answer to these questions by the biochemical means and the *in vitro* analysis is very expensive and time consuming. Indeed, some tasks, such as the determination of the protein three-dimensional structure, can extend over months and even years whereas the biological sequences quantity generated by the various sequencing programs knows an exponential growth. Henceforth, the challenge is not the gathering of biological data but rather their exploration in a faster and efficient way making it faster to reveal the secrets of the cell . This explosive growth of the amount of biological data requires, therefore, the use of computer resources for the storage, organization, maintenance and analysis. In order to make biological data available to scientists in computer-readable forms, many generalized and specialized databases have been constructed, and been growing exponentially. Moreover a panoply of computational tools have been developed to analyze biological data, especially for the search of similarities between biological data.

1.1.2 Protein classification issue in bioinformatics

Due to their crucial importance, proteins have been the subject of thorough studies in bioinformatics. Proteins play crucial roles in almost every biological process and they are responsible in one form or another for a variety of physiological functions including enzymatic catalysis, binding, transport and storage, immune protection, control of growth, etc. This important position of proteins in the mechanisms has made the analysis and interpretation of proteins a fundamental task in bioinformatics. Classification and prediction techniques have been utilized as one way to deal with such task [Bhaskar 2005].

In bioinformatics, the inference of new knowledge from significant similarity has become a considerably reliable routine [Pearson 2005]. Alignment has become the main technique used by biologists to look for similarity between structures, and hence to classify new ones into already known families/classes. Whenever two protein sequences or protein structures are similar, they can be considered to belong to the same class. However, the inference of classes from alignment may include some weakness such as the *orphan proteins* issue [Ekman 2010], the lack of discriminative models taking into account the classification scope and the disuse of additional information (contextual, topological..). This explains the recourse to the use of alternative means from other fields namely from data mining. Indeed, data mining provides a panoply of algorithms and techniques that can help with the problem of protein classification.

1.1.3 Data mining and preprocessing issue

Bioinformatics is a data-rich field but lacks a comprehensive theory of life's organization, at the molecular level that allows to effectively analyze biological data. In the framework of data mining, many software solutions were developed for the extraction of knowledge from tabular data (which are typically obtained from relational databases). These solutions could help with the investigation of bioinformatics data.

In fact, protein classification has been cast as a problem of data mining, in which an algorithm classifies new structures based on what it learns from an already available classification (For example the SCOP database [Andreeva 2004]). Work on protein classification has been ongoing for over a decade using data mining classifiers, such as neural networks [Cai 2000, Ding 2001, Huang 2003, Ie 2005] and support vector machines (SVM) [Chen 2006, Melvin 2007, Shamim 2011]. However, knowing that protein data are presented in complex formats and that mining tools often process data under the relational format, it will not be possible to apply these tools directly on such data, *i.e.*, a preprocessing step is seen essential.

The solutions to address the problem of format come from data mining itself. Methodological extensions of data preprocessing have been proposed to deal with data initially obtained from non-tabular sources, *e.g.*, in the context of natural language (text mining) and image (image mining). Data mining has thus evolved following a scheme instantiated according to the type of the underlying data (tabular data, text, images, etc.), which, at the end, always leads to working on the classical double entry tabular format where instances are encoded based on a set of attributes. Feature extraction (or motif extraction) is one major way to address the attribute creation. However, the main challenge in any preprocessing process is the loss of information that accompanies the format change.

1.2 Contributions

This thesis deals with the protein data preprocessing as a preparation step before their classification. We present motif extraction as one way to address this task. The extracted motifs are used as descriptors to encode proteins into feature vectors. This enables using known data mining classifiers which require this format. However, designing a suitable feature space, for a set of proteins, is not a trivial task due to the complexity of the raw data. We deal with two kinds of protein data *i.e.*, sequences and tri-dimensional structures.

1.2.1 First axis: sequential protein data

In the first axis *i.e.*, protein sequences, we propose a novel encoding method, termed *DDSM* that uses amino-acid substitution matrices to define similarity between motifs during the extraction step. We demonstrate the efficiency of such approach by comparing it with several encoding methods using some data mining classifiers. We also propose new metrics to study the robustness of some of these methods when perturbing the input data. These metrics allow to measure the ability of the method to reveal any change occurring in the input data and also its ability to target the interesting motifs.

1.2.2 Second axis: spatial protein data

The second axis is dedicated to 3D protein structures which are recently seen as graph of amino acids. We make a brief survey on the most used graph-based representations and we propose a naïve method to help with the protein graph making. We show that some existing and widespread methods present remarkable weaknesses and do not really reflect the real protein conformation. Besides, we have been interested in discovering recurrent sub-structures in proteins which can give important functional and structural insights. We propose a novel algorithm to find spatial motifs, termed *ant-motifs*, from protein. The extracted motifs obey a well-defined shape which is proposed based on a biological basis. We compare ant-motifs with sequential motifs and spatial motifs of recent related works.

1.3 Outline

This thesis is organized as follows. In Chapter 2, we provide the required material to understand the basic notions of our two research fields, namely data mining and bioinformatics. We also give a panorama of the main biological applications of data mining. This chapter is mainly dedicated to readers who are not familiar with biological terms.

In Chapter 3, we introduce the problem of protein classification seen within a data mining framework. We overview the classification concept and we present its most known algorithms, evaluation techniques and metrics. Meanwhile, we present the importance of protein classification in bioinformatics and we explain the necessity of preprocessing relative to the complexity and the format of bioinformatics data under consideration.

In Chapter 4, we deal with the motif-based preprocessing of protein sequences for their classification. We propose a novel encoding method that uses amino-acid substitution matrices to define similarity between motifs during the extraction step. We carry out a detailed experimental comparison (in terms of classification accuracy and number of attributes) between several encoding methods using various kinds of classifiers (C4.5 decision tree, naïve bayes NB, support vertor machines SVM and nearest neighbour NN), the Hidden-Markov-Model-based approach as well as the standard approach based on alignment.

In Chapter 5, we introduce the notion of stability of the generated motifs in order to study the robustness of motif extraction methods. We express this robustness in terms of the ability of the method to reveal any change occurring in the input data and also its ability to target the interesting motifs. We use these criteria to experimentally evaluate and compare four existing extraction methods for biological sequences.

In Chapter 6, we make a brief survey on various existing graph-based representations and propose some tips to help with the protein graph making since a key step of a valuable protein structure learning process is to build concise and correct graphs holding reliable information. We, also, show that some existing and widespread methods present remarkable weaknesses and do not really reflect the real protein conformation.

In Chapter 7, we propose a novel algorithm to find spatial motifs from protein structures by extending the Karp-Miller-Rosenberg (KMR) repetition finder dedicated to sequences. The extracted motifs obey a well-defined shape which is proposed based on a biological basis. These spatial motifs are used to perform various supervised classification tasks on already published data. Experimental results show that they offer considerable benefits, in protein classification, over sequential motifs and spatial motifs of recent relative works. We also show that it is better to enhance the data preprocessing rather than to focus on the optimization of classifiers.

In Chapter 8, we conclude this thesis by summarizing our contributions and highlighting some prospects.

1.4 Main Assumptions of the thesis

To allow a better understanding to the reader, we list the main assumptions that we adopt in this thesis. These assumptions are given in order of appearance in the manuscript. They can be seen as the dimensions of the thesis.

1. This thesis is **not** about classification, but about preprocessing for the classification.
2. The more complex the data are, the more required the preprocessing is.
3. Any preprocessing is accompanied by a loss of information contained in the raw data.
4. Motif extraction can efficiently contribute in protein preprocessing for classification, where motifs are used as features.
5. The more reliable the set of features is, the higher the classification performance is.
6. The more reliable the feature extraction method is, the more sensitive to variations in data it is.
7. 3D protein structures contain useful spatial information that can be expressed by graph, *i.e.*, a protein can be seen as a graph of amino acids.
8. Spatial information can be wasted if proteins are not parsed into graph in a "judicious" way.
9. It is more judicious to limit the spatial motifs to a specific shape, rather than frequent subgraphs.

1.5 Interchangeably used terms

In this thesis, as well as in literature, many terms are used interchangeably even if slight subtleties, related to the context, may exist between them. In Table 1.1, we list these terms into clusters and we give to each cluster a general meaning.

1.6 Appendices and glossary

Four appendices are provided at the end of the manuscript. In Appendix A, we describe the bioinformatics data formats and tools we used in our experiments. In Appendix B, we describe the SeqCod library. This library comprises methods (comprising DDSM) to encode biological sequences (DNA and protein) into relational or binary formats. Methods have been developed

Table 1.1: Interchangeably used terms.

General meaning	**Terms**
A piece of data that describes an object	Motif, feature, descriptor, attribute, pattern
A group of objects	Class, family, group, cluster
A process of extracting useful knowledge from data	Data mining, DM, knowledge discovery in data, KDD
Data, in computer formats, issued from biology	Bioinformatics data, biological data
Spatial data	3D structure, tertiary structure, spatial structure
Motif represented as graph	Spatial motif, frequent subgraph
A set of objects described by the same set of attributes	Relational format, tabular format, object-attribute table, context
Assigning a label, from a set of label, to an object	Classification, supervised classification, prediction, affiliation

in C language. In Appendix C, we explain the bases of the original KMR algorithm. In Appendix D, we provide a description of the Protein Graph Repository (PGR), our online repository mainly dedicated to protein graphs. In Appendix E, we explain how to use our software of spatial motif (ant-motif) extraction implemented in java language.

Chapter 2

Bioinformatics & Data Mining: Basic Notions

Contents

Goals

This chapter introduces our two intersecting research fields, namely bioinformatics and data mining. It is dedicated to present, in a simplified way, the basic notions related to these fields. We mainly focus on defining bioinformatics data, we show their complexity, give an idea about their usual tools of storage and processing. We also overview the main tasks performed by data mining techniques in bioinformatics. Those who are familiar with these notions can skip this chapter.

2.1 Bioinformatics

Bioinformatics is made up of all the concepts and techniques necessary to interpret biological data by computer. Several fields of application or subdisciplines of bioinformatics have been formed [Ouzounis 2003]:

- Sequence bioinformatics, which deals with the analysis of data from the genetic information contained in the sequence of DNA or the protein it encodes. This branch is particularly interested in identifying the similarities between the sequences, the identification of genes or biologically relevant regions in the DNA or protein, based on the sequence or sequence of elementary components (nucleotides, amino acids).

- Structural bioinformatics, which deals with the reconstruction, the prediction or analysis of the 3D structures or the folding of biological macromolecules (proteins, nucleic acids), using computer tools.

- Network bioinformatics, which focuses on interactions between genes, proteins, cells, organisms, trying to analyze and model the collective behavior of sets of building blocks of living. This part of bioinformatics in particular feeds of data from technologies for high-throughput analysis such as proteomics and transcriptomics to analyze gene flow or metabolic.

- Statistical bioinformatics and population bioinformatics, whose the ultimate goal is to statistically identify significant changes in biological processes and data for the purpose of answering biological questions.

In other words, it is about analyzing, modeling and predicting biological information from experimental data. In a broader sense, the concept of bioinformatics may include the development of tools for information processing based on biological systems, for example, the use of combinatorial properties of the genetic code for the design of DNA computers to solve complex algorithmic problems [Kahan 2008].

2.2 Bioinformatics data

Bioinformatics data revolve around three biological macromolecules. The central dogma of molecular biology, detailed in [Tortora 2006], describes these biological macromolecules and the flow of genetic information between them (Fig. 2.1). There exist three kinds of bioinformatics data related to the three mentioned macromolecules, namely, DNA, RNA and protein. DNA is transcribed into RNA and the RNA is then translated into proteins. From a

Table 2.1: Bioinformatics data and their alphabets.

Type	Data	Alphabet
Nucleic	DNA	{A, T, C, G}
	RNA	{A, U, C, G}
Protein	Protein	{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}

computational perspective, these data can be seen as computer-readable structures defined within given alphabets detailed in the following subsections and summarized in Table 2.1.

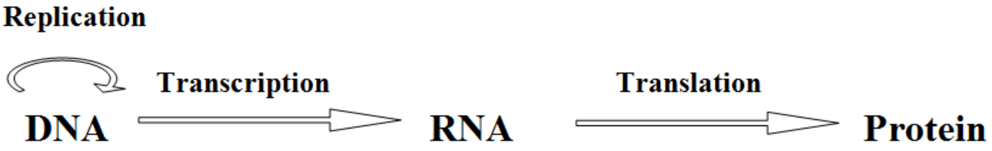


Figure 2.1: Simplified process of transcription and translation. The circular arrow around DNA denotes its ability to replicate.

2.2.1 Nucleic data: DNA and RNA

2.2.1.1 DNA

DNA (deoxyribonucleic acid) has a double helical twisted structure. Each side of the spiral of DNA is a polymer constructed of four parts, called nucleotides (or bases): A, T, C, and G (abbreviations for adenine, the thymine, cytosine and guanine). Both sides of the DNA are complementary, *i.e.*, whenever there is an edge of T, there is A in the corresponding position on the other side, so if there is a G on one side, there is a C in the corresponding position of the other (Fig. 2.2). DNA can be represented by a sequence of four nucleotides.

2.2.1.2 RNA

Such as DNA, RNA is a long but usually simple molecule , except when it folds in on itself. It differs chemically from DNA by containing the sugar ribose instead of deoxyribose and containing the base uracil (U) instead of thymine. Thus, the four RNA bases are A, C, G and U.

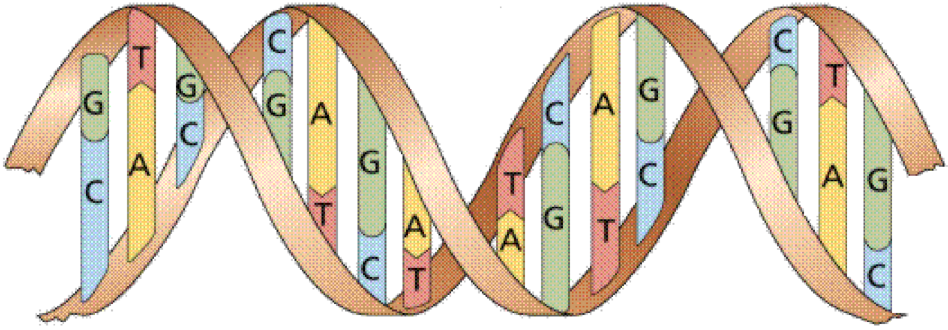


Figure 2.2: DNA structure.

2.2.2 Protein data

Proteins are biological macromolecules formed by concatenation of 20 distinct amino acids into long chains. They play crucial roles in almost every biological process. They are responsible in one form or another for a variety of physiological functions including enzymatic catalysis, binding, transport and storage, immune protection, control of growth, etc.

The sequence of the amino acid residues in these chains is termed the protein *primary structure*. These chains can fold to form complex 3D structures due to a combination of chemical interactions with the existence of some standard sub-structures called *secondary structures* (α helix and β sheet). In the final folded state of a protein *i.e.*, *tertiary structure*, residues that are far away in the chain can be very close in space. Often, proteins are composed of several chains of amino acids. This is the case of hemoglobin, which contains four protein chains, or insulin which has two chains linked by disulfide bonds (see Chapter 4). The combination of these chains that each has a tertiary structure, is the quaternary structure of these proteins, also called *oligomeric structure* (Fig. 2.3).

A protein consists of a set of 20 amino acids. Each amino acid is represented by a letter: alanine (A), cysteine (C), aspartic acid (D), glutamic acid (E), phenylalanine(F), glycine (G), histidine (H), isoleucine (I) , lysine (K), leucine (L), methionine (M), asparagine (N), proline (P), glutamine (Q), arginine (R), serine (S), threonine (T), valine (V), tryptophan (W) and tyrosine (Y). All amino acids share a common structural scheme. An amino acid is composed of a central (but not the centroid) carbon atom called C_α and four chemical groups attached to C_α: a hydrogen atom, an amino group, a carboxyl group and a side chain or radical R (Fig. 2.4). It is the side chain that differentiates one amino acid from another and gives it its physico-chemical properties. The common parts between the amino acids compose the so called backbone [Brandon 1991]. 2.4).

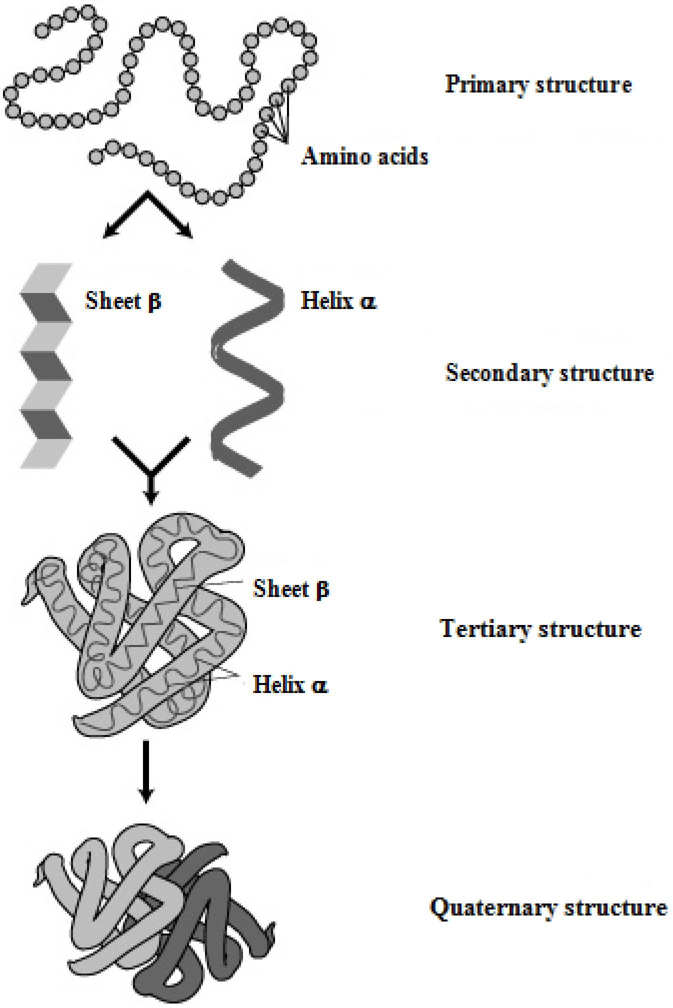


Figure 2.3: Protein structures.

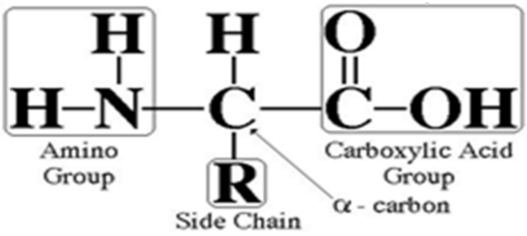


Figure 2.4: Amino acid structure.

2.3 Databases

Recently, the collection of biological data has increased at explosive rates, due to the improvements of existing technologies and the introduction of new technologies such as microarrays [Mohapatra 2011]. These technological advances have helped conduct experiments and research programs on a large scale. An important example is the human genome project (HGP) [Baetu 2012], which was founded in October 1990 by the the Department of Energy and the National Institutes of Health (NIH) of the United States. This project was completed in 2003 and has seen the collaboration of other countries such as France, Germany and Canada.

The explosive growth of the amount of biological data requires, therefore, the use of computer resources for the storage, organization, maintenance and analysis of these data. In order to make biological data available to scientists in computer-readable forms, many generalized and specialized databases have been constructed, and been growing exponentially. Fig. 2.5 illustrates the exponential growth of some known databases.

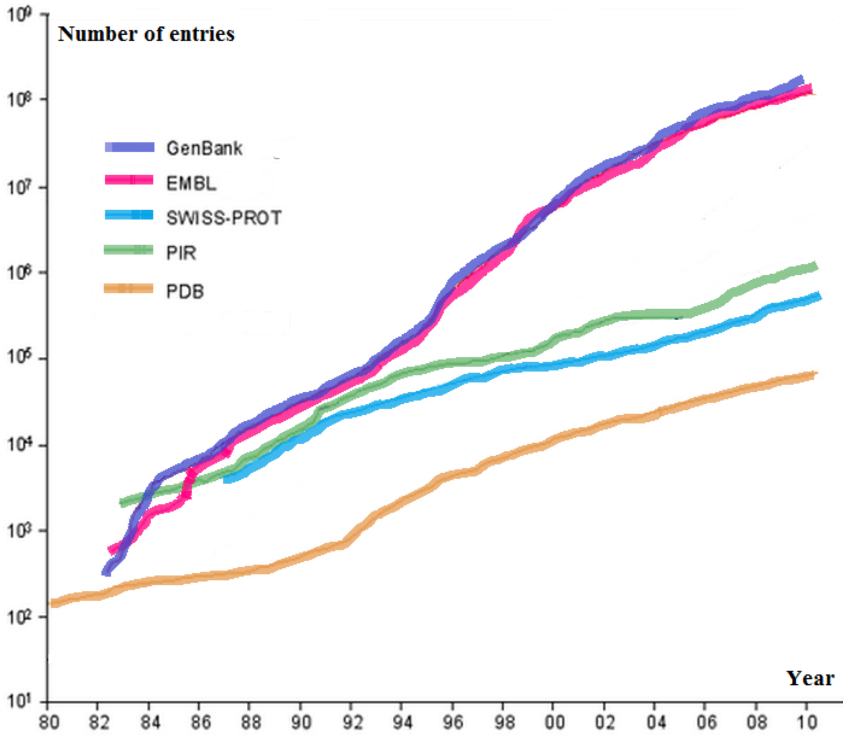


Figure 2.5: Growth of biological databases.

For specific requirements related to the activities of research groups, many specific databases have been created in laboratories. Some have continued to be developed; others have not been updated and disappeared as they represented a specific need. Still others are unknown or poorly known and waiting to be operated more. All these specialized databases of interest are very diverse and the mass of data they represent may vary considerably from one base to another. Generally, they aim to:

- Identify families of sequences around specific biological characteristics such as regulatory signals, the promoters of genes, peptide signatures or identical genes from different species.
- Group specific classes of sequences such as cloning vectors, restriction enzymes, and all sequences of the same genome.

In fact, these databases are, in many cases, improvements or combinations compared to data from the general bases. For example, the Protein Data Bank (PDB) consists of molecules whose 3D coordinates were obtained by magnetic resonance or X-ray diffraction [Berman 2007]. These structures can be easily visualized using 3D visualization software. Below are some other examples of specialized databases:

- ECD: nucleic sequences of Escherichia coli [Kroger 1998].
- TFD: nucleic consensus motifs [Ghosh 2000].
- PROSITE: protein motifs with biological activity [Sigrist 2010].
- SCOP: structural classification of proteins [Andreeva 2004].
- CATH: hierarchical classification of proteins [Orengo 2002].
- IMGT: immunoglobulin sequences and T-receptors [Lefranc 2003].
- GENATLAS: mapping information of human genes [Frézal 1998].

These databases are replete with standard formats for representing biological data. Those standards that have been successfully adopted by the bioinformatics community are associated with software tools which can perform analysis, integration and visualization of data which comply with community-accepted formats. Many formats have been created over the years. The FASTA format is the most common for sequential data and the PDB format is the most common to represent 3D-structures (see Appendix A).

2.4 Similarity search

2.4.1 Similarity and homology

The *similarity* is the resemblance between two or more structures. It can be measured, in a simple way, as the percentage of identical elements in these structures. The *homology* implies that structures derive from a common ancestral structure and have the same evolutionary history (retained functions for example). A high similarity is taken as evidence of homology on the existence of a common ancestor.

The search for similarities between structures is a fundamental operation which is often the first step of biological data analysis. It is widely used in the search of motifs, the characterization of common or similar regions between two or more structures, the comparison of a structure with all or a subset of a sequence database, or even the analysis of molecular evolution. The similarity search operation can be performed by an *alignment* program.

2.4.2 Alignment

Alignment is a procedure used to identify identical or very similar regions between two or more structures, and to distinguish those that are meaningful and correspond to biological meanings from those observed by chance. Formally an alignment can be defined as follows:

Definition 1 (Alignment) *Let $\mathcal{S} = \{S_1, .., S_k\}$ be a set of k structures defined within a given alphabet Σ such that $S_i = \langle x^i_1, .., x^i_{|S_i|}\rangle$, $1 \leq i \leq k$. An alignment $\mathcal{A}(S_1, .., S_k)$ is a matrix:*

$$\mathcal{A}(\mathcal{S}) = \begin{bmatrix} a^1_1 & .. & a^1_q \\ \vdots & & \vdots \\ a^k_1 & .. & a^k_q \end{bmatrix}$$

such that:

$$\begin{cases} a^i_j \in \Sigma \cup \{-\},\ \textit{where}\ 1 \leq i \leq k\ \textit{and}\ 1 \leq j \leq q \\ \textit{For a given}\ j\ \in [1, q],\ a^i_j \in \{a\} \cup \{-\},\ \textit{where}\ \ a \in \Sigma \\ \max(|S_i|) \leq q \leq \sum_{i=1}^{k} |S_i| \\ \nexists\ j \in [1, q]\ \mid\ \forall\ i\ \in [1, k]\ a^i_j\ \ = - \\ \langle a^i_1, .., a^i_q\rangle \setminus \{a^i_j = -\} = S_i \end{cases}$$

Example 1 *Let be $\mathcal{S} = \{S_1, S_2, S_3\}$ such that $S_1 = \langle A, G, V, S, I, L, N, Y, A\rangle$, $S_2 = \langle V, S, I, L, Y, A, K, R\rangle$ and $S_3 = \langle A, G, I, L, A, K, R, F\rangle$. An alignment example $\mathcal{A}$ of $\mathcal{S}$ is:*

$$\mathcal{A}(\mathcal{S}) = \begin{bmatrix} A & G & V & S & I & L & N & Y & A & - & - & - \\ - & - & V & S & I & L & - & Y & A & K & R & - \\ A & G & - & - & I & L & - & - & A & K & R & F \end{bmatrix}$$

	A	**C**	**G**	**T**
A	1	0	0	0
C	0	1	0	0
G	0	0	1	0
T	0	0	0	1

	A	**C**	**G**	**T**
A	3	0	0	2
C	0	3	2	0
G	0	2	3	0
T	2	0	0	3

Figure 2.6: Two examples of DNA substitution matrix.

2.4.3 Scoring and substitution matrices

Generally, an alignment score is calculated to qualify and quantify the similarity between structures. It can measure either the distance or the closeness of structures. This score is computed based on elementary scores that take into account all possible states according to the alphabet used in the description of the structures. These matrices are called *substitution matrices*.

Example 2 (Simple scoring) *Let be $S_1 = \langle V, S, I, L, Y, A, K, R \rangle$, $S_2 = \langle A, G, I, L, A, K, R \rangle$ and an alignment example $\mathcal{A}$ of S_1 and S_2:*

$$\mathcal{A}(S_1, S_2) = \begin{bmatrix} V & S & I & L & Y & A & K & R \\ A & G & I & L & - & A & K & R \end{bmatrix}$$

A simple way to score this alignment is to reward matches by x, and penalize mismatches by y. Hence, the score of this alignment is $5x - 3y$.

Definition 2 (Substitution matrix) *Given an alphabet Σ, a substitution matrix $\mathcal{M}$ over Σ is the function defined as below:*

$$\mathcal{M} : \left| \begin{array}{ccc} \Sigma^2 & \longrightarrow & [\bot, \top] \subset \mathbb{R} \\ (x, x') & \longmapsto & s \end{array} \right. \tag{2.1}$$

The higher the value of s is, the more possible the substitution of x' by x is. If $s = \bot$ then the substitution is impossible, and if $s = \top$ then the substitution is certain. The values $\bot$ and $\top$ are optional and user-specified. They may appear or not in $\mathcal{M}$.

2.4.3.1 Nucleic matrices

There are few matrices for nucleic acids because there are only four symbols in their alphabet. The most frequently used is the unitary matrix (or identity matrix), where all bases are considered equivalent (see Fig. 2.6).

2.4.3.2 Protein matrices

The most frequently used matrices for proteins are PAM and BLOSUM:

PAM matrices This mutation matrix corresponds to a substitution accepted for 100 sites in a particular time of evolution, *i.e.*, a mutation that does not destroy the activity of the protein. This is known as a *one-percent-accepted-mutation matrix* (1-PAM) . If we multiply the matrix by itself a few times, we obtain a matrix X-PAM that gives the probabilities of substitution for larger evolutionary distances. To be more easily used in sequence comparison programs, each X-PAM matrix is transformed into a matrix of similarities PAM-X called mutation matrix of Dayhoff [Dayhoff 1978]. This transformation is performed by considering the relative frequencies of mutation of amino acids and by taking the logarithm of each element of the matrix.

Simulation studies have shown that PAM-250 seems best to distinguish related proteins of those with similarity due to chance [Schwartz 1979]. Therefore, the matrix PAM-250 has become the standard substitution matrix among Dayhoff ones.

BLOSUM matrices A different approach was undertaken to highlight the substitution of amino acids. While PAM matrices derive from global alignments of very similar proteins, here the degree of substitution of amino acids is measured by observing blocks of amino acids from more distant proteins. Each block is obtained by multiple alignment from short and highly conserved regions. These blocks are used to group all segments of sequences having a minimum percentage of identity within their block. The frequency of substitution is deduced for each pair of amino acids and then calculate a logarithmic probability matrix called BLOSUM (BLOcks SUbstitution Matrix). Every percentage of identity is a particular matrix. For instance, the BLOSUM-62 matrix is obtained by using a threshold of 62% identity. Henikoff and Henikoff [Henikoff 1992] conducted such process from a database containing more than 2000 blocks.

Choice of protein matrices The effectiveness of protein matrices depends on the type of experiments and results used for alignment. Although many comparative studies have been conducted [Yu 2005, Brick 2008, Mount 2008, Zimmermann 2010], there is no ideal matrix. But it is clear from these studies that the matrices rather based on comparisons of sequences or 3D structures usually give better results than those based primarily on the model of Dayhoff. Higher BLOSUM matrices and lower PAM matrices are used to compare sequences that are relatively close and short while to compare more divergent and longer sequences, it is better to use lower BLOSUM or higher PAM. The latest versions of BLAST and FASTA programs can choose from several BLOSUM and PAM matrices and no longer use the PAM250 matrix as default but BLOSUM-62 (Fig. 2.7).

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

Figure 2.7: The amino acids substitution matrix BLOSUM-62.

2.5 Mining in bioinformatics data

2.5.1 Data mining

The concept of data mining is often used to term the process of Knowledge Discovery in Data (KDD) [Fayyad 1997]. However, the former is considered as one part of the latter. Indeed, the process of KDD has two other major parts, one preceding and one following the step of data mining, namely the phase of preprocessing and that of post-processing (Fig. 2.8)

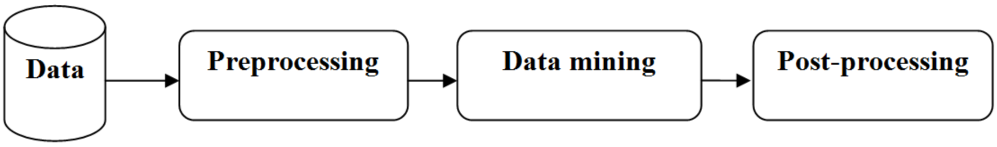


Figure 2.8: Steps of the KDD process.

The term *knowledge* is often used interchangeably with the terms *motif* or *pattern*.

Definition 3 (Motif/Pattern) *In general, a motif (or pattern) consists of a non-null finite feature that can characterize a given population P of objects. This motif may be identified based to its high frequency in P, its rarity in other populations or based on other parameters.*

Definition 4 (Knowledge discovery in data) *It is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.*

Definition 5 (Data mining) *It consists of applying computational techniques that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns (or models) over the data.*

Definition 6 (Preprocessing) *It comprises all necessary procedures to prepare and parse data into adequate format for the data mining step.*

Definition 7 (Post-processing) *It includes the evaluation, interpretation, validation and possible utilization of the mined knowledge.*

This decomposition of KDD in three major phases, generalizes several other more detailed decompositions found in the literature. Table 2.2 [Andrassyova 1999] lists some specific steps presented from four different sources that deal with data mining [Brachman 1996, Simoudis 1996, Fayyad 1997, Mannila 1997]. Terms belonging to the same line refer to the same task. It is also noteworthy that some tasks of the preprocessing stage

Table 2.2: KDD decomposition in literature.

	[Fayyad 1997]	[Brachman 1996]	[Simoudis 1996]	[Mannila 1997]
Preprocessing	Application domain learning	Tasks discovery		Domain understanding
	Data targeting	Data discovery	Data selection	
	Data cleaning and preprocessing	*Data Cleaning*	*Data transformation*	*Data preparation*
	Data Reduction and projection	*Data model construction*		
	data mining function selection			
Data mining	Algorithm selection	Data analysis	Data Mining	Motif discovery (data mining)
	Data mining			
Post-processing	Interpretation	Result generation	Result interpretation	Discovered motif Post-processing
	Discovered knowledge usage			Result usage

Table 2.3: Common tasks in data mining.

Predictive	**Descriptive**
Classification : Assigning predefined classes to data objects.	**Association rules** : Generating rules describing causal relationships between data.
Regression : Predicting the value of an numerical variable.	**Clustering** : Grouping similar data together.

use data mining techniques, especially in the case of the transformation of data (in italic font).

Data mining uses algorithms and techniques from statistics, artificial intelligence and databases. Some of the most popular tasks are classification, clustering and retrieval of association rules. Depending on the nature of the data as well as the desired knowledge, there are many algorithms for each task. All these algorithms try to adapt a model to the data [Dunham 2002]. Such a model can be predictive or descriptive. A predictive model makes a prediction about the data using known examples, while a descriptive model identifies relationships between data. Table 2.3 presents the most common tasks in data mining [Dunham 2002].

2.5.2 Application of data mining in bioinformatics

Although enormous progress has been made over the years, many fundamental problems in bioinformatics, such as protein structure or gene classification and finding, are still open. The field of data mining has emerged with the promise to provide the useful tools, technical knowledge and experience. Thus, data mining methods play a fundamental role in understanding gene expression, drug design and other emerging problems in genomics and proteomics.

The application of data mining in bioinformatics is quite difficult, since data are not often encoded in adequate format. Moreover, the data space for most bioinformatics problems is huge, infinite and demands highly efficient and heuristic algorithms. Many data mining algorithms have been utilized for the prediction and classification of various protein properties, such as active sites, junction sites, stability, shape, protein domains, etc [Cannataro 2010]. Data mining methods have been also applied for protein secondary and tertiary structure prediction. This problem has been studied over many years and many techniques have been developed [Tzanis 2007]. Initially, statistical approaches were adopted to deal with this problem. Later, more accurate techniques based on information theory, nearest neighbors, and neural networks were developed. Combined methods such as integrated sequence alignments with nearest neighbor approaches have improved prediction accuracy.

Other important problems of structural bioinformatics that utilize data mining methods are the RNA secondary structure prediction, the inference of a protein's function from its structure, the identification of protein-protein interactions and the efficient design of drugs, based on structural knowledge of their target.

The aim of applying data mining on bioinformatics is to discover global knowledge giving a meaning to the biological data and associating them with understandable relationships. The main challenge that opposes this goal is the complex aspect of data issued from bioinformatics.

2.5.3 Complexity of bioinformatics data

A complex data type is usually a composite of other existing similar or distinct data types, whose processing requires different kinds of expert knowledge. In [Ras 2008], authors mentioned five dimensions of complex data that must be taken into account in new data mining strategies

1. **Different kinds.** The data associated to an object are of different types. Besides classical numerical, categorical or symbolic descriptors, text, image or audio/video data are often available. For example biological sequences are textual data.

2. **Diversity of the sources.** The data come from different sources. For instance, a protein structure may often be stored in several databases, each one of them producing specific information.

3. **Evolving and distributed.** It often happens that the same object is described according to the same characteristics at different times or different places. For example, the description of a protein in the PDB database may vary over time with new identifers and new information.

4. **Linked to expert knowledge.** Intelligent data mining should also take into account external information, also called expert knowledge. Bioinformatics data are strongly linked to biologists' knowledge such as chemical characteristics, evolution, substitution, etc. These information could be taken into account by means of descriptive structures such as substitution matrices.

5. **Dimensionality of the data.** The association of different data sources at different moments multiplies the points of view and therefore the number of potential descriptors. The resulting high dimensionality is the cause of both algorithmic and methodological difficulties.

2.6 Conclusion

In this chapter, we presented two emerging research areas in which our work is located, namely bioinformatics, which encompasses all the technologies and data related to biology and data mining that extracts useful knowledge from data. Data mining is particularly suited for the analysis of bioinformatics data due to the panoply of algorithms and techniques it presents, that can address many known issues in bioinformatics. However, the complex nature of these data remains a real obstacle to overcome. In the next chapter, we introduce a specific problem in bioinformatics in a data mining view, *i.e.*, protein classification.

Chapter 3

Classification of Proteins in a Data Mining Framework: Preprocessing issue

Contents

Goals

This chapter introduces the problem of protein classification seen within a data mining framework. We overview the classification concept and we cite its most known algorithms, and evaluation techniques and metrics. Meanwhile, we present the importance of protein classification in bioinformatics and we explain the necessity of preprocessing relative to the complexity of bioinformatics data under consideration.

3.1 Classification of proteins based on data mining

In this section we define the concept of classification in data mining and we overview its application in bioinformatics, precisely for protein data. Moreover, we present and detail the basic ideas of a bench of the most known classifiers.

3.1.1 Classification in data mining

Classification, also termed supervised classification, refers to the process of assigning an object (or more generally an instance) into a given set of affiliations (or classes or labels), where the affiliations are a priori known [Han 2006]. Contrariwise, in clustering (referred also to unsupervised classification) the affiliations are missing and have to be created based on one or many criteria of similarity between instances. Formally, classification is defined as follows:

Definition 8 (Classification) *Given a set of objects $\mathcal{O}$ and a set of labels (or classes) $\mathcal{C}$, a classification Φ over $\mathcal{O}$ is a discrete value-output function defined as below:*

$$\Phi : \left| \begin{array}{ccc} \mathcal{O} & \longrightarrow & \mathcal{C} \\ o & \longmapsto & c \end{array} \right. \tag{3.1}$$

Learning to classify is a central problem in both natural and artificial intelligence [Cornuéjols 2010]. Intuitively, a classification rule is a cognitive act or procedure allowing affect to an object the family where it belongs, *i.e.*, recognizing it. This is how a child learns to classify animals into cats and dogs, plates into sugary and salty, etc. Analogously, some computer programs that are able to recognize handwriting, have learned rules allowing them to distinguish and classify the different traced signs; other programs are able to classify sounds, etc. These data are, generally, presented in a relational format.

Definition 9 (Relational format) *Data, concerning a set of objects $\mathcal{O}$, are said to be in a relational format if all objects of $\mathcal{O}$ are defined in the same dimension, i.e., described by the same attributes. This format is also said* tabular *format and* object-attribute *format.*

In general, classification can be seen as a succession of two steps: learning and prediction. The first step consists in analyzing a set of instances, namely the learning set, where these instances belong to already known classes. Meanwhile, a set of rules is generated defining the classification function. The second consists in applying the defined function on a set of unknown instances, where each unknown instance is affiliated to a class, based on the already generated function.

3.1.2 Classifiers

Most of the classifiers attempt to find a model that explain the relationships between the input data and the output classes. This reasoning method is called inductive since it inducts knowledge (model) from input data (instances and attributes) and outputs (classes). This model allows class prediction for new instances. Thus, a model is as good as the correctness of its prediction. In Table 3.1, we report a comparison between four known classifiers namely decision tree (DT) [Li 2008], naïve bayes (NB) [Wasserman 2004], nearest neighbour (NN) [Weiss 1990] and support vector machines (SVM) [Vapnik 1995, Bi 2003]. More details can be found in the review [Kotsiantis 2007].

3.1.3 Evaluation techniques

The evaluation of classifiers is a recurrent issue in supervised learning. Resampling techniques allow us to answer this question [Kohavi 1995].

3.1.3.1 Holdout

The data set is separated into two sets: the training set and the testing set. The classifier creates a model using the training set only. Then, the created model is used to predict the output values for the data in the testing set (it has never seen these output values before). The errors it makes are accumulated as before to give the mean absolute test set error, which is used to evaluate the model. The advantage of this method is that it is usually preferable to the residual method and takes no longer computation time than the next techniques. However, its evaluation can have a high variance. The evaluation may depend heavily on which data points end up in the training set and which end up in the test set, and thus the evaluation may be significantly different depending on how the division is made.

3.1.3.2 K-fold cross validation

This technique (CV) is one way to improve over the holdout method. The data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other $k - 1$ subsets are put together to form a training set. Then the average error across all k trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set $k-1$ times. The variance of the resulting estimate is reduced as k is increased. The disadvantage of this method is that the training algorithm has to be rerun from scratch k times, which means it takes k times as much computation to make an evaluation. A variant of this

Table 3.1: Comparing some classifier algorithms. Scoring: **** stars represent the best and * star the worst performance.

	DT	**NB**	**NN**	**SVM**
Accuracy in general	**	*	**	****
Speed of learning with respect to number of attributes and the number of instances	***	****	****	*
Speed of classification	****	****	*	****
Tolerance to missing values	***	****	*	**
Tolerance to irrelevant attributes	***	**	**	****
Tolerance to redundant attributes	**	*	**	***
Tolerance to highly interdependent attributes (*e.g.* parity problems)	**	*	*	***
Dealing with discrete/binary/continuous attributes	****	*** (not continuous)	*** (not directly discrete)	** (not discrete)
Tolerance to noise	**	***	*	**
Dealing with danger of over-fitting	**	***	***	**
Attempts for incremental learning	**	****	****	**
Explanation ability/transparency of knowledge/classifications	****	****	**	*
Model parameter handling	***	****	***	*

method is to randomly divide the data into a test and training set k different times. The advantage of doing this is that one can independently choose how large each test set is and how many trials you average over.

3.1.3.3 Leave one out

Leave one out (LOO) is K-fold cross validation taken to its logical extreme, with K equal to N, the number of data points in the set. That means that N separate times, the classifier is trained on all the data except for one point and a prediction is made for that point. As before, the average error is computed and used to evaluate the model. The evaluation given by leave-one-out cross validation error (LOO-E) is good, but at first pass it seems very expensive to compute. Fortunately, locally weighted learners can make LOO predictions just as easily as they make regular predictions. That means computing the LOO-E takes no more time than computing the residual error and it is a much better way to evaluate models.

3.1.3.4 Bootstrap

Given a dataset of size n, a bootstrap sample is created by sampling n instances uniformly from the data with replacement. In the simplest form of bootstrapping, instead of repeatedly analyzing subsets of the initial dataset, one repeatedly analyzes subsamples of the data. Each subsample (a bootstrap sample) is a random sample with replacement from the full dataset. Then, the evaluation is performed as in cross-validation.

3.1.4 Classification performance metrics

In order to evaluate the performance of the classification, many metrics were proposed in literature. Actually, the formulas of most of these metrics are based on four parameters namely true positive (TP), false positive (FP), true negative (TN) and false negative (FN).

Definition 10 (True positive) *is when the example is correctly classified as positive.*

Definition 11 (False positive) *is when the example is incorrectly classified as positive, when it is in fact negative.*

Definition 12 (True negative) *is when the example is correctly classified as negative.*

Definition 13 (False negative) *is when the example is incorrectly classified as negative, when it is in fact positive.*

The mentioned parameters are usually shown in what is called a confusion matrix.

In the following we define and present a bench of the most used metrics.

3.1.4.1 Sensitivity

Called also *recall rate*, sensitivity represents the percentage of correctly classified as positive instances from all those classified as positive. This measure is computed using the following formula:

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.2}$$

3.1.4.2 Specificity

Specificity represents the percentage of correctly classified as negative instances from all those classified as negative. This measure is computed using the following formula:

$$Specificity = \frac{TN}{TN + FP} \tag{3.3}$$

3.1.4.3 Accuracy

Accuracy is the percentage of the correctly classified instances. It is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.4}$$

3.1.4.4 Precision

Precision or positive predictive value is the percentage of correctly classified as positive from all the positive instances. It is calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3.5}$$

3.1.4.5 F-measure

The F-measure is used as a single measure of performance of the test. It considers both the precision and the recall, and is computed using the following formula:

$$F - measure = 2 * \frac{precision * recall}{precision + recall} \tag{3.6}$$

3.1.4.6 ROC curve

A receiver operating characteristic, shortly a ROC curve, is a plot of the true positive rate (sensitivity) against the false positive rate (1 - specificity) for the different possible cut-points of a diagnostic test. It shows the tradeoff between sensitivity and specificity (Fig. 3.1.4.6. The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test [Zweig 1993]. It is possible to derive a synthetic indicator from the ROC curve, known as the AUC (Area Under Curve - Area Under the Curve). The AUC indicates the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. There exists a threshold value: if we classify the instances at random, the AUC will be equal to 0.5, so a significant AUC must be superior to this threshold.

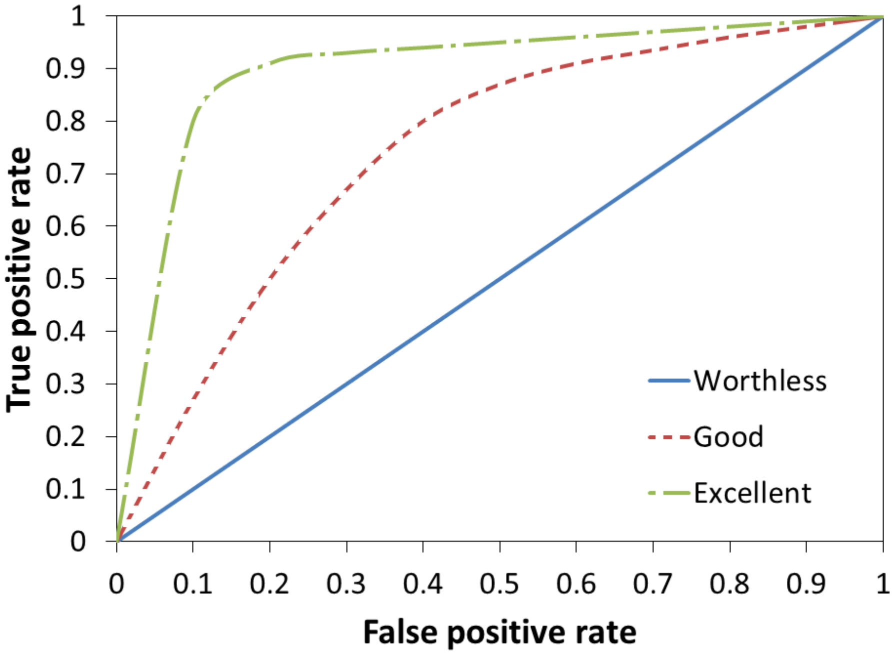


Figure 3.1: Comparing ROC curves.

3.1.4.7 E-value

The Expect value (shortly E-value, called also expectation) is a parameter that describes the number of hits one can expect to see by chance when searching a database of a particular size. It decreases exponentially as the score of the match increases. In other words, the E-value allows for example to measure the fairness of a game of chance and is then equal to the sum of the gains (or losses) weighted by the probability of gain (or loss). When the expectation is equal to 0, the game is fairly stated.

3.1.5 Classification of biological data: the case of proteins

3.1.5.1 Why classifying proteins?

Proteins are biological macromolecules that play crucial roles in almost every biological process. They are responsible in one form or another for a variety of physiological functions including enzymatic catalysis, binding, transport and storage, immune protection, control of growth, etc. Analysis and interpretation of proteins is a fundamental task in bioinformatics. Classification and prediction techniques are one way to deal with such task [Bhaskar 2005]. In fact, biologists are often interested in identifying the family to which an unknown protein belongs [Bernardes 2008]. This makes it possible to study the evolution of this protein and to discover its biological functions. Furthermore, the study and the prediction of oligomeric proteins (quaternary structures) are very useful in biology and medicine for many reasons [Klotz 1975]. Indeed, they often intervene in terms of bio-macromolecules functional evolution, reparation of misfolds and defects [Price 1994, Corrales 1996]. They are also involved in many important biological processes such as chromosome replication, signal transduction, folding pathway and metabolism [Terry 1998]. Biologists also seek, for instance, to identify active sites in proteins and enzymes [Slama 2008], to classify parts of DNA sequences into coding or non-coding zones or to determine the function of the nucleic sequences such as the identification of the promoter sites and the junction sites [Mephu Nguifo 1993, Lemoine 1999, Vignal 1997]. All these purposes can be seen in a classification framework where the identification of the classes of unknown biological data may yield further understanding.

Protein classification has several goals depending on the nature of the task *e.g.*, structural, taxonomic, functional or any other affiliation. In order to gain a better understanding of the functions of proteins and their relationship, existing databases specialized in classification of proteins should be updated frequently. Unfortunately, this is no longer possible with the exponential growth in the number of newly discovered protein structures. Indeed, the PDB database [Berman 2000] continues to expand tremendously comprising so far more than 72000 protein structures. For instance SCOP [Andreeva 2008], being manually built, is updated only every 6 months. This is due to the intensive work required in visual inspection which makes it the most reliable database for structural classification. Hence, accurate computational and machine learning tools may offer considerable boosting to meet the increasing load of data [Muggleton 2006]. One way to evaluate automated methods is to compare their results with well-known databases often considered as gold standard of protein classification.

3.1.5.2 From alignment to data mining

Biological databases and alignment programs have revolutionized the practice of biochemistry, molecular and evolutionary biology. Early sequence comparisons revealed extraordinary evolutionary relationships [Pearson 2005]. Since then, the inference of new knowledge from significant similarity has become routine and considerably more reliable. Therefore, alignment has become the main technique used by biologists to look for similarity between structures, and hence to classify new ones into already known families/classes. Whenever two protein sequences or protein structures are similar, the similarity can be explained by one of two alternatives [Pearson 2005] :

1. The two proteins are similar because they are homologous, *i.e.*, both are descendants from a common ancestor.

2. The proteins are not related, *i.e.*, they are similar because some set of structural or functional constraints caused them to converge from independent origins to the observed similarity.

Sequential and structural alignments have proved to be useful. For instance, SCOPMap [Cheek 2004] uses sequence comparison methods such as BLAST [Altschul 1990], PSI-BLAST [Altschul 1997], and structure comparison methods such as DALI [Holm 1993] and MAMMOTH [Ortiz 2002], and a manually set similarity threshold to make classification assignments. However, the inference of classes from alignment may include some weaknesses. This should be expected since alignment depends basically on pairwise similarity and this reveals three major problems:

1. **Orphan proteins**. Unknown proteins that have no detectable similar known proteins have been termed *orphan* proteins [Ekman 2010]. Hence, they can not be classified.

2. **Classification scope**. The kind of classification is not taken into account. The unknown protein is simply affiliated into the class of its most similar known protein whatever was the objective of classification. For example, moving from structural to functional or to taxonomic classification does not change the alignment results. More concretely, two proteins may belong to the same class with respect to a given scope, whereas they belong to different classes with respect to another scope. This issue yields classification errors.

3. **Disuse of external information**. Biological data are generally accompanied by a worth of information. These information can serve as

descriptive characteristics *e.g.*, topological measures, volume, etc or related to their environment *e.g.*, contextual information such as pH, cell concentration, etc. The alignment process does not integrate the possibility to use such information.

In this context, the use of data mining techniques is suited for protein classification. It even represents a rational choice to address that problem for the following reasons:

- Data mining classifiers have proved to be efficient in several application fields, *e.g.*, finance, commerce, marketing, etc.
- The generated knowledge are reliable since they are evaluated based on statistically efficient sampling techniques.
- Data mining classifiers are generic and can be applied to different fields without having deep expert's knowledge about the application domain.
- Data mining offers a panoply of classification algorithms.

Indeed, protein classification have been cast as a problem of data mining, in which an algorithm classifies new structures based on what it learns from an already available classification (For example the SCOP database). Work on protein classification has been ongoing for over a decade using data mining classifiers, such as neural networks [Cai 2000, Ding 2001, Huang 2003, Ie 2005] and support vector machines (SVM) [Chen 2006, Melvin 2007, Shamim 2011]. However, knowing that protein data are presented in complex formats and that mining tools often process data under the relational format, it will not be possible to apply these tools directly on such data, *i.e.*, a preprocessing step is seen essential.

3.2 Preprocessing of proteins data for classification

Classification performance is heavily dependent on the quality of the input data. Yet, real application data are usually incomplete (missing values, abbreviated data..), noisy (errors, exceptions..) or inconsistent (naming, coding format..). These anomalies can be caused by a lot of factors that can be due to human mistakes like writing mistakes or to program dysfunction like network connection interruption or even the complex nature of raw data. Thus, preprocessing is crucial to conduct an efficient classification [Zhang 2003b].

Data preprocessing is a very important yet often neglected step in the data mining process (or KDD process). This task usually takes substantial project time (between 70% and 80%), especially when many tasks are required [Feelders 2000]. Preprocessing mainly relies on discovering a set of features to convert raw data, that may be present in different scales and dimensions, into a feature space within one same dimension.

3.2.1 Feature discovery

There is broad interest in feature discovery among practitioners from statistics, pattern recognition, and data mining. Feature discovery includes a set of techniques that transform and simplify data so as to make data mining tasks easier. These techniques can be clustered into three main categories namely, *feature extraction*, *feature construction* and *feature selection*.

3.2.1.1 Feature / motif extraction

Feature extraction is a process that extracts a set of new features from the raw data. These features have the particularity that they are subsets, regions, parts of the raw data. In other words, they are particular pieces of data that generally have the same nature and type of the raw data; they are also termed *motifs*.

An important motivation behind feature extraction is that when we deal with complex data and there is a lack of knowledge on them, we can describe them by certain parts of them. This kind of features can provide good quality description allowing recognizing forms and objects. For example, in the celebrity face recognition game, the candidate relies on few parts of the celebrity face to recognize the person.

Feature extraction is not totally an independent issue from feature construction and selection. For example, feature construction and feature selection can be viewed as two complementary tasks of feature extraction. We can consider features as a representation language. In some cases where this language contains more features than necessary, the selection helps simplify the language; in other cases where this language is not sufficient to describe the problem, feature construction helps enrich the language [Liu 1998].

3.2.1.2 Feature construction

Feature construction is a process that generates new features either from other features or from descriptions on raw data [Hasan 2009]. In the first case, feature construction can be performed by combining features to form compound ones [Liu 1998]. In the second case, descriptive characteristics can be built by applying statistical operators on data [Zhang 2003a, Chen 2006]. In addition the feature construction process may include discretization, normalization and space embedding.

Discretization In many cases, data can contain a large number of possible feature values. This contributes to slow and ineffective process of inductive machine learning. Aiming to resolve this issue, discretization is used to reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals. Hence, interval labels can be used to replace actual data values. However, the choice of interval borders for the discretization of a numerical value range remains an open problem in numerical feature handling.

Normalization Normalization aims to scale attributes to fall within a specified range. Mainly, three normalization techniques are used in the literature. The first technique is *min-max normalization* applied on original data values based on this formula:

$$V' = \frac{V - Min}{Max - Min} * (Max' - Min') + Min' \tag{3.7}$$

where V' is the new value, V is the original value, Max and Min are the old maximum and minimum values, Max' and Min' are the new maximum and minimum values. In the case when min and max are unknown or when there are outliers, normalization can be performed using *Z-score normalization.* Here, values are scaled using mean and standard deviation:

$$V' = \frac{V - Mean}{StDev} \tag{3.8}$$

The third technique is *normalization by decimal scaling.* The idea of this technique is to move V by j positions such that j is the minimum number of positions moved so that absolute maximum value falls in [0..1]. Normalization by decimal scaling is performed based on this formula:

$$V' = \frac{V}{10^j} \tag{3.9}$$

Space embedding Given a set of features, it is possible to project or embed the data into a lower dimensional space while keeping as much information as possible. Hence, a smaller feature set is generated. *Principal component analysis* is one classical technique to construct such features [Ding 2004].

3.2.1.3 Feature selection

Feature selection consists in choosing a subset of n features from the original set of N features ($n \leq N$) so that the feature set is reduced according to certain parameters [Liu 2007]. Although feature selection is mainly dedicated to select relevant features, it can have other motivations , including:

- General data reduction, to limit storage requirements and increase computation speed.

- Feature set reduction, to save resources in the next round of data collection or during utilization.

- Performance improvement, to gain in predictive accuracy.
- Data understanding, to gain knowledge and improve the comprehensibility about the learning results or the visualization .

Mainly, there exist three approaches for feature selection explained in the following:

- **Embedded approaches:** feature selection is included as part of the mining algorithm.
- **Filter approaches:** features are first extracted then selected before running the mining algorithm.
- **Wrapper approaches:** these approaches try to find the best attributes subset without enumerating all subsets

3.2.2 Preprocessing framework for protein classification

Proteins are usually presented in two kinds of file format, namely the sequential formats (*e.g.*, FASTA format) for the primary structure data and the spatial formats (*e.g.*, PDB format) for 3D data (See Appendix A):

- **Sequential format**: presents one or a list of protein's primary structures. Each one is defined by string of characters where each character is an abbreviation of the name of an amino acid.
- **Spatial format**: contains spatial coordinates of atoms present in a protein structure.

Since these structures are not represented in a relational format, their representation does not generally enable the use of well-known classification techniques such as decision trees (DT), naïve bayes (NB), support vector machines (SVM) and nearest neighbour (NN) which have proved to be very efficient in real data mining tasks [Han 2006]. We recall that these classifiers rely on data described in a relational format. Therefore, a set of attributes must be created to describe the proteins to obtain a set of feature vectors, where each vector represents a protein. Feature extraction (or motif extraction) is one major way to address this issue. Different studies have been devoted to feature discovery in biological data, especially to motif extraction [Huan 1998, Nevill-Manning 1998, Maddouri 2004, Yongqiang 2006b, Yongqiang 2006a, Lopes 2008]. Motif extraction methods are generally based on the assumption that the significant regions are better preserved during the evolution because of their importance in terms of structure and/or function of the molecule [Nevill-Manning 1998], and thus that they appear more frequently than it is expected. In [Maddouri 2004], authors have shown that feature discovery can efficiently contribute to the use of data mining algorithms for the classification of biological data. In this case, the classification obeys the knowledge discovery in data (KDD) process and hence comprises three major steps namely, preprocessing step, mining step and postprocessing step.

3.2.2.1 Preprocessing step

The preprocessing is composed of two main parts. The first part consists in building a feature set allowing a reliable description of data, depending on the case of study. The second part is concerned with how this description is performed; in other words a function defining the relation between features and instances is formulated. At the end of the preprocessing step, all instances are encoded into vectors in the same dimension. The relational table comprising all these vectors is called a *learning context* or *feature space* (Fig. 3.2).

Feature set One generic way to build the feature set is to follow these three procedures:

1. Motifs are discovered using a feature extraction technique.
2. External characteristics can be constructed based on observation, statistical metrics or motif processing.
3. The feature set is reduced using a feature selection technique in order to keep uninteresting features.

At least one of the two first procedures must be performed to build a feature set. The third procedure is not compulsory but it allows to reduce the feature set dimension and to target the most interesting features. It can be either separated or embedded in one of the two previous procedures.

Instance Encoding Preprocessing data with reference to a feature set allows the use of motifs as well as external characteristics to describe the instances under consideration. The nature of encoding may differ depending on the nature of features.

Encoding using motifs The encoding in this case is dependent on the presence of the motif in the instances. This presence can be expressed either by incidence or by frequency.

Definition 14 (Incidence-based encoding) *Given a set of objects $\mathcal{O}$ and a set of features $\mathcal{F} = \{f_1, f_2, .., f_d\}$, an incidence-based encoding IbE over $\mathcal{O}$*

is the function defined as below:

$$IbE : \left| \begin{array}{lcl} \mathcal{O} & \longrightarrow & \mathcal{O}' \\ o & \longmapsto & o' = [x_1, x_2, .., x_d] \end{array} \right. \tag{3.10}$$

such that

$$x_i = \begin{cases} 1 & iff_i \ occurs \ in \ o, i = 1..d \\ 0 & otherwise \end{cases} \tag{3.11}$$

The main strengths (✓) and weaknesses (×) of this encoding are:

- ✓ Extremely simple model.
- ✓ Well suited for most classifiers *e.g.*, naïve bayes, support vector machine, nearest neighbour.
- ✓ Ease of interpretation.
- ✓ Suited for other tasks rather than classification *e.g.*, association rules.
- × Feature abundance / paucity is not taken into account.
- × Feature position is not taken into account.

Definition 15 (Frequency-based encoding) *Given a set of objects $\mathcal{O}$ and a set of d features $\mathcal{F} = \{f_1, f_2, .., f_d\}$, a frequency-based encoding FbE over $\mathcal{O}$ is the function defined as below:*

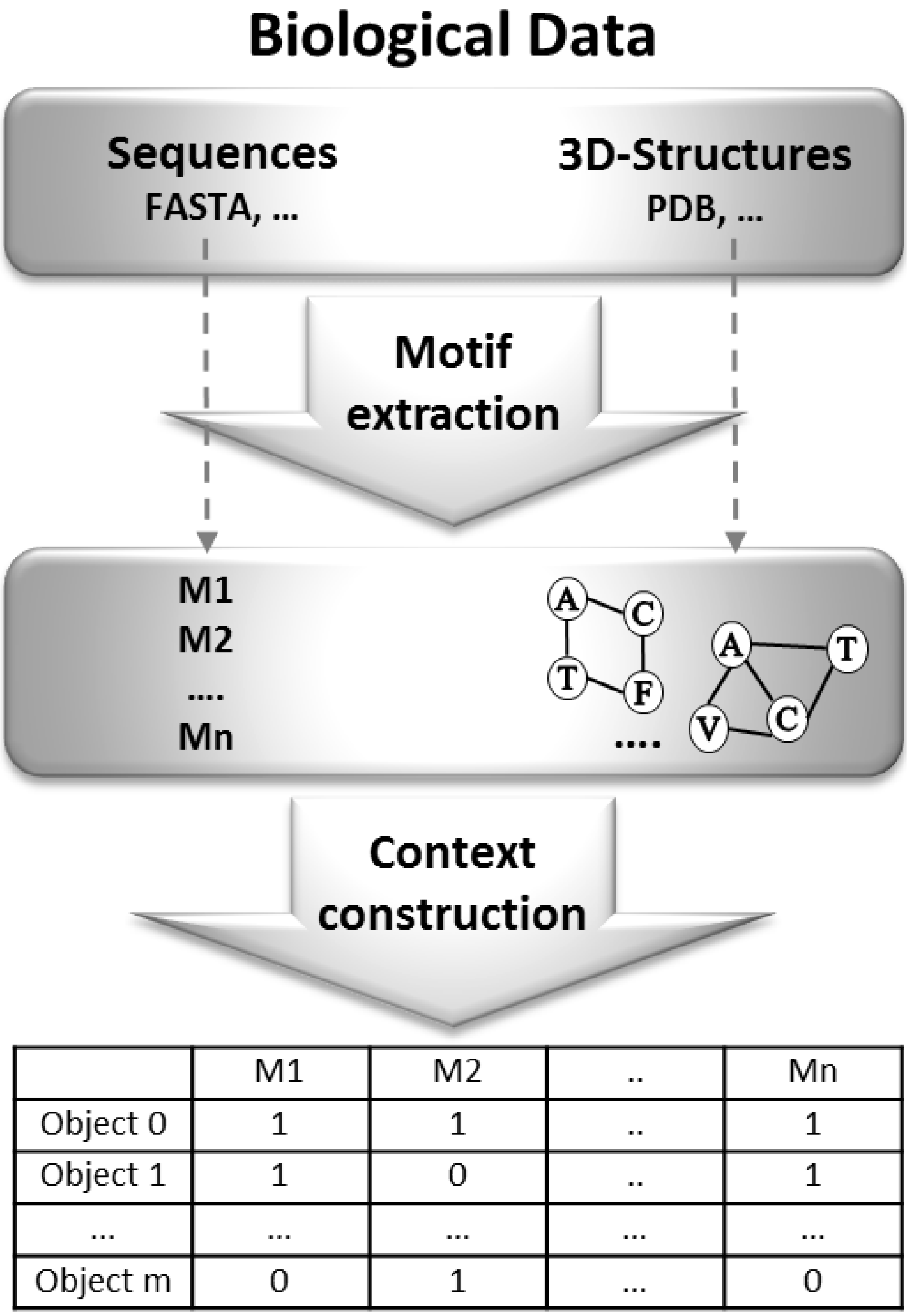
$$FbE : \left| \begin{array}{lcl} \mathcal{O} & \longrightarrow & \mathcal{O}' \\ o & \longmapsto & o' = [x_1, x_2, .., x_d] \end{array} \right. \tag{3.12}$$

such that

$$x_i = \begin{cases} n_i = the \ number \ of \ occurrences \ of \ f_i \ in \ o, i = 1..d \\ or \\ \frac{n_i}{\sum_{j=1}^{d} n_j} = the \ frequency \ of \ f_i \ in \ o, i = 1..d \end{cases} \tag{3.13}$$

The main strengths (✓) and weaknesses (×) of this encoding are:

- ✓ Feature abundance / paucity is taken into account.
- ✓ The instance description is more precise.
- ✓ Instance size is taken into account.
- ✓ Suited for multinomial bayesian classifier .
- × Not suited for many classifiers relying on symbolic encoding.
- × Not suited when the instances are imbalanced in terms of size.



	M1	M2	..	Mn
Object 0	1	1	..	1
Object 1	1	0	..	1
...	...	...	...	...
Object m	0	1	...	0

Figure 3.2: Preprocessing based on motif extraction. This figure describes the process of protein encoding. The extracted motifs are used as attributes to build a binary context where each row represents a protein.

Encoding using external characteristics In this case, the encoding can be simply done by describing instances using the provided characteristics, be they numeric or symbolic.

3.2.2.2 Mining step

In the mining step, a classifier is applied to the learning context to generate a classification model. The reliability of the produced model depends not only on the classifier but also on the quality of the preprocessing and mainly the interestingness of the feature space.

3.2.2.3 Postprocessing step

The latter model is used to classify other instances in the postprocessing step. These instances are also encoded into a relational format using the same features as for the learning context *i.e.*, *test context*. In addition to the evaluation of the classifier, this step allows also to evaluate the interestingness of the feature space. In other words, the more reliable the feature space is, the better the classifier performs.

3.3 Conclusion

In this chapter, we have introduced one of the most important problems in bioinformatics, which is the classification of protein data. We have presented this problem in a data mining framework, where we have mentioned a bench of known classifiers and some techniques and metrics of evaluation. The raised issue is that the data are described in unusual formats for the use of data mining classification algorithms. To overcome this obstacle, a preprocessing step should be implemented. One way to preprocess these data is based on the extraction of features that will play the role of attributes. In the next chapter, we will establish a comparative study of a preprocessing method of protein sequences, we propose, with other literature methods.

Chapter 4

Substitution-Matrix-based Feature Extraction for Protein Sequence Preprocessing

Contents

Goals

This chapter deals with the motif-based preprocessing of protein sequences for their classification. We propose a novel encoding method that uses amino-acid substitution matrices to define similarity between motifs during the extraction step. We carry out a detailed experimental comparison (in terms of classification accuracy and number of attributes) between several encoding methods using various kinds of classifiers (C4.5 decision tree, NB, SVM and NN) as well as the standard approach based on alignment and the Hidden-Markov-Model-based approach. The outcomes of our comparative experiments confirm the efficiency of our encoding method to represent protein sequences in classification tasks. The subject of this chapter has been published in [Saidi 2010b].

4.1 Background and related works

The sequential motif mining problem was first introduced by Agrawal and Srikant in [Agrawal 1995]: *Given a set of sequences, where each sequence consists of a list of elements, and given a user-specified support threshold, sequential motif extraction is to find all of the frequent subsequences, i.e., the subsequences whose occurrence frequency in the set of sequences is no less than the support threshold* . Analogously, protein primary structures are commonly known as strings of character (or sequences), where each character represents an amino acid. Finding motifs of conserved amino acid residues in sets of sequences is an important problem in computational biology, particularly in the study of functionally related proteins [Wang 1994, Ollivier 1991]. In our case (protein sequences), and based on the definition of motif provided in Chapter 2, we can make the following definition:

Definition 16 (Sequential motif) *This consists of a motif whose composing residues are contiguous in the primary structure i.e., it is a sub-chain extracted from the protein chain.*

In this section we present a bench of five existing methods of features discovery: the N-Grams (NG), the Active Motifs (AM), the Amino Acid Composition (AAC), the Functional Domain Composition (FDC) and the Discriminative Descriptors (DD). After this, we describe our approach which consists of modifying the DD method by the use of a substitution matrix (DDSM)[Saidi 2010b].

4.1.1 N-Grams

The simplest approach is that of the N-Grams, known also as N-Words or length N fenestration [Leslie 2002]. The motifs to be built have a predefined length. The N-gram is a subsequence composed of N characters, extracted from a larger sequence. For a given sequence, the set of the N-grams which can be generated is obtained by sliding a window of N characters on the whole sequence. This movement is carried out character by character. With each movement a subsequence of N characters is extracted. This process is repeated for all the analyzed sequences (Fig. 4.1). Then, only the distinct N-grams are kept.

The N-Grams are widely used in information retrieval and natural languages processing [Khreisat 2009, Mesleh 2007]. They are also used in local alignment by several alignment systems such as Blast [Altschul 1990]. The N-Grams extraction can be done in a $O(m * n * N)$ time, where m is the maximum length of a sequence, n is the number of sequences in question and N is the motif length.

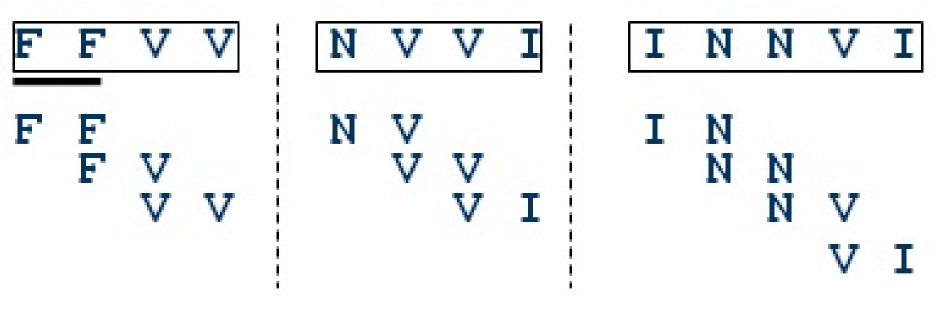


Figure 4.1: Extraction of 2-grams from the 3 sequences. FFVV, NVVI and INNVI. For each sequence of length m, the number of extracted N-Grams is: $m - N + 1$

4.1.2 Active Motifs

This method allows extracting the commonly occurring motifs whose lengths are longer than a specified length, called Active Motifs, in a set of biological sequences. The activity of a motif is the number of matching sequences given an allowed number of mutations [Wang 1994]. The motif extraction is based on the construction of a Generalized Suffix Tree (GST) which is an extension of the suffix tree [Hui 1992] and is dedicated to represent a set of n sequences indexed each one by $i = 1..n$. Each suffix of a sequence is represented by a leaf (in the shape of a rectangle) labelled by the index i of this sequence. It is composed by the concatenated sub-sequences labelled on the root-to-leaf i path. Each non-terminal node (in the shape of a circle) is labelled by the number of sequences to which belongs its corresponding sub-sequence composed by the concatenation of the sub-sequences labelled on the arcs which bind it to the root (Fig. 4.2). The candidate motifs are the prefixes of strings labelled on root-to-leaf paths which satisfy the length minimum. Then, only motifs having an acceptable activity will remain.

There are several algorithms used for the construction of the GST. Wang affirms that the GST can be built in a $O(m * n)$ time [Wang 1994], where m the maximum size of a sequence and n the number of sequences in question. To extract the motifs which satisfy the conditions of research, it is necessary to traverse the entire tree. That is to say a complexity of $O((m * n)^2)$.

4.1.3 Amino Acid Composition

According to the classic definition of this method, the feature set consists of 20 components, representing the 20 native amino acids in proteins. The amino acid composition refers to the occurrence frequency of each of these 20 components in a given protein. Since the information in the primary sequence is greatly reduced by considering the amino acid composition alone, other considerations have been taken into account within sev-

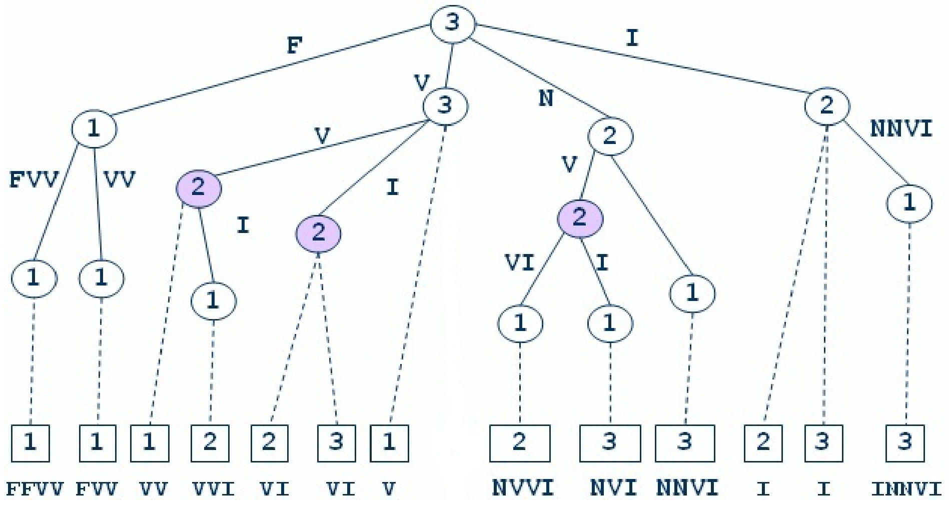


Figure 4.2: GST illustration for the 3 sequences. FFVV, NVVI and INNVI. If we suppose that only exactly coinciding segments occurring in at least two sequences and having a minimum length of 2 are considered as active. Then we have 3 active motifs: VV, VI and NV.

eral studies such as the sequence-order correlation factors *i.e.*, new features were added to the 20 original which yielded several AAC variants [Zhang 1995, Zhou 1998, Chou 2003, Zhang 2003a, Chen 2006].

4.1.4 Functional Domain Composition

Biological databases, such as PFAM [Finn 2010] and ASTRAL, contain large collections of multiple sequence alignments and Hidden Markov Model (HMM) profiles covering many common protein domains and families [Johnson 2006, Finn 2010]. Functional domains are determined using computational means, especially HMM profiles, combined with biologist knowledge and other databases information. Since they allow variable length gaps between several components, where each component is a simple motif, functional domains can be considered as structured motifs [Yongqiang 2006b, Yongqiang 2006a]. But they are more reliable since they obey the expert assessment.

4.1.5 Descriminative Descriptors

Given a set of n sequences, assigned to P families (or classes) $F_1, F_2, .., F_P$, this method consists of building substrings called Discriminative Descriptors DD which allow to discriminate a family F_i from other families F_j, with $i = 1..P$ and $i \neq j$ [Maddouri 2004]. This method is based on an adaptation of the Karp, Miller and Rosenberg (KMR) algorithm [Karp 1972] (see Appendix C). This algorithm identifies the repeats in character strings, trees or tables. The

extracted repeats are then filtered in order to keep only the discriminative and minimal ones.
A substring X is considered to be discriminative between the family F_i and the other families F_j, with $i = 1..P, j = 1..P$ and $i \neq j$ if:

$$\frac{\textit{number of sequence of } F_i \textit{ where } X \textit{ appears}}{\textit{total number of sequences of } F_i} \geq \alpha \tag{4.1}$$

$$\frac{\textit{number of sequence of } F_j \textit{ where } X \textit{ appears}}{\textit{total number of sequences of } F_j} \leq \beta \tag{4.2}$$

where α and β are user-specified thresholds between 0 and 1.

4.2 Descriminative Descriptors with Substitution Matrix

In the case of protein, the Discriminative Descriptors method neglects the fact that some amino acids have similar properties and that they can be therefore substituted by each other while changing neither the structure nor the function of the protein [Henikoff 1992]. Indeed, we can find several motifs in the set of the attributes generated by the DD method, which are similar and can derive all from a single motif. In the same way, during the construction of the context (binary table), we are likely to lose information when we denote by 0 the absence of a motif while another one, that can replace it, already exists [Saidi 2010b].

As mentioned, the similarity between motifs is based on the similarity between the amino acids which constitute them. Indeed, there are various degrees of similarity between amino acids. Since there are 20 amino acids, the mutations between them are scored by a 20 × 20 matrix called a substitution matrix [Henikoff 1992, Leslie 2002, Malde 2008].

4.2.1 Terminology

Let Ω be a set of n motifs, denoted each one by $\Omega[p], p = 1..n.$ can be divided into m clusters. Each cluster contains a main motif M^* and probably other motifs which can be substituted by M^*.

Definition 17 (Main motif) *The main motif is the one which can substitute all motifs in its cluster and has the highest mutation probability.*

Definition 18 (Mutation probability) *The mutation probability of a motif M is its probability of mutating to another motif in its cluster. For a motif M of k amino acids, this probability, noted $P_m(M)$, is based on the probability*

$P_i(i = 1..k)$ that each amino acid $M[i]$ of the motif M does not mutate to any other amino acid. We have:

$$P_m = 1 - \prod_{i=1}^{k} P_i \tag{4.3}$$

P_i is calculated based on the substitution matrix according to the following formula:

$$P_i = \frac{S(M[i], M[i])}{\sum_{j=1}^{20} S^+(M[i], AA_i)} \tag{4.4}$$

$S(x, y)$ is the substitution score of the amino acid y by the amino acid x as it appears in the substitution matrix. $S^+(x, y)$ indicates a positive substitution score. AA_j is the amino acid of index j among the 20 amino acids.

Definition 19 (Motif substitution) *For our purpose, a motif M substitutes a motif M' if:*

1. *M and M' have the same length k,*
2. *$S(M[i], M'[i]) \geq 0$, $i = 1..k$,*
3. *$SP(M, M') \geq T$, T is a user-specified threshold such that $0 \leq T \leq 1$.*

Definition 20 (Substitution probability) *We denote by $SP(M, M')$ the substitution probability of the motif M' by the motif M having the same length k. It measures the possibility that M mutates to M':*

$$SP(M, M') = \frac{S_m(M, M')}{S_m(M, M)} \tag{4.5}$$

$S_m(X, Y)$ is the substitution score of the motif Y by the motif X. It is computed according to the following formula:

$$S_m(X, Y) = \sum_{i=1}^{k} S(X[i], Y[i]) \tag{4.6}$$

Lemma 1 *For a motif M, if $P_m(M) = 0$ then M is a main motif belonging to a singleton.*

Proof 1 *According to equation (4.3) in definition 18, $P_m(M) = 0$ if $\prod_{i=1}^{k} P_i = 1$. That means, according to equation (4.4), that every amino acid in M has no substitute in the substitution matrix but itself. Therefore, M can not be substituted by any other motif in Ω and it composes a singleton.*

Lemma 2 *There is only one best motif which can substitute a motif M i.e, itself.*

Proof 2 *According to any substitution matrix, the amino acids which constitute a motif M are better substituted by themselves i.e., $S(a, a) \geq S(a, b)$, a and b are amino acids. Therefore $S_m(M, M) \geq S_m(M, M')$, M' is another motif of same length.*

Proposition 1 *If two motifs M and M' satisfy the substitution conditions in definition 19 then the substitution probability $SP(M, M')$ is between 0 and 1.*

Proof 3 *Lemma 2 and equation (4.5) induce that $SP(M, M') \leq 1$. The second condition in definition 19, equation (4.5) and equation (4.6) induce that $SP(M, M') \geq 0$.*

4.2.2 Methodology

The encoding method is composed of two parts. First, the number of extracted motifs will obviously be reduced because we will keep only one motif for each cluster of substitutable motifs of the same length. Second, we will modify the context construction rule. Indeed, we will denote by 1 the presence of a motif or of one of its substitutes. The first part can be also divided into two phases: (1) identifying clusters' main motifs and (2) filtering. (1) The main motif of a cluster is the one that is the most likely to mutate to another in its cluster. To identify all the main motifs, we sort Ω in a descending order by motif lengths, and then by P_m. For each motif M' of Ω, we look for the motif M which can substitute M' and that has the highest Pm (probability of mutation to another motif). The clustering is based on the computing of the substitution probability between motifs. We can find a motif which belongs to more than one cluster. In this case, it must be the main motif of one of them. (2) The filtering consists of keeping only the main motifs and removing all the other substitutable ones. The result is a smaller set of motifs which can represent the same information as the initial set.

4.2.2.1 Algorithm

The main motifs identification and the filtering are performed by the simplified Algorithm 1.

4.2.2.2 Complexity

Suppose Ω contains n motifs of l different lengths and suppose k is the maximum motif length. Ω can be sorted in $O(n \log n)$. Searching for the main motifs requires browsing Ω and for each motif, browsing at worst all motifs of the same length to check the substitution (definition 19). This can be done in $O((n^2/l) * k)$. Deleting non-main motifs can be done in $O(n)$. Hence, the time complexity of this algorithm is $O((n^2/l) * k)$.

Algorithm 1: MainMot

```
Data: $\Omega$: set of $n$ initial motifs.
Result: $\Omega^*$: set of main motifs.
begin
    sort $\Omega$ in a descending order by (motif length, $P_m$);
    foreach $\Omega[i] \in \Omega$, from $i = n$ to 1 do
        if $P_m(\Omega[i]) = 0$ then
            $\Omega[i]$ becomes a main motif;
        else
            $x \longleftarrow$ position of the first motif having the same length as $\Omega[i]$;
            for each $\Omega[j]$ from $j = x$ to $i$ do
                if $\Omega[j]$ substitutes $\Omega[i]$ or $j = i$ then
                    $\Omega[j]$ becomes a main motif;
                    break;
    foreach each motif $M \in \Omega$ do
        if $M$ is not a main motif then
            delete M;
    $\Omega^* \longleftarrow \Omega$;
```

Table 4.1: Motif clustering example. Ω is a set of motifs sorted by their lengths and P_m. The third row shows the cluster main motifs.

Ω	**LLK**	**IMK**	**VMK**	**GGP**	**RI**	**RV**	**RF**	**RA**	**PP**
P_m	0.89	0.87	0.86	0	0.75	0.72	0.72	0.5	0
Main motif	LLK	LLK	LLK	GGP	RI	RI	RI	RV	PP

4.2.3 Illustrative example

Given a Blosum62 substitution matrix and the following set of motifs (Table 4.1) sorted by their lengths and Pm, we assign each motif to a cluster represented by its main motif. We get 5 clusters illustrated by the diagram shown in Fig. 4.3.

4.3 Experiments

4.3.1 Aims and datasets

NG, AM, DD and DDSM encoding methods are implemented in C language and gathered into a DLL library (Appendix B). The accepted format of the input files is the FASTA format for biological sequences files. The library code

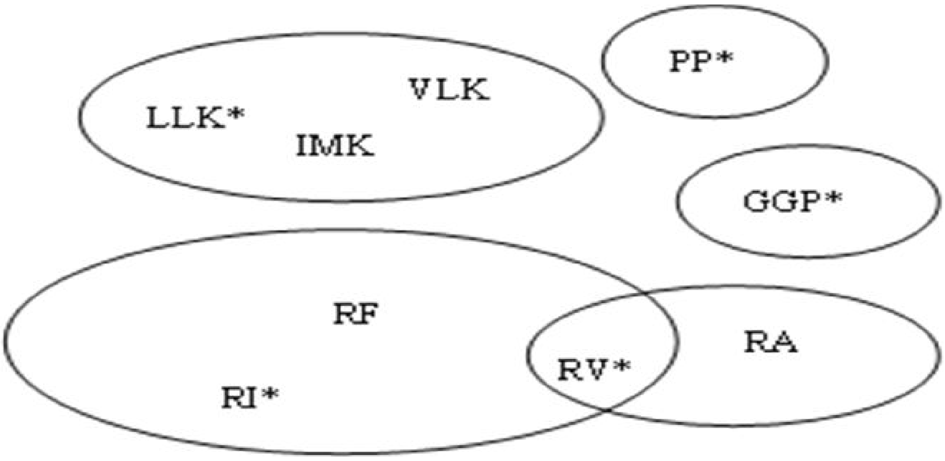


Figure 4.3: Motif clustering example. This figure illustrates the set of clusters and main motifs obtained from the data of Table 1 after application of our algorithm. RV belongs to 2 clusters and is the main motif of one of them.

that we have implemented generates relational files under various formats such as the ARFF format used by the workbench WEKA [Witten 2005] and the DAT format used by the system DisClass [Maddouri 2004].

Our experiments are divided into 2 parts. In the first one, we make a detailed comparison between NG, AM, DD and DDSM encoding methods. We perform the sequence classification using DT, SVM, NB and NN algorithms. We also conduct classification experiments using Blast [Altschul 1990] and the HHM tool HMMER [Johnson 2006, Eddy 2008]. For Blast, we assign to a protein query the class of the reference sequence with the best hit score. As for HMMER, the key idea is very similar to the alignment based approach *i.e.*, we first create an HMM-profile for each protein group (class), then we score the query sequence against all the created profiles, and the query protein sequence takes the class of the HMM-profile having the best hit score. Our method (DDSM) constructs the features using the substitution matrix Blosum62. The choice of this substitution matrix is not based on preliminary experiments, but instead on the fact that it is the most used by alignment tools, especially the widespread Blast. We examine three aspects:

1. The effect of each encoding method on the four classifiers to deduce which one is the best in terms of accuracy and number of generated attributes.

2. The comparison of the four classifiers while varying the encoding methods.

3. The comparison with Blast and HMM results.

In the second part, we try to assess the effect of varying the substitution matrices on our method and on the classification quality and hence to determine whether there is a substitution matrix which could be recommended. Then we compare our feature-construction method with other ones presented in [Zhou 1998, Yu 2006, Chen 2006], which means that we compare with nine related works [Zhou 1998, Cai 2000, Cai 2001, Cao 2006, Chou 1989, Feng 2005, Nakashima 1986, Yu 2006, Chen 2006].

4.3.1.1 Part 1

To perform our experiments, we use 5 datasets comprising 1604 protein sequences from Swiss-Prot [Bairoch 2000] and SCOP [Andreeva 2004] described in Table 4.2.

Table 4.2: Experimental data. IdP: Idenity Percentage, Tot: Total.

Dataset (source)	IdP	Family/class	Size	Tot
DS1 (Swiss-prot)	48%	High-potential Iron-Sulfur Protein	19	60
		Hydrogenase Nickel Incorporation Protein HypA	20	
		Hlycine Dehydrogenase	21	
DS2 (Swiss-prot)	48%	Chemokine	255	510
		Melanocortin	255	
DS3 (Swiss-prot)	25%	Monomer	208	717
		Homodimer	335	
		Homotrimer	40	
		Homotetramer	95	
		Homopentamer	11	
		Homohexamer	23	
		Homooctamer	5	
DS4 (Swiss-prot)	28%	human TLR	14	40
		Non-human TLR	26	
DS5 (SCOP)	84%	All-α domain	70	277
		All-β domain	61	
		α/β domain	81	
		$\alpha + \beta$ domain	65	

We try to conduct our experiments on various kinds of datasets. These datasets differ from one another in terms of size, number of class, class distribution, complexity and sequence identity percentage. The first dataset DS1 contains 3 distinct and distant protein families. We suppose that classification in this case will be relatively easy since each family will probably have preserved patterns which are different from those of other families [Nevill-Manning 1998]. DS2 represents a bigger dataset comprising two subfamilies of protein sequences belonging to the Rhodopsin Like/Peptide family. However, the datasets DS3 and DS4 present more difficult classification problems. DS3 contains seven classes that represent seven categories of quaternary (4D) protein structure with a sequence identity of 25%. The problem here lies in recognizing the 4D structure category from the primary structure. In this case, an important question is to be answered: does the primary structure contain sufficient information to identify the 4D structure? The task relative to DS4 is that of distinguishing between the human Toll-like Receptors (TLR) protein sequences and the non-human ones. The difficulty is due to the structural and functional similarity of the two groups. The choice of this dataset came after Biologists of Pasteur Institute of Tunis (PIT) asked to help them in identifying TLR families especially human ones among the 40 TLR that exist. DS5 consists of 277 domains: 70 all-α domains, 61 all-β domains, 81 α/β domains, and 65 $\alpha+\beta$ domains from SCOP [Andreeva 2004]. This challenging dataset was constructed by Zhou [Zhou 1998] and has been extensively used to address structural class prediction [Cai 2000, Cai 2001, Cao 2006, Chou 1989, Feng 2005, Nakashima 1986, Chen 2006, Zhou 1998].

4.3.1.2 Part 2

In this part, we consider again the datasets DS3, DS4 and DS5 since they are considered to be delicate classification tasks and can thus reveal valuable information about the efficiency of the classifiers and the feature-construction methods. We try to investigate the effect of the substitution matrices variation on the quality of our encoding method and hence on the classification quality using C4.5, SVM, NB and NN algorithms. We employ all the substitution matrices used by the standalone version of Blast and belonging to the two well-known families: Blosum [Henikoff 1992] and Pam [Dayhoff 1978] *i.e.*, Blosum45, Blosum62, Blosum80, Pam30, Pam70, Pam 250.

Since DS3 is the same dataset as in [Yu 2006], these experiments allow us to compare our encoding method with other related ones presented in that paper, where the nearest neighbour algorithm NN was coupled with each of the following methods: functional domain composition FDC, amino acid composition AAC and Blast alignment tool [Altschul 1990], to predict the quaternary structures categories of the proteins. In fact, the investigation of the quaternary structures prediction using computational tools remains a task with important implications for many reasons. First, these structures are involved in many biological processes and have direct link with known diseases like sickle-cell anaemia. Second, the in vitro methods are very slow and costly in spite of being accurate. This comparison allows us to assess whether our feature-construction method could offer any benefits over the above-mentioned methods quoted in [Yu 2006] while using the same classifier (NN) and learning technique (leave-one-out).

Since prior information on the structure of a protein can provide useful information about its function, many other works similar to [Yu 2006] have investigated this topic [Cai 2000, Cai 2001, Cao 2006, Chou 1989, Chou 2000, Chou 2003, Chou 2004, Feng 2005, Nakashima 1986, Song 2004, Zhang 2003a, Zhang 2006, Chen 2006, Zhou 1998]. These works often use different kinds of amino acid composition or functional domain composition to deal with the prediction of oligomeric proteins or protein structural classes. DS5 represents a challenging dataset that has been extensively used to address structural class prediction [Chen 2006]. This allows us to compare our method with several works existing in the literature.

4.3.2 Protocol

The computations are carried out on a computer with an Intel Centrino 1.6 GHz CPU and 1GB of main memory. Results are shown in the next subsections tables. Best accuracies, for each dataset, are shown in bold and results below minimum accepted values results are underlined. The minimum accepted value (MAV) is obtained by assigning all the sequences of a dataset to its biggest class. Hence, we have 35%, 50%, 46.7%, 65% and 29.2% as MAVs respectively for DS1, DS2, DS3, DS4 and DS5. We also show the number of attributes generated by each method.

In the classification process, we use the leave-one-out technique [Han 2006] also known as jack-knife test. For each dataset (comprising n instances), only one instance is kept for the test and the remaining part is used for the training. This action is repeated n times. The leave-one-out is considered to be the most objective test technique compared to the other ones *i.e.*, hold-out, n-cross-validation. Indeed the leave-one-out test allows to obtain the same classification results regardless of the number of runs, which is not the case for the other tests (see the monograph [Mardia 1979] for the mathematical principle and [Chen 2006] for a comprehensive discussion). For the encoding methods, we use default parameters *i.e.*, NG ($N = 3$), AM ($min-length = 3$, $activity = 25\%$), DD and DDSM ($\alpha = 0, \beta = 0$ except for DS3 where $\beta = 1$ to reduce the runtime), DDSM (substitution matrix = Blosum62, substitution probability threshold $T = 0.9$). These parameters can also be specified by users.

We recall that in part 1, we use the following classifiers: C4.5 decision tree, support vector machine SVM, naïve bayes NB and nearest neighbour algorithm NN of the workbench WEKA [Witten 2005]. We generate and test the classification models; then we report the classification accuracy (rate of correctly classified sequences). Moreover, we conduct the leave-one-out test on the same datasets using Blast as already explained in Section 4.3.1. In part 2, we investigate any potential effect of the substitution matrix variance on the features building and the classification quality, and then we compare it with other classification systems quoted in [Zhou 1998, Yu 2006, Chen 2006].

4.4 Results and discussion

4.4.1 Part 1 results

The experimental results vary according to the input data (Table 4.3 and Table 4.4). The classification of the datasets DS1 and DS2 was relatively easy, as expected. Each family probably has its own motifs which characterize and distinguish it from the others. This explains the high accuracies reached by all the classifiers with all the encoding methods. But it is notable that the N-Grams encoding gave the best results although it is the simplest method to use. Moreover, since this kind of classification is easy, it does not require any sophisticated preprocessing and can simply be addressed by using alignment tools; indeed Blast arrived at full accuracy and and HMM scored almost as well as Blast(Table 4.4).

As for DS3, classification represents a real challenge. In fact, it is comprised of 717 sequences unequally distributed into seven classes which represent seven quaternary protein structure categories. It is a question of predicting the 4D structure based only on the primary structure without any complementary information. The AM method could not be used because it generates a great number of attributes (dashes in Table 4.3). The obtained accuracies with the NG and the DD methods were below the MAV (within 20.9% and 43.2%), in the same way HMMER failed to provide an acceptable result (28.7%). The result obtained by Blast was acceptable (69.60%) while the best accuracy reached (79.2%) was obtained with the DDSM method (Fig. 4.4 illustrates a sample of ROC curves [Zweig 1993] of the NB classifier based on the DDSM, DD and NG encoding methods with Homotetramer as the positive class from DS3).

The dataset DS4 was not as easy to classify as DS1 and DS2 since the human TLR and the non-human TLR resemble each other in terms of function and structure. Indeed the two classes share many similar parts, making it difficult to discriminate them. That is why alignment based classification (using Blast) didn't reach full accuracy as it did for the two first datasets. HMM allowed to abtain an accuracy above the MAV but it was less efficient than Blast. The NG and the AM encoding seem to be inefficient since they gave accuracies below the MAV with two classifiers. The DD method outperforms the two previous methods (NG and AM). Since it adopts a discriminating approach to build the attributes, it allowed a better distinction between the human TLR and the non-human TLR. But, to improve classification in the

Table 4.3: Data mining classifiers coupled with encoding methods. Mtr: Metric, Clfr: Classifer, CA: Classification Accuracy(%), NA: Number of Attributes.

			Encoding method			
Data	Mtr	Clfr	NG	AM	DD	DDSM
DS1	CA	C4.5	96.7	95	95	96.7
		SVM	**96.7**	93.3	**96.7**	**96.7**
		NB	86.7	90	81.7	80
		NN	63.3	78.3	60	61.7
	NA		4935	2060	4905	2565
DS2	CA	C4.5	99.6	99.4	99.8	99.4
		SVM	**100**	99.4	**100**	**100**
		NB	**100**	74.7	**100**	**100**
		NN	**100**	**100**	**100**	98.8
	NA		6503	7055	10058	1312
DS3	CA	C4.5	36.4	-	36.7	**79.2**
		SVM	43.2	-	43.2	78.94
		NB	43.2	-	43.1	59.4
		NN	20.9	-	21.3	77
	NA		7983	-	8403	508
DS4	CA	C4.5	60	57.5	77.5	82.5
		SVM	67.5	65	87.5	87.5
		NB	57.5	40	92.6	**95**
		NN	52.5	60	80	80
	NA		5561	3602	7116	5505
DS5	CA	C4.5	75.5	75.1	67.9	73.3
		SVM	84.1	81.2	82.3	82.3
		NB	77.3	63.7	84.5	**85.9**
		NN	80.5	79.4	78	78
	NA		6465	2393	13830	13083

dataset DS4, it is necessary to take into account the phenomenon of mutation and substitution between the amino acids which constitute the protein sequences. Indeed, the DDSM method made it possible to reach the highest precisions with all the classifiers, while reducing the number of generated attributes.

Experimental results obtained with DS5 show a good performance for all the encoding methods, though no full accuracy was reached. We can notice

Table 4.4: Comparison between Blast, Hmmer and DDSM in terms of accuracy (%).

Dataset	Blast	Hmmer	(DDSM & SVM)	Best of DDSM
DS1	100	100	96.7	96.7
DS2	100	99.21	100	100
DS3	69.60	28.73	78.94	79.2
DS4	78.57	70	87.5	95
DS5	78.3	82.76	82.3	85.9

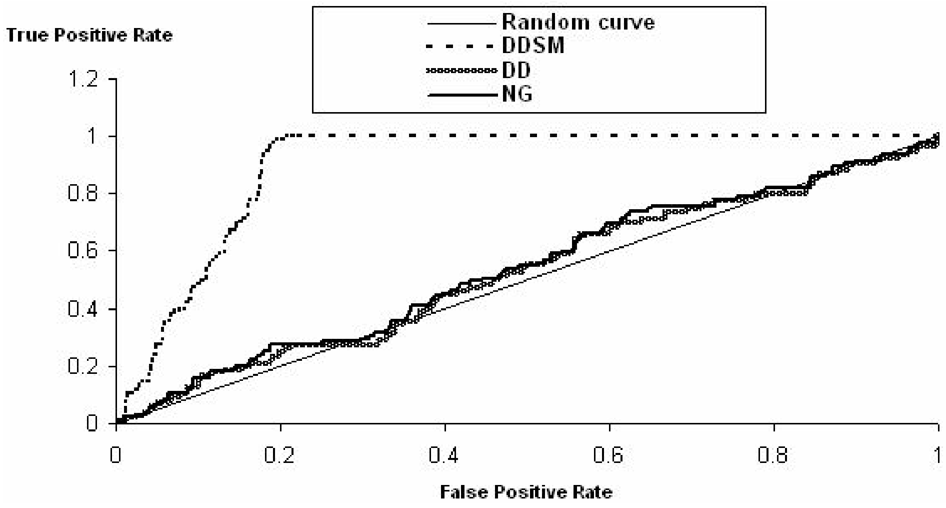


Figure 4.4: ROC curve samples for the NB classifier in the dataset DS3 with the DDSM, DD and NG encoding methods. The positive class is Homotetramer. This figure shows a sample of ROC curves of the NB classifier based on the DDSM, DD and NG encoding methods with Homotetramer as the positive class (DS3). It appears that the DDSM based ROC curve is obviously higher than the two other ones.

that NG performed very well and allowed to improve results with the classifiers C4.5, SVM and NN. Blast allowed also to obtain good accuracy which is due to the high identity percentage within the dataset and the result was even better with HMMER. But, the best accuracy was obtained with DDSM ($\simeq 86\%$).

4.4.2 Part 2 results

In this section, we study the effect of the substitution matrice variation on the classification by applying some of the most often used substitution matrices belonging to the two well-known families: Blosum and Pam [Henikoff 1992, Dayhoff 1978]. These matrices are the same used by the standalone version of Blast [Altschul 1990].

Substitution scoring is based on the substitution frequencies seen in multiple sequence alignments, yet it differs from Pam to Blosum. Whereas the Pam matrices have been developed from global alignments of closely related proteins, the Blosum matrices are based on local multiple alignments of more distantly related sequences. This would have an effect on the representation size. Indeed, the number of constructed features varies from a substitution matrix to another. Blosum matrices with low numbers and Pam matrices with higher numbers allow the building of fewer features since they score highly the substitution between amino acids. This would yield larger clusters of substitutable motifs, and hence fewer main motifs *i.e.*, fewer features (see Sections 4.2.2 and 4.2.3). However, the variances of accuracies are slight when varying the substitution matrices with the same classifier (Table 4.5, Table 4.6 and Table 4.7). Moreover, no substitution matrix allows obtaining the best accuracy for all the classifiers. We can even notice contradicting results; indeed, in DS3 and DS4, NN algorithm performs worse when coupled with Pam30, while the same matrix allows SVM to reach its best accuracy. The same phenomenon is noticed in DS5 with the classifiers C4.5 and SVM and the matrix Pam250. If one looks for reduced-size representation, Blosum matrices with low numbers and Pam matrices with higher numbers are recommended.

Since we used the same dataset (DS3) and the same assessment technique (leave-one-out) as in [Yu 2006], we compare our feature-building method (DDSM with default parameter values: $\alpha = 0$, $\beta = 0$, substitution matrix = Blosum62, substitution probability threshold T = 0.9) with the ones studied in [Yu 2006] (FDC, AAC, and Blast coupled each one with the nearest neighbor algorithm NN). Comparative results are reported in Table 4.8. We can notice that the worst results were obtained with the AAC method. Indeed, the obtained results were below the MAV 46.7%. Blast arrived at better results, but the accuracy was not very high. In fact, an analysis of the Protein Data Bank (PDB) [Berman 2000], where the protein structures are deposited, reveals that proteins with more than 30% pairwise sequence identity have similar 3D structures [Sander 1991]. But in our case we process a dataset with

Table 4.5: Experimental results per substitution matrix for DS3.

Substitution matrix	**Attributes**	**Accuracy (%)**			
		C4.5	**SVM**	**NB**	**NN**
Blosum45	377	78.5	79.2	59.4	77.7
Blosum62	508	79.2	78.9	59.4	77
Blosum80	532	77.6	80.5	60	77.6
Pam30	2873	77.8	82	60.3	76.7
Pam70	802	78.1	80.5	60.5	77
Pam250	1123	77.3	79.4	59.6	78.7

Table 4.6: Experimental results per substitution matrix for DS4.

Substitution matrix	**Attributes**	**Accuracy (%)**			
		C4.5	**SVM**	**NB**	**NN**
Blosum45	5095	82.5	85	95	80
Blosum62	5505	82.5	87.5	95	80
Blosum80	5968	72.5	87.5	92.5	80
Pam30	7005	82.5	92.5	92.5	65
Pam70	5846	82.5	85	92.5	80
Pam250	1948	82.5	77.5	95	80

Table 4.7: Experimental results per substitution matrix for DS5.

Substitution matrix	**Attributes**	**Accuracy (%)**			
		C4.5	**SVM**	**NB**	**NN**
Blosum45	12603	69.3	82.3	85.9	78
Blosum62	13083	73.3	82.3	85.9	78
Blosum80	13146	70.1	82.3	84.1	78
Pam30	13830	69.3	82.3	84.5	78
Pam70	13822	70.4	82.3	84.5	78
Pam250	1969	66.1	85.2	79.4	78

Table 4.8: Comparison with results reported in [Yu 2006] for DS3.

Methods	Accuracy %	Correctly classified sequences
DDSM & C4.5	79.2	568
DDSM & SVM	78.9	588
DDSM & NB	59.4	434
DDSM & NN	77	564
FDC & NN	75.2	539
AAC & NN	41.4	297
Blast-based	69.6	499

a sequence identity of 25%. The FDC method seems to be promising since it allowed reaching an accuracy of 75.2%. But our method was quite better and enabled to reach the highest accuracy rates among the mentioned methods and also coupled with the same classifier *i.e.*, NN algorithm (77%). If we look for better classification systems we can consider the combinations (DDSM & C4.5) or (DDSM & SVM). In addition, higher accuracy can be obtained by using the combination (DDSM & SVM) and the matrix Pam30 which enabled to reach an accuracy of 82% (Table 4.8). This indicates that SVM coupled with our encoding method DDSM represents an efficient system for protein classification.

In the same way, the use of the same dataset (DS5) and the same validation technique (leave-one-out) as in [Chen 2006, Zhou 1998] allowed us to compare our method with these two works as well as six others [Cai 2000, Cai 2001, Cao 2006, Chou 1989, Feng 2005, Nakashima 1986]. In these studies, variants of the amino acid composition AAC have been proposed to encode protein sequences and then coupled with a classifier to predict the protein structural classes. These works are based on the assumption that there is a strong correlation between the AAC and the structural class of a protein. In Table 4.9, we report the results obtained by our method (DDSM with default parameter values: $\alpha = 0, \beta = 0$, substitution matrix = Blosum-62, substitution probability threshold $T = 0.9$) coupled with C4.5, SVM, NB and NN as well as the results of the related works (in Table 4.9, AACx means the AAC variant presented in the paper x). We can claim that our encoding method generally outperforms any AAC encoding method proposed by the above-mentioned works. In [Chen 2006], authors coupled three kinds of AAC with SVM *i.e.*, (AAC & SVM), (pair-coupled AAC & SVM) and (PseAAC & SVM). In the best case, they reached an accuracy of 80.5%, whereas the combinations (DDSM & SVM) and (DDSM & NB) allowed reaching respectively 82.3% and 85.9% of accuracy. To enhance their results, authors in [Chen 2006] proposed a fusion network that combines the results obtained by the three proposed combinations and they arrived at an accuracy of 87.7%. Although, this result is slightly superior to ours, it does not mean that their encoding method outperforms DDSM. Indeed, the improvement of their results comes from the fusion network classifier and not from the AAC variants they use. Moreover, in most of these related works [Chen 2006, Zhou 1998, Cai 2000, Cai 2001, Cao 2006, Chou 1989, Feng 2005, Nakashima 1986], authors perform a fine-tuning to look for the classifier parameter values allowing to get the best results, whereas we just use the default parameter values of both our encoding method and the classifiers as found in WEKA [Witten 2005]. This fine tuning allowed to reach competitive accuracies which is the case of the combination (AAC & LogitBoost) [Feng 2005]. We believe that we can also reach higher accuracies if we perform a fine-tuning of the parameters of our method and the classifiers. But, we chose to just use the default parameter values to make it easier for users who may have no prior knowledge on what these parameters mean or how to specify them.

4.5 Conclusion

In this chapter we have proposed a new method of feature extraction to preprocess protein sequences. We have demonstrated its efficiency by comparing it with existing methods mainly in terms of classification accuracy. In the next chapter we introduce another aspect of comparison. We explore the robustness of motif extraction methods with respect to perturbations within input data. We propose new metrics to measure that robustness.

Table 4.9: Comparison with results reported in [Chen 2006] and [Zhou 1998] for DS5.

Methods	**Accuracy %**	**Correctly classified seq**
DDSM & C4.5	73.3	203
DDSM & SVM	82.3	228
DDSM & NB	85.9	238
DDSM & NN	78	216
Blast-based	78.3	220
AAC$_{\text{[Chen 2006]}}$ & SVM	80.5	223
pair-coupled AAC$_{\text{[Chen 2006]}}$ & SVM [Chen 2006]	77.6	215
PseAAC$_{\text{[Chen 2006]}}$ & SVM [Chen 2006]	80.5	223
SVM fusion [Chen 2006]	87.7	243
AAC$_{\text{[Zhou 1998]}}$ & Component coupled [Zhou 1998]	79.1	219
AAC$_{\text{[Chou 1989]}}$ & City-block distance [Chou 1989]	59.9	166
AAC$_{\text{[Nakashima 1986]}}$ & Euclidean distance [Nakashima 1986]	55.2	153
AAC$_{\text{[Cai 2000]}}$ & Neural network [Cai 2000]	74.7	206
AAC$_{\text{[Cai 2001]}}$ & SVM [Cai 2001]	79.4	219
AAC$_{\text{[Feng 2005]}}$ & LogitBoost [Feng 2005]	84.1	233
AAC$_{\text{[Cao 2006]}}$ & Rough Sets [Cao 2006]	79.4	219

Chapter 5

New stability Metrics for Feature Extraction in protein Sequences

Contents

Goals

Several previous works have described feature extraction methods for biosequence classification, but none of them discussed the robustness of these methods when perturbing the input data. In this chapter, we introduce the notion of stability of the generated motifs in order to study the robustness of motif extraction methods. We express this robustness in terms of the ability of the method to reveal any change occurring in the input data and also its ability to target the interesting motifs. We use these criteria to evaluate and experimentally compare four existing extraction methods for biological sequences. The subject of this chapter has been published in [Saidi 2010a, Saidi 2012a].

5.1 Background and related works

Several motif extraction methods have been proposed. Meanwhile, various studies have made assessments and comparisons between these methods and have tried to study the impact of one method or another on the quality of the learning task to be performed (classification, prediction, shape recognition, etc). So, the best methods are those that allow having the best values of quality metrics such as accuracy rate in the case of supervised classification.

In this chapter, we introduce the concept of stability to compare motif extraction methods. We call stability of a motif extraction method from a dataset, the non-variability in its set of motifs, when applying a technique of variation on the input data. The robustness of a method is the coupling of the non-stability and the ability to retain or improve the quality of the associated data mining task. In our case, we will use the supervised classification accuracy as a quality measure.

Concrete motivations behind the above-mentioned stability can be found within distributed systems and grid computing environments that are mining huge amounts of data. In such environments, the variation of data is a common fact. This variation can be due to various events such as failed transfer of data portions, loss of communication between nodes of the distributed system, data updating, etc. Another motivating application is information retrieval from biological databases (such as GenBank [Benson 2009], EMBL[Stoesser 2002], UniProt[Bairoch 2005]). The problem here lies in the fact that every database has its own terminology and procedures which sometimes yield related but not identical data [Zhao 2008]. Since the retrieved data are the main materials of several delicate processes in both industry and research like disease management and drug development, it is crucial for database researcher, bioscience user and bioinformatics practitioner to be aware of any change in the preparation data samples [Topaloglou 2004]. In addition, with the exploding amounts of data submitted to the biological databases, there is an increasing possibility of finding erroneous data. In such conditions, it is important to make sure that the motif extraction methods, which are the start point of any mining process, are robust enough to detect even slight variations in input data like does any good sensor when describing its context environment.

The topic of stability with respect to motif extraction methods has not been studied in the literature. However, this aspect was slightly studied in a very close field to the extraction which is the feature selection [Pavel 2007, Yvan 2008, Dunne 2002, Kalousis 2007, Somol 2008, Yu 2008].

In [Pavel 2007], authors propose a measure which assesses the stability of feature selection algorithms with respect to random perturbation in data. In this work, the stability of feature selection algorithms can be assessed through the properties of the generated probability distributions of the selected feature subsets. The interestingness is, of course, in feature selection algorithms that produce probability distributions far from the uniform and close to the peak one. Given a set of features, all possible feature combinations of size k are considered achieving n feature subsets. The frequencies F of selected feature subsets are recorded during data perturbation in a histogram. For a size k, the stability S_k is measured based on the Shannon entropy:

$$S_k = -\sum_{i=1}^{n} F_i \log F_i \ . \tag{5.1}$$

In [Yvan 2008], authors perform an instance sub-sampling to simulate data perturbation. Feature selection is performed on each of the n sub-samples, and a measure of stability is calculated. The output f of the feature selection applied on each sub-sample is compared to the outputs of the other sub-samples using Pearson correlation coefficient [Cohen 1988], the Spearman rank correlation coefficient [Yule 1950] and the Jaccard index [Real 1996] as similarity measure noted S. The more similar all outputs are, the higher the stability measure will be. The overall stability can then be defined as the average over all pairwise similarity comparisons:

$$S_{total} = \frac{2 \times \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} S(f_i, f_j)}{n(n-1)} \ . \tag{5.2}$$

In [Dunne 2002], different sub-samples (or training sets) are created using the same generating distribution. Stability quantifies how different training sets affect the feature selection output. Authors take into account three types of representations for feature subsets. In the first type, a weight or score is assigned to each feature indicating its importance. The second type of representation, ranks are assigned to features. The third type consists only of sets of features in which no weighting or ranking is considered. Measuring stability requires a similarity measure for feature representations. This obviously depends on the representation used by a given feature selection algorithm to describe its feature subset. The authors used three similarity measures: the Pearson's correlation coefficient, the Spearman rank correlation coefficient and the Tanimoto distance.

5.2 Robustness of motif extraction methods

5.2.1 Motivations

The application of the above-presented measures of stability is not convenient in our case (motif extraction). This originates from the nature of input data used by feature selection methods [Ma 2008, Saeys 2007, Liu 2007, Sebban 2002]. In fact, these methods use an original set of features (motifs)

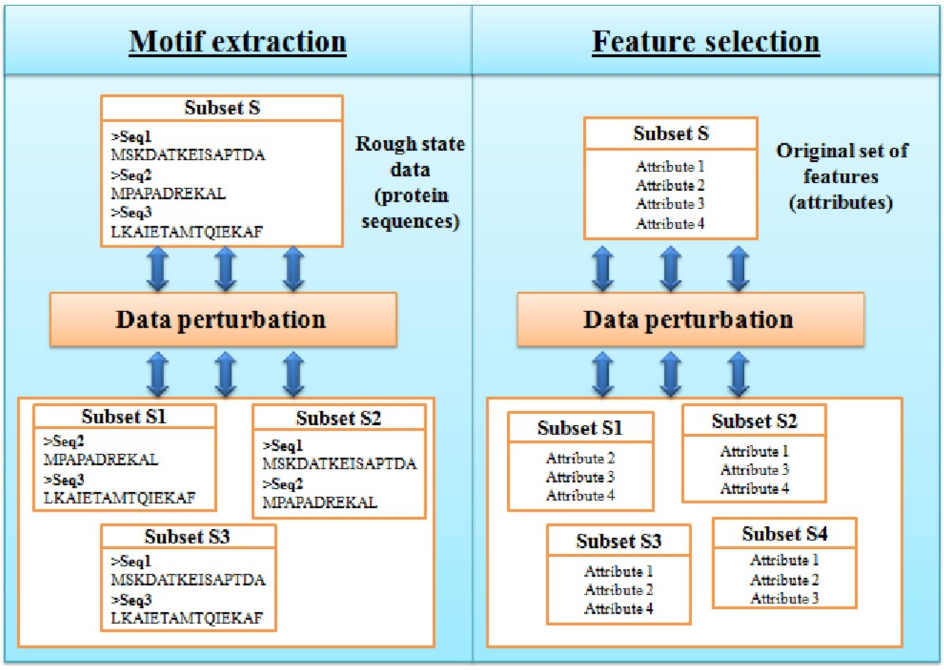


Figure 5.1: Data perturbation in feature selection and motif extraction

as input and try to merely select a subset of relevant features. The perturbation of data is applied to the original set of features. In the case of motif extraction, the input data are still in raw state and the data perturbation step is applied directly to raw data (before motif extraction step)(Fig. 5.1).

In our work, the motivation behind exploring the motif extraction method stability is to provide evidence that even slight changes in the data must also be followed by changes in the output results (extracted motifs). These changes must concern the motifs that are no longer significant for the perturbed input data; which means that the motifs that have been conserved must prove to be interesting *i.e.*, help with better classification. Since the set of features are not known a priori we can not apply the measures quoted in the feature selection related works. For our purposes let be the two following assumptions:

Assumption 1. We consider that a motif extraction method allows a reliable description of input data if any variation within these data affects the set of the generated motifs. That is to say that it reveals any change occurring in the input data.

Assumption 2. After changes in the set of generated motifs, the motifs that are conserved should be interesting *i.e.*, help with better classification.

In the next subsection we define and describe the terms we use to formally express our assumptions and to evaluate the robustness of motif extraction methods.

5.2.2 Terminology

Based on Assumption 1, we introduce the concept of sensibility . This concept reflects the ability to produce a different set of motifs *i.e.*, a different data description, whenever we make a variation within the input data. The sensitivity criterion can be studied by means of the conserved motifs called stable motifs. It is also interesting to test the Assumption 2, that is to say the quality of the stable motifs, by assessing their benefits in an artificial learning task.

Below we formally define the terms used in this paper. Consider the following:

- A dataset D, divided into n subsets D_1, D_2,.., D_n.
- A motif extraction method M applied to D on one side and to D_1, D_2,.., D_n on the other side and respectively generating the sets of motifs SM from D and SM_1, SM_2, .., SM_n from D_1, D_2,.., D_n.
- An artificial learning task T and a quality metric Mtr of T. Let $Mtr^T(SM)$ denote the value of the metric obtained if T is performed using the set SM as a feature space.

We define the following:

Definition 21 (Motif Stability) *A motif x is said to be stable if and only if its occurrence rate in all SM_i, $i = 1..n$, exceeds a threshold . The occurrence rate is simply the ratio of the number of SM_i, $i = 1..n$, where x appears to n. Formally:*

$$\frac{Number\ of\ SM_i/x \in SM_i}{n} \geq \tau, with\ i = 1..n\ . \tag{5.3}$$

Definition 22 (Rate of stable motifs) *The rate of stable motifs (RSM) of a method M is the ratio of the number of stable motifs to the number of distinct motifs of all SM_i, $i = 1..n$. Formally:*

$$RSM = \frac{Number\ of\ stable\ motifs}{|\bigcup_{i=1}^{n} SM_i|}\ . \tag{5.4}$$

Definition 23 (Method sensibility) *A method M_1 is more sensible than another method M_2 if and only if for the same changes within the same dataset, the rate of stable motifs of M_1 is lower than that of M_2. Thus, the sensibility S of a method is complementary to its rate of stable motifs. It may be noted:*

$$S = 1 - RSM\ . \tag{5.5}$$

Definition 24 (Conservation of the quality metric value) *A motif extraction method M conserves the quality metric value of a data mining task*

T if the use of the set of stable motifs SSM in T preserves the quality metric values for this task as when we use the set of motifs SM generated from the original dataset D. However, it is noteworthy that we can not judge that conservation unless the method is already sensible. Indeed, an insensible method tends to generate the same motifs even after perturbations in the input data indicating that its extraction approach is rigid and does not adopt a concept of "choice". This conservation C can be measured by:

$$C = 1 - |Mtr^T(SM) - Mtr^T(SSM)| \ . \tag{5.6}$$

Definition 25 (Interestingness of a set of stable motifs) *A set of stable motifs SSM is considered to be interesting if it allows interesting values of conservation and sensibility. Formally, we can measure this interestingness I by:*

$$I = 2 \times \frac{S \times C}{S + C} \ . \tag{5.7}$$

This measure is inspired from the F1-Score which is a statistical measure of a test's accuracy that combines Precision and Recall. The F1-score can be interpreted as a weighted average of the precision and recall, where an F1-score reaches its best value at 1 and worst score at 0. In our case, we combine conservation and sensibility to quantify the interestingness of stable motifs.

5.2.3 Illustrative example

Considering a dataset D in a supervised classification task T. The data perturbation of D generates three subsets D_1, D_2 and D_3. The application of a motif extraction method M to D on one side and to D_1, D_2 and D_3 on another side generates the sets of motifs SM from D and SM_1, SM_2 and SM_3 from D_1, D_2 and D_3 respectively :
SM =\{$m1$, $m2$, $m3$, $m4$, $m5$, $m6$, $m7$, $m8$, $m9$, $m10$\}
SM_1=\{$m1$, $m4$, $m5$, $m6$\}
SM_2=\{$m1$, $m2$, $m3$\}
SM_3=\{$m1$, $m6$, $m7$, $m8$\}

Using τ such that $\tau = 65\%$, the motifs $m1$ and $m6$ are considered stable since they appear in more than 65% of the motifs subsets.
We can easily calculate the rate of stable motifs $RSM = 0.25$, which is two over the set of eight motifs. We consider that $m9$ and $m10$ are noise, and thus are not relevant for the classification task.
The sensibility of M is calculated by $S = 0.75$.

In this case, the set of stable motifs $SSM_1 = \{m1, m6\}$.

Using $\tau > 66\%$, only the motif $m1$ are considered stable since it appears in more than $\tau\%$ of the motifs subsets.
We can easily calculate the rate of stable motifs $RSM = 0.125$.
The sensibility of M is calculated by $S = 0.875$, and $SSM_2 = \{m1\}$.

Suppose we use sets of motifs SM and SSM_1 as variables space to measure the accuracy rate (Mtr^T) of the supervised classification task T. Let consider the following obtained values with Mtr^T :
$Mtr(SM) = 0.85$
$Mtr(SSM) = 0.80$

The set of stable motifs SSM_1 enables a conservation of the quality metric value $C = 0.95$ Finally, we can measure the interestingness of stable motifs by $I = 0.83$.

5.3 Experiments

In this section, we describe an experimental study conducted on four motif selection methods quoted in [Saidi 2010b]. Calculations were run on a duo CPU 1.46GHz PC with 2GB memory, operating on Linux. The following is a presentation of the input datasets and the used tools.

5.3.1 Aims and datasets

We used four datasets containing 1327 protein sequences extracted from Swiss-Prot [Bairoch 2000] and described in Table 5.1. These datasets differ from one another in terms of size, number of class, class distribution, complexity and sequence identity percentage. The change in the nature of the datasets allows us to avoid specific outcomes to data and to have better interpretations. More description of data can be found in Chapter 4.

We compare the motif extraction methods quoted in chapter 4 [Saidi 2010b], *i.e.*, n-grams NG [Leslie 2002], active motifs AM [Wang 1994], discriminative descriptors DD [Maddouri 2004] and discriminative descriptors with substitution matrix DDSM [Saidi 2010b]. In our experiments, we use the same default settings as in [Saidi 2010b].

Comparisons made in [Saidi 2010b] between these methods revealed that DDSM performs the best to help in problems of protein sequences classification even in difficult cases where other methods fail to produce reliable descriptors for an accurate classification. In this work, we try to find a relationship between this performance and the concepts introduced in Section 5.2.2.

5.3.2 Protocol

In our experiments, we perturb each input dataset in a systematic way and we observe the impact of this perturbation on the set of generated motifs. To do this, several perturbation techniques can be adopted :

Table 5.1: Experimental data.

Dataset	Family / class
DS1	High-potential Iron- Sulfur Protein Hydrogenase Nickel Incorporation Protein HypA Glycine Dehydrogenase
DS2	Human TLR Non-human TLR
DS3	Chemokine Melanocortin
DS4	Monomer Homodimer Homotrimer Homotetramer Homopentamer Homohexamer Homooctamer

1. Removing and/or adding of sequences: This technique can be simulated by eliminating some sequences from the dateset.

2. Perturbing the sequences: This can be done by modifying amino acids within sequences.

In our experiments, we perturb each input dataset in a systematic way and we observe the impact of this perturbation on the set of generated motifs. To do this, we use the 10-cross-validation (10CV) and leave-one-out techniques (LOO) [Han 2006]. Therefore, the variation of a dataset containing n sequences consists in removing a partition (one tenth with 10-cross-validation and a single sequence with leave-one-out) from the dataset and the rest is used to generate a set of motifs. This is done several times (ten times with 10-cross-validation and n times with leave-one-out). At each iteration, the number of occurrences of generated motifs is updated. As already defined in Section 5.2.2, the technique we adopt to measure the sensibility of motif extraction methods from protein sequences is based on the rate of stable motifs. Whereas the sensibility is related to the amount of stable motifs, the interestingness of stable motifs is related to their quality. In other words, if these motifs are generated by the extraction method to appear often enough then they should be "interesting". We measure the interestingness in our experiments by their usefulness in a supervised classification task. Once the stable motifs are generated, they are used to convert protein sequences into binary vectors where the value '1' denotes the presence of motif in the sequence and '0' its

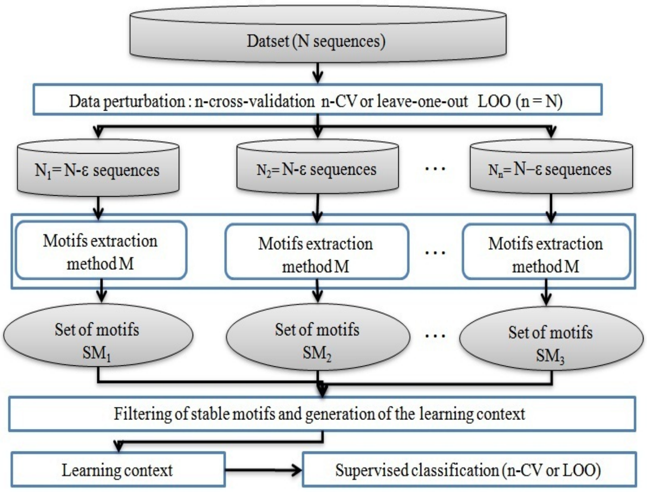

Datset (N sequences)
Data perturbation : n-cross-validation n-CV or leave-one-out LOO (n = N)
N1=N-ε sequences
N2=N-ε sequences
...
Nn=N−ε sequences
Motifs extraction method M
Motifs extraction method M
...
Motifs extraction method M
Set of motifs SM1
Set of motifs SM2
...
Set of motifs SM3
Filtering of stable motifs and generation of the learning context
Learning context
Supervised classification (n-CV or LOO)

Figure 5.2: Experimental process

absence, all these binary vectors compose what is called a learning context. Thus the classification of proteins in this new format is now possible with data mining tools. To do this, we use the support vector machine classifier SVM of WEKA workbench [Witten 2005]. The classification is performed based on 10-cross-validation (10CV) and leave-one-out (LOO) techniques. Hence, our experiments are conducted using the following four combinations for data variation and classification: (LOO; LOO), (LOO; 10CV), (10CV; LOO) and (10CV; 10CV).

5.4 Results and discussion

We show in Table 5.2 the classification results of our datasets using the four motif extraction methods. The classification is performed without making any perturbation on our datasets using the SVM classifier of WEKA [Witten 2005] based on 10-cross-validation (10CV) and leave-one-out (LOO). Comparing these results with those obtained using the stable motifs allows us to better evaluate the studied methods and test Assumption 1 and 2.

The experimental results are presented in Table 5.3 and 5.4. Table 5.3 contains the results obtained with a LOO based variation and Table 5.4 with 10CV based variation. For each dataset and for each value of τ, we note the rate of stable motifs and their corresponding accuracy if we use these motifs to classify protein sequences of that dataset based on 10-cross-validation and leave-one-out tests. We can notice that the classification test technique *i.e.*, 10CV or LOO does not affect the obtained results (the interestingness rates are almost the same). Using results from Table 5.2, 5.3 and 5.4, we draw the interestingness of stable motifs histogram corresponding to dataset (Fig. 5.3

Table 5.2: Accuracy rate of the studied methods using datasets without modification.

Method	DS1		DS2		DS3		DS4	
	10CV	LOO	10CV	LOO	10CV	LOO	10CV	LOO
NG	96.7	96.7	67.5	67.5	100	100	44.9	45.5
AM	100	100	72.5	65	100	100	-	-
DD	96.7	96.7	82.5	80	100	100	43.5	43.5
DDSM	96.7	96.7	95	95	100	100	82.5	87.5

and 5.4).

We notice that NG is virtually insensible to variations in data. Indeed, its rate of stable motifs is often equal or very close to 100%. Therefore, the variation of input data has no bearing on the generated motifs. In other words, we often obtain the same motifs even in the presence of variations in input data. In this case, we can not evoke the interestingness of stable motifs (see Fig. 5.3 and 5.4).

The AM method follows almost the same fluctuating behavior for all datasets (except for DS4 where we could not conduct our experiments due to lack of memory). In fact, below $\tau = 0.7$, AM is insensible (RSM is equal or close to 100%). Beyond this value, AM becomes sensible. This sensibility varies depending on dataset and the variation technique (10CV or LOO). It is very significant for DS1, average for DS2 and slight for DS3. For example in Table 5.3, for $\tau = 0.7$, the rate of stable motifs are 32.5, 83.6 and 98.3%, respectively for DS1, DS2 and DS3. Similarly, the interestingness of stable AM motifs is very fluctuating and varies as well depending on the dataset (see Fig. 5.3 and 5.4). This method is sometimes completive to DD and DDSM. But, we note that it is greedy in memory and can not handle large datasets as it is the case with the dataset DS4 (see Table 5.3 and 5.4).

The approach adopted by the DD method offers it a sensible nature. In fact, according to this method, each motif must satisfy the conditions of discrimination and minimality (see Chapter 4). Therefore, it is likely that a disruption of input yields not meeting these conditions and thus the elimination of some existing motifs and/or addition of new ones. At the same time, this method generates sets of interesting stable motifs with all the data samples and different values of τ . Indeed, it generally allows better interestingness rates than NG and AM (see Fig. 5.3 and 5.4).

The DDSM method is an extension of DD, which adopts a competitive approach among the motifs to generate. Indeed, to be chosen, a motif must be the most mutable among other ones of equal size. This constraint remarkably increases the sensibility of the method vis-a-vis the changes in the input data.

Table 5.3: Rate of stable motifs and their classification accuracy using LOO-based variation.

		Rate of stable motifs				**Classification accuracy rate**							
DS	τ	**NG**	**AM**	**DD**	**DDSM**	**NG**		**AM**		**DD**		**DDSM**	
						10CV	**LOO**	**10CV**	**LOO**	**10CV**	**LOO**	**10CV**	**LOO**
DS1	**0.5**	100	100	81	63	96.7	96.7	100	100	96.7	96.7	100	100
	0.6	100	100	81	60.5	96.7	96.7	100	100	96.7	96.7	100	100
	0.7	100	32.5	81	59.7	96.7	96.7	100	100	96.7	96.7	100	100
	0.8	100	31.6	80.8	59.1	96.7	96.7	100	100	96.7	96.7	100	100
	0.9	100	31.6	80.7	58.6	96.7	96.7	100	100	96.7	96.7	100	100
	1	69.3	1.8	0.01	0.02	96.7	96.7	93.3	93.3	60	28.3	30	0
DS2	**0.5**	100	100	79	62.8	67.5	67.5	72.5	65	82.5	80	95	95
	0.6	100	100	79	62.4	67.5	67.5	72.5	65	82.5	80	95	95
	0.7	100	83.6	76.9	58.4	67.5	67.5	65	62.5	80	77.5	92.5	92.5
	0.8	100	83.6	76.8	57.7	67.5	67.5	65	62.5	80	77.5	90	92.5
	0.9	100	83.6	76.7	55.5	67.5	67.5	65	62.5	80	77.5	92.5	92.5
	1	74.6	0.4	0.05	0. 01	65	67.5	62.5	62.5	65	65	65	65
DS3	**0.5**	100	100	95.6	86.7	100	100	100	100	100	100	99.8	99.8
	0.6	100	100	95.5	85.5	100	100	100	100	100	100	99.8	99.8
	0.7	100	98.3	95.2	83.9	100	100	100	100	100	100	99.8	99.8
	0.8	100	98.3	95.1	83.1	100	100	100	100	100	100	99.8	99.8
	0.9	100	98.3	94.7	81	100	100	100	100	100	100	99.8	99.8
	1	87.9	1.23	0.01	0.1	100	100	97	97.1	95.3	95.3	49	0
DS4	**0.5**	100	-	98.8	84.1	44.9	45.5	-	-	45	46	70.2	69.2
	0.6	100	-	98.8	84.1	44.9	45.5	-	-	45	46	70.2	69.2
	0.7	100	-	98.8	84.1	44.9	45.5	-	-	45	46	70.2	69.2
	0.8	100	-	98.8	83.6	44.9	45.5	-	-	45	46	69.9	69.5
	0.9	100	-	98.7	73.2	44.9	45.5	-	-	45	46	69.8	70.9
	1	90	-	0.01	69	44.9	45.5	-	-	46.7	46.7	70.6	70

Table 5.4: Rate of stable motifs and their classification accuracy using 10CV-based variation.

		Rate of stable motifs				Classification accuracy rate							
DS	τ	NG	AM	DD	DDSM	NG		AM		DD		DDSM	
						10CV	LOO	10CV	LOO	10CV	LOO	10CV	LOO
DS1	**0.5**	100	98,4	79,9	62,1	96.7	96.7	98.3	98.3	96.7	96.7	100	100
	0.6	100	96	79.7	59.9	96.7	96.7	98.3	98.3	96.7	96.7	100	100
	0.7	100	38,5	74,5	49,2	96.7	96.7	98.3	98.3	96.7	96.7	100	100
	0.8	100	28.1	66.1	37.3	96.7	96.7	100	100	96.7	96.7	100	100
	0.9	64,6	14,8	30,8	14,5	96.7	96.7	100	100	96.7	96.7	100	100
	1	64.6	14.8	30.8	14.5	96.7	96.7	100	100	96.7	96.7	100	100
DS2	**0.5**	100	94,3	64,6	49,1	65	67.5	62.5	65	92.5	92.5	62.5	62.5
	0.6	100	80	64	46.6	65	67.5	60	65	92.5	92.5	65	67.5
	0.7	100	60,6	54,4	31,2	65	67.5	65	67.5	95	95	65	65
	0.8	100	46.5	38.2	16.7	65	67.5	67.5	70	92.5	95	57.5	65
	0.9	100	32,1	9,5	4,1	65	67.5	67.5	70	92.5	92.5	37.5	35
	1	70.1	32.1	9.5	4.1	65	67.5	67.5	70	92.5	92.5	37.5	35
DS3	**0.5**	100	97,1	80,3	62,8	100	100	100	100	100	100	99.8	99.8
	0.6	100	69.3	80	60.9	100	100	100	100	100	100	99.8	99.8
	0.7	100	52,4	75,8	51,8	100	100	100	100	100	100	99.8	99.8
	0.8	100	48.5	63.8	38	100	100	100	100	100	100	99.8	99.8
	0.9	84,3	40,5	33	17,1	100	100	100	100	100	100	99.8	99.8
	1	84.3	40.5	33	17.1	100	100	100	100	100	100	99.8	99.8
DS4	**0.5**	100	-	100	86.2	44.89	45.47	-	-	45.19	45.89	70.2	69.2
	0.6	100	-	100	86.2	44.9	45.5	-	-	45.2	45.9	70.2	69.2
	0.7	100	-	100	83.5	44.9	45.5	-	-	45.2	45.9	70.4	69.3
	0.8	100	-	100	78.8	44.9	45.5	-	-	45.2	45.9	69.5	70
	0.9	99,4	-	99,4	72.8	44.9	45.5	-	-	45.2	45.9	70.6	71
	1	99.4	-	99.4	72.8	44.9	45.5	-	-	45.2	45.9	70.6	71

This can be noticed by the decreasing rates of stable motifs compared to the DD method. In addition, this high sensibility is always accompanied by a set of very interesting stable motifs manifested by generally allowing the highest interestingness rates. However, we note that for $\tau = 1$, DDSM does not often have the best rates of interestingness especially with 10CV based variation (we recall that this value of τ means that the stable motifs are those that appear in all variations of data). This is because the substitution, which is a fundamental criterion in the process of DDSM, is not taken into account in the construction of the set of stable motifs. Hence, similar forms of a given motif may be ignored. But by relaxing the condition of $\tau = 1$ and moving to smaller values of τ we see that the interestingness rates get improved considerably. This method reveals both the property of sensibility and interestingness of its stable motifs (see Fig. 5.3 and 5.4), which allows it to redescribe well the input data, which is in accordance with results of Chapter 4 showing the efficiency of this method for feature extraction in protein sequences.

5.5 Conclusion

In this chapter, we introduced the notions of stability and sensibility as new criteria to compare motif extraction methods from biological sequences. The sensibility of a method is its ability to produce a different set of motifs, so a different description, whenever a perturbation is made in the dataset. This criterion must be accompanied by a set of interesting stable motifs. This concept of interestingness arises when a method eliminates certain motif and conserves others following a change in the input data and that the conserved motifs are useful if used in a data mining task. The experimental study shows that the DDSM method is more sensible compared to the other methods. This sensibility is usually accompanied by sets of stable interesting motifs. This confirms the results of Chapter 4 that show the contribution of the DDSM method in supervised classification tasks. In the next chapter, we explore the second axis of this thesis by studying the different approaches of graph-representation of proteins.

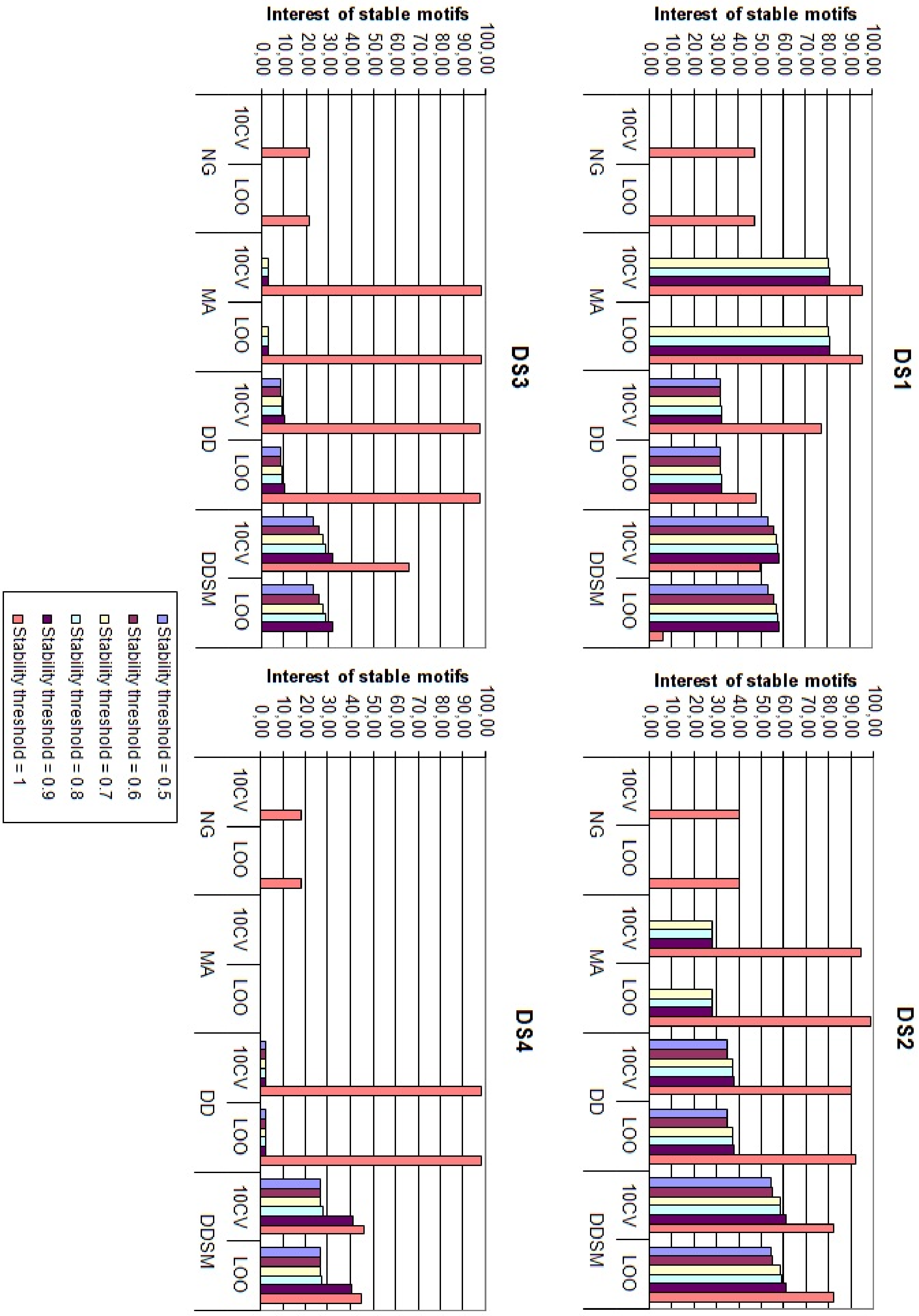


Figure 5.3: Interestingness of stable motifs using LOO-based variation

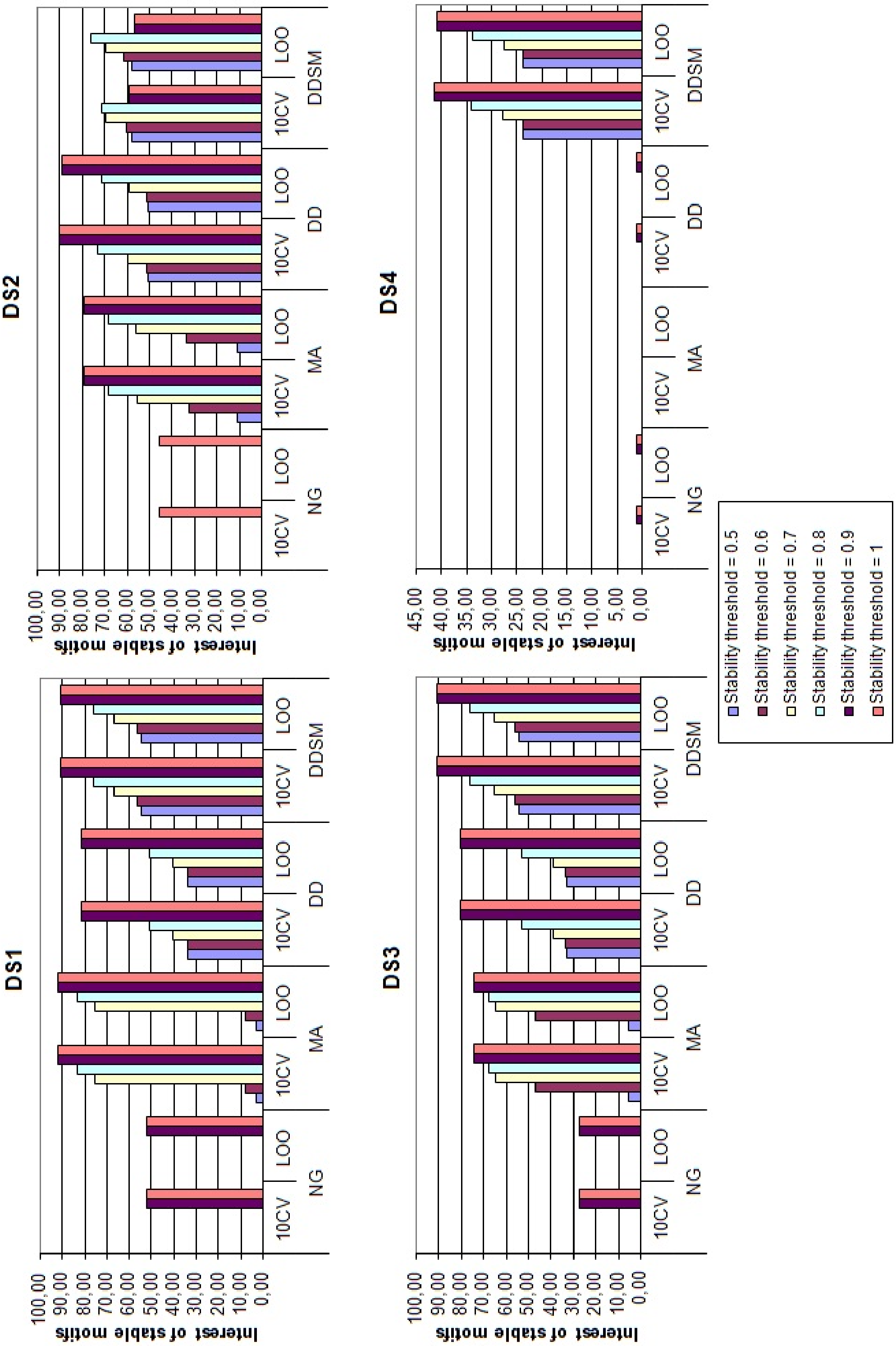


Figure 5.4: Interestingness of stable motifs using 10CV-based variation

Chapter 6

Graph-based representations of protein structures

Contents

Goals

In this chapter, we make a brief survey on various existing graph-based representations and propose some tips to help with the protein graph making since a key step of a valuable protein structure learning process is to build concise and correct graphs holding reliable information. We, also, show that some existing and widespread methods present remarkable weaknesses and do not really reflect the real protein conformation. The subject of this chapter has been published in [Saidi 2009].

6.1 Background

The structural database PDB [Berman 2000] continues to expand tremendously comprising so far more than 77000 protein structures. Hence, new methods are required to analyze, study and compare protein structures. Whereas the recognition, comparison, and classification of sequences is now more or less a solved problem, accurate computational and data mining tools for the study of the proteins in their native state are still not abundant [Kleywegt 1999, Doppelt 2007].

The investigation of 3D protein structures can give important functional and structural insights whereas the sequences fail to provide full accurate information especially in function prediction tasks [Doppelt 2007]. Indeed, during the evolution some distantly related proteins may lose sequence homology while retaining some common folding. A primordial step to any mining process or any computational study of proteins is to look for a convenient representation of their spatial conformation *i.e.*, a preprocessing step is necessary to yield computer-analyzable data [Saidi 2010b]. Since proteins are macromolecules that can be viewed as a set of related elements (amino-acids, atoms, etc), they can be translated into graphs where vertices may vary from atoms to coarser sub-structures like secondary structure fragments. In most works that studied proteins as graph structures, vertices have been represented by amino acids, the basic building blocks of proteins. In fact, amino acids are better to express any conformation homology between protein structures than using atoms or secondary structure elements. Besides, that may give valuable extensions to the previous works on sequences like motif discovery [Kleywegt 1999][Doppelt 2007].

We review several existing methods and show that they comprise fundamental points of criticisms. Based on these criticisms, we try to enhance the protein graph making. It is very important to build a graph-based representation that reflects the real protein conformation since it will be the starting point of any later processing and analysis.

In the next section, we try to make the matching between proteins and graphs in a structural view while explaining what makes proteins perform their 3D shape and introducing any technical term we use in the protein graph building. Section 6.3 exhibits the most used preprocessing methods of graph making from proteins as well as our suggestions to enhance this task. In Section 6.4, we experimentally compare two methods. Some recommendations, concluding points can be found in Section 6.5.

6.2 Proteins and graphs

A graph is a finite set of vertices (or nodes) and edges (or arcs) defined as couples $G = (V, E)$ where V is the set of vertices and E is the set of edges. Each element of E *i.e.*, edge is a pair of adjacent vertices of V. Graphs have extensively been used to investigate various real applications such as communication and transportation networks, electrical circuits, chemical components and recently to investigate the protein structure analysis [Vishveshwara 2002].

Proteins are biological macromolecules formed by concatenation of 20 distinct amino acids into long chains. They play crucial roles in almost every biological process. They are responsible in one form or another for a variety of physiological functions including enzymatic catalysis, binding, transport and storage, immune protection and control of growth, etc.

The sequence of the amino acid residues in these chains is termed the protein primary structure. These chains can fold to form complex 3D structures due to a combination of chemical interactions with the existence of some standard sub-structures called secondary structures. In the final folded state of a protein *i.e.*, tertiary structure, residues that are far away in the chain can be very close in space.

From a computer science point of view, the protein structure can be viewed as a set of elements. Each element can be an atom, an amino acid residue or a secondary structure fragment. Hence, several graph representations have been developed to preprocess protein structure, ranging from coarse representations in which each vertex is a secondary structure fragment to fine representations in which each vertex is an atom [Vishveshwara 2002].

In this chapter, we consider the graph representations that use amino acids as vertices, since amino acids are the basic structural building units of any protein which give it its properties and specify its spatial conformation. Besides, any spatial motif is more valuable if expressed in term of amino acids than atoms or secondary structure elements. This kind of spatial motifs give more information about distant proteins sharing similar functions.

6.2.1 Amino Acids and vertices

The amino acids, which are supposed to be the vertices of the graph, represent the basic building units of proteins. We recall that there exist 20 amino acids sharing a common structural scheme. An amino acid is composed of a central (but not the centroid) carbon atom called C_α and four chemical groups attached to C_α: a hydrogen atom, an amino group, a carboxyl group and a side chain or radical R (Fig. 6.1). It is the side chain that differentiates one amino acid to another and gives it its physico-chemical properties. The common parts between the amino acids compose the so called backbone [Brandon 1991].

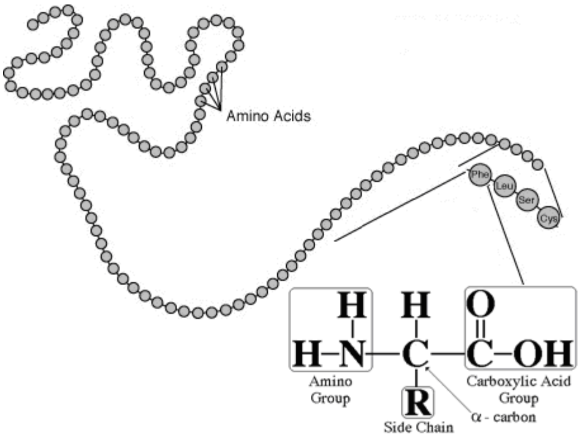


Figure 6.1: Amino acid structure and protein primary structure.

As we have mentioned, amino acids are linearly linked together to form the protein sequence called primary structure [Brandon 1991]. But the protein does not keep that linearity and continue to fold into a 3D compact shape while having local standard fragments called secondary structure due to hydrogen bonds between backbone atoms (see Chapter 2). The determination of the folded structure of a protein is a lengthy process, involving complicated methods like X-ray crystallography. X-ray method is the most accurate and most used method; it covers 86.5% of the released protein structures but it does not make hydrogen atoms visible.

6.2.2 Chemical Interactions that stabilize Proteins

Chemical interactions are the forces that hold atoms and residues together, forming molecules. The chemical interactions that stabilize proteins and give them their 3D shape can be divided into five groups: covalent bonds in which atoms share electrons, ionic bonds, hydrogen bonds, hydrophobic interactions, and Van der Waals forces [Brandon 1991].

When talking about protein graph building, these interactions are supposed to be, in one form or another, the chemical analogues of the graph edges.

6.2.2.1 Covalent Bonds

Covalent bonds involve the sharing of a pair of valence electrons by two atoms and they are the strongest chemical bonds contributing to the protein structure. In fact, protein chains are held together by covalent bonds linking between neighboring amino acids that compose the primary structure, they are also called peptide bonds and formed when the carboxyl group of one amino acid reacts with the amino group of the other amino acid, thereby releasing a molecule of water (H_2O). Peptide bonds have a typical distance of 1.5 . As well, covalent bonds between cysteine side chains can be important determinants of protein structure like in human insulin (2HUI in the PBD). Cysteine is the one and only amino acid whose side chain can form covalent bonds with other cysteine side chains. This type of covalent bond is called disulfide bridges and its length vary from to 1.8 to 3 [Steudel 1975].

6.2.2.2 Ionic Bonds

Ionic bonds, also called salt bridges, are distance-depending electrostatic attractions between oppositely charged components [Brandon 1991]. The closer the charged components are, the stronger the attraction is. Ionic bonds are rare, however, they can be important to protein structure since they allow potent electrostatic attractions approaching the covalent bond strength. Typical ionic bonds bridges have lengths of around 3.0 .

6.2.2.3 Hydrogen bonds

Hydrogen bonds (or shortly H-bonds) arise when two partially negatively charged atoms share a partially positively charged hydrogen [Brandon 1991]. Many combinations of H-bond are possible:

- Atoms on two different amino acid side-chains.
- Atoms on amino acid side-chains and protein backbone atoms.
- Backbone atoms on two different amino acids (like in secondary structures).

The range of this bond, which is the distance between the two atoms that share the hydrogen atom, is typically around 3.5 .

6.2.2.4 Hydrophobic interactions

Hydrophobic interactions are the most important non-covalent forces that make the compact shape of the protein structure. They arise when hydrophobic amino acids in the protein closely associate their side chains together in an aqueous solvent, forming interior hydrophobic protein core shielded from interactions with water [Brandon 1991].

6.2.2.5 Van der Waals forces

The Van der Waals forces are a transient, weak electrical attraction of one atom for another when electrons are fluctuating. Electrons fluctuation yields a temporary electric charge which induces a complementary dipole in another atom. Van der Waals forces are short range bonds with a radius of about 2 [Bondi 1964].

6.3 Building graphs from protein structures

In the representations that use amino acids as vertices, edges are usually defined based on distance between these vertices. However, some works expressed that edge by the strength of interaction between two amino acid side chains [Brinda 2005]. Hereafter, we mention the most widespread methods of protein graph building and try to give some tips to help with the enhancement of such task. Proteins are described by PDB files [Berman 2000] including the atoms coordinates among other information.

We assume that the correctness of a protein graph is directly linked to how much the existing edges reflect the really existing interactions that stabilize the protein even the interactions responsible of maintaining secondary structures.

Initially, we connect the vertices corresponding to the consecutive amino acids in the primary structure.

6.3.1 Main atom

Generally, works that use graphs as a means to represent proteins, abstract each amino acid u to a sole main atom of u denoted by MA_u and characterized by its spatial coordinates (x_u, y_u, z_u). This main atom can be real like C_α or C_β^1 atoms or virtual like the amino acid centroid or the side chain centroid [Lovell 2003, Huan 2005]. Two vertices u and v are said linked by an edge $e(u, v) = 1$ if the euclidian distance between their two main atoms $\Delta(MA_u, MA_v)$ is below a threshold distance δ . Formally:

$$e(u,v) = \begin{cases} 1 & if \Delta(MA_u, MA_v) \leq \delta \\ 0 & otherwise \end{cases} \tag{6.1}$$

$$\Delta(MA_u, MA_v) = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2 + (z_u - z_v)^2} \tag{6.2}$$

The time complexity of such methods is $O(n^2)$ where n is the number of amino acids.

By reviewing the literature, we found that many works use C_α atoms as main atoms and sometimes C_β with usually $\delta \geq 7$ on the argument that C_α atoms define the overall protein conformation [Huan 2005]. We experimentally tried to assess these ways of graph building and to check whether they can find already known links/bonds in proteins by making the graphs of some PDB files namely, the human insulin (2HIU). We noticed that to discover some true edges (disulfide bridges, secondary structures, etc) using C_α, δ must not be below 7 . However, such threshold would yield many false edges depending

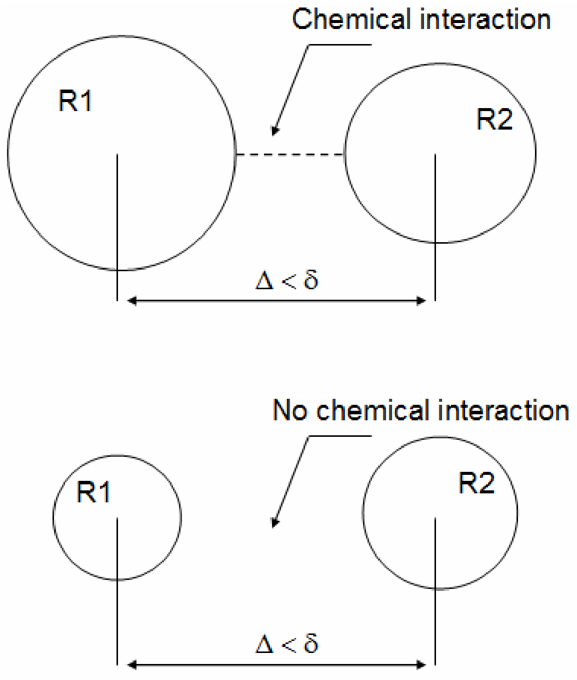


Figure 6.2: Distance between the centroids is not enough. Size of side chains must be taken into account: for the same Δ, we notice that in the upper figure, the fact that $\Delta \geq \delta$ matches with the existence of a chemical bond since the side chain boundaries are enough close to each other which is not the case for the lower figure.

on the positions and orientations of amino acids. Indeed, it is obvious that we can find many amino acids, whose C_α atoms are within the range of 7 and even less, which are not concerned by any chemical interaction especially when their side chains are far away to each other. Using high threshold greatly increases the number of edges without really matching with existing chemical interactions *i.e.*, false edges. Whereas using low threshold does not help with finding true edge especially in the case of amino acids having big side chains making C_α atoms far away from each other. The same criticisms can be induced concerning the use of C_β.

As for the use of the side chain centroid (or radical centroid RC), this method is better to detect bonds between side chains. But, it is weak concerning bonds linking backbones to backbones or backbones to side chains. It presents another weakness since it does not take into account the size of side chains which differs from an amino acid to another, so a threshold δ that allows to reflect a real chemical interaction between two side chains does not necessarily do the same for two other side chains especially when they are small in size (see Fig. 6.2). We recommend assimilating the side chain into a sphere with a ray ρ, to use lower threshold and to replace Formula 6.1 by the following:

$$e(u,v) = \begin{cases} 1 & if \Delta(RC_u, RC_v) - (\rho_u + \rho_v) \leq \delta \\ 0 & otherwise \end{cases} \tag{6.3}$$

The use of the amino acid centroid may be better than using C_α or C_β. But this method is also concerned by the remarks on the importance of taking into account the size of the amino acids. So, similar formula to Formula 6.3 can be utilized.

6.3.2 All atoms

Since atoms are directly concerned by bonding, one can think of building edges by examining distances between all atoms of two given amino acids. We did not find in the literature any work that uses this method to build graphs of amino acids, but it has been used with graphs of atoms [Jacobs 2001, Keating 2009]. The principle is similar to the one in Section 6.3.1: Two vertices u and v are said linked by an edge $e(u,v) = 1$ if there exist two atoms A_u, A_v belonging respectively to u and v and whose euclidian distance $\Delta(A_u, A_v)$ is below a threshold distance. Formally:

$$e(u,v) = \begin{cases} 1 & if \; \exists \; A_u \; \in \; u \; and \; A_v \; \in \; v : \Delta(A_u, A_v) \leq \delta \\ 0 & otherwise \end{cases} \tag{6.4}$$

The time complexity of such methods is $O(n^2 * m^2)$ where n is the number of amino acids and m is the number of atoms.

This method reduces the possibility of obtaining false edges but it seems to be slower than the ones of Section 6.3.1. Based on the information about the interactions that stabilize proteins, we propose tips that reduce the complexity of all-atoms method to $O(n^2)$ while keeping almost the same output. First, we simplify an amino acid residue into a block of three main components: a side chain, a positively charged N backbone atom and a negatively charged O backbone atom all linked to $C\alpha$ atom (Fig. 6.3). We know that interactions between amino acids u and v can take the following schemes:

- side-chain R_u with side chain R_v,
- side chain with backbone atom (O or N),
  - R_u — O_v
  - R_u — N_v
    - R_v — O_u
    - R_v — N_u
- backbone atoms of u and v (O and N).

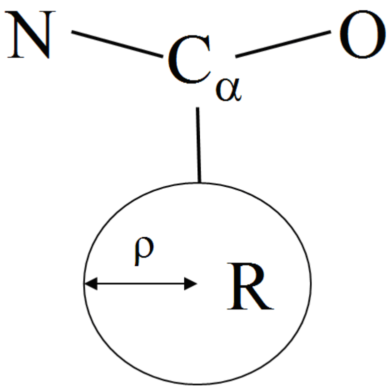


Figure 6.3: The main three components involved in interactions stabilizing the protein: O and N backbone atoms and side chain with ray ρ

- N_u — O_v
- N_v — O_u

Hence, there is no need to check the distances between all atoms but just the seven possibilities mentioned above. If we denote by RC the side chain centroid and by ρ its ray, the edges of the graph are built based on the following formula:

$$e(u,v) = \begin{cases} 1 & if \Delta(RC_u, RC_v) - (\rho_u + \rho_v) \leq \delta 1 \\ 1 & if \Delta(RC_u, O_v) - \rho_u \leq \delta 2 \\ 1 & if \Delta(RC_u, N_v) - \rho_u \leq \delta 2 \\ 1 & if \Delta(RC_v, O_u) - \rho_v \leq \delta 2 \\ 1 & if \Delta(RC_v, N_u) - \rho_v \leq \delta 2 \\ 1 & if \Delta(N_u, O_v) \leq \delta 3 \\ 1 & if \Delta(N_v, O_u) \leq \delta 3 \\ 0 & otherwise \end{cases} \tag{6.5}$$

Interactions between side-chains can be any of the mentioned ones in Section 6.2.2 whereas interactions between a side chain and a backbone atom (O or N) or between O and N of two different amino acids, can be an ionic attraction or a h-bond. Then, we can use a value from 3.5 to 4 for $\delta 1$, $\delta 2$ and $\delta 3$

6.3.3 Triangulation

The triangulation is a process that transforms a geometric object P, composed of a set of points, to a set of simplices with all points from P are being among the vertices of the triangulation [De Berg 2008]. In particular, if P is a plan (respectively or a 3D-space), then, a triangulation is a way to subdivide P into a collection of triangles (respectively or tetrahedra).

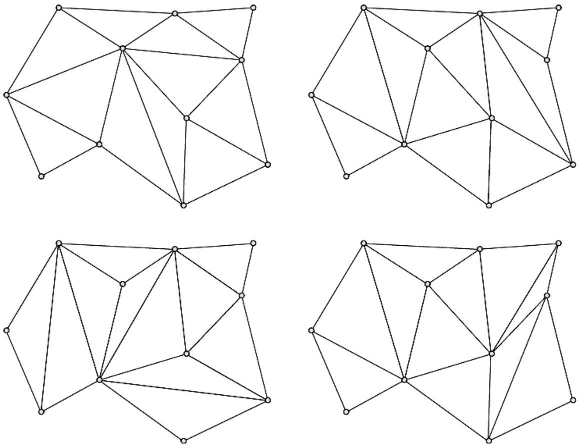


Figure 6.4: Four different triangulations of the same point set.

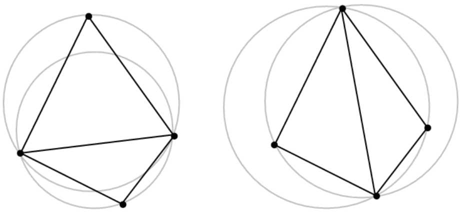


Figure 6.5: Triangulation samples in a 2D space. Left: triangulation does not meet the Delaunay condition (empty circum-circle). Right: represents a Delaunay triangulation sample.

Triangulation is used in various applications, such as nearest neighbour research, navigation, astronomy, reconstruction of geometrical spaces, determining the properties of a topological space. It is notable that several triangulations of a same geometric object are possible (see Fig. 6.4). A special way of triangulation is the Delaunay tessellation [Delaunay 1934, De Berg 2008] which is the geometric dual of the Voronoi diagram. There exist subsets of the Delaunay triangulation [Stout 2008] which are the Gabriel graph, nearest neighbour graph and the minimal spanning tree and extensions such as the Almost Delaunay Tessellation [Bandyopadhyay 2004]. Several works have used Delaunay triangulation to build protein graph [Huan 2005, Stout 2008, Bostick 2004]. They mainly used one of the main atoms described in Section 6.3.1. It is about creating tetrahedra such that no main atom is inside the circum-sphere of any tetrahedron *i.e.*, empty sphere property (Fig. 6.5). Many algorithms have been proposed to perform the Delaunay triangulation with various time complexities. See [Amenta 2007] for more details.

In addition to the criticisms in Section 6.3.1 concerning the use of main atoms, this method presents other weaknesses. In fact, we can find very far vertices in the protein graph which are linked especially at the surface of the protein where the circum-spheres are getting out of the cloud of atoms. Besides, Delaunay triangulation omits many true edges and does not allow to, exactly, describe hydrophobic cores which must be seen as highly cross linked sub-graphs. This is fundamentally due to the empty sphere property that does not allow to one vertex to make edges with other vertices out of its tetrahedron sphere even in the presence of interactions. Possible enhancements could be the use of a threshold δ to eliminate long edges and taking into account the remarks done in Section 6.3.1. But in general, we do not recommend using the Delaunay tessellation to build protein graphs.

6.4 Experimental comparison

In Section 6.3, we report that the most widespread method, which is the one based on distance between C_α atoms (CA method with a threshold $\delta = 7$), may contain several false edges. We also recommend building protein graphs by considering distances between all atoms (AA method) with a threshold $\delta = 3.6$.

In this section, we try to experimentally compare the two methods *i.e.*, CA and AA. To evaluate the quality of the graph building methods, we tried to extract discriminative features (DF, see definition below) from two datasets. The datasets are retrieved using the advanced search of the Protein Data Bank [Berman 2000] with no more than 50% pair-wise sequence similarity in order to remove highly homologous proteins. To ensure using high quality data, we selected X-Ray proteins with resolution ≤ 3 Å. The first dataset (DS1) includes two distant protein families that belong to two different SCOP classes. The first family is the nuclear receptor ligand-binding domain proteins (NLB) from the all alpha class and the second one is the prokaryotic protease family (PP) from the all beta class. The second dataset (DS2) includes two close families of eukaryotic proteases (EP) and the prokaryotic proteases (PP the same one of DS1). These two families belong to the same superfamily (see Table 6.1 for datasets description).

Table 6.1: Experimental data.

Dataset	Family	Size (#proteins)	#Amino acids
DS1	NLB	15	5545
	PP	12	4931
DS2	PS	12	4931
	PP	17	5570

We use Subdue system [Ketkar 2005] to extract DFs. We denote DF(DS, F, GBM) which means the DF of the family F in the dataset DS under the graph building method GBM. We define a DF of a family F as a subgraph, having at least three vertices, that only appears within F. For each DF, Subdue computes a score S defined by:

$$S(DF) = \frac{NPos(DF) + NNeg(DF)}{|DS|} \tag{6.6}$$

S measures the specificity of DF to a family F where NPos(DF) is the number of positive proteins (belonging to F) containing DF, NNeg(DF) is the number of negative proteins not containing DF and |DS| is the size of the dataset. For each family, we just consider the largest discriminative feature (LDF) *i.e.*, the DF containing the greatest number of vertices and edges and having the highest score S. We assume that the better is the graph building method, the better is the quality of the extracted LDFs. However, the S metric does not allow comparing LDF of different sizes since it is obvious that larger LDF would have lower S. Then, we define a new comparative score (CS) that takes into account the size and the S of a LDF:

$$CS(LDF) = |LDF| \times +S(LDF) \tag{6.7}$$

|LDF| represents the LDF size *i.e.*, the number of vertices of LDF. The CS indicates how much a LDF can be as large as possible while being specific to its family. Then, it can be compared with other LDFs of different sizes within the same family. It is majorated by the number of vertices of the smallest graph of each family (case where the smallest graph of a family F is a subgraph of all the graphs of F); so it can be normalized if one would get a probabilistic value between 0 and 1. Before evaluating the graph-making methods, it is notable to report some observations. First, the graphs built using AA method contain fewer edges. They also structurally differ from the ones built using CA method and so are the extracted DFs. The fact that CA and AA methods allow to obtain structurally different DFs, makes us strongly ask a fundamental question about which DFs are biologically meaningful. The second observation concerns DS2. It is well known that prokaryotic proteins are more primitive than eukaryotic ones which contain more structural components. Indeed, results show that the PP family has no DF discriminating it from EP whereas the latter has even large and frequent DFs. Experimental results are summarized in tables 6.2 and 6.3. We note that AA method always enables to reach the best comparative scores CS. That means the extracted LDFs, under the AA graph representation, represent the best tradeoffs between the size and the specificity. On another side, we reported in Section 6.3.2 that every edge under AA method is likely to represent a chemical interaction between two amino acids. Hence, the comparison between LDF(DS2, EP, CA) and LDF(DS2, EP, AA) shows that CA method fails to represent

existing chemical interactions as edges and does not allow to extract larger LDF containing true edges.

Table 6.2: Results for DS1

Dataset	DS1			
Family	NLB		PP	
Method	CA	AA	CA	AA
#edges	21379	20055	20595	14361
#DFs	16	6	4	8
LDF shape	(6v,6e)	(4v,3e)	(4v,3e)	(6v,5e)
S	0.59	0.96	1.00	0.70
CS	3.54	3.84	4.00	4.20

Table 6.3: Results for DS2

Dataset	DS2			
Family	NLB		PP	
Method	CA	AA	CA	AA
#edges	20595	14361	22424	15985
#DFs	0	0	21	43
LDF shape	\	\	(17v,20e)	(20v,25e)
S	\	\	0.72	0.66
CS	\	\	12.24	13.20

6.5 Conclusion

Protein structures are complex data that represent an important area to be discovered and mined, but it must first be encoded into a computer-analyzable data. Recently, proteins have been seen as graphs, mainly, graphs of amino acids. We have reviewed the most used existing methods of protein graph making, but before that, we tried to make the matching between proteins and graphs and to exhibit the various interactions that make some amino acids get close to each other and make the protein fold in its 3D shape. Based on that, we made several criticisms about the reviewed methods and proposed one possible and simple enhancement. We assume that the quality of any protein graph is directly depending on how much that graph reflects the real conformation of the protein *i.e.*, how much the existing edges reflect the really existing interactions that stabilize the protein including the interactions responsible of maintaining hydrophobic cores and secondary structures. We conducted an experimental comparison, based on the largest discriminative feature extraction, between two graph-making methods *i.e.*, CA and AA. We noticed that the extracted features vary in terms of composition, size and quality according to the used method.

We have implemented the mentionned methods and other ones in java language into a jar file available upon request or on my home page http://fc.isima.fr/~saidi. The program accepts protein PDB files as input and outputs graph files of amino acids and edges between them under several format. We also implemented a web repository for graph-represented proteins [Dhifli 2010] (see Appendix D)

In the next chapter, we propose a new method to extract spatial motifs from protein graphs which can be a way to assess the quality of graph building and on other side a means to perform some machine learning tasks such as classification.

Chapter 7

Ant-Motifs: Novel Spatial Motifs

Contents

Goals

In this chapter, we propose a novel algorithm to find spatial motifs from protein structures by extending the Karp-Miller-Rosenberg (KMR) repetition finder dedicated to sequences. The extracted motifs obey a well-defined shape proposed based on a biological basis. These motifs are used to perform various supervised classification tasks on already published data. Experimental results show that they offer considerable benefits, in protein classification, over sequential motifs, spatial motifs of recent related works and alignment-based approaches. We also show that it is worthy to enhance the data preprocessing rather than only focussing on the optimization of classifiers. The subject of this chapter has been published in [Saidi 2012b].

7.1 Background and related works

An essential starting point for any mining process or any computational study of proteins is to define a convenient computer-analyzable representation of their internal components and the existing links between them. Proteins are commonly known as strings of characters (or sequences), where each character represents an amino acid. This linear representation has been very useful in bioinformatics and data mining applications [Dominic A. Clark 1990, Mephu Nguifo 1993, Lemoine 1999, Yongqiang 2006b, Yongqiang 2006a, Battaglia 2009, Saidi 2010b]. However, it fails to provide full accurate information especially in function prediction and classification tasks, whereas the investigation of the spatial shape of proteins can give important functional and structural insights [Cootes 2003, Clark 1991]. Indeed, proteins have been recently seen within graph representations and studied based on graph theory concepts.

In this regard, many topics have been explored. Some works have been interested in the study of protein structures based on their graph properties and involve the use of topological classifications as in [Bartoli 2007] where it has been shown that proteins can be considered as small world networks of amino acids. Other works have looked for identifying residues that play the role of hubs in the protein graph that stabilize the structure [Vallabhajosyula 2009], predicting pathways from biological networks [Faust 2010], inferring protein-protein interaction networks [Brouard 2011, Brouard 2012], performing structural classification by means of graph based clustering [Santini 2010]. Another current trend in many recent studies focuses on the subject of discovering motifs from protein structures and using them as features to perform protein classification[Fei 2010].

Our work explores this last topic. For our purposes, we recall the definitions of a *motif* and a *sequential motif*, and we give the definition of a *spatial motif*:

Definition 26 (Motif) *In general, a motif (or pattern) consists of a non-null finite feature that can characterize a given population P of objects. This motif may be identified based to its high frequency in P, its rarity in other populations or based on other parameters. In our case, a motif is considered to be a significant set of linked amino acid residues found in proteins.*

Definition 27 (Sequential motif) *This consists of a motif whose composing residues are contiguous in the primary structure i.e., it is a sub-chain extracted from the protein chain.*

Definition 28 (Spatial motif) *This consists of a motif whose composing residues are not necessarily contiguous in the primary structure i.e., it contains linked residues that are far away in the chain, termed distant residues.*

In literature, many classification methods based on tertiary structures mainly use spatial motifs as features to characterize protein groups. Since protein tertiary structures can be interpreted as graphs, many contributions around the frequent subgraph discovery were used in the pre-processing step of the classification process, whereas other contributions focused on the learning process and proposed new boosting algorithms using discovered motifs as base learners [Fei 2010].

Many algorithms have been proposed on frequent subgraph discovery [Krishna 2011] which are generally classified into two main categories, namely the Apriori-based approaches and the pattern growth approaches. This classification of frequent subgraph discovery methods was basically dependent on the use or non-use of the apriori information in the extension of subgraphs during the search step [Hong Cheng 2010]. Apriori-based approaches, for instance AGM [Inokuchi 2000], FSG [Kuramochi 2001], FFSM [Huan 2003], etc, starts with small-size subgraphs and proceeds in a bottom-up manner. New subgraphs are generated by joining two slightly different frequent subgraphs among those already generated. Hence, the size of newly discovered frequent subgraphs is increased iteratively by one. The last step of each iteration is to check the frequency of the newly formed subgraph. However, pattern growth approaches, for instance MoFa [Borgelt 2002], Gspan [Yan 2002], Gaston [Nijssen 2004], etc, do not perform expensive join operations since they just extend a frequent subgraph directly by adding a new edge in every possible position. The problem of this method is that the same graph can be discovered multiple times, so these approaches are supposed to run a pruning procedure to avoid duplicates.

Meanwhile, other algorithms have been proposed as boosting methods where discovered motifs were used as base learners. Kudo et al. [Saigo 2009] proposed gboost, a boosting method devoted to labeled graphs classification where, based on a weak classifier called decision stump, they iteratively construct multiple weak classifiers on weighted training instances using subgraphs as classification features. Saigo et al. [Saigo 2008] suggested another approach called gPLS which, in the same time, uses Partial Least Square regression to mine graph data, iteratively performs feature selection and classifier construction. gPLS is very similar to gboost in the mining approach. A structural leap search approach called LEAP [Yan 2008] was proposed by Yan et al. for mining the most significant subgraph patterns. Two new mining concepts were explored in this approach namely structural leap search and frequency descending mining; these two concepts reduce the search space and thus mine patterns faster since both of them are related to specific properties in pattern search space. COM [Jin 2009], proposed by Ning *et al.*, is another classification method where they derive graph classification rules based on pattern co-occurrence. A very recently proposed pattern based graph classification method is LPGBCMP [Fei 2010], proposed by Fei and Huan. The main idea of this method is to develop a boosting algorithm where base learners *i.e.*, subgraphs, have structural relationships in the functional space.

Though spatial motifs in graph format comprise important information that sequential ones fail to provide, the main disadvantage, of using subgraphs as features is the high complexity of their extraction. Indeed, it is known to be NP-complete. In addition, the number of extracted features is expected to be very high since proteins are very dense molecules. Hence, the use of such motifs may hinder the classification process. For these reasons, we propose a novel way to represent spatial motifs. This representation simplifies the existing graph format of motifs while taking into account the links between distant amino acids in the primary sequence. Our spatial motifs are termed *ant-motifs*.

7.2 Ant motif

7.2.1 Biological basis

It is commonly known that the tertiary structure of a protein depends on its primary structure [Gille 2000]. Thus, two homologous proteins with high sequence similarity ($>$ 80% identical amino acids) will also have very similar structures, whereas the reverse is not necessarily true. The prediction of tertiary structures from primary structures has been an active field of research in bioinformatics. Additionally, many methods use exactly the homology between proteins to achieve their predictions. It has also long been known that certain amino acids favor the formation of certain folds over others [Pearson 2001]. For example, proline and glycine have a very low propensity to form α helices. These spatial links allow to obtain the native tertiary structure starting from the sequence.

In fact, many bioinformatics methods use only the protein sequence to predict the tertiary structure by means of sequential motifs [Gille 2000]. Usually, sequential motifs have functional meanings. Some of them are easy to recognize (*e.g.*, zinc finger motif) since they are uninterrupted; this is not the case with many other motifs, in which the spacing between their elements and even their order can vary considerably [Gille 2000]. Moreover, structure is more stable than sequence. Indeed, preserved sequential regions can be lost across the evolutionary time, whereas spatial links persist longer.

Given this, it would be judicious to keep the preserved sub-sequences as bases of motifs and feed them with spatial information.

7.2.2 Shape and definition

Based on what we have mentioned above, we propose a novel shape of a spatial motif preserving a sequential part from the primary structure and abstracting the spatial information by links with distant residues, which give the motif an ant-like shape (see Figure 7.1 for the shape and see Figure 7.2 for a real ant-motif example, sampled from our experiments). The sequential part represents the largest preserved sub-sequence while the spatial links indicate the types of distant residues to which the sequential part residues are connected. To extract this kind of motifs we propose an adaptation of the Karp, Miller and Rosenberg algorithm [Karp 1972] (for more details see Appendix C). The following definition gives a more formal description of ant-motifs:

Definition 29 (Ant motif) *Let be a population of proteins represented as graphs of amino acids. The nodes of each graph are ordered based on their occurrences in the primary structure. An ant motif is a spatial motif composed of a sequential motif, called* sequential part*, and other edges such that each node of index i in the sequential part may be connected to other nodes whose indices are strictly greater than i+1.*

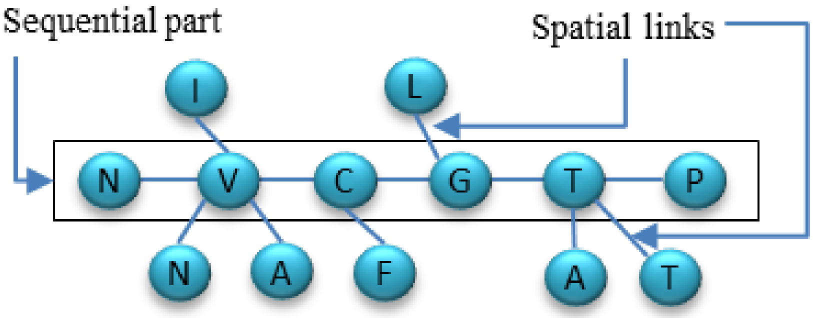


Figure 7.1: Shape of an ant-motif. The sequential part is composed of amino acids from the primary structure and the spatial links represent their connections to distant residues.

7.2.3 Algorithm details

7.2.3.1 Preliminary

In our case, ant-motif extraction differs from the problem solved by KMR. Indeed, the notion of equivalence in KMR is limited to contiguous elements in a sequence *i.e.*, for a given k there may be no more than one k-equivalence E_k between two positions. We change the definition of the equivalence to

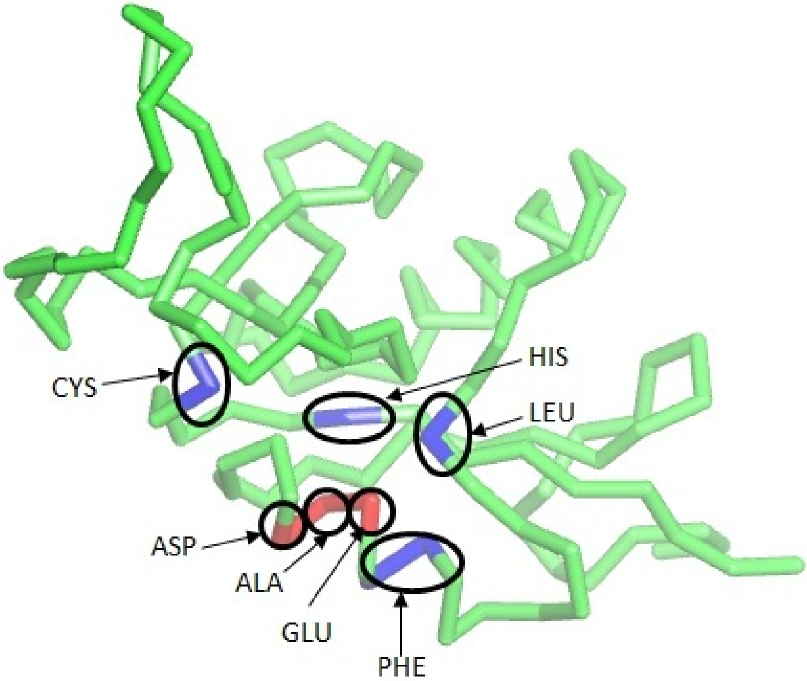


Figure 7.2: An example of an ant-motif sampled from our experiments (in C-type lectin domains (DS4)). Red residues (ASP, ALA and GLU) represent the sequential part and the blue ones are distant residues.

be adapted to our needs. Hence, two positions may have many equivalence relations (Figure 7.3).

Definition 30 (Equivalence in graphs) *Two positions i and j in a graph G are k-equivalent, we note $i\ E_k\ j$, if and only if $G[i] = G[j]$ and $G[i]$ is linked to k nodes identical to other k nodes linked to $G[j]$. If i and j have more than one equivalence, we note $i\ E_k^r\ j$, where r is the equivalence number (see Figure 7.3).*

The construction of ant-motifs is done incrementally throughout the sequential part, using the KMR lemma (see Appendix C) and according to the KMR definition of equivalence (see Appendix C). But before applying KMR lemma, all spatial links must be built for each node of the sequential part. When two positions have more than one equivalence relation, it is necessary to merge the resulting motifs. Hence a larger equivalence is inducted using the following lemma:

Lemma 3

$$i\ E^1_{k_1}\ j\ \&\ i\ E^2_{k_2}\ j\ \&\ ...\&\ i\ E^n_{k_n}\ j\ \Leftrightarrow i\ E_{((\sum^i_n k_i)-(n-1))}\ j \tag{7.1}$$

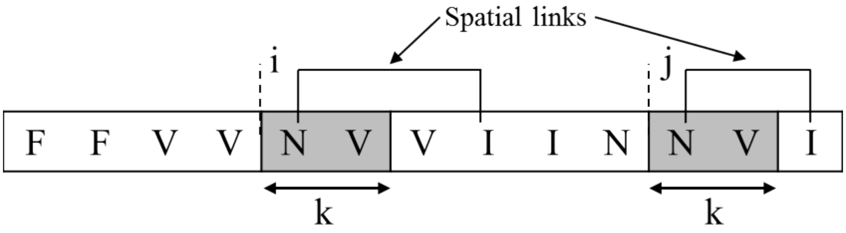


Figure 7.3: Illustration of two 2-equivalences between positions $i = 5$ and $j = 11$: $i \; E_2^1 \; j \; \rightarrow \; \{ N, V \}$ and $i \; E_2^2 \; j \; \rightarrow \; \{ N, I \}$.

Proof 4 *Let be a graph G and two positions i and j having n equivalence relations $E_{k_1}..E_{k_n}$. The first equivalence E_{k_1} allows to build a motif of size k_1, we note it M_1. Each equivalence relation E_{k_i} of the remaining $(n-1)$ equivalences will increase the size of M_1 by new $k_i - 1$ nodes since M_1 already contains the node corresponding to positions i and j. This implies that the final motif will be of size $((\sum k_i)-(n-1)), i = 1..n$. Hence $i \; E_{((\sum_n^i k_i)-(n-1))} \; j$. For example in Figure 7.3, the two 2-equivalences between $i = 5$ and $j = 11$ allow to induce a new 3-equivalence.*

7.2.3.2 Algorithm and data structures

Graph nodes are ordered according to their positions in the protein primary structure. To avoid processing raw graphs, we parse them into new sequential codes. An index table describing all edges existing in the graph accompanies each graph sequential code. The latter is constructed by inserting between two contiguous nodes (two contiguous residues in the primary structure) with indices i and $i + 1$ all the distant residues with higher indices, having edges with the node i (illustrative example in Section 7.2.4). The node $i + 1$ and the mentioned distant nodes spatially linked to node i are termed *successor* nodes. All index tables are concatenated to form a global index table, GIT.

Motifs are incrementally built using KMR lemma and Lemma 3. In each level we adopt a stack-based implementation to look for common positions allowing concatenating or merging motifs of low levels. In each level two stack families P and Q are used. The number of stacks in each family is the number of motifs extracted in the previous level (see AntMot algorithm).

7.2.3.3 Complexity

The time complexity of KMR algorithm for a string of characters or an array has been proved to be $O(x) \log x$ where x is the size of the data [Karp 1972]. If we do not take into account the spatial aspect of motifs the complexity of our

Algorithm 2: AntMot

```
Data: $GIT$.
Result: $\Omega$: set of ant-motifs.
begin
    $M_1 \longleftarrow$ set of distinct node labels;
    $k \longleftarrow 2$;
    $\Omega \longleftarrow M_1$;
    while $k \neq -1$ do
        construct $|M_{k-1}|$ $P$ and $|M_{k-1}|$ $Q$ stacks;
        for $i = 1$ to $|GIT|$ do
            push $i$ in $P_{GIT[i]}$;
        foreach stack $I \in P$ stacks do
            foreach element $i \in I$ do
                pop $i$;
                $s \longleftarrow$ number of successors of $GIT[i]$;
                for $j = 1$ to $s$ do
                    push $i$ in $Q_{GIT[i+j]}$;
            foreach stack $O$ in the $Q$ stacks do
                if $|O| \geq 2$ then
                    construct a motif based on KMR lemma;
            if $k = 2$ then
                merge motifs satisfying Lemma 3 ;
        if $M_k = \varnothing$ then
            $k \longleftarrow -1$ ;
        else
            $\Omega \longleftarrow \Omega \cup M_k$ ;
            remove redundant motifs from $\Omega$ ;
```

algorithm can be reduced to the same complexity of KMR, with $x = n * m$ where n is the number of proteins and m is the number of amino acids of the biggest protein. Taking into account Lemma 3 (spatial links) increases the data size *i.e.*, the size of GIT will be multiplied by δ, where δ is the maximum number of spatial link. Therefore GIT size can be approximated by $n * m * \delta$. In the worst case, *i.e.*, the case of complete graphs, δ is equal to m. However this is practically impossible and $\delta << m$ for steric constraints [Ramachandran 1968]. Since no proved theorem has shown that δ is bounded by a constant, so the time complexity of AntMot is $O(n * m^2)\log(n * m^2)$.

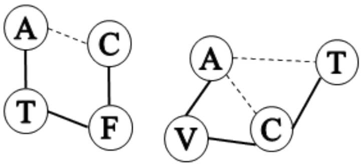
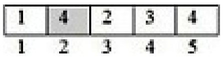


Figure 7.4: Toy proteins. Left: protein 1, right: protein 2. Continuous lines represent the primary structure and the dashed lines represent the spatial links.

7.2.4 Illustrative example

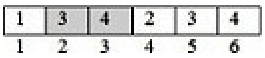
Given the following toy protein structures (see Figure 7.4) from which we attempt to discover common motifs:

- Protein 1:
  - Primary structure 1: ATFC
  - Graph 1 : Figure 7.4 left
  - Graph sequential code 1: ACTFC
  - Index table 1: the grey box indicates that node 1 is spatially linked to node 4.



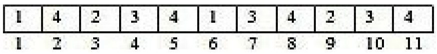
1	4	2	3	4
1	2	3	4	5

- Protein 2:
  - Primary structure 2: AVCT
  - Graph 2: Figure 7.4 right
  - Graph sequential code 2: ACTVCT
  - Index table 2: the grey boxes indicate that node 1 is spatially linked to node 3 and node 4.



1	3	4	2	3	4
1	2	3	4	5	6

- Global index table GIT:



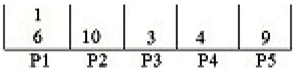
1	4	2	3	4	1	3	4	2	3	4
1	2	3	4	5	6	7	8	9	10	11

- M_1 : set of motifs of level 1 → M1 contains 5 motifs. Each one contains a node corresponding to an amino acid letter:

- f 1: Nodes {A} - Edges {}
- Motif 2: Nodes {C} - Edges {}
- Motif 3: Nodes {T} - Edges {}
- Nodes {F} - Edges {}
- Nodes {V} - Edges {}

- Building M_2

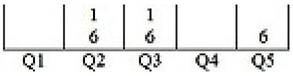
  - Stacking in P stacks :



P1	P2	P3	P4	P5
1				
6	10	3	4	9

  - Unstacking from P into Q stacks :

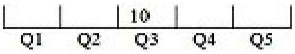
    - Unstacking P1



Q1	Q2	Q3	Q4	Q5
	1	1		
	6	6		6

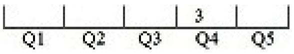
→ New motifs:
- Nodes {A, C} - Edges {1-2} - positions: 1, 6
- Nodes {A, T} - Edges {1-2} - positions: 1, 6

    - Unstacking P2



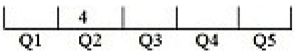
Q1	Q2	Q3	Q4	Q5
		10		

    - Unstacking P3



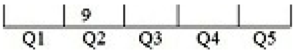
Q1	Q2	Q3	Q4	Q5
			3	

    - Unstacking P4



Q1	Q2	Q3	Q4	Q5
	4			

    - Unstacking P5



Q1	Q2	Q3	Q4	Q5
	9			

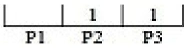
→The fusion of motifs sharing common nodes in the primary structure generates the motif below (lemma 3):
- Nodes {A, C, T} - Edges {1-2, 1-3} - positions: 1, 6

→ M_2:

- Motif 1: Nodes {A, C} - Edges {0-1}- positions: 1, 6
- Motif 2: Nodes {A, T} - Edges {0-1}- positions: 1, 6
- Motif 3: Nodes {A, C, T} - Edges {0-1, 0-2}- positions: 1, 6

- Building M_3

- Stacking in P stacks :



- Unstacking from P into Q stacks :

○ Unstacking P1

○ Unstacking P2

○ Unstacking P3

→ No new motifs are found in level 3.
→The motifs of level 2 kept after eliminating all redundant submotifs are:

- Motif 3: Nodes {A, C, T} - Edges {0-1, 0-2}- Positions: 1, 6
- The two other motifs of level 2 are discarded because they are included in the motif 3.

Remark 1 *If we had used only the sequence representation, we would have found no motifs.*

7.3 Experiments

7.3.1 Aims

The experimental study is composed of three parts. In the first part, we are interested in two aspects *i.e.*, the interestingness of motifs when used in the protein classification and the runtime of their extraction. To study the first aspect, we carry out motif-based classification experiments on various datasets. We study the effect of three types of motifs on the classification performance, namely, the sequential motifs (SM), the frequent-subgraph motifs (FSM) and the ant-motifs (AM). To study the second aspect, we compare our algorithm AntMot and a state of the art method of frequent subgraph extraction in terms of runtime. We study the behaviour of both methods when increasing the amount of data and varying the frequency threshold of motifs. In both criteria, *i.e.*, interestingness and runtime, the number of motifs is an influencing factor that must be discussed.

In the second part, we experimentally investigate the impact of the graph building method on the quality of extracted motifs and consequently on the classification performance. To do that, we use a different method to construct graphs of amino acids (the AA method recommended in Chapter 6) and we study if it has an obvious impact on the classification performance.

In the third part, we compare ant-motif-based classification with other classification approaches that use alignment or that focus on optimizing the classifier rather than enhancing the preprocessing.

7.3.2 Datasets

To perform our experiments, we use the same six datasets of protein structures as in [Fei 2010]. In each dataset, positive proteins are sampled from a selected protein family whereas negative proteins are randomly sampled from the Protein Data Bank [Berman 2000]. A detailed description of the data collection process can be found in [Jin 2009]. Table 7.1 summarizes the characteristics of the six datasets : the related protein family ID in the SCOP database [Andreeva 2008], the description of the protein family, the number of positive and negative samples.

Table 7.1: Experimental data from [Fei 2010]. ID: identifier of protein family in SCOP, Pos: positive proteins sampled from a selected protein family, Neg: negative proteins randomly sampled from the PDB

Dataset	ID	Family name	Pos	Neg
DS1	48623	Verteb. phospho. A2	29	29
DS2	52592	G proteins	33	33
DS3	48942	C1 set domains	38	38
DS4	56437	C-type l. domains	38	38
DS5	56251	Proteasome subunits	35	35
DS6	88854	Kin., cata. subunits	41	41

7.3.3 Settings

Our algorithm, AntMot, is implemented in java (see Appendix E for details). It allows extracting ant-motifs (AM) as well as sequential motifs (SM) from the protein primary structures. Since the existing algorithms of frequent subgraph discovery are supposed to extract the same number of motifs, we choose to use Gaston [Nijssen 2004], known to be the fastest among them, to extract frequent-subgraph motifs (FSM). We use the java implementation of Gaston available in the Parsemis graph-mining suite (available at https://www2.informatik.uni-erlangen.de/EN/research/ParSeMiS/index.html).

The experiments were conducted on a 3 Ghz quad core CPU. The memory management depends on the algorithm used. Indeed, Gaston is memory consuming and requires, in some tests, tens of GB of RAM; whereas 1 GB is largely enough for AntMot.

7.3.3.1 Graph building settings

Proteins are parsed into graphs where nodes represent amino acid residues and are labeled with the amino acid type. We use two methods quoted in Chapter 6 to construct the edges. The first one is based on the euclidian distance between C_α atoms of amino acids (we term it CA) whereas the second takes into account the distance between all atoms in the amino acids (we term it AA). Hereafter we recall the two methods:

CA method Proteins are parsed into graphs where nodes represent amino acid residues and are labeled with the amino acid type. Two nodes u and v are linked by an edge $e(u, v) = 1$ if the Euclidean distance between their two C_α atoms $\Delta(C_\alpha(u), C_\alpha(v))$ is below a threshold distance δ. Formally:

$$e(u,v) = \begin{cases} 1, & if\ \Delta(C_\alpha(u), C_\alpha(v)) \leq \delta \\ 0, & otherwise \end{cases} \tag{7.2}$$

In the literature, most works use this method with usually $\delta \geq 7$ Åon the argument that C_α atoms define the overall shape of the protein conformation [Huan 2005]. In our experiments, we use $\delta = 7$ as in [Fei 2010].

AA method Two nodes u and v are said to be linked by an edge $e(u, v) = 1$ if there exist two atoms A_u and A_v belonging respectively to u and v and whose Euclidian distance $\Delta(A_u, A_v)$ is below a threshold δ. Formally:

$$e(u,v) = \begin{cases} 1, & if\ \exists\ A_u\ \in\ u\ and\ A_v\ \in\ v\ :\ \Delta(A_u, A_v) \leq \delta \\ 0, & otherwise \end{cases} \tag{7.3}$$

We have proposed AA method in Chapter 6 based on the assumption that peripheral atoms are directly involved in bonding within the protein rather than C_α atoms [Saidi 2009]. In our experiments, we use $\delta = 4$ Å.

7.3.3.2 Classification settings

The classification process is illustrated by Figure 7.5. The input data are protein structures in PDB format [Berman 2000]. First, proteins are parsed into graphs as described in Section 7.3.3.1. Then, we use a motif extraction method to find features. Each protein is encoded as a binary feature vector indexed by the extracted motifs with values indicating the presence (1) or absence (0) of the related motif. Finally, we apply a classifier on the binary data. We use 5 replicates of 5-cross validation as evaluation technique. Each of the six datasets is partitioned into five folds. Four are used for training and one fold is reserved for testing. AM, SM and FSM motifs are generated with a minimum frequency = 0.3 and motif sizes between 3 and 7 nodes. In order to show that the obtained classification results are not biased by the classifier, we use two different classifiers from the workbench Weka [Witten 2005] with default parameters, namely support vector machine (SVM) and naïve bayes (NB).

7.3.3.3 Runtime test settings

To better study the variation of runtime with a larger amount of data, we gathered all distinct graphs of the six datasets previously used, to form a single dataset of 392 graphs. We run AntMot and Gaston on this dataset to discover respectively AM and FSM while varying the minimum frequency threshold τ of motifs from 0.1 to 1. In this experiment, no constraints on the motif size are put. In other words, each algorithm must find all its related motifs satisfying τ.

7.4 Results and discussion

7.4.1 Comparing motif extraction methods

In this subsection the CA method is used to build graphs from proteins.

7.4.1.1 Motif interestingness in classification

In this section, we report the accuracy, sensitivity and specificity of the combination of SVM and NB with the three types of motifs namely SM, AM and FSM. We recall that sensitivity is (TP / (TP+FN)), specificity is (TN / (TN+FP)) and accuracy is ((TP+TN) / S), where TP stands for true positive, TN stands for true negative, FP stands for false positive, FN stands for

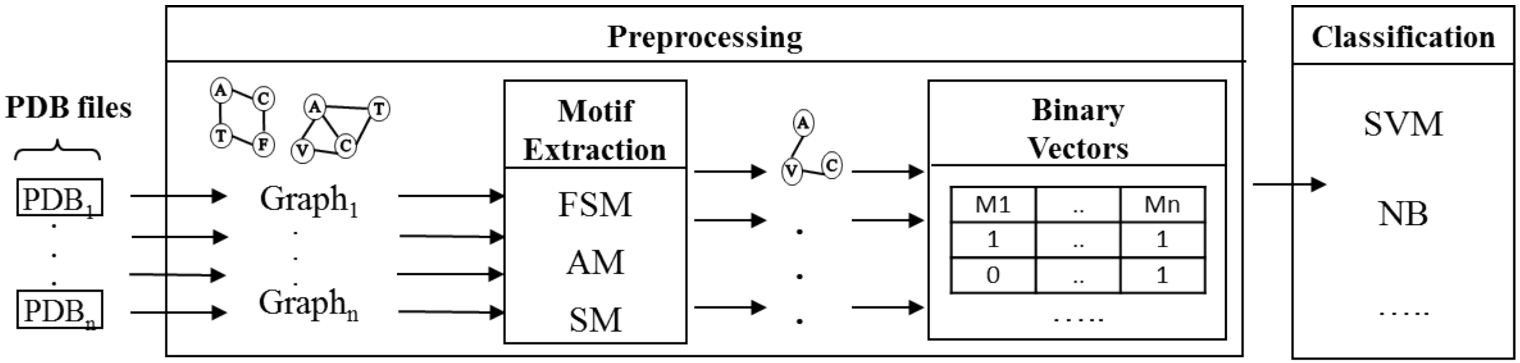


Figure 7.5: Classification process. First, proteins are parsed into graphs. Then, motifs are extracted and used as features. to encode proteins into binary vectors with values indicating the presence (1) or absence (0) of the related motif. Finally, binary vectors are used as input for classifiers.

false negative and S stands for the number of samples. Since the standard deviation is around 0% and 7% for all these methods, we do not list it here.

Experimental results can be found in Table 7.2 and Fig. 7.6. We notice that GASTON generates a large number of motifs, especially with DS1 and DS6. This illustrates the case of *information overload* [Hasan 2009], a real obstacle for any further use of motifs. That is why the classification performance was negatively affected compared to the other approaches. Meanwhile, FSM motifs perform better with SVM than NB. This is explained by the selective ability of SVM. SM motifs, which are motifs extracted from the protein primary structures, present a reduced number. Yet, they allow better classification performance than FSM in most cases. This highlights the fact that the gain in spatial information offered by the FSM motifs, is promptly lost due to their large number. The number of AM motifs is reasonable compared to FSM. This comes with the best classification performance with both SVM and NB. This is due to the AM structure which combines recurrent sequential regions from the primary structures enriched by recurrent spatial links (see Figure 7.2 for a real ant-motif example). Generally, AM comes first, followed by SM and FSM.

To better understand the accuracy differences, we plot the average sensitivity and specificity of all methods in Figure 7.7. It is obviously clear that AM outperforms FSM and SM in terms of sensitivity and specificity with almost all datasets. Overall, FSM and SM seem to have a good compromise between sensitivity and specificity.

Table 7.2: Number of motifs with frequencies more than 30% and having between 3 and 7 nodes.

Method	Number of discovered motifs					
	DS1	DS2	DS3	DS4	DS5	DS6
SM	23	11	54	12	15	26
AM	1994	2007	2132	1366	1152	1725
FSM	1081501	680525	196867	80290	764356	951657

7.4.1.2 Runtime

As mentioned in Section 7.3.3.3, the six datasets were gathered into one single dataset of 392 graphs. AM and GM motifs were extracted from this dataset with respect to different frequencies varying from 0.1 to 1, without any limitation in size.

In Figure 7.8 (top), we notice that the runtime for both ANTMOT and GASTON is inversely proportional to τ (the frequency threshold). Indeed, lower τ yield more motifs (see Figure 7.8 (bottom)). Hence, too much comput-

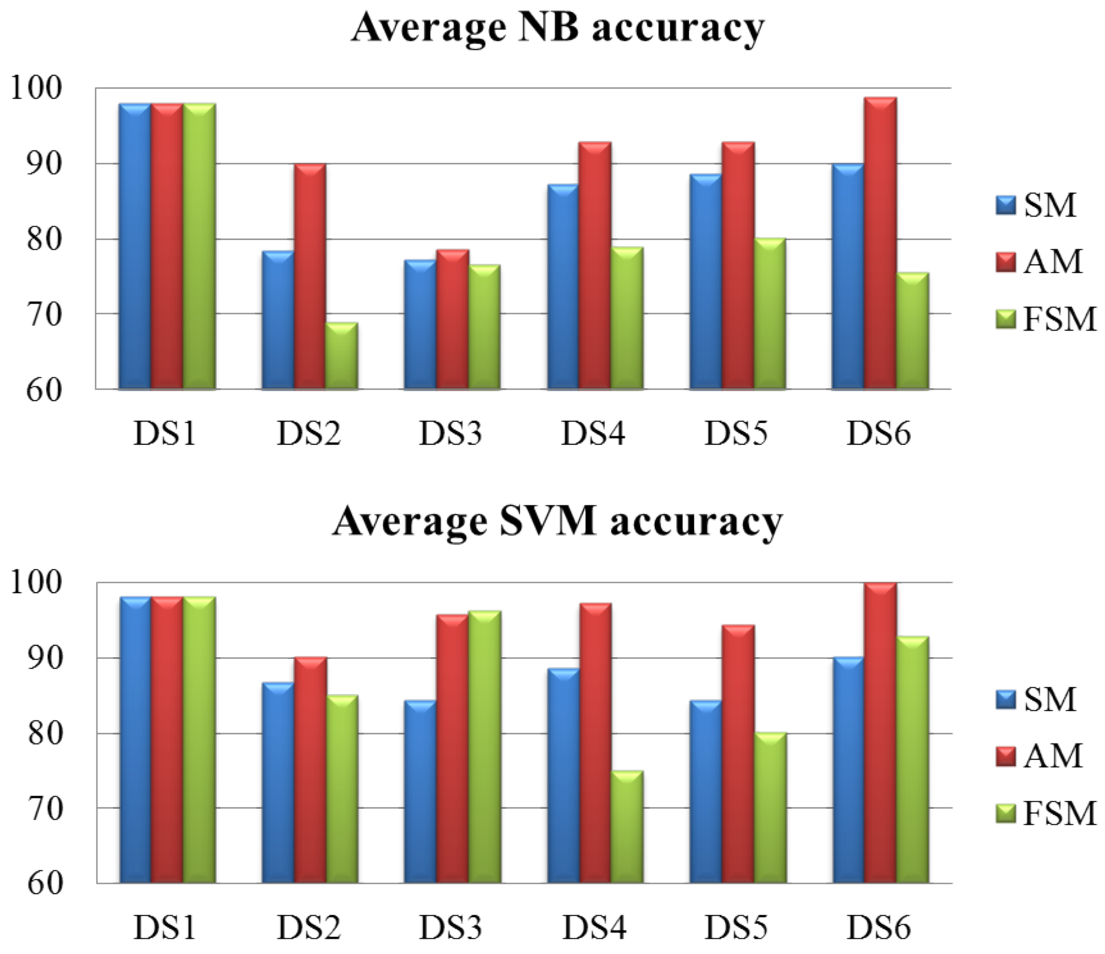


Figure 7.6: Average accuracy results using SM, AM and FSM as features and SVM and NB as classifiers.

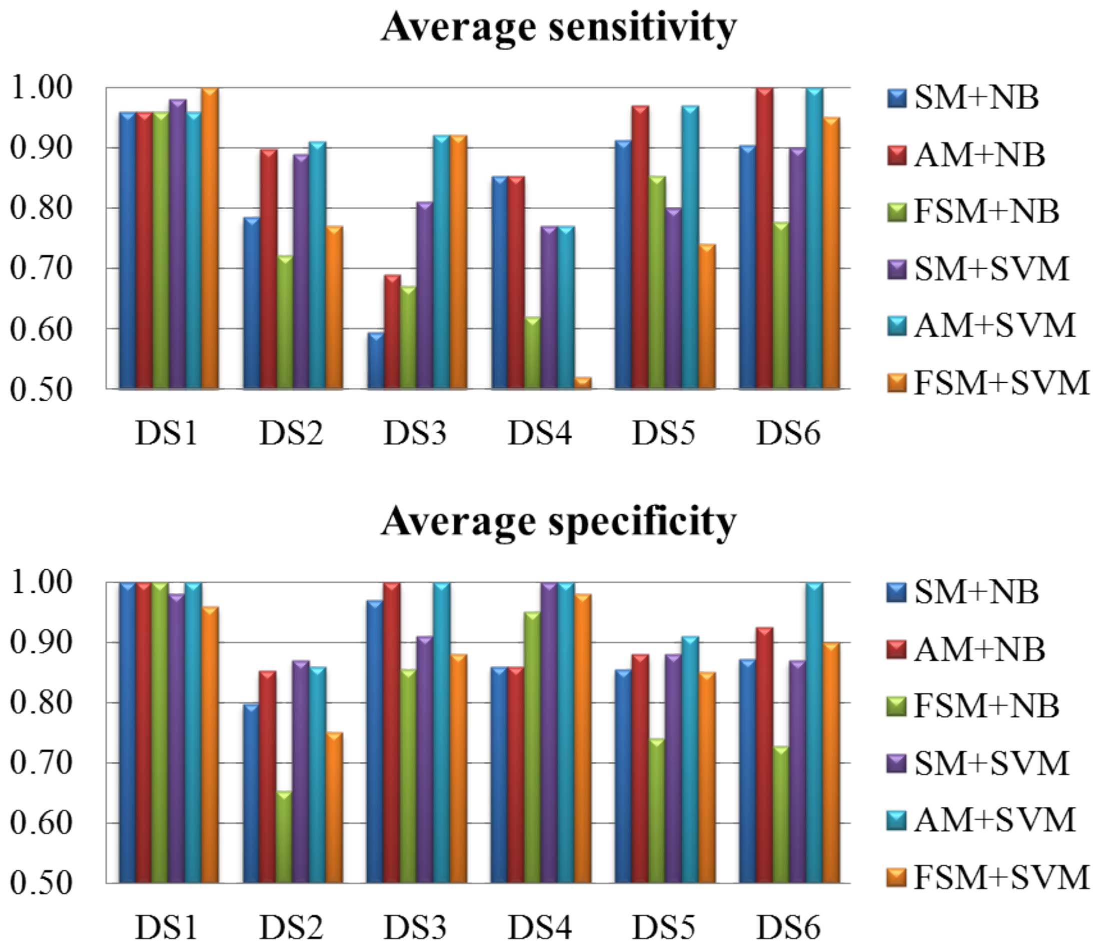


Figure 7.7: Average sensitivity and specificity results using SM, AM and FSM as features and SVM and NB as classifiers.

ing is required. Although GASTON is slightly faster than ANTMOT for higher τ, the latter scales remarkably better to lower τ. Indeed, ANTMOT runtime increases almost linearly with an approximate gradient of 1.75, whereas GASTON runtime increases exponentially to reach more than 3 days with $\tau = 0.1$. The runtime behaviours of both algorithms are consistent with the evolution of the number of motifs extracted by each of them. Indeed, The difference in runtime between the two algorithm derives from the huge difference in the number of discovered motifs.

7.4.2 Impact of the graph building method

In this section we evaluate the effect of the graph building on the representation of proteins and consequently on the classification performance. We perform classification on the same six datasets and using the same protocol and settings previously detailed. Yet, we change only the graph building method. We use the AA method to build the protein graphs, then we compare the classification results with those previously obtained using CA. To do so, we extract again spatial motifs, *i.e.* AM, then we use them for classification using NB and SVM.

The results obtained from our experiments are illustrated in Fig. 7.9. It is clearly remarkable that the use of AA enhanced the classification accuracy compared to CA, especially with DS1 where it reaches full accuracy. This fit with our claim in Chapter 6 that AA gives a better graph representation of the considered protein structures. Hence, the generated graphs with AA are closer to the real spatial conformation of proteins than those built with CA. This is simply due to the fact that AA consider the distances between all the atoms of each amino acid of the protein to build the edges, whereas C_α consider only the distances between the C_α atoms and thus edges between amino acids that are based on atoms other than C_α are missing in the graph representation. In addition, it is true that when using AA to build the graphs, our motif extraction method AM still outperforms FSM. However, the enhancement of the classification performance came both with AM as well as with FSM. This proves that the protein-graphs built using AA have a generic representation and are not suited only to our method AM. In addition, according to the previous experiments, sequential motifs SM performed better in classification compared to the FSM motifs using CA as graph building method. Yet, the use of AA for graph building enabled FSM to outperform SM in classification and thus enhanced the quality of the discovered spatial motifs.

7.4.3 Comparison with other classification approaches

Dealing with protein classification requires, above all, the comparison with the most widespread and favorite tool of biologists, *i.e.*, the alignment. It would be

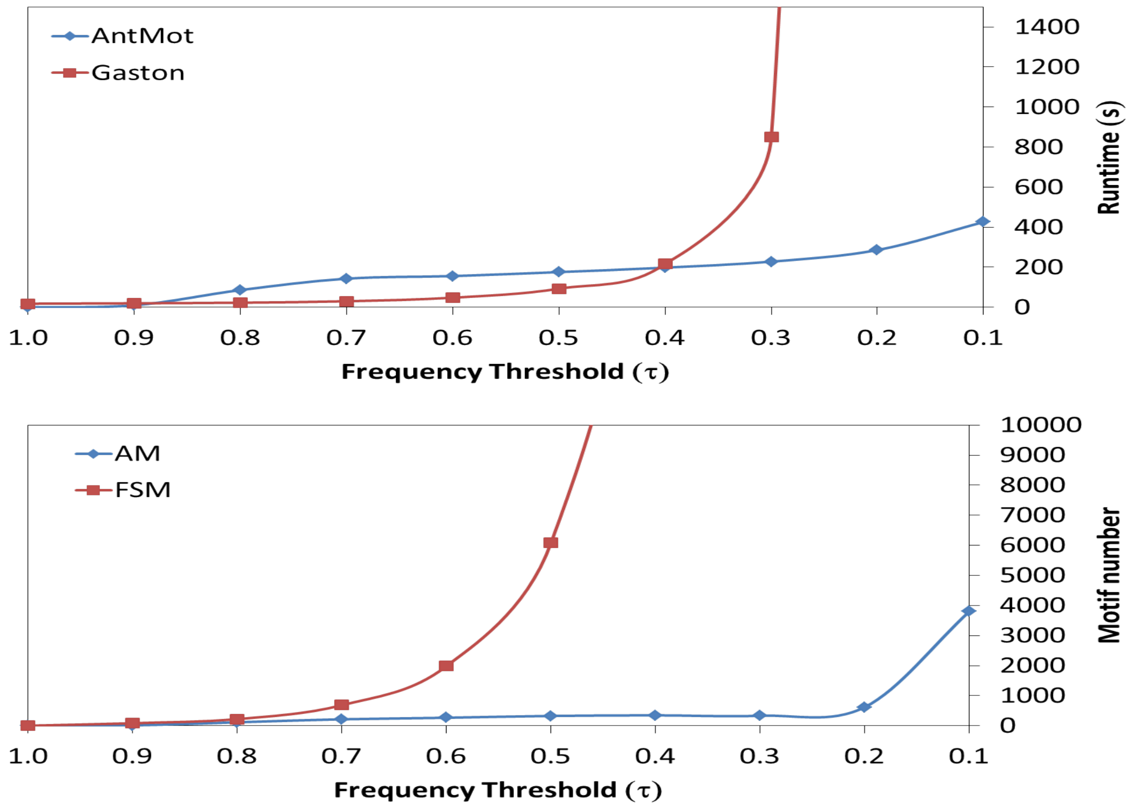


Figure 7.8: Evolution of runtime and number of motifs depending on the minimum frequency.

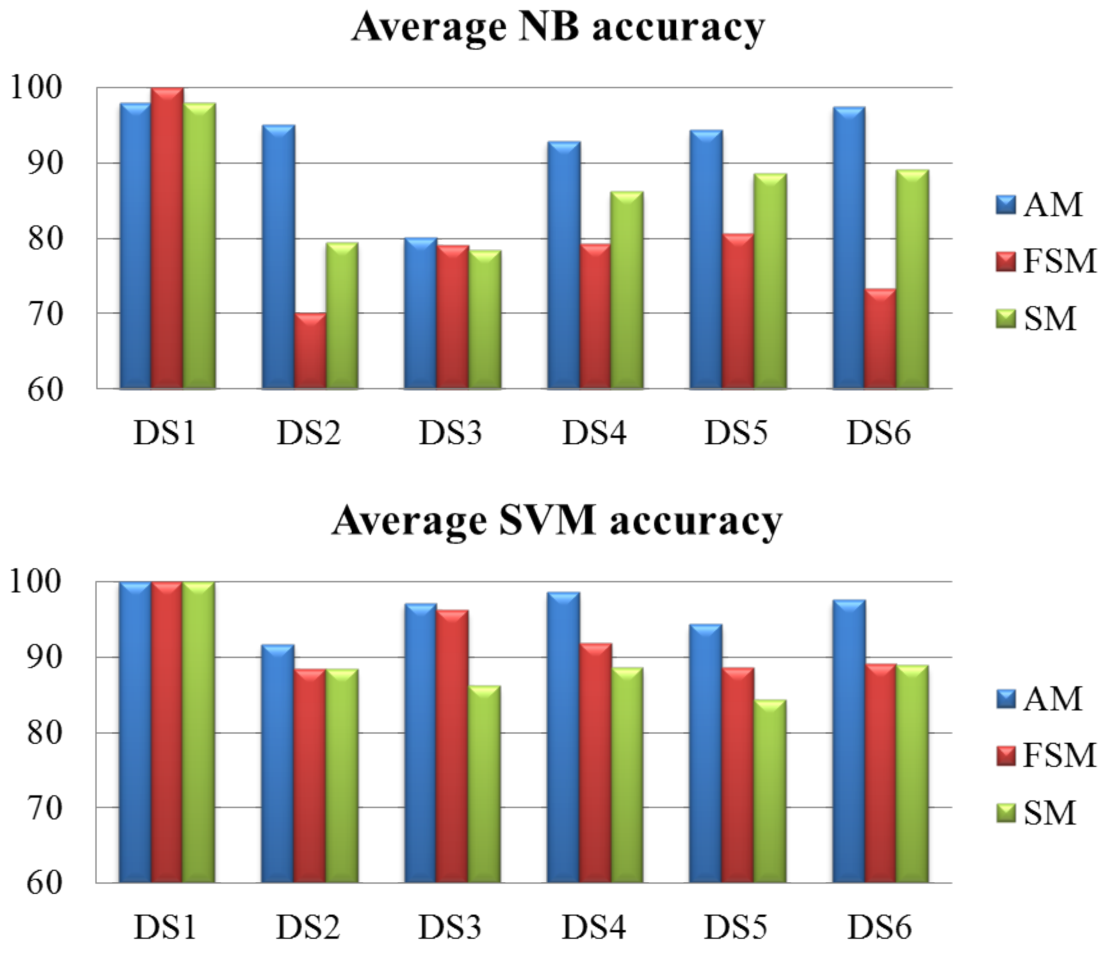


Figure 7.9: Comparison between graph building methods: CA and AA

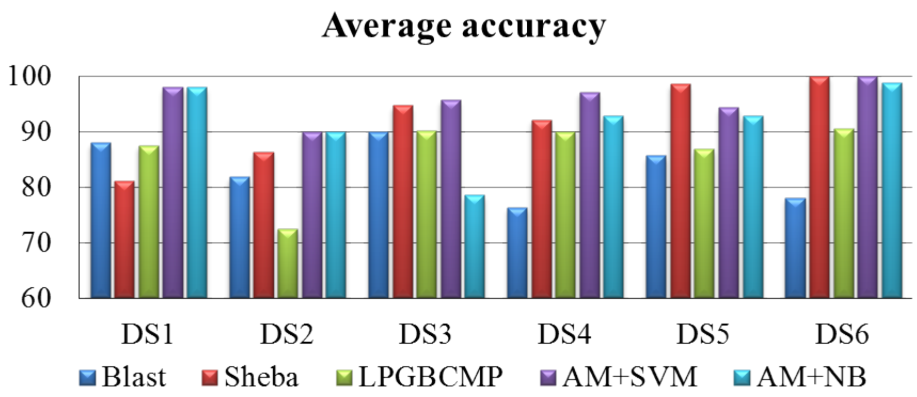


Figure 7.10: Classification accuracy comparison between alignment-based approach, LPGBCMP-based approach and AM-based approach.

also interesting to compare with other approaches that propose sophisticated classifiers.

We carried out the classification on our six datasets based on two types of alignment. One is sequential using Blast [Altschul 1990] and the other is structural using Sheba [Jung 2000]. For each protein in a given dataset, we make an alignment search against all the others. We assign to the query protein the class of the subject protein with the best hit score. In addition, we report the results of a related work [Fei 2010] that dealt with the classification of the same datasets and used the same experimental settings as our work. Authors in that study devoted more efforts to improve the classifier and to optimize its parameters rather than enhancing the quality of features. They proposed a boosting algorithm termed $LPGBCMP$, where base learners, *i.e.*, subgraphs, have structural relationships in the functional space. Their classification system outperformed other related ones [Fei 2010, Jin 2009]. It also outperforms Blast-based classification in most cases (DS3, DS4, DS5, DS6). Sheba, the structural alignment tool, allowed to reach better accuracies compared to Blast and LPGBCMP. But in general, ant-motif-based classification outperforms all the above mentioned approaches.

7.5 Conclusion

In this chapter, we proposed a novel algorithm to extract spatial motifs obeying a certain shape *i.e.*, ant-motifs. The use of already published datasets allowed us to carry out a comparison with several related works. The outcomes of this comparison confirm the reliability of our algorithm and the interestingness of our motifs in classification tasks. By this chapter, we finish with the second axis of this thesis. However, several other extensions are open and under development. We give more details about these ongoing works in the concluding chapter

Chapter 8

Conclusion and Prospects

Contents

Goals

In this chapter, we conclude the thesis by summarizing our contributions and highlighting some prospects. Contributions are briefly surveyed; whereas we give much more details about the ongoing works we are conducting in extension to this thesis.

8.1 Summary of contributions

After our study of the motif-based preprocessing of complex data, we recall the main lines that trace the results of our research.

8.1.1 DDSM method [Saidi 2010b]

We have proposed a novel encoding method for protein sequence classification, termed DDSM, that uses amino-acid substitution matrices to define similarity between motifs during the extraction step. We have demonstrated its efficiency by comparing it with existing methods mainly in terms of accuracy. Thorough experiments have been carried out using several classifiers and known bioinformatics tools namely, Blast and HMMER. The goals of this experiments were mainly to study the impact of the encoding method on the classification performance and to study the effect of the substitution matrix on the output of our method. The outcomes of our comparative experiments confirm the efficiency of our encoding method to represent protein sequences in classification tasks.

8.1.2 Sensibility metrics [Saidi 2010a, Saidi 2012a]

We have introduced the concept of stability to compare motif extraction methods. We call stability of a motif extraction method from a dataset, the non-variability in its set of motifs, when applying a technique of variation on the input dataset. The robustness of a method is the coupling of the non-stability (or sensibility) and the ability to retain or improve the quality of the associated data mining task. In our case, we used the supervised classification accuracy as a quality measure.

The topic of stability with respect to motif extraction methods has not been studied in the literature. However, this aspect was slightly studied in a very close field to the extraction which is the feature selection [Pavel 2007, Yvan 2008, Dunne 2002, Kalousis 2007, Somol 2008, Yu 2008]. The application of the above-cited approaches of stability is not convenient in our case (motif extraction). This originates from the nature of input data used by feature selection methods. In fact, these methods use an original set of features (motifs) as input and try to merely select a subset of relevant features. The perturbation of data is applied to the original set of features. In the case of motif extraction, the input data are still in raw state and the data perturbation step is applied directly to raw data (before extracting motif). The motivation behind exploring the stability of motif extraction methods is to provide evidence that even slight changes in the input data must also be followed by changes in the output results (extracted motifs). These changes must concern the motifs that are no longer significant for the perturbed input data; which means that the motifs that have been conserved must prove to be interesting *i.e.*, help with better classification. Since the set of features are not known a priori we can not apply the measures quoted in the feature selection related works.

In this scope, we have proposed metrics to measure both the sensibility of motif extraction methods and the interestingness of their motifs. The experimental study shows that the DDSM method is more sensible compared to the other methods. This sensibility is usually accompanied by sets of stable interesting motifs. These results are in accordance with those of Chapter 4 that show the contribution of the DDSM method in supervised classification tasks.

8.1.3 Graph representation of proteins [Saidi 2009]

We have made the matching between proteins and graphs and exhibited the various interactions that make some amino acids get close to each other and make the protein fold in its 3D shape. We also have reviewed the most used existing methods of protein graph making, made several criticisms about them and proposed one possible and simple enhancement. We conducted an experimental comparison, based on the largest discriminative feature extraction, between two graph-making methods *i.e.*, CA and AA. We noticed that the extracted features vary in terms of composition, size and quality according to the used method.

We have implemented the cited methods and other ones in java language into a jar file available upon request or on my home page http://fc.isima.fr/~saidi. The program accepts protein PDB files as input and outputs graph files of amino acids and edges between them under several format.

8.1.4 Ant motifs [Saidi 2012b]

We have proposed a novel algorithm to find spatial motifs from protein structures by extending the Karp-Miller-Rosenberg (KMR) repetition finder dedicated to sequences. The extracted motifs, termed *ant-motifs*, obey a well-defined shape which is proposed based on a biological basis. The body of our proposed motifs contains two parts. The first and main part, termed *sequential part*, is composed of contiguous nodes in the primary structure. The second part is the set of spatial edges indicating the nodes connected to the primary structure. This composition gives our motifs an ant-like shape. None of the existing approaches of subgraph mining can extract this kind of motifs.

Experimental results show that ant-motifs offer considerable benefits in protein classification over sequential motifs, frequent-subgraph motifs and alignment-based approaches. The programs and data related to this contribution are freely available at http://fc.isima.fr/~mephu/FILES/AntMotif/.

8.2 Ongoing works and prospects

We are currently working on three major axis. The first axis aims to provide a large-scale pipeline for the functional affiliation of metagenomics data. In the second axis, we are exploiting the substitution idea previously presented in DDSM to summarize large sets of frequent subgraphs to a smaller sets of representative subgraphs. In the third axis, we are working on generalizing the AntMot algorithm to cover the class of *traceable graphs*.

8.2.1 Large-scale many-classes learning

With the continuously increasing amounts of biological data, the need for automated, accurate and rapid classification has become all the more urgent. This need is challenging when the number of classes is large. The inefficiency of the alignment-based solution in many cases has raised the question whether it is possible to benefit from data mining to address that task. In this case, the number of classes is a very considerable constraint. We describe and compare the alignment-based approach (ABA) and the data-mining-based approach (DMBA). Then, we recommend a two-phase approach coupling hidden Markov models (HMM) with standard classifiers and we experimentally compare it with two known alignment tools.

As mentioned during this thesis, DMBA benefits from the panoply of developed classifiers that have shown high efficiency as decision aid tools in several fields such as finance, trade, medicine, etc, due to their strong discrimination and generalization. In general, DMBA requires data in relational format *i.e.*, object-attribute table. Thus, two elements must be provided: a set of reliable attributes to be used as descriptors, and a reliable function of description *e.g.*, frequency, incidence, etc. DMBA faces many problems when dealing with biological data classification. On one side, protein data do not respect the format required by DMBA. On another side, the number of classes has an important impact on any learning task. Indeed, the discrimination ability of any classifier decreases with increasing numbers of classes especially in the case of unbalanced data. This problem has not been deeply investigated [Madani 2008] whereas many efforts have been devoted to address large scale learning in term of number of instances [Madani 2008].

8.2.1.1 Proposed approach

Training For each class we create an HMM profile then we build a binary model using a discriminative classifier. This model is trained on a dataset comprising that class and the other classes' consensuses, after being encoded into relational format using a motif based approach (Chapter 4). This model discriminates between each class and the rest of the training set. A class consensus is a unique sequence which abstracts its class and is generated using the class HMM-profile. This allows us to bypass the unbalanced data and the many-class problems. Henceforth, each class is represented by a probabilistic model (its HMM-profile) and a discriminative model (its binary classifier model).

Prediction Each query sequence is scanned against the HMM-profiles. Hence, some classes are suggested as potential targets sorted by their scores. The number of suggested classes is considerably below the total number of classes. At this level, we use the binary models corresponding to the suggested classes to confirm or refute the HMM results. The final sustained class is the one having the best score and confirmed by the binary model. The combination of the probabilistic and the discriminative aspects preserves an acceptable rapidity while enhancing the sensitivity of the prediction. Furthermore, the memory consumption in our approach is moderate compared to Blat since models are processed separately.

8.2.1.2 Experimental Comparison

To evaluate the above described methods, we utilized four protein datasets taken from the KEGG [Kanehisa 2000] (Table 8.1). The datasets are characterized by a large number of sequences (from 12192 to 44572) and a large number of classes (from 25 to 100). Each class refers to an ortholog (functional) group [Kanehisa 2000] of less than 45% of identity. Experiments were conducted on a PC with a 3 Ghz duo core CPU / 3.25GB RAM. We used the hold-out technique to evaluate the classification approaches *i.e.*, a third is reserved to test and the rest is used for training (for DMBA) or as reference base (for ABA). For our approach we use HMMER [Johnson 2006, Eddy 2008] as HMM tool, N-grams (Chapter 4) as encoding method and SVM as classifier. It is noteworthy that the functional groups in the KEGG base are built using many techniques including alignment. This explains the full accuracy reached by Blast. It is much more accurate than Blat; whereas Blat is much faster. Our approach represents a tradeoff between Blast and Blat *i.e.*, tradeoff between accuracy and speed with the ability to deal with other kinds of classification rather than the functional one *e.g.*, taxonomic, structural.

8.2.2 Substitution for spatial motifs

As previously mentioned in this thesis, 3D protein structures have been recently seen as graphs of amino acids and studied based on graph theory con-

Table 8.1: Comparison with Blast and Blat.

Data	Sequence#	Classe#	Accuracy(%)			Time (mn)		
			Blast	Blat	MLBA	Blast	Blat	MLBA
DS1	12192	25	100	79	88	94	4	3
DS2	24301	50	100	90	92	187	6	8
DS3	33814	75	100	87	90	267	9	13
DS4	44572	100	100	87	90	392	15	18

cepts. Indeed, algorithms of frequent subgraph discovery have been applied on proteins to find motifs that could be interesting in any further analysis. However, when the support threshold is low, the number of frequent subgraphs is expected to be very large which may hinder rather than help (Chapter 7).

In this perspective, we claim that in the set of the generated frequent subgraphs, there exist subgraphs that can substitute several others and hence can summarize the whole set. This claim is based on the same biological facts explored in DDSM, that some amino acids have similar properties and can thus be substituted by each other, without changing the structure or the function of proteins. We are attempting to exploit the substitution idea previously presented in DDSM to summarize the set of discovered frequent subgraphs to a smaller set of representative subgraphs. However, DDSM is dedicated only to sequences and does not contain enough properties to fit frequent subgraphs. Indeed, dealing with sequential motifs consists only on considering the amino acids that compose the motifs. Yet, dealing with spatial motifs represented in the form of subgraphs requires also taking into account the links between amino acids. For this purpose, we extend DDSM with a novel constraint to take into account the shape of the motifs of interest. Specifically, it verifies, without considering the nodes labels, whether two spatial motifs represented as subgraphs are isomorphic. Formally:

Definition 31 *Let $G = (V_G, E_G, L)$ and $G' = (V'_G, E'_G, L)$ be two subgraphs. G (respectively G') collection of nodes V_G (respectively V'_G) and a collection of edges E_G (respectively E'_G). The vertices of V_G (respectively V'_G) are labelled within an alphabet L. G and G' are said to have the same shape, we note shape$(P, P') = true$, iff:*

- *G and G' have the same order, i.e., $|V_G| = |V'_G|$,*
- *G and G' have the same size, i.e., $|E_G| = |E'_G|$,*
- *According to a specific order, $\forall\ 1 \leq i \leq |V_G|$ and $1 \leq j \leq |V_G|$ if $(G[i], G[j]) \in E_P$ then $(G'[i], G'[j]) \in E'_G$.*

Definition 32 (Subgraph substitution) *A subgraph G substitutes a pattern G', we note $subst(G, G', \tau) = true$, iif:*

1. *G and G' have the same shape,*
2. *$S(G, G') \geq \tau$, τ is a user-specified threshold such that $0 \leq \tau \leq 1$ and S is a substitution score similar to that of DDSM.*

A first implementation of this approach termed UnSubPatt was proposed in [Dhifli 2012a, Dhifli 2012b] and the results seem to be promising both in terms of the number of selected motifs and their interestingness. Fig. 8.1 and Fig. 8.2 show the first outcomes of this approach using four datasets (DS1, DS2, DS3 and DS4).

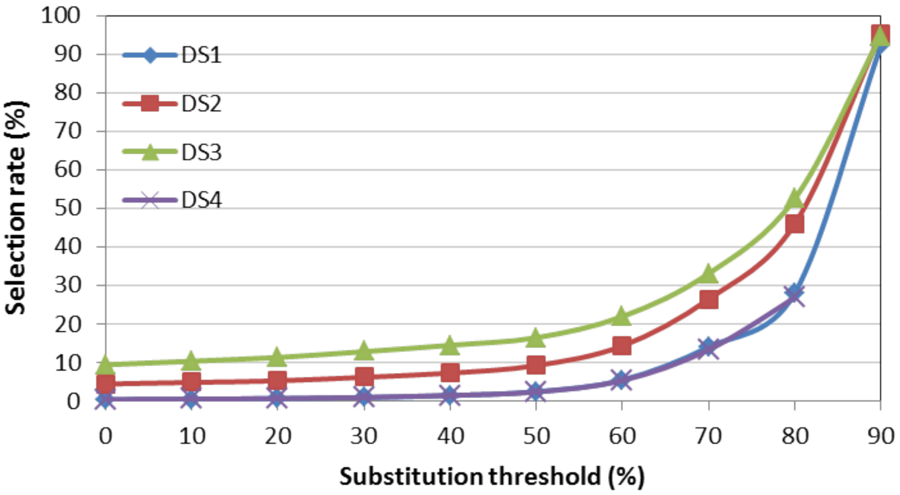


Figure 8.1: Rate of the selected motifs from the initial set depending on the substitution threshold.

8.2.3 Traceable graphs

There exists a specific type of graphs, called *traceable graph*, possessing a path between two nodes that visits each node in the graph exactly once. This path is called *Hamiltonian path*, or also *Hamilton path*. Seen as graphs of amino acids, proteins are known to contain a remarkable Hamiltonian path which is its primary structure. Many algorithms for frequent subgraph discovery have been proposed [Krishna 2011]. But none of them has taken into account the Hamiltonian path information to reduce the number of frequent subgraphs in the case of traceable graphs. Our ongoing work lies within this scope.

The Hamiltonian path forms a sort of backbone that maintains all the graphs nodes linked. That is why we consider it as a *concise description* of

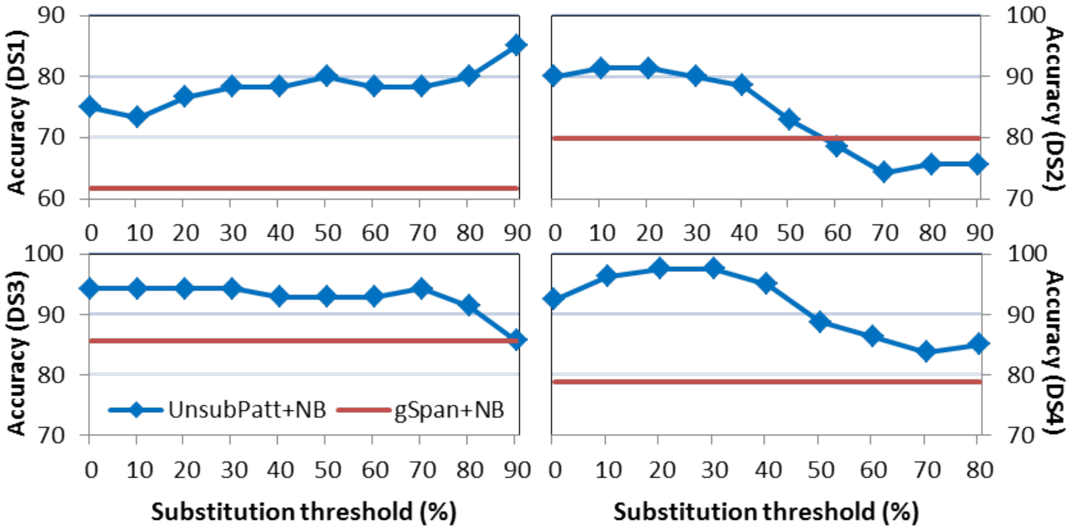


Figure 8.2: Comparison of the classification accuracies between UnsubPatt-motifs and gSpan-motifs using naïve bayes.

a traceable graph. Hence, we claim that if a given graph is traceable, then it would be easier to process it based on its Hamiltonian path. For our purpose, we call a *Hamiltonian edge*, noted *H-edge*, every edge belonging to the selected Hamiltonian path. Otherwise, we term it *non-Hamiltonian edge*, noted *NH-edge* (see Table 8.2 for correspondence with terms in Chapter 7). We can obviously notice that the Hamiltonian path of a traceable graph is the source and the target of every NH-edge. In other words, every NH-edge links two nodes that are not contiguous in the Hamiltonian path. Our purpose is to make the AntMot algorithm generic in order to cover any traceable graph rather than proteins. However, a traceable graph may contain more than one hamiltonian path. Hence, the choice of the "best" one is an issue that must be addressed.

Table 8.2: Correspondence with terms in Chapter 7.

Term (From Chapter 7)	**Traceable graph term**
Primary structure	Hamiltonian path
Sequential link	Hamiltonian edge
Spatial link	Non-Hamiltonian edge
Ant-motif	Hamiltonian motif

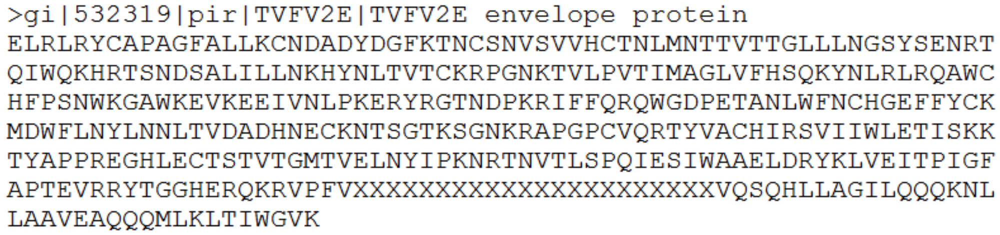
Appendix A

Used Bioinformatics Data Formats and Tools

A.1 Data formats

A.1.1 FASTA format

FASTA is a textual format used to represent biological sequences as a text file. This format is very commonly used in bioinformatics. In FASTA format, each nucleotide or amino acid sequence of the biological is represented by a character. A biological sequence is represented by a string of characters representing the successive nucleotides or amino acids of the sequence (Fig. A.1).

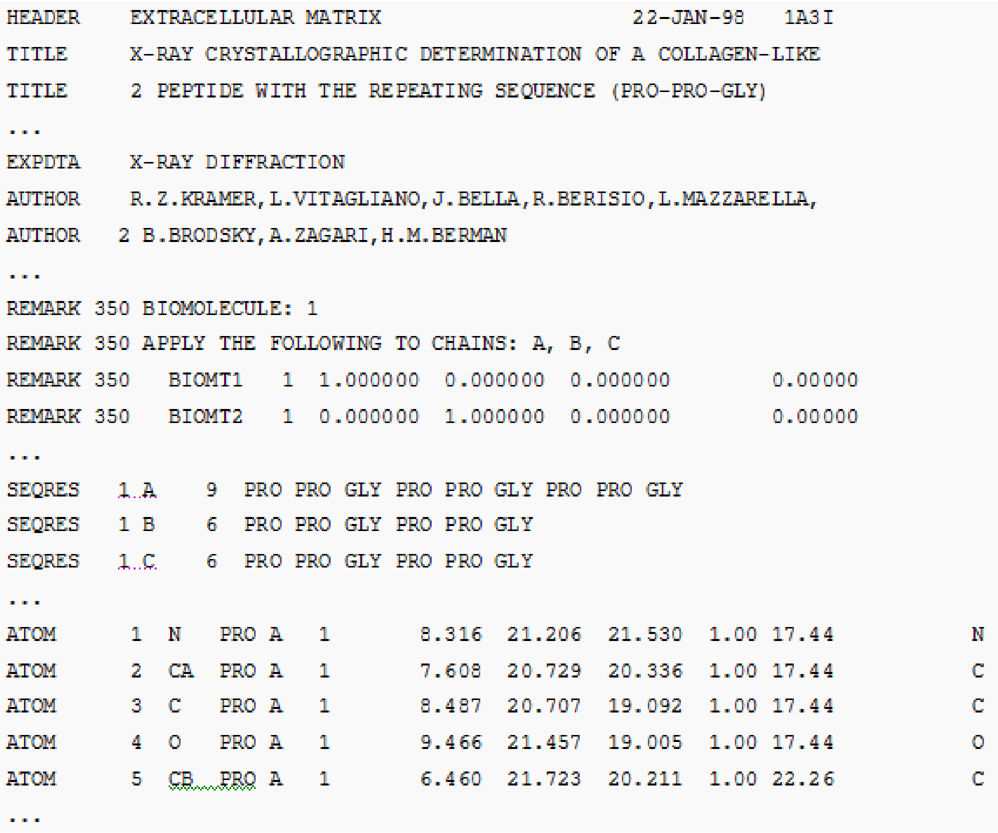


```
>gi|532319|pir|TVFV2E|TVFV2E envelope protein
ELRLRYCAPAGFALLKCNDADYDGFKTNCSNVSVVHCTNLMNTTVTTGLLLNGSYSENRT
QIWQKHRTSNDSALILLNKHYNLTVTCKRPGNKTVLPVTIMAGLVFHSQKYNLRLRQAWC
HFPSNWKGAWKEVKEEIVNLPKERYRGTNDPKRIFFQRQWGDPETANLWFNCHGEFFYCK
MDWFLNYLNNLTVDADHNECKNTSGTKSGNKRAPGPCVQRTYVACHIRSVIIWLETISKK
TYAPPREGHLECTSTVTGMTVELNYIPKNRTNVTLSPQIESIWAAELDRYKLVEITPIGF
APTEVRRYTGGHERQKRVPFVXXXXXXXXXXXXXXXXXXXXXXVQSQHLLAGILQQQKNL
LAAVEAQQQMLKLTIWGVK
```

Figure A.1: FASTA format. The first line of a FASTA file describes the sequence. This line begins with a ">" and the rest of the description (the original database of the sequence, the sequence identifier in the database, a description) must be adjoined to the sign ">". The other lines consist of characters representing the nucleotides or amino acids of the sequence. The lines representing the sequence have a maximum size. The maximum limit is 120 characters per line, but for historical reasons, the maximum length of the line is generally 80 characters.

A.1.2 PDB format

PDB is a textual format describing the position of atoms in a molecule in a three-dimensional space. To reduce file size, the hydrogen atoms are missing from the description files of macromolecules. Even for small molecules, the double bonds are rarely present. A typical PDB file describing a protein consists of hundreds to thousands of lines like the example in Fig. A.2.



```
HEADER    EXTRACELLULAR MATRIX                    22-JAN-98   1A3I
TITLE     X-RAY CRYSTALLOGRAPHIC DETERMINATION OF A COLLAGEN-LIKE
TITLE     2 PEPTIDE WITH THE REPEATING SEQUENCE (PRO-PRO-GLY)
...
EXPDTA    X-RAY DIFFRACTION
AUTHOR    R.Z.KRAMER,L.VITAGLIANO,J.BELLA,R.BERISIO,L.MAZZARELLA,
AUTHOR   2 B.BRODSKY,A.ZAGARI,H.M.BERMAN
...
REMARK 350 BIOMOLECULE: 1
REMARK 350 APPLY THE FOLLOWING TO CHAINS: A, B, C
REMARK 350   BIOMT1   1  1.000000  0.000000  0.000000        0.00000
REMARK 350   BIOMT2   1  0.000000  1.000000  0.000000        0.00000
...
SEQRES   1 A    9  PRO PRO GLY PRO PRO GLY PRO PRO GLY
SEQRES   1 B    6  PRO PRO GLY PRO PRO GLY
SEQRES   1 C    6  PRO PRO GLY PRO PRO GLY
...
ATOM      1  N   PRO A   1       8.316  21.206  21.530  1.00 17.44           N
ATOM      2  CA  PRO A   1       7.608  20.729  20.336  1.00 17.44           C
ATOM      3  C   PRO A   1       8.487  20.707  19.092  1.00 17.44           C
ATOM      4  O   PRO A   1       9.466  21.457  19.005  1.00 17.44           O
ATOM      5  CB  PRO A   1       6.460  21.723  20.211  1.00 22.26           C
...
```

Figure A.2: PDB format.The file describes the coordinates of the atoms that are part of the protein. For example, the first ATOM line above describes the alpha-N atom of the first residue of peptide chain A, which is a proline residue; the first three floating point numbers are its x, y and z coordinates and are in units of Ångströms. The next three columns are the occupancy, temperature factor, and the element name, respectively.

A.2 Bioinformatics tools for classification

A.2.1 Sequential alignment: BLAST

The interestingness of the algorithm is that its design is based on a statistical model established by Karlin and Altschul [Altschul 1990]. The basic unit of BLAST is the HSP (High-scoring Segment Pair). It is a pair of fragments identified on each of the compared sequences, of equal length but not predefined, and which has a significant score. In other words, a HSP corresponds to a common segment, as long as possible, between two sequences having a score greater than or equal to a threshold score.

The BLAST algorithm performs in four steps illustrated by Fig. A.3:

1. The query sequence is cut into words of fixed size w (default: $w = 3$ for

proteins, and $w = 3$ for nucleic data). For each of these words, a list of similar words is created, using a substitution matrix.

2. Each word from the list of similar words is searched for similarity (a hit) against all sequences of the database. A hit is also defined by a score value that must be higher than a fixed value.

3. The similarity is extended starting from the common word, in both directions along the matching sequence. The extension will be finished when either: the cumulated score decreases of a fixed amount compared to the maximum value previously reached, the cumulated score becomes equal to 0, or the extremity of one of the two sequences is reached. Finally, the longest fragment found is called a High Scoring Pairs (HSP).

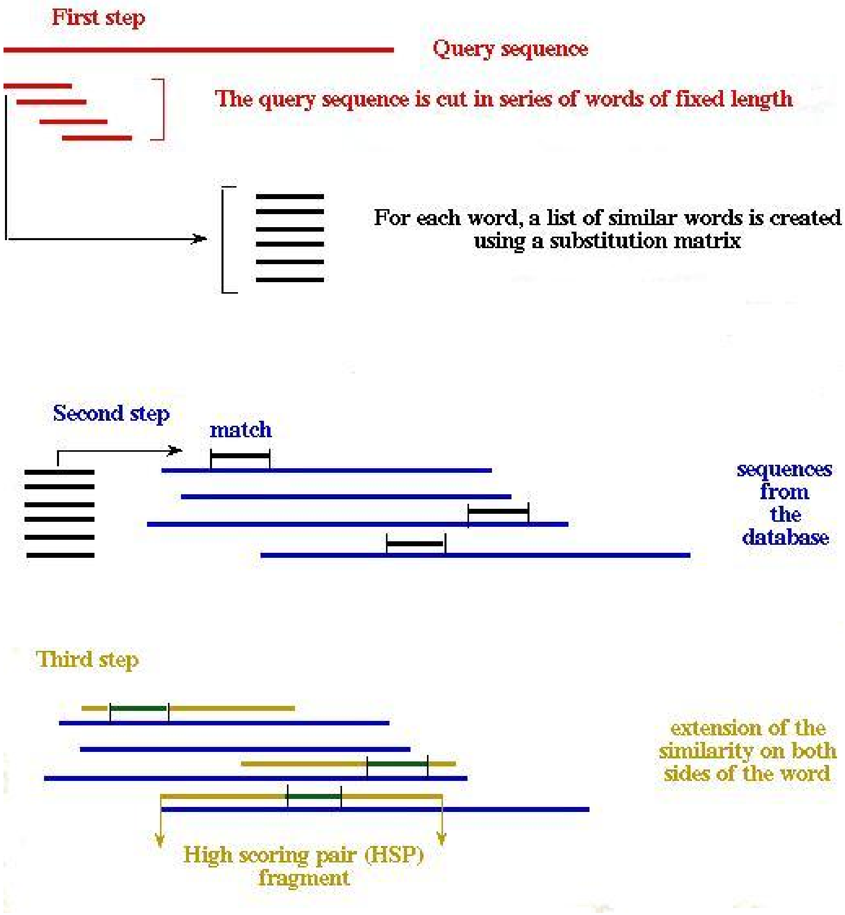

First step
Query sequence
The query sequence is cut in series of words of fixed length
For each word, a list of similar words is created using a substitution matrix
Second step
match
sequences from the database
Third step
extension of the similarity on both sides of the word
High scoring pair (HSP) fragment

Figure A.3: Steps of BLAST alignment.

A.2.2 Spatial alignment: Sheba

SHEBA [Jung 2000] is a protein structure alignment procedure. The initial alignment is made by comparing a one-dimensional list of primary, secondary and tertiary structural profiles of two proteins, without explicitly considering the geometry of the structures. The alignment is then iteratively refined in the second step, in which new alignments are found by three-dimensional superposition of the structures based on the current alignment. SHEBA can do pair-wise (one-to-one) alignment or multiple (one-to-many) alignment. It also has several different output options:

- For pair-wise alignment: alignment statistics, corresponding sequence alignments, formatted column output, a list of aligned residue numbers, the transformation matrix, and the transformed coordinates in PDB-like format.
- For multiple alignment: corresponding sequence alignments and multiple alignment statistics.

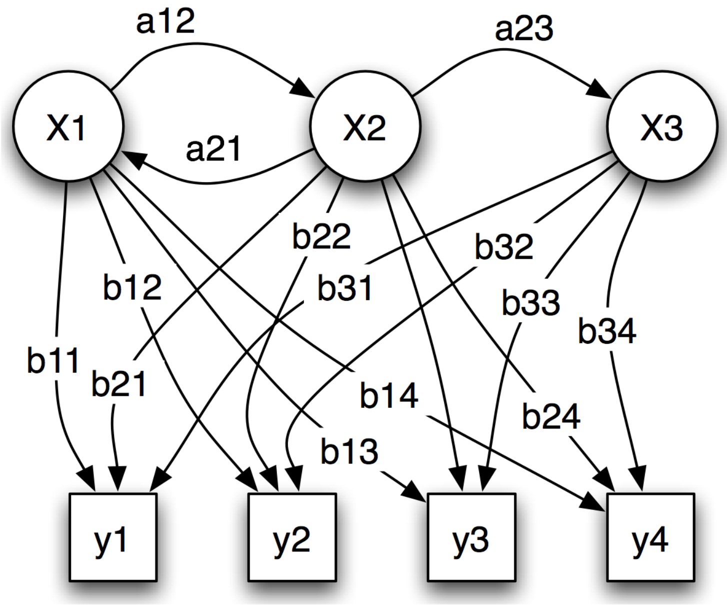


Figure A.4: Example of a hidden Markov model. X1,X2 and X3 present the hidden states of the HMM profile and y1, y2, y3 and y4 present the symbols of alphabet that can be emitted from the hidden states. The labelled arrows a12, a21 and a23 present the evolution probabilities of the system from one hidden state to another where for instance X2 may evolve to X3 or X1 respectively with probability values of a23 and a21. However, the labelled arrows b11, b12, ..., b33 and b34 are the emission probabilities, for instance, the state X1 may emit the observation symbol y1 with a probability value of b11 and so on.

A.2.3 Hidden Markov models

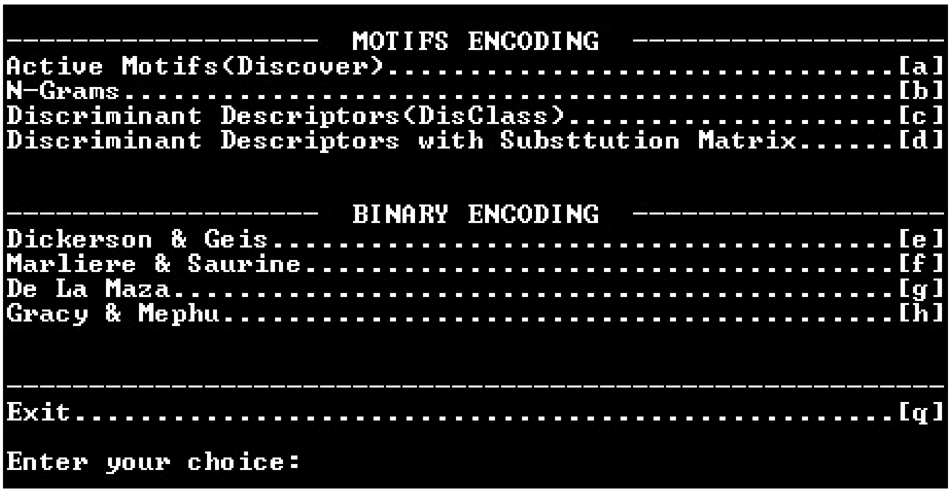
Hidden Markov Models or HMM [Johnson 2006] are generative models of sequences defined by a set of states, a discrete alphabet of symbols, a matrix of transition probabilities between states and a matrix of emissions probability of each symbol of the alphabet from each state (Fig.A.4. The system randomly evolves from one state to another according to the transition probabilities, emitting symbols of the alphabet. These models are mainly used to address three issues namely: the evaluation of the probability of emitting a given sequence of observations, finding the most probable path that generated a sequence of observations, and creating and calibrating models also called HMM profiles.

Appendix B

Relational and Binary Coding Methods Library for Biological Sequences

B.1 Description

This library comprises methods to re-encode biological sequences (DNA and protein) into relational or binary formats. Methods have been developed in C language and can be called by the following interface:



```
-------------------  MOTIFS ENCODING  -------------------
Active Motifs(Discover)...............................[a]
N-Grams...............................................[b]
Discriminant Descriptors(DisClass)....................[c]
Discriminant Descriptors with Substtution Matrix......[d]


-------------------  BINARY ENCODING  -------------------
Dickerson & Geis......................................[e]
Marliere & Saurine....................................[f]
De La Maza............................................[g]
Gracy & Mephu.........................................[h]


---------------------------------------------------------
Exit..................................................[q]

Enter your choice:
```

Figure B.1: Main interface.

The menu consists of two sections:

B.1.1 Motifs based encoding methods

- Active Motifs

- N-Grams

- Discriminant Descriptors

- Discriminant Descriptors with Substitution Matrices

The generated files by these methods are under relational format, namely ARFF format (Attribute Relation File Format) used by the workbench Weka.

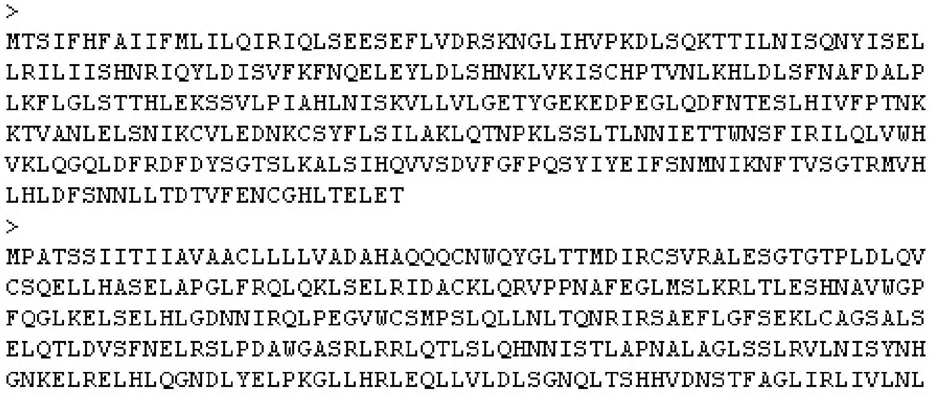
B.1.2 Binary encoding methods

- Dickerson & Geis
- Marliere & Saurine
- De La Maza
- Gracy & Mephu

B.2 How to use

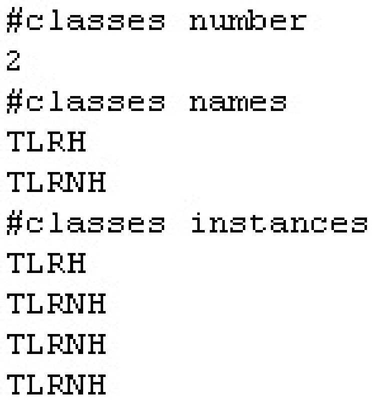
- Two files are needed *i.e.*, SeqCod.exe and DLL_SeqCod.dll, to run the application
- Sequence file(s) in fasta format
- Classification file(s) describing the sequences file (for methods based on motifs)
- Select a method to apply

B.2.1 Fasta format



```
>
MTSIFHFAIIFMLILQIRIQLSEESEFLVDRSKNGLIHVPKDLSQKTTILNISQNYISEL
LRILIISHNRIQYLDISVFKFNQELEYLDLSHNKLVKISCHPTVNLKHLDLSFNAFDALP
LKFLGLSTTHLEKSSVLPIAHLNISKVLLVLGETYGEKEDPEGLQDFNTESLHIVFPTNK
KTVANLELSNIKCVLEDNKCSYFLSILAKLQTNPKLSSLTLNNIETTWNSFIRILQLVWH
VKLQGQLDFRDFDYSGTSLKALSIHQVVSDVFGFPQSYIYEIFSNMNIKNFTVSGTRMVH
LHLDFSNNLLTDTVFENCGHLTELET
>
MPATSSIITIIAVAACLLLLVADAHAQQQCNWQYGLTTMDIRCSVRALESGTGTPLDLQV
CSQELLHASELAPGLFRQLQKLSELRIDACKLQRVPPNAFEGLMSLKRLTLESHNAVWGP
FQGLKELSELHLGDNNIRQLPEGVWCSMPSLQLLNLTQNRIRSAEFLGFSEKLCAGSALS
ELQTLDVSFNELRSLPDAWGASRLRRLQTLSLQHNNISTLAPNALAGLSSLRVLNISYNH
GNKELRELHLQGNDLYELPKGLLHRLEQLLVLDLSGNQLTSHHVDNSTFAGLIRLIVLNL
```

Figure B.2: Fasta format. Each sequence starts by >.



```
#classes number
2
#classes names
TLRH
TLRNH
#classes instances
TLRH
TLRNH
TLRNH
TLRNH
```

Figure B.3: Classification file.

B.2.2 Classification file

The classification file above describes a fasta file containing 4 biological sequences belonging to 2 classes: the 1st belongs to TLRH class and the 3 others belong to TLRNH. To make the generation of such file easier, an application has been developed: ClassFileGen.exe http://fc.isima.fr/~saidi.

B.2.3 Motif encoding methods

B.2.3.1 Common parameters

- Enter the FASTA file name (do not forget file extension, *e.g.*: seq_file.txt),

- Enter the classification file name (do not forget file extension, *e.g.*: seq_file_class.txt),

- If there exists a test file then enter its name and the name of its classification file,

- When all parameters are set, enter the name of the output file (do not forget file extension, *e.g.*: out_file.arff).

B.2.3.2 Active Motifs

- Select motif shape: *X* (for simple motifs, *e.g.*: RSMT) or *X*Y* (for compound motifs: with gap, e.g. RSMT*VFF),

- Set the minimum length of motifs,

- Set the minimum occurrence number of motifs,

- Set the number of allowed mutations (*e.g.*, if number of allowed mutations = 1 then the motifs RSMT and RSVT are considered the same).

B.2.3.3 N-Grams

- Enter the length of motifs (it is a fixed length: 3 by default: 3-grams).

B.2.3.4 Discriminative Descriptors

- Set alpha threshold of motifs: minimum occurrence rate of motifs within a defined sequence family F (*e.g.*: 0.9),
- Set beta threshold of motifs: maximum occurrence rate of motifs within all sequence families excluding F, *i.e.*, other families than the family F (*e.g.*: 0.08),
- *E.g.*, RSMT is a considered as motif of a family F iff it occurs in at least 90% of the sequences of F and at most 8% of the database sequences excluding F, *i.e.*, other families than the family F.

B.2.3.5 Discriminative Descriptors with Substitution Matrices

- Set alpha and beta thresholds (as in Discriminant Descriptors section),
- Select the substitution matrix number (*e.g.*: 2 for Blosum62),
- Set the similarity score threshold (or substitution probability): We consider that a motif X substitutes a motif Y if their substitution probability is higher than a given threshold.

B.2.4 Binary encoding methods

- Enter the FASTA file name (do not forget file extension, *e.g.*: seq_file.txt),
- Enter the name of the output file.

Appendix C

Karp-Rosenberg-Miller Algorithm

The algorithm of Karp, Miller and Rosenberg (KMR) is a method to detect repetitions in a data structure (strings, tables). Richard Karp, Raymond Miller and Arnold Rosenberg proposed it in 1972 [Karp 1972]. The original version of the KMR algorithm is dedicated to one string and its complexity is almost linear in the size of the structure as input. The original version has been the kernel of other algorithms [Pisanti 2005] and it has been adapted in a parallelized version [Crochemore 1991]. The KMR algorithm is based on the following notion of equivalence:

Definition 33 *Two positions i and j in a string S of length m are k-equivalent, we note $i\ E_k\ j$, if and only if the two substrings of length k $S[i, i+k-1]$ and $S[j, j+k-1]$ are identical [Karp 1972].*

We also say that the positions i and j belong to the same equivalence class in the level k. An equivalence relation E_k (or a level) k $1 \leq k \leq m$, can be represented by a vector $V_k[1..m-k+1]$, where each component $V[i]$ of this vector, $1 \leq i \leq m-k+1$, represents the number of the equivalence class to which position i belongs to the equivalence relation E_k. Figure C.1 illustrates a case of 2-equivalence between positions $i = 5$ and $j = 11$. We note that it is a repeated substring "NV" of length $k = 2$ identified in positions $i = 5$ and $j = 11$. This repeated substring is one of equivalence class of the relation E_2 (or level 2).

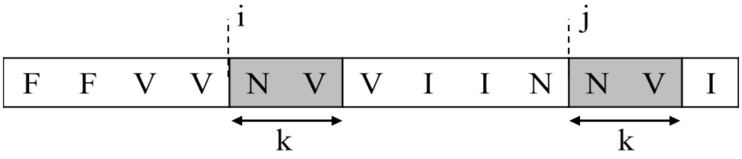


Figure C.1: Illustration of 2-equivalence between positions i=5 and j=11

Given this, KMR provides a characterization of $E_{k+k'}$ in terms of E_k and $E_{k'}$. So that it constructs inductively larger sets E_L by setting $k' = 1$ or $k' = k$. That is, it increases the length of the substrings by concatenating two substrings from the previous iteration using the following lemma:

Lemma 4 ([Karp 1972])

$$i\ E_k\ j\ \&\ (i+k)\ E_{k'}\ (j+k)\ \Leftrightarrow i\ E_{k+k'}\ j \tag{C.1}$$

Appendix D

Protein Graph Repository

D.1 Description

Protein Graph Repository (PGR) is an online repository mainly dedicated to protein graphs. The core of this online repository is developed using both JAVA and PHP as a programing languages and MySQL as a database management system. In addition, PGR was deployed using the latest web technologies and respecting the web standardization specifications.

D.2 How to use

The general operation schema is as follow :

D.2.1 Parser

This tool allows the transformation of PDB protein files [Berman 2000] into graphs. Many graph formats could be generated enabling the use of panoply of existant tools namely Biolyout [Theocharidis 2009], Network Workbench [Börner 2010], GraphClust [Recupero 2008] . . .
Several methods of graph construction are supported. The use of the parser is very simple:

- The user upload his list of PDB files
- Specify : the graph construction method, the appropriate parameters values, and the output format
- Run the parser

A more detailed description is reported in the site.

D.2.2 Repository

The repository represents a protein graph data bank easily reached online for PGR users. This repository is coupled with a selection tool allowing the filtering and targeting of a specific population of protein graphs. The repository is fed each time the parser is run. A download option is enabled making the existent protein graphs available for any further purpose.

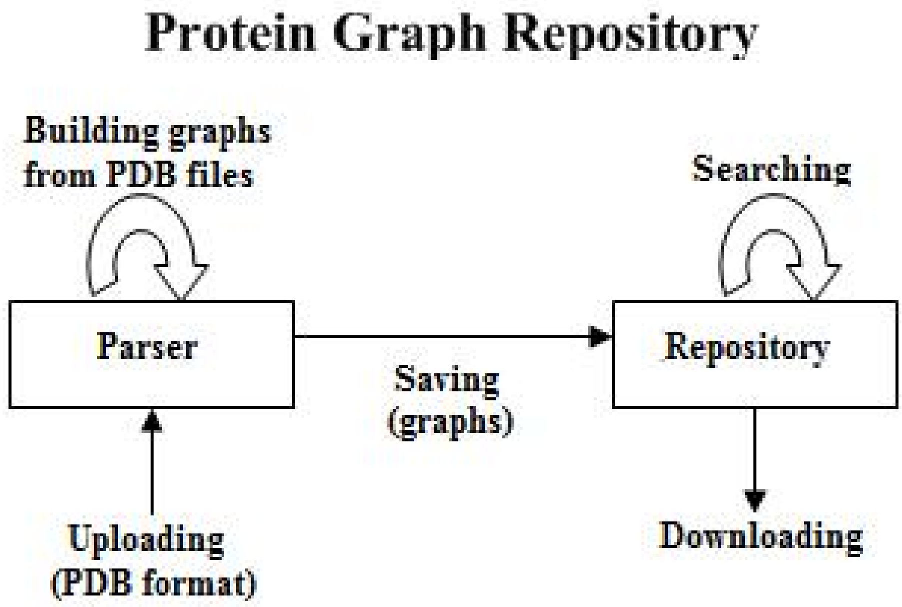
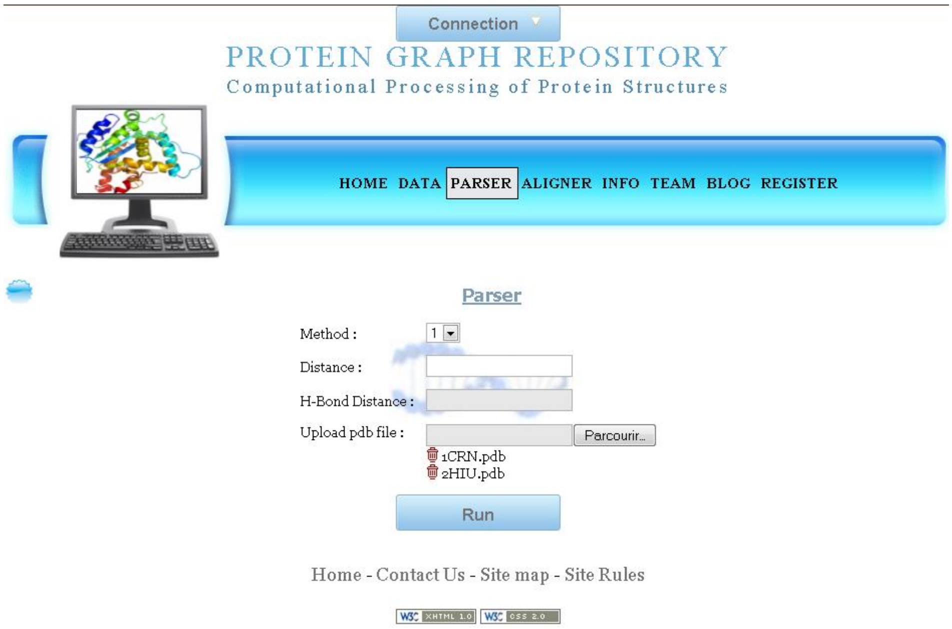


Figure D.1: PGR general schema.



PROTEIN GRAPH REPOSITORY

Computational Processing of Protein Structures

Parser

Method : 1

Distance :

H-Bond Distance :

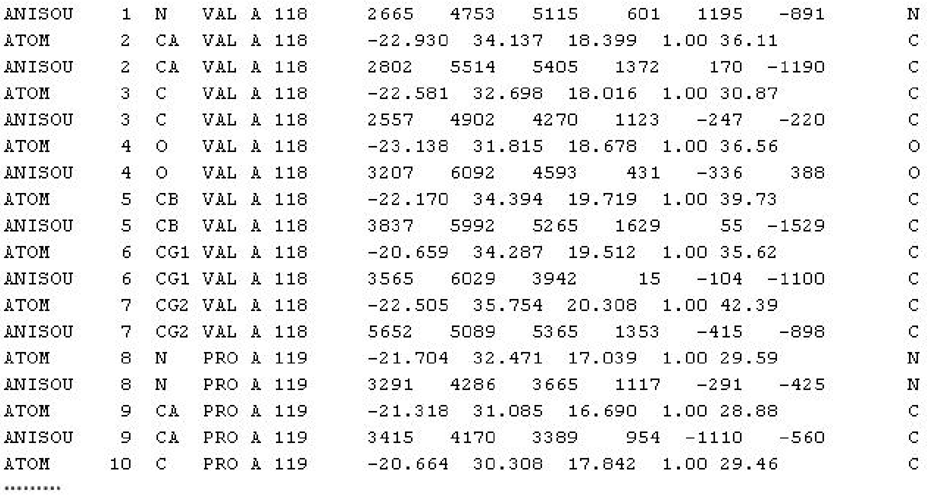
Upload pdb file :

1CRN.pdb

2HIU.pdb

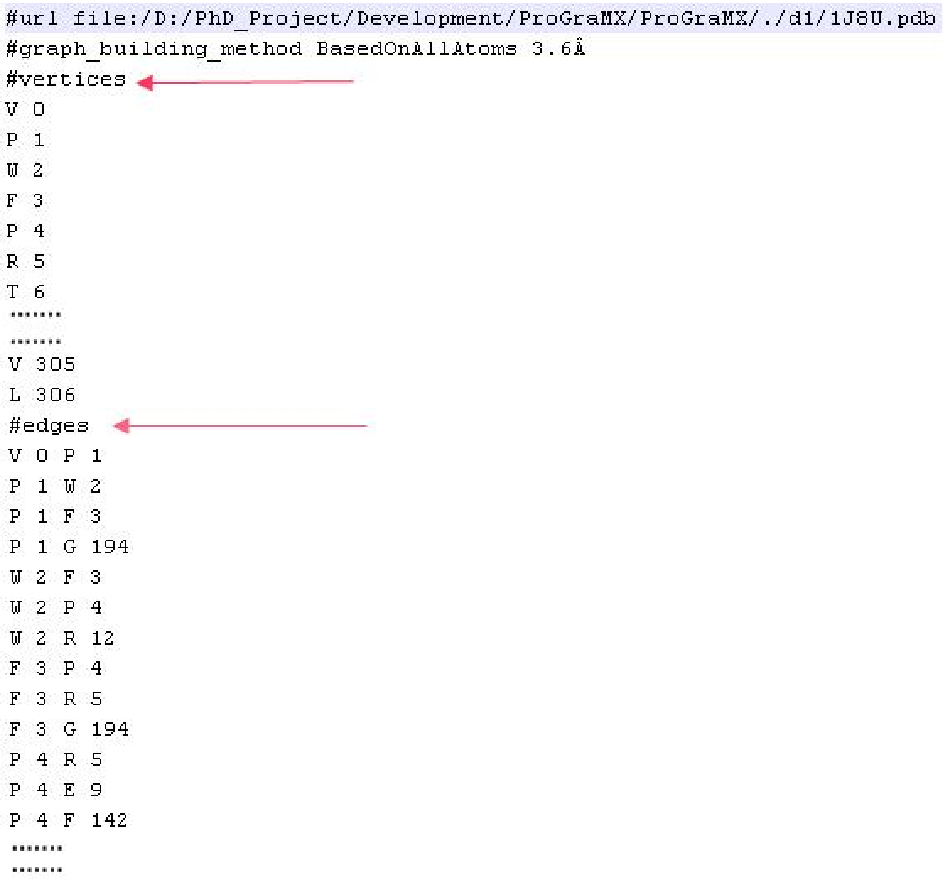
Home - Contact Us - Site map - Site Rules

Figure D.2: Parser.



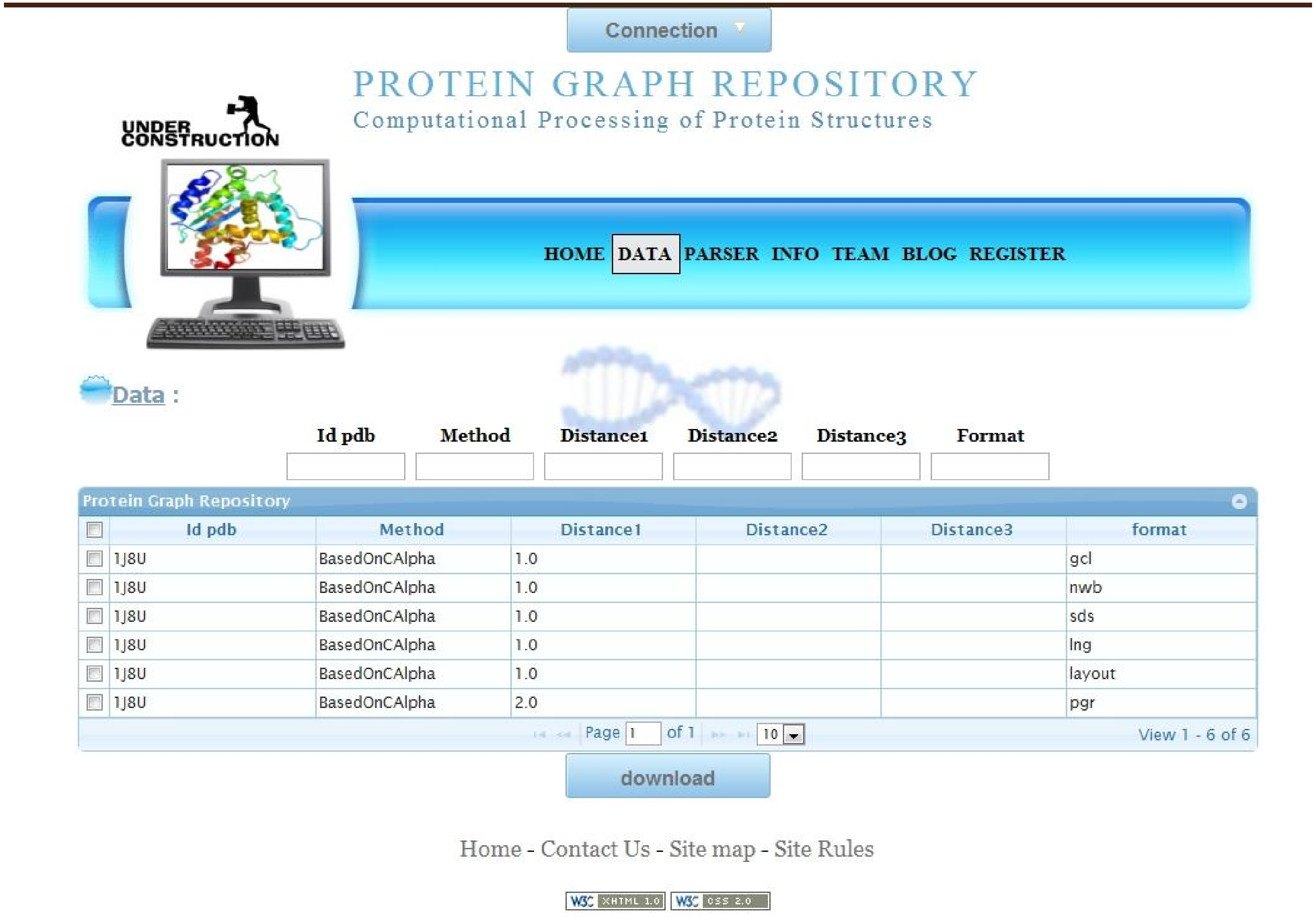
```
ANISOU    1  N   VAL A 118     2665   4753   5115    601   1195   -891       N
ATOM      2  CA  VAL A 118     -22.930  34.137  18.399  1.00 36.11           C
ANISOU    2  CA  VAL A 118     2802   5514   5405   1372    170  -1190       C
ATOM      3  C   VAL A 118     -22.581  32.698  18.016  1.00 30.87           C
ANISOU    3  C   VAL A 118     2557   4902   4270   1123   -247   -220       C
ATOM      4  O   VAL A 118     -23.138  31.815  18.678  1.00 36.56           O
ANISOU    4  O   VAL A 118     3207   6092   4593    431   -336    388       O
ATOM      5  CB  VAL A 118     -22.170  34.394  19.719  1.00 39.73           C
ANISOU    5  CB  VAL A 118     3837   5992   5265   1629     55  -1529       C
ATOM      6  CG1 VAL A 118     -20.659  34.287  19.512  1.00 35.62           C
ANISOU    6  CG1 VAL A 118     3565   6029   3942     15   -104  -1100       C
ATOM      7  CG2 VAL A 118     -22.505  35.754  20.308  1.00 42.39           C
ANISOU    7  CG2 VAL A 118     5652   5089   5365   1353   -415   -898       C
ATOM      8  N   PRO A 119     -21.704  32.471  17.039  1.00 29.59           N
ANISOU    8  N   PRO A 119     3291   4286   3665   1117   -291   -425       N
ATOM      9  CA  PRO A 119     -21.318  31.085  16.690  1.00 28.88           C
ANISOU    9  CA  PRO A 119     3415   4170   3389    954  -1110   -560       C
ATOM     10  C   PRO A 119     -20.664  30.308  17.842  1.00 29.46           C
.........
```

Figure D.3: PDB file.



```
#url file:/D:/PhD_Project/Development/ProGraMX/ProGraMX/./d1/1J8U.pdb
#graph_building_method BasedOnAllAtoms 3.6Å
#vertices
V 0
P 1
W 2
F 3
P 4
R 5
T 6
.......
.......
V 305
L 306
#edges
V 0 P 1
P 1 W 2
P 1 F 3
P 1 G 194
W 2 F 3
W 2 P 4
W 2 R 12
F 3 P 4
F 3 R 5
F 3 G 194
P 4 R 5
P 4 E 9
P 4 F 142
.......
.......
```

Figure D.4: PGR file.



PROTEIN GRAPH REPOSITORY

Computational Processing of Protein Structures

Data :

Id pdb	Method	Distance1	Distance2	Distance3	Format

Protein Graph Repository

□	Id pdb	Method	Distance1	Distance2	Distance3	format
□	1J8U	BasedOnCAlpha	1.0			gcl
□	1J8U	BasedOnCAlpha	1.0			nwb
□	1J8U	BasedOnCAlpha	1.0			sds
□	1J8U	BasedOnCAlpha	1.0			lng
□	1J8U	BasedOnCAlpha	1.0			layout
□	1J8U	BasedOnCAlpha	2.0			pgr

Home - Contact Us - Site map - Site Rules

Figure D.5: Data

Appendix E

AntMot Read Me

E.1 Command line

```
java −heap −jar antmot.jar names pdb_path graph_build dist
min_size max_sizemin_intra_freq max_extra_freq minimality
 arff_path motif_path
```

E.2 Example

```
java −Xmx1024m −jar antmot.jar test.txt ./pdb 2 7 3 7 0.5 0 false
    test.arff motifs_test.txt
```

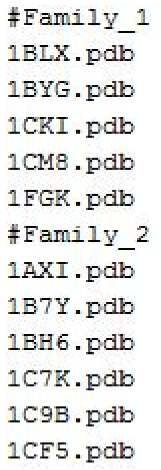
E.3 Paramaters

E.3.1 Input

heap: java heap memory for example Xmx1024m
names: file containing pdb names in the format illustrated by figure 1
pdb_path: folder where the pdb files are saved
graph_build: method of graph building, values=1 , 2, 1 for AllAtoms method and 2 for CarbonAlpha method.
dist: distance used by the method of graph building (graph_build)
min_size: minimum number of vertices in motifs
max_size: maximum number of vertices in motifs
min_intra_freq: minimum frequency of a motif within a given class
max_extra_freq: maximum frequency of a motif in an outer class
minimality: boolean parameter to stop building the motif if it satisfies min_intra_freq and max_extra_freq. This parameter is not yet implemented in our program, so the current and default value is false.

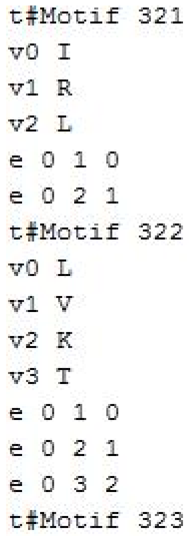
E.3.2 Output

arff_path: arff file used to perform classification with Weka workbench.
motif_path: file where motifs are saved (figure 2)



```
#Family_1
1BLX.pdb
1BYG.pdb
1CKI.pdb
1CM8.pdb
1FGK.pdb
#Family_2
1AXI.pdb
1B7Y.pdb
1BH6.pdb
1C7K.pdb
1C9B.pdb
1CF5.pdb
```

Figure E.1: File containing the names of concerned pdb files belonging to two families.



```
t#Motif 321
v0 I
v1 R
v2 L
e 0 1 0
e 0 2 1
t#Motif 322
v0 L
v1 V
v2 K
v3 T
e 0 1 0
e 0 2 1
e 0 3 2
t#Motif 323
```

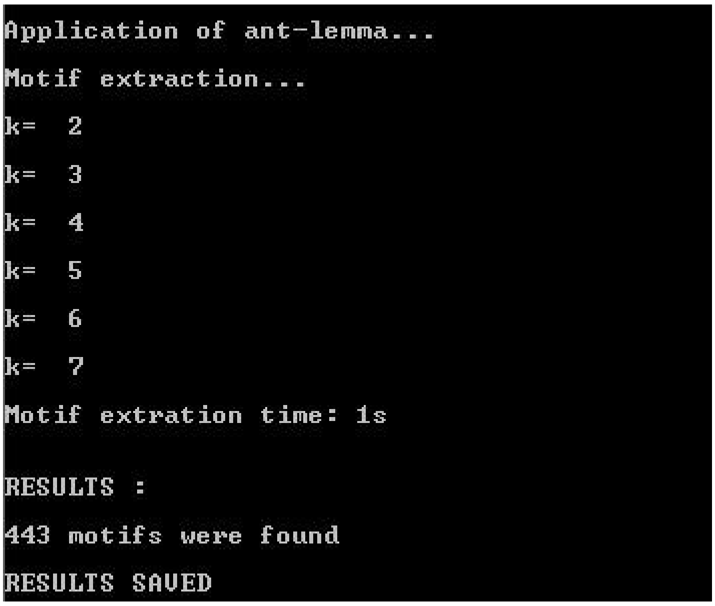
Figure E.2: Sample of an output file containing ant motifs.

E.4 Redundancy and runtime

The time needed to extract spatial motifs is strongly proportional with their sizes. If the proteins are very similar (for example they share identical chains), the runtime increases considerably.

It is well advised to find spatial motifs in distinct chains. That is why works dealing with finding spatial motifs from proteins process **one**-chain-protein structures or at worst protein structures that do not contain redundant chains. In order to check this redundancy, one can make a blast alignment. If there exist very similar chains then the runtime is expected to be long.

The redundancy can also be detected more easily using our program, by affecting the value "0" to the parameter **dist**. That means that the program will only consider the primary structures. The program will terminate **very rapidly** and show the size of extracted motifs (the final value of **k**, see Fig. E.4). A high final value of **k** indicates that there exist very similar chains.



```
Application of ant-lemma...
Motif extraction...
k=  2
k=  3
k=  4
k=  5
k=  6
k=  7
Motif extration time: 1s

RESULTS :
443 motifs were found
RESULTS SAVED
```

Figure E.3: Screenshot of the program running.

E.5 Memory and result recovery

The program manages the allocated memory and if it is not enough it will indicate it and recover the lately extracted motifs since the start till the memory-lack termination.

Bibliography

[Agrawal 1995] Rakesh Agrawal and Ramakrishnan Srikant. *Mining Sequential Patterns.* In Philip S. Yu and Arbee L. P. Chen, editeurs, ICDE, pages 3–14. IEEE Computer Society, 1995. (Cited on page 44.)

[Altschul 1990] S. Altschul, W. Gish, W. Miller, E. Myers and D. Lipman. *Basic local alignment search tool.* Journal of Molecular Biology, vol. 215, pages 403–410, 1990. (Cited on pages 33, 44, 51, 53, 58, 114 and 126.)

[Altschul 1997] Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller and David J. Lipman. *Gapped Blast and PsiBlast: a new generation of protein database search programs.* Nucleic Acids Research, vol. 25, no. 17, pages 3389–3402, 1997. (Cited on page 33.)

[Amenta 2007] Nina Amenta, Dominique Attali and Olivier Devillers. *Complexity of Delaunay triangulation for points on lower-dimensional polyhedra.* In Proc. 18th Annual ACM-SIAM Sympos. Discrete Algo., pages 1106–1113, 2007. (Cited on page 88.)

[Andrassyova 1999] Eva Andrassyova, Jan Paralic, Eva Andrfissyovfi Msc and Jfin Parali Phd. *Knowledge Discovery in Databases: A Comparison of Different Views.*, 1999. (Cited on page 20.)

[Andreeva 2004] Antonina Andreeva, Dave Howorth, Steven E. Brenner, Tim J. P. Hubbard, Cyrus Chothia and Alexey G. Murzin. *SCOP database in 2004: refinements integrate structure and sequence family data.* Nucleic Acids Research, vol. 32, pages 226–229, 2004. (Cited on pages 3, 15, 52 and 53.)

[Andreeva 2008] Antonina Andreeva, Dave Howorth, John-Marc Chandonia, Steven E. Brenner, Tim J. P. Hubbard, Cyrus Chothia and Alexey G. Murzin. *Data growth and its impact on the SCOP database: new developments.* Nucleic Acids Research, vol. 36, no. suppl 1, pages D419–D425, 2008. (Cited on pages 32 and 104.)

[Baetu 2012] Tudor M. Baetu. *Genes after the human genome project.* Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences, vol. 43, no. 1, pages 191–201, March 2012. (Cited on pages 2 and 14.)

[Bairoch 2000] Amos Bairoch and Rolf Apweiler. *The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000.* Nucleic Acids

Research, vol. 28, no. 1, pages 45–48, January 2000. (Cited on pages 52 and 69.)

[Bairoch 2005] A. Bairoch, R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, D. A. Natale, C. O'Donovan, N. Redaschi and L. S. Yeh. *The Universal Protein Resource (UniProt).* Nucleic acids research, vol. 33, pages 154–159, 2005. (Cited on page 64.)

[Bandyopadhyay 2004] Bandyopadhyay and Snoeyink Jack. *Almost-Delaunay simplices: Robust neighbor relations for imprecise points.* ACM-SIAM Symposium On Distributed Algorithms, pages 403–412, 2004. (Cited on page 88.)

[Bartoli 2007] L. Bartoli, P. Fariselli and R. Casadio. *The effect of backbone on the small-world properties of protein contact maps.* Phys. Biol., vol. 4, no. 4, pages L1+, December 2007. (Cited on page 94.)

[Battaglia 2009] Giovanni Battaglia, Roberto Grossi, Roberto Marangoni and Nadia Pisanti. *Mining Biological Sequences with Masks.* In A Min Tjoa and Roland Wagner, editeurs, DEXA Workshops, pages 193–197. IEEE Computer Society, 2009. (Cited on page 94.)

[Benson 2009] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell and E. W. Sayers. *GenBank.* Nucleic acids research, vol. 37, pages D26–31, 2009. (Cited on page 64.)

[Berman 2000] Helen M. Berman, John D. Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov and Philip E. Bourne. *The Protein Data Bank.* Nucleic Acids Research, vol. 28, no. 1, pages 235–242, 2000. (Cited on pages 32, 58, 80, 84, 89, 104, 106 and 137.)

[Berman 2007] Helen M. Berman, Kim Henrick, Haruki Nakamura and John L. Markley. *The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data.* Nucleic Acids Research, vol. 35, no. Database-Issue, pages 301–303, 2007. (Cited on page 15.)

[Bernardes 2008] J Bernardes, J Fernandez and A Vasconcelos. *Structural descriptor database: a new tool for sequence based functional site prediction.* BMC Bioinformatics, vol. 9, page 492, 2008. (Cited on page 32.)

[Bhaskar 2005] H Bhaskar, DC Hoyle and S Singh. *Machine learning in bioinformatics: A brief survey and recommendations for practitioners.* Computers in Biology and Medicine, vol. 36, pages 1104–1125, 2005. (Cited on pages 3 and 32.)

[Bi 2003] Jinbo Bi and Vladimir Vapnik. *Learning with Rigorous Support Vector Machines.* In COLT, pages 243–257, 2003. (Cited on page 27.)

[Bondi 1964] A. Bondi. *van der Waals Volumes and Radii.* The Journal of Physical Chemistry, vol. 68, no. 3, pages 441–451, 1964. (Cited on page 84.)

[Borgelt 2002] C Borgelt and M R Berthold. *Mining molecular fragments: finding relevant substructures of molecules.* 2002 IEEE International Conference on Data Mining 2002 Proceedings, vol. 2, pages 51–58, 2002. (Cited on page 95.)

[Börner 2010] Katy Börner, Weixia Huang, Micah Linnemeier, Russell J. Duhon, Patrick Phillips, Nianli Ma, Angela Zoss, Hanning Guo and Mark A. Price. *Rete-netzwerk-red: analyzing and visualizing scholarly networks using the Network Workbench Tool.* Scientometrics, vol. 83, no. 3, pages 863–876, 2010. (Cited on page 137.)

[Bostick 2004] David L Bostick, Min Shen and Iosif I Vaisman. *A simple topological representation of protein structure: implications for new, fast, and robust structural classification.* Proteins, vol. 56, no. 3, pages 487–501, 2004. (Cited on page 88.)

[Brachman 1996] Ronald J. Brachman and Tej Anand. *The Process of Knowledge Discovery in Databases.* In Advances in Knowledge Discovery and Data Mining, pages 37–57. 1996. (Cited on pages 20 and 21.)

[Brandon 1991] C. Brandon and J. Tooze. Introduction to protein structure. Garland Publishing, 1991. (Cited on pages 12, 81, 82 and 83.)

[Brick 2008] Kevin Brick and Elisabetta Pizzi. *A novel series of compositionally biased substitution matrices for comparing Plasmodium proteins.* BMC Bioinformatics, vol. 9, no. 1, pages 236+, 2008. (Cited on page 18.)

[Brinda 2005] KV Brinda and S Vishveshwara. *A network representation of protein structures: implications for protein stability.* Biophys J., vol. 89, no. 6, pages 4159–4170, 2005. (Cited on page 84.)

[Brouard 2011] Céline Brouard, Florence d'Alché Buc and Marie Szafranski. *Semi-supervised Penalized Output Kernel Regression for Link Prediction.* In Lise Getoor and Tobias Scheffer, editeurs, ICML, pages 593–600. Omnipress, 2011. (Cited on page 94.)

[Brouard 2012] Céline Brouard, Marie Szafranski and Florence d'Alché Buc. *A novel approach of spatial motif extraction to classify protein structures.* In Proc. 13th Journées Ouvertes en Biologie, Informatique et Mathématiques (JOBIM), pages 133–136, 2012. (Cited on page 94.)

[Cai 2000] Yu-Dong Cai and Guo-Ping Zhou. *Prediction of protein structural classes by neural network*. Biochimie, vol. 82, no. 8, pages 783 – 785, 2000. (Cited on pages 3, 34, 52, 53, 54, 60, 61 and 62.)

[Cai 2001] Yu D. Cai, Xiao J. Liu, Xue B. Xu and Guo P. Zhou. *Support Vector Machines for predicting protein structural class*. BMC Bioinformatics, vol. 2, no. 1, pages 3+, June 2001. (Cited on pages 52, 53, 54, 60, 61 and 62.)

[Cannataro 2010] Mario Cannataro, Pietro Hiram Guzzi and Pierangelo Veltri. *Using RDF for managing protein-protein interaction data*. In BCB, pages 664–670, 2010. (Cited on page 22.)

[Cao 2006] Youfang Cao, Shi Liu, Lida Zhang, Jie Qin, Jiang Wang and Kexuan Tang. *Prediction of protein structural class with Rough Sets*. BMC Bioinformatics, vol. 7, no. 1, pages 20+, 2006. (Cited on pages 52, 53, 54, 60, 61 and 62.)

[Cheek 2004] Sara Cheek, Yuan Qi, S. Sri Krishna, Lisa N. Kinch and Nick V. Grishin. *SCOPmap: Automated assignment of protein structures to evolutionary superfamilies*. BMC Bioinformatics, vol. 5, page 197, 2004. (Cited on page 33.)

[Chen 2006] Chao Chen, Xibin Zhou, Yuanxin Tian, Xiaoyong Zou and Peixiang Cai. *Predicting protein structural class with pseudo-amino acid composition and support vector machine fusion network*. Analytical Biochemistry, vol. 357, no. 1, pages 116 – 121, 2006. (Cited on pages v, 3, 34, 35, 46, 52, 53, 54, 55, 60, 61 and 62.)

[Chou 1989] P Y Chou. Prediction of protein structure and the principles of protein conformation, chapitre Prediction of protein structural classes from amino acid composition, pages 549–586. Plenum Press, 1989. (Cited on pages 52, 53, 54, 60, 61 and 62.)

[Chou 2000] K. C. Chou. *Prediction of protein structural classes and subcellular locations*. Current protein & peptide science, vol. 1, no. 2, pages 171–208, September 2000. (Cited on page 54.)

[Chou 2003] Kuo-Chen Chou and Yu-Dong Cai. *Predicting protein quaternary structure by pseudo amino acid composition*. Proteins-structure Function and Bioinformatics, vol. 53, pages 282–289, 2003. (Cited on pages 46 and 54.)

[Chou 2004] Kuo-Chen Chou and Yu-Dong Cai. *Predicting protein structural class by functional domain composition*. Biochemical and Biophysical Research Communications, vol. 321, no. 4, pages 1007 – 1009, 2004. (Cited on page 54.)

[Clark 1991] D A Clark, J Shirazi and C J Rawlings. *Protein topology prediction through constraint-based search and the evaluation of topological folding rules.* Protein Engineering, vol. 4, no. 7, pages 751–760, 1991. (Cited on page 94.)

[Cohen 1988] Jacob Cohen. Statistical power analysis for the behavioral sciences. Hillsdale, NJ: Erlbaum, 2nd édition, January 1988. (Cited on page 65.)

[Cootes 2003] Adrian P Cootes, Stephen H Muggleton and Michael J E Sternberg. *The automatic discovery of structural principles describing protein fold space.* Journal of Molecular Biology, vol. 330, pages 839–850, 2003. (Cited on page 94.)

[Cornuéjols 2010] A. Cornuéjols and L. Miclet. Apprentissage artificiel: concepts et algorithmes. Algorithmes. Eyrolles, 2nd edition édition, 2010. (Cited on page 26.)

[Corrales 1996] FJ Corrales and AR Fersht. *Kinetic significance of GroEL14.(GroES7)2 complexes in molecular chaperone activity.* Folding & Design, vol. 1, pages 265–273, 1996. (Cited on page 32.)

[Crochemore 1991] Maxime Crochemore and Wojciech Rytter. *Usefulness of the Karp-Miller-Rosenberg Algorithm in Parallel Computations on Strings and Arrays.* Theor. Comput. Sci., vol. 88, no. 1, pages 59–82, 1991. (Cited on page 135.)

[Dayhoff 1978] M. O. Dayhoff, R. M. Schwartz and B. C. Orcutt. *A model of evolutionary change in proteins.* Atlas of protein sequence and structure, vol. 5, no. suppl 3, pages 345–351, 1978. (Cited on pages 18, 53 and 58.)

[De Berg 2008] Mark De Berg, Otfried Cheong, M Van Kreveld and M Overmars. Computational geometry: Algorithms and applications, volume 85. Springer, 2008. (Cited on pages 87 and 88.)

[Delaunay 1934] B. Delaunay. *Sur la sphère vide. A la mémoire de Georges Vorono.* Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et naturelles, vol. 6, pages 793–800, 1934. (Cited on page 88.)

[Dhifli 2010] Wajdi Dhifli and Rabie Saidi. *Protein graph repository.* In Yahia & Petit [Yahia 2010], pages 641–642. (Cited on page 92.)

[Dhifli 2012a] Wajdi Dhifli, Rabie Saidi and Engelbert Mephu Nguifo. *Frequent subgraph summarization by substitution matrices.* In Proc. 13^{th}

Journées Ouvertes en Biologie, Informatique et Mathématiques (JOBIM), pages 403–404, 2012. (Cited on page 123.)

[Dhifli 2012b] Wajdi Dhifli, Rabie Saidi and Engelbert Mephu Nguifo. *Novel Approach for Mining Representative Spatial Motifs of Proteins.* In Proc. ACM Conference on Bioinformatics, Computational Biology and Biomedicine (ACM BCB), page accepted, 2012. (Cited on page 123.)

[Ding 2001] Chris H. Q. Ding and Inna Dubchak. *Multi-class Protein Fold Recognition Using Support Vector Machines and Neural Networks.* Bioinformatics, vol. 17, pages 349–358, 2001. (Cited on pages 3 and 34.)

[Ding 2004] Chris H. Q. Ding and Xiaofeng He. *K-means clustering via principal component analysis.* In Carla E. Brodley, editeur, ICML, volume 69 of *ACM International Conference Proceeding Series.* ACM, 2004. (Cited on page 36.)

[Dominic A. Clark 1990] Geoffrey J. Baiton Dominic A. Clark Christopher J. Rawlings and Iain Archer. *Knowledge-Based Orchestration of Protein Sequence Analysis and Knowledge Acquisition for Protein Structure Prediction.* AAAI Spring Symposium, pages 28–32, 1990. (Cited on page 94.)

[Doppelt 2007] Olivia Doppelt, Fabrice Moriaud, Aurélie Bornot and Alexandre G de Brevern. *Functional annotation strategy for protein structures.* Bioinformation, vol. 9, no. 1, pages 357–359, 2007. (Cited on page 80.)

[Dunham 2002] Margaret H. Dunham. Data mining: Introductory and advanced topics. Prentice Hall, 1 édition, September 2002. (Cited on page 22.)

[Dunne 2002] Kevin Dunne, Padraig Cunningham and Francisco Azuaje. *Solutions to Instability Problems with Sequential Wrapper-Based Approaches To Feature Selection.* Rapport technique TCD-CD-2002-28, Department of Computer Science, Trinity College, Dublin, Ireland, 2002. (Cited on pages 64, 65 and 118.)

[Eddy 2008] Sean R. Eddy. *A probabilistic model of local sequence alignment that simplifies statistical significance estimation.* PLoS Comput Biol, vol. 4, no. 5, May 2008. (Cited on pages 51 and 121.)

[Ekman 2010] Diana Ekman and Arne Elofsson. *Identifying and Quantifying Orphan Protein Sequences in Fungi.* Journal of Molecular Biology, vol. 396, no. 2, pages 396–405, February 2010. (Cited on pages 3 and 33.)

[Faust 2010] Karoline Faust, Pierre Dupont, Jérôme Callut and Jacques van Helden. *Pathway discovery in metabolic networks by subgraph extraction.* Bioinformatics, vol. 26, no. 9, pages 1211–1218, May 2010. (Cited on page 94.)

[Fayyad 1997] Usama M. Fayyad. *Knowledge Discovery in Databases: An Overview.* In Inductive Logic Programming ILP, pages 3–16, 1997. (Cited on pages 20 and 21.)

[Feelders 2000] A. J. Feelders, H. A. M. Daniels and Marcel Holsheimer. *Methodological and practical aspects of data mining.* Information & Management, vol. 37, no. 5, pages 271–281, 2000. (Cited on page 34.)

[Fei 2010] Hongliang Fei and Jun Huan. *Boosting with structure information in the functional space: an application to graph classification.* In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '10, pages 643–652, New York, NY, USA, 2010. ACM. (Cited on pages v, 94, 95, 96, 104, 105 and 114.)

[Feng 2005] Kai-Yan Feng, Yu-Dong Cai and Kuo-Chen Chou. *Boosting classifier for predicting protein domain structural class.* Biochemical and Biophysical Research Communications, vol. 334, pages 213–217, 2005. (Cited on pages 52, 53, 54, 60, 61 and 62.)

[Finn 2010] Robert D. Finn, Jaina Mistry, John G. Tate, Penny C. Coggill, Andreas Heger, Joanne E. Pollington, O. Luke Gavin, Prasad Gunasekaran, Goran Ceric, Kristoffer Forslund, Liisa Holm, Erik L. L. Sonnhammer, Sean R. Eddy and Alex Bateman. *The Pfam protein families database.* Nucleic Acids Research, vol. 38, pages 211–222, 2010. (Cited on page 46.)

[Frézal 1998] J. Frézal. *Genatlas database, genes and development defects.* Comptes rendus de l'Académie des sciences. Série III, Sciences de la vie, vol. 321, no. 10, pages 805–817, October 1998. (Cited on page 15.)

[Ghosh 2000] David Ghosh. *Object-oriented Transcription Factors Database (ooTFD).* Nucleic Acids Research, vol. 28, no. 1, pages 308–310, 2000. (Cited on page 15.)

[Gille 2000] C Gille, A Goede, R Preissner, K Rother and C Frommel. *Conservation of substructures in proteins: interfaces of secondary structural elements in proteasomal subunits.* Journal of Molecular Biology, vol. 299, no. 4, pages 1147–1154, 2000. (Cited on page 96.)

[Han 2006] J Han and M Kamber. Data mining: Concepts and techniques. isbn 1-55860-901-6. Morgan Kaufmann Publishers: www.mkp.com, 2006. (Cited on pages 26, 37, 54 and 70.)

[Hasan 2009] Mohammad Al Hasan and Mohammed J. Zaki. *Output Space Sampling for Graph Patterns.* PVLDB, vol. 2, no. 1, pages 730–741, 2009. (Cited on pages 35 and 108.)

[Henikoff 1992] Steven Henikoff and Jorja G. Henikoff. *Amino acid substitution matrices from protein blocks.* Proceedings of The National Academy of Sciences, vol. 89, pages 10915–10919, 1992. (Cited on pages 18, 47, 53 and 58.)

[Holm 1993] L. Holm and C. Sander. *Protein structure comparison by alignment of distance matrices.* J. Mol. Biol, vol. 233, pages 123–138, 1993. (Cited on page 33.)

[Hong Cheng 2010] Xifeng Yan Hong Cheng and Jiawei Han. *Mining Graph Patterns.* In Managing and Mining Graph Data, pages 365–392. 2010. (Cited on page 95.)

[Huan 1998] L Huan and H Motoda. Feature extraction, construction and selection: A data mining perspective. isbn: 978-0-7923-8196-9. Kluwer Academic Publishers, Norwell, MA, 1998. (Cited on page 37.)

[Huan 2003] Jun Huan, Wei Wang 0010 and Jan Prins. *Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism.* In ICDM, pages 549–552, 2003. (Cited on page 95.)

[Huan 2005] Jun Huan, Deepak Bandyopadhyay, Wei Wang 0010, Jack Snoeyink, Jan Prins and Alexander Tropsha. *Comparing Graph Representations of Protein Structure for Mining Family-Specific Residue-Based Packing Motifs.* Journal of Computational Biology, vol. 12, no. 6, pages 657–671, 2005. (Cited on pages 84, 88 and 105.)

[Huang 2003] Chuen-Der Huang, I-Fang Chung, Nikhil R. Pal and Chin-Teng Lin. *Machine Learning for Multi-class Protein Fold Classification Based on Neural Networks with Feature Gating.* In Artificial Neural Networks and Neural Information Processing - ICANN/ICONIP 2003, Joint International Conference ICANN/ICONIP 2003, Istanbul, Turkey, June 26-29, 2003, Proceedings, pages 1168–1175, 2003. (Cited on pages 3 and 34.)

[Hui 1992] LCK Hui. *Color Set Size Problem with Applications to String Matching.* In Z. Galil A. Apostolico M. Crochemore and U. Manber, editeurs, Combinatorial Pattern Matching, volume 644 of *Lec-*

ture Notes in Computer Science, pages 230–243. Springer-Verlag, 1992. (Cited on page 45.)

[Ie 2005] Eugene Ie, Jason Weston, William Stafford Noble and Christina Leslie. *Multi-class protein fold recognition using adaptive codes.* In Proceedings of the 22nd International Conference on Machine Learning, pages 329–336. ACM, 2005. (Cited on pages 3 and 34.)

[Inokuchi 2000] Akihiro Inokuchi, Takashi Washio and Hiroshi Motoda. *An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data.* In PKDD, pages 13–23, 2000. (Cited on page 95.)

[Jacobs 2001] D J Jacobs, A J Rader, L A Kuhn and M F Thorpe. *Protein flexibility predictions using graph theory.* Proteins, vol. 44, no. 2, pages 150–165, 2001. (Cited on page 86.)

[Jin 2009] Ning Jin, Calvin Young and Wei Wang 0010. *Graph classification based on pattern co-occurrence.* In CIKM, pages 573–582, 2009. (Cited on pages 95, 104 and 114.)

[Johnson 2006] S. Johnson. *Remote protein homology detection using hidden Markov models.* PhD thesis, Washington University School of Medicine, 2006. (Cited on pages 46, 51, 121 and 129.)

[Jung 2000] Jongsun Jung and Byungkook Lee. *Protein structure alignment using environmental profiles.* Protein Eng., vol. 13, no. 8, pages 535–543, August 2000. (Cited on pages 114 and 128.)

[Kahan 2008] M Kahan, B Gil, R Adar and E Shapiro. *Towards molecular computers that operate in a biological environment.* Physica D: Nonlinear Phenomena, vol. 237, no. 9, pages 1165–1172, 2008. (Cited on page 10.)

[Kalousis 2007] Alexandros Kalousis, Julien Prados and Melanie Hilario. *Stability of feature selection algorithms: a study on high-dimensional spaces.* Knowledge and Information Systems, vol. 12, pages 95–116, 2007. 10.1007/s10115-006-0040-8. (Cited on pages 64 and 118.)

[Kanehisa 2000] M. Kanehisa and S. Goto. *KEGG: Kyoto Encyclopedia of Genes and Genomes.* Nucleic Acids Research, vol. 28, no. 1, pages 27–30, January 2000. (Cited on page 121.)

[Karp 1972] Richard M. Karp, Raymond E. Miller and Arnold L. Rosenberg. *Rapid identification of repeated patterns in strings, trees and arrays.* In ACM Symposium on Theory of Computing, pages 125–136, 1972. (Cited on pages 46, 97, 99 and 135.)

[Keating 2009] KS Keating, SC Flores, MB Gerstein and LA. Kuhn. *Stone-Hinge: hinge prediction by network analysis of individual protein structures.* Protein Science, vol. 18, no. 2, pages 359–371, 2009. (Cited on page 86.)

[Ketkar 2005] Nikhil S. Ketkar. *Subdue: compression-based frequent pattern discovery in graph data.* In Proceedings of the 1st international workshop on open source data mining OSDM, pages 71–76. ACM Press, 2005. (Cited on page 89.)

[Khreisat 2009] Laila Khreisat. *A machine learning approach for Arabic text classification using N-gram frequency statistics.* J. Informetrics, vol. 3, no. 1, pages 72–77, 2009. (Cited on page 44.)

[Kleywegt 1999] G. J. Kleywegt. *Recognition of spatial motifs in protein structures.* Journal of molecular biology, vol. 285, no. 4, pages 1887–1897, 1999. (Cited on page 80.)

[Klotz 1975] C Klotz, MC Aumont, JJ Leger and B Swynghedauw. *Human cardiac myosin ATPase and light subunits: A comparative study.* Biochim Biophys, vol. 386, pages 461–469, 1975. (Cited on page 32.)

[Kohavi 1995] Ron Kohavi. *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection.* In IJCAI, pages 1137–1145. Morgan Kaufmann, 1995. (Cited on page 27.)

[Kotsiantis 2007] Sotiris B. Kotsiantis. *Supervised Machine Learning: A Review of Classification Techniques.* Informatica (Slovenia), vol. 31, no. 3, pages 249–268, 2007. (Cited on page 27.)

[Krishna 2011] Varun Krishna, N N R R Suri and G Athithan. *A comparative survey of algorithms for frequent subgraph discovery.* Current Science, vol. 100, no. 2, page 190, 2011. (Cited on pages 95 and 123.)

[Kroger 1998] Manfred Kroger and Ralf Wahl. *Compilation of DNA sequences of Escherichia coli K12: description of the interactive databases ECD and ECDC.* Nucleic Acids Res, vol. 26, pages 46–9, 1998. (Cited on page 15.)

[Kuramochi 2001] Michihiro Kuramochi and George Karypis. *Frequent Subgraph Discovery.* In ICDM, pages 313–320, 2001. (Cited on page 95.)

[Lefranc 2003] Marie-Paule Lefranc. *IMGT, the international ImMunoGeneTics database.* Nucleic Acids Research, vol. 31, no. 1, pages 307–310, 2003. (Cited on page 15.)

[Lemoine 1999] E Lemoine, D Merceron, J Sallantin and E Mephu Nguifo. *Improving the Efficiency of a User-Driven Learning System with Reconfigurable Hardware. Application to DNA Splicing.* In Pacific Symposium on Biocomputing, pages 290–301, 1999. (Cited on pages 32 and 94.)

[Leslie 2002] Christina S. Leslie, Eleazar Eskin and William Stafford Noble. *The Spectrum Kernel: A String Kernel for SVM Protein Classification.* In Pacific Symposium on Biocomputing, pages 566–575, 2002. (Cited on pages 44, 47 and 69.)

[Li 2008] Pei-Pei Li, Xuegang Hu and Xindong Wu. *Mining Concept-Drifting Data Streams with Multiple Semi-Random Decision Trees.* In ADMA, pages 733–740, 2008. (Cited on page 27.)

[Liu 1998] Huan Liu and Hiroshi Motoda. *Feature Transformation and Subset Selection.* IEEE Intelligent Systems, vol. 13, pages 26–28, 1998. (Cited on page 35.)

[Liu 2007] Huan Liu and Hiroshi Motoda. Computational methods of feature selection (chapman & hall/crc data mining and knowledge discovery series). Chapman & Hall/CRC, 2007. (Cited on pages 36 and 65.)

[Lopes 2008] F Lopes, D Martins and R Cesar. *Feature selection environment for genomic applications.* BMC Bioinformatics, vol. 9, page 451, 2008. (Cited on page 37.)

[Lovell 2003] Simon C. Lovell, Ian W. Davis, W. Bryan Arendall, Paul I. W. de Bakker, J. Michael Word, Michael G. Prisant, Jane S. Richardson and David C. Richardson. *Structure validation by Calpha geometry: Phi, Psi and Cbeta deviation.* Proteins: Structure, Function, and Bioinformatics, vol. 50, no. 3, pages 437–450, 2003. (Cited on page 84.)

[Ma 2008] Shuangge Ma and Jian Huang. *Penalized feature selection and classification in bioinformatics.* Briefings in bioinformatics, vol. 9, no. 5, pages 392–403, September 2008. (Cited on page 65.)

[Madani 2008] Omid Madani and Michael Connor. *Large-Scale Many-Class Learning.* In Proceedings of the SIAM International Conference on Data Mining, SDM 2008, April 24-26, 2008, Atlanta, Georgia, USA, pages 846–857, 2008. (Cited on page 120.)

[Maddouri 2004] Mondher Maddouri and Mourad Elloumi. *Encoding of primary structures of biological macromolecules within a data mining perspective.* Journal of Computer Science and Technology (JCST). Allerton Press. USA, vol. 19, no. 1, pages 78–88, 2004. (Cited on pages 37, 46, 51 and 69.)

[Malde 2008] Ketil Malde. *The effect of sequence quality on sequence alignment.* Bioinformatics/computer Applications in The Biosciences, vol. 24, pages 897–900, 2008. (Cited on page 47.)

[Mannila 1997] Heikki Mannila. *Methods and Problems in Data Mining.* In Database Theory - ICDT '97, 6th International Conference, Delphi, Greece, January 8-10, 1997, Proceedings, pages 41–55, 1997. (Cited on pages 20 and 21.)

[Mardia 1979] K. V. Mardia, J. M. Bibby and J. T. Kent. Multivariate analysis / k. v. mardia, j. t. kent, j. m. bibby. Academic Press, London ; New York :, 1979. (Cited on page 54.)

[Melvin 2007] Iain Melvin, Eugene Ie, Rui Kuang, Jason Weston, William Stafford Noble and Christina S. Leslie. *SVM-Fold: a tool for discriminative multi-class protein fold and superfamily recognition.* BMC Bioinformatics, vol. 8, no. S-4, 2007. (Cited on pages 3 and 34.)

[Mephu Nguifo 1993] E Mephu Nguifo and J Sallantin. *Prediction of Primate Splice Junction Gene Sequences with a Cooperative Knowledge Acquisition System.* In International Conference on Intelligent Systems for Molecular Biology ISMB, pages 292–300, 1993. (Cited on pages 32 and 94.)

[Mesleh 2007] Abdelwadood Moh'd A. Mesleh. *Chi Square Feature Extraction Based Svms Arabic Text Categorization System.* In Joaquim Filipe, Boris Shishkov and Markus Helfert, editeurs, ICSOFT (PL/DPS/KE/MUSE), pages 235–240. INSTICC Press, 2007. (Cited on page 44.)

[Mohapatra 2011] Saroj K. Mohapatra and Arjun Krishnan. *Microarray data analysis.* Methods in molecular biology (Clifton, N.J.), vol. 678, pages 27–43, 2011. (Cited on page 14.)

[Mount 2008] David W. Mount. *Comparison of the PAM and BLOSUM Amino Acid Substitution Matrices.* Cold Spring Harbor Protocols, vol. 2008, no. 7, pages pdb.ip59+, June 2008. (Cited on page 18.)

[Muggleton 2006] Stephen H. Muggleton. *2020 Computing: Exceeding human limits.* Nature, vol. 440, no. 7083, pages 409–410, March 2006. (Cited on page 32.)

[Nakashima 1986] Hiroshi Nakashima, Ken Nishikawa and Tatsuo Ooi. *The Folding Type of a Protein Is Relevant to the Amino Acid Composition.* J Biochem, vol. 99, no. 1, pages 153–162, January 1986. (Cited on pages 52, 53, 54, 60, 61 and 62.)

[Nevill-Manning 1998] CG Nevill-Manning, TD Wu and DL Brutlag. *Highly specific protein sequence motifs for genome analysis.* In Proceedings of the National Academy of Sciences of the United States of America, volume 95(11), pages 5865–5871, 1998. (Cited on pages 37 and 53.)

[Nijssen 2004] Siegfried Nijssen and Joost N. Kok. *A quickstart in frequent structure mining can make a difference.* In KDD, pages 647–652, 2004. (Cited on pages 95 and 105.)

[Ollivier 1991] E. Ollivier, Henry Soldano and Alain Viari. *'Multifrequency' location and clustering of sequence patterns from proteins.* Computer Applications in the Biosciences, vol. 7, no. 1, pages 31–38, 1991. (Cited on page 44.)

[Orengo 2002] Christine A Orengo, James E Bray, Daniel W A Buchan, Andrew Harrison, David Lee, Frances M G Pearl, Ian Sillitoe, Annabel E Todd and Janet M Thornton. *The CATH protein family database: A resource for structural an functional annotation of genomes.* Proteomics, vol. 2, no. 1, pages 11–21, 2002. (Cited on page 15.)

[Ortiz 2002] AR Ortiz, CE Strauss and O Olmea. *MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison.* Protein science : a publication of the Protein Society, vol. 11, pages 2606–21, 2002 Nov 2002. (Cited on page 33.)

[Ouzounis 2003] Christos A. Ouzounis and Alfonso Valencia. *Early bioinformatics: the birth of a discipline - a personal view.* Bioinformatics, vol. 19, no. 17, pages 2176–2190, 2003. (Cited on page 10.)

[Pavel 2007] K. Pavel, K. Josef and H. Václav. *Improving stability of feature selection methods.* In Proceedings of the 12th international conference on Computer analysis of images and patterns, CAIP'07, pages 929–936, Berlin, Heidelberg, 2007. Springer-Verlag. (Cited on pages 64 and 118.)

[Pearson 2001] William R Pearson and Jordan Hall. *Protein sequence comparison and Protein evolution Tutorial - ISMB2000.* Biochemistry, no. 804, pages 1–53, 2001. (Cited on page 96.)

[Pearson 2005] William R. Pearson and Michael L. Sierk. *The limits of protein sequence comparison?* Current opinion in structural biology, vol. 15, no. 3, pages 254–260, June 2005. (Cited on pages 3 and 33.)

[Pisanti 2005] Nadia Pisanti, Henry Soldano and Mathilde Carpentier. *Incremental Inference of Relational Motifs with a Degenerate Alphabet.* In Alberto Apostolico, Maxime Crochemore and Kunsoo Park, editeurs, CPM, volume 3537 of *Lecture Notes in Computer Science*, pages 229–240. Springer, 2005. (Cited on page 135.)

[Price 1994] NC Price. Mechanisms of protein folding, chapitre Assembly of multi-subunit structure, pages 160–193. Oxford University Press, Oxford, 1994. (Cited on page 32.)

[Ramachandran 1968] G. N. Ramachandran and V. Sasisekharan. Conformation of polypeptides and proteins, volume 23 of *Advances in Protein Chemistry*, pages 283–437. 1968. (Cited on page 100.)

[Ras 2008] Zbigniew W. Ras, Shusaku Tsumoto and Djamel A. Zighed, editeurs. Mining complex data, ecml/pkdd 2007 third international workshop, mcd 2007, warsaw, poland, september 17-21, 2007, revised selected papers, volume 4944 of *Lecture Notes in Computer Science*, 2008. (Cited on page 23.)

[Real 1996] R Real and J M Vargas. *The Probabilistic Basis of Jaccard's Index of Similarity.* Systematic Biology, vol. 45, no. 3, pages 380–385, 1996. (Cited on page 65.)

[Recupero 2008] Diego Reforgiato Recupero, Rodrigo A. Gutiérrez and Dennis Shasha. *Graphclust: a Method for Clustering Database of Graphs.* JIKM, vol. 7, no. 4, pages 231–241, 2008. (Cited on page 137.)

[Saeys 2007] Yvan Saeys, Iñaki Inza and Pedro Larrañaga. *A review of feature selection techniques in bioinformatics.* Bioinformatics, vol. 23, no. 19, pages 2507–2517, October 2007. (Cited on page 65.)

[Saidi 2009] Rabie Saidi, Mondher Maddouri and Engelbert Mephu Nguifo. *Comparing graph-based representations of protein for mining purposes.* In KDD StReBio, pages 35–38, 2009. (Cited on pages ix, 79, 106, 117 and 119.)

[Saidi 2010a] Rabie Saidi, Sabeur Aridhi, Mondher Maddouri and Engelbert Mephu Nguifo. *Etude de stabilité de méthodes d'extraction de motifs à partir des séquences protéiques.* In Yahia & Petit [Yahia 2010], pages 703–704. (Cited on pages ix, 63, 117 and 118.)

[Saidi 2010b] Rabie Saidi, Mondher Maddouri and Engelbert Mephu Nguifo. *Protein sequences classification by means of feature extraction with substitution matrices.* BMC bioinformatics, vol. 11, no. 1, pages 175+, April 2010. (Cited on pages ix, 43, 44, 47, 69, 80, 94, 117 and 118.)

[Saidi 2012a] Rabie Saidi, Sabeur Aridhi, Mondher Maddouri and Engelbert Mephu Nguifo. *Feature Extraction in Protein Sequences Classification:*

A New Stability Measure. In Proc. ACM Conference on Bioinformatics, Computational Biology and Biomedicine (ACM BCB), WRSBS Workshop, page accepted, 2012. (Cited on pages ix, 63, 117 and 118.)

[Saidi 2012b] Rabie Saidi, Wajdi Dhifli, Mondher Maddouri and Engelbert Mephu Nguifo. *A novel approach of spatial motif extraction to classify protein structures.* In Proc. 13^{th} Journées Ouvertes en Biologie, Informatique et Mathématiques (JOBIM), pages 209–216, 2012. (Cited on pages ix, 93, 117 and 119.)

[Saigo 2008] Hiroto Saigo, Nicole Krämer and Koji Tsuda. *Partial least squares regression for graph mining.* In KDD, pages 578–586, 2008. (Cited on page 95.)

[Saigo 2009] Hiroto Saigo, Sebastian Nowozin, Tadashi Kadowaki, Taku Kudo and Koji Tsuda. *A mathematical programming approach to graph classification and regression.* Machine Learning, vol. 75, no. 1, pages 69–89, 2009. (Cited on page 95.)

[Sander 1991] Chris Sander and Reinhard Schneider. *Database of homology-derived protein structures and the structural meaning of sequence alignment.* Proteins: Structure, Function, and Bioinformatics, vol. 9, no. 1, pages 56–68, 1991. (Cited on page 58.)

[Santini 2010] Guillaume Santini, Henry Soldano and Joël Pothier. *Use of ternary similarities in graph based clustering for protein structural family classification.* In Aidong Zhang, Mark Borodovsky, Gultekin Özsoyoglu and Armin R. Mikler, editeurs, BCB, pages 457–459. ACM, 2010. (Cited on page 94.)

[Schwartz 1979] R. Schwartz and M. Dayhoff. Matrices for detecting distant relationships, pages 353–358. National Biomedical Research Foundation, 1979. (Cited on page 18.)

[Sebban 2002] Marc Sebban and Richard Nock. *A hybrid filter/wrapper approach of feature selection using information theory.* Pattern Recognition, vol. 35, no. 4, pages 835 – 846, 2002. (Cited on page 65.)

[Shamim 2011] Mohammad Tabrez Anwar Shamim and Hampapathalu A. Nagarajaram. *Svm-Based Method for protein Structural Class Prediction Using Secondary Structural Content and Structural Information of amino acids.* J. Bioinformatics and Computational Biology, vol. 9, no. 4, pages 489–502, 2011. (Cited on pages 3 and 34.)

[Sigrist 2010] Christian J. A. Sigrist, Lorenzo Cerutti, Edouard De Castro, Petra S. Langendijk-Genevaux, Virginie Bulliard, Amos Bairoch and

Nicolas Hulo. *PROSITE, a protein domain database for functional characterization and annotation.* Nucleic Acids Research, vol. 38, no. Database-Issue, pages 161–166, 2010. (Cited on page 15.)

[Simoudis 1996] Evangelos Simoudis. *Reality Check for Data Mining.* IEEE Expert, vol. 11, no. 5, pages 26–33, 1996. (Cited on pages 20 and 21.)

[Slama 2008] P Slama, I Filippis and M Lappe. *Detection of protein catalytic residues at high precision using local network properties.* BMC Bioinformatics, vol. 9, page 517, 2008. (Cited on page 32.)

[Somol 2008] Petr Somol and Jana Novovicová. *Evaluating the Stability of Feature Selectors That Optimize Feature Subset Cardinality.* In Niels da Vitoria Lobo, Takis Kasparis, Fabio Roli, James Kwok, Michael Georgiopoulos, Georgios Anagnostopoulos and Marco Loog, editeurs, Structural, Syntactic, and Statistical Pattern Recognition, volume 5342 of *Lecture Notes in Computer Science*, pages 956–966. Springer Berlin / Heidelberg, 2008. (Cited on pages 64 and 118.)

[Song 2004] Jie Song and Huanwen Tang. *Accurate Classification of Homodimeric vs Other Homooligomeric Proteins Using a New Measure of Information Discrepancy.* Journal of Chemical Information and Computer Sciences, vol. 44, no. 4, pages 1324–1327, 2004. PMID: 15272840. (Cited on page 54.)

[Steudel 1975] Ralf Steudel. *Properties of Sulfur-Sulfur Bonds.* Angewandte Chemie International Edition in English, vol. 14, no. 10, pages 655–664, 1975. (Cited on page 83.)

[Stoesser 2002] G. Stoesser, W. Baker, A. van den Broek, E. Camon, M. Garcia-Pastor, C. Kanz, T. Kulikova, R. Leinonen Q. Lin, V. Lombard, R. Lopez, N. Redaschi, P. Stoehr, M. A. Tuli, K. Tzouvara and R. Vaughan. *The EMBL Nucleotide Sequence Database.* Nucleic acids research, vol. 30, no. 1, pages 21–26, 2002. (Cited on page 64.)

[Stout 2008] Michael Stout, Jaume Bacardit, Jonathan D. Hirst, Robert E. Smith and Natalio Krasnogor. *Prediction of topological contacts in proteins using learning classifier systems.* Soft Comput., vol. 13, pages 245–258, October 2008. (Cited on page 88.)

[Terry 1998] BF Terry and MC Richard. *Determination of protein-protein interactions by matrix-assisted laser desorption/ionization mass spectrometry.* J. Mass Spectrom, vol. 33, pages 697–704, 1998. (Cited on page 32.)

[Theocharidis 2009] Athanasios Theocharidis, Stjin van Dongen, Anton J. Enright and Tom C. Freeman. *Network visualization and analysis of gene expression data using BioLayout Express(3D).* Nature protocols, vol. 4, no. 10, pages 1535–1550, October 2009. (Cited on page 137.)

[Topaloglou 2004] T. Topaloglou. *Biological data management: research, practice and opportunities.* In Proceeding of the 30th VLDB conference, pages 1233–1236, Toronto, 2004. (Cited on page 64.)

[Tortora 2006] Gerard J. Tortora, Berdell R. Funke and Christine L. Case. Microbiology: An introduction, 9th edition. Benjamin Cummings, 9th édition, March 2006. (Cited on page 10.)

[Tzanis 2007] G. Tzanis, C. Berberidis and P. I. Vlahavas. *MANTIS: A Data Mining Methodology for Effective Translation Initiation Site Prediction.* Proc. of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, Lyon, France, pages 6344–6348, 2007. (Cited on page 22.)

[Vallabhajosyula 2009] Ravishankar R. Vallabhajosyula, Deboki Chakravarti, Samina Lutfeali, Animesh Ray and Alpan Raval. *Identifying Hubs in Protein Interaction Networks.* PLoS ONE, vol. 4, no. 4, page e5344, 04 2009. (Cited on page 94.)

[Vapnik 1995] Vladimir Vapnik and Corinna Cortes. *Support-Vector Networks.* Machine Learning, vol. 20, no. 3, pages 273–297, 1995. (Cited on page 27.)

[Vignal 1997] L Vignal, Y D'Aubenton-Carafa, F Lisacek E Mephu Nguifo, PRouze P, J Quinqueton and C Thermes C. *Exon prediction in eucaryotic genomes.* Biochimie, vol. 78, pages 327–334, 1997. (Cited on page 32.)

[Vishveshwara 2002] S Vishveshwara, KV Brinda and N Kannan. *Protein Structure: Insights from Graph Theory.* Jl Th Comp Chem., vol. 1, no. 1, pages 187–211, 2002. (Cited on page 81.)

[Wang 1994] J. T. L. Wang, T. G. Marr, D. Shasha, B. A. Shapiro and G. W. Chirn. *Discovering Active Motifs in Sets of Related Protein Sequences and Using Them for Classification.* Nucleic Acids Research, vol. 22, no. 14, pages 2769–2775, 1994. (Cited on pages 44, 45 and 69.)

[Wasserman 2004] L. Wasserman. All of statistics: A concise course in statistical inference. Springer Texts in Statistics. Springer, 2004. (Cited on page 27.)

[Weiss 1990] Sholom M. Weiss and Casimir A. Kulikowski. Computer systems that learn: Classification and prediction methods from statistics, neural nets, machine learning and expert systems. Morgan Kaufmann, 1990. (Cited on page 27.)

[Witten 2005] Ian H Witten and Eibe Frank. Data mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2005. (Cited on pages 51, 54, 61, 71 and 106.)

[Yahia 2010] Sadok Ben Yahia and Jean-Marc Petit, editeurs. Extraction et gestion des connaissances (egc'2010), actes, 26 au 29 janvier 2010, hammamet, tunisie, volume RNTI-E-19 of *Revue des Nouvelles Technologies de l'Information*. Cépaduès-Éditions, 2010. (Cited on pages 149 and 158.)

[Yan 2002] Xifeng Yan and Jiawei Han. *gSpan: Graph-based substructure pattern mining*. Order A Journal On The Theory Of Ordered Sets And Its Applications, vol. 02, pages 721–724, 2002. (Cited on page 95.)

[Yan 2008] X Yan, H Cheng, J Han and P S Yu. *Mining significant graph patterns by leap search*. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pages 433–444. ACM SIGMOD, 2008. (Cited on page 95.)

[Yongqiang 2006a] Z Yongqiang and MJ Zaki. *EXMOTIF: efficient structured motif extraction*. Journal of Algorithms for Molecular Biology, BioMed Central, vol. 1, page 21, 2006. (Cited on pages 37, 46 and 94.)

[Yongqiang 2006b] Z Yongqiang and MJ Zaki. *SMOTIF: efficient structured pattern and profile motif search*. Journal of Algorithms for Molecular Biology, BioMed Central, vol. 1, page 22, 2006. (Cited on pages 37, 46 and 94.)

[Yu 2005] Yi-Kuo Yu and Stephen F. Altschul. *The construction of amino acid substitution matrices for the comparison of proteins with non-standard compositions*. Bioinformatics, vol. 21, no. 7, pages 902–911, April 2005. (Cited on page 18.)

[Yu 2006] Xiaojing Yu, Chuan Wang and Yixue Li. *Classification of protein quaternary structure by functional domain composition*. BMC Bioinformatics, vol. 7, pages 187–192, 2006. (Cited on pages v, 52, 53, 54, 55, 58 and 60.)

[Yu 2008] Lei Yu, Chris Ding and Steven Loscalzo. *Stable feature selection via dense feature groups*. In Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '08, pages 803–811, New York, NY, USA, 2008. ACM. (Cited on pages 64 and 118.)

[Yule 1950] G.U. Yule and M.G. Kendall. An introduction to the theory of statistics. Griffin, London, 1950. (Cited on page 65.)

[Yvan 2008] S. Yvan, A. Thomas and P. Yves. *Robust Feature Selection Using Ensemble Feature Selection Techniques.* In Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II, ECML PKDD '08, pages 313–325, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on pages 64, 65 and 118.)

[Zhang 1995] Chun Ting Zhang, Kuo-Chen Chou and G. M. Maggiora. *Predicting protein structural classes from amino acid composition: application of fuzzy clustering.* Protein Engineering Design & Selection, vol. 8, pages 425–435, 1995. (Cited on page 46.)

[Zhang 2003a] Shao Wu Zhang, Quan Pan, Hongcai Zhang, Yun long Zhang and Hai yu Wang. *Classification of protein quaternary structure with support vector machine.* Bioinformatics/computer Applications in The Biosciences, vol. 19, pages 2390–2396, 2003. (Cited on pages 35, 46 and 54.)

[Zhang 2003b] Shichao Zhang, Chengqi Zhang and Qiang Yang. *Data Preparation for Data Mining.* Applied Artificial Intelligence, vol. 17, no. 5-6, pages 375–381, 2003. (Cited on page 34.)

[Zhang 2006] S.-W. Zhang, Q. Pan, H.-C. Zhang, Z.-C. Shao and J.-Y. Shi. *Prediction of protein homo-oligomer types by pseudo amino acid composition: Approached with an improved feature extraction and Naive Bayes Feature Fusion.* Amino Acids, vol. 30, pages 461–468, 2006. 10.1007/s00726-006-0263-8. (Cited on page 54.)

[Zhao 2008] Z. Zhao, J. Wang, H. Liu, J. Ye and Y. Chang. *Identifying biologically relevant genes via multiple heterogeneous data sources.* In Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 839–847, USA, 2008. (Cited on page 64.)

[Zhou 1998] GP Zhou. *An intriguing controversy over protein structural class prediction.* J. Protein Chem, vol. 17, pages 729–738, 1998. (Cited on pages v, 46, 52, 53, 54, 55, 60, 61 and 62.)

[Zimmermann 2010] Karel Zimmermann and Jean F. Gibrat. *Amino acid "little Big Bang": Representing amino acid substitution matrices as dot products of Euclidian vectors.* BMC Bioinformatics, vol. 11, no. 1, pages 4+, 2010. (Cited on page 18.)

[Zweig 1993] MH Zweig and G Campbell. *Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine [published erratum appears in Clin Chem 1993 Aug;39(8):1589].* Clin Chem, vol. 39, no. 4, pages 561–577, 1993. (Cited on pages 31 and 55.)

Motif Extraction from Complex Data: Case of Protein Classification

Abstract: The classification of biological data is one of the significant challenges in bioinformatics, as well for protein as for nucleic data. The presence of these data in huge masses, their ambiguity and especially the high costs of the in vitro analysis in terms of time and resources, make the use of data mining rather a necessity than a rational choice. However, the data mining techniques, which often process data under the relational format, are confronted with the inappropriate format of the biological data. Hence, an inevitable step of pre-processing must be established.

This thesis deals with the protein data preprocessing as a preparation step before their classification. We present motif extraction as a reliable way to address that task. The extracted motifs are used as descriptors to encode proteins into feature vectors. This enables the use of known data mining classifiers which require this format. However, designing a suitable feature space, for a set of proteins, is not a trivial task.

We deal with two kinds of protein data *i.e.*, sequences and tri-dimensional structures. In the first axis *i.e.*, protein sequences, we propose a novel encoding method that uses amino-acid substitution matrices to define similarity between motifs during the extraction step. We demonstrate the efficiency of such approach by comparing it with several encoding methods, using some classifiers. We also propose new metrics to study the robustness of some of these methods when perturbing the input data. These metrics allow to measure the ability of the method to reveal any change occurring in the input data and also its ability to target the interesting motifs. The second axis is dedicated to 3D protein structures which are recently seen as graphs of amino acids. We make a brief survey on the most used graph-based representations and we propose a naïve method to help with the protein graph making. We show that some existing and widespread methods present remarkable weaknesses and do not really reflect the real protein conformation. Besides, we are interested in discovering recurrent sub-structures in proteins which can give important functional and structural insights. We propose a novel algorithm to find spatial motifs from proteins. The extracted motifs match a well-defined shape which is proposed based on a biological basis. We compare with sequential motifs and spatial motifs of recent related works. For all our contributions, the outcomes of the experiments confirm the efficiency of our proposed methods to represent both protein sequences and protein 3D structures in classification tasks.

Software programs developed during this research work are available on my home page http://fc.isima.fr/~saidi.

Keywords: Preprocessing, motif/feature extraction, protein classification, protein structures, sequential motif, spatial motif.

Extraction de Motifs des Données Complexes: Cas de la Classification des Protéines

Abstract: La classification est l'un des défis important en bioinformatique, aussi bien pour les données protéiques que nucléiques. La présence de ces données en grandes masses, leur ambiguïté et en particulier les coûts élevés de l'analyse *in vitro* en termes de temps et d'argent, rend l'utilisation de la fouille de données plutôt une nécessité qu'un choix rationnel. Cependant, les techniques fouille de données, qui traitent souvent des données sous le format relationnel, sont confrontés avec le format inapproprié des données biologiques. Par conséquent, une étape inévitable de prétraitement doit être établie.

Cette thèse traite du prétraitement de données protéiques comme une étape de préparation avant leur classification. Nous présentons l'extraction de motifs comme un moyen fiable pour répondre à cette tâche. Les motifs extraits sont utilisés comme descripteurs, en vue de coder les protéines en vecteurs d'attributs. Cela permet l'utilisation des classifieurs connus. Cependant, la conception d'un espace appropié d'attributs, n'est pas une tâche triviale.

Nous traitons deux types de données protéiques à savoir les séquences et les structures 3D. Dans le premier axe, *i.e.*, celui des séquences, nous proposons un nouveau procédé de codage qui utilise les matrices de substitution d'acides aminés pour définir la similarité entre les motifs lors de l'étape d'extraction. En utilisant certains classifieurs, nous montrons l'efficacité de notre approche en la comparant avec plusieurs autres méthodes de codage. Nous proposons également de nouvelles métriques pour étudier la robustesse de certaines de ces méthodes lors de la perturbation des données d'entrée. Ces métriques permettent de mesurer la capacité d'une méthode de révéler tout changement survenant dans les données d'entrée et également sa capacité à cibler les motifs intéressants. Le second axe est consacré aux structures protéiques 3D, qui ont été récemment considérées comme graphes d'acides aminés selon différentes représentations. Nous faisons un bref survol sur les représentations les plus utilisées et nous proposons une méthode naïve pour aider à la construction de graphes d'acides aminés. Nous montrons que certaines méthodes répandues présentent des faiblesses remarquables et ne reflètent pas vraiment la conformation réelle des protéines. Par ailleurs, nous nous intéressons à la découverte, des sous-structures récurrentes qui pourraient donner des indications fonctionnelles et structurelles. Nous proposons un nouvel algorithme pour trouver des motifs spatiaux dans les protéines. Ces motifs obéissent à un format défini sur la base d'une argumentation biologique. Nous comparons avec des motifs séquentiels et spatiaux de certains travaux reliés. Pour toutes nos contributions, les résultats expérimentaux confirment l'efficacité de nos méthodes pour représenter les séquences et les structures protéiques, dans des tâches de classification.

Les programmes developpés sont disponibles sur ma page web http://fc.isima.fr/~saidi.

Mots-clés: Prétraitement, extraction de motif, classification de proteins, structure protéique, motif séquentiel, motif spatial.

إِشتِقَاق المُمَيِّزَات مِن المُعطيَات المُعَقَّدَة: حَالَة تَصنِيف البرُوتِينَات

مُلَخَّص: يُعتَبَر التَصنِيف أَحَد أَهَمّ التَحَدِيَّات فِي مَجَال المَعلُومَاتِيَّة الحَيَوِيَّة، سَوَاء بِالنِسبَة لِلبرُوتِينَات أَو الأَحمَاض النَوَوِيَّة. وُجُود هَذِهِ المُعطَيَات بِكِمِيَّات كَبِيرَة، إِضَافَة إِلَى تَعقِيدِهَا وَ خَاصَّة التَّكلُفَة المَادِيَّة وَ الزَمَنِيَّة اَلبَاهِضَة لِتَحلِيلِهَا مَخبَرِيًا، تَجعَل مِن إِعتِمَاد التَنقِيب فِي المُعطِيَات ضَرُورَة قَبلَ أَن يَكُون اختِيَارًا عَقلَانيًا. إِلَّا أَنَّ تَقنِيَات التَنقِيب فِي المُعطَيَات غَالِبًا مَا تَقتَضِي أَن تَكُون البيَانَات فِي شَكل عَلَائِقِي، مِمَّا يَستَوجِب مُعَالجَة أَوَّلِيَّة لِلمُعطَيَات البيُولُجِيَّة. هَذِهِ الأَطرُوحَة تَتَطَرَّق لِلمُعَالَجَة الأَوَلِيَّة لِلبرُوتِينَات كَمَرحَلَة تَحضِرِيَّة قَبل تصنِيفِهَا. نَعرض فِيهَا اشتِقَاق المُمَيِّزَات كَطَرِيقَة نَاجِعَة لِتَأمِين هَذِه العَمَلِيَّة، حَيث تُستَعمَل المُمَيِّزَات المُستَخرَجَة لِتَوصِيف البرُوتِينَات وَ تَرمِيزِهَا فِي شَكل مُتَّجِهَات ذَات نَفس البُعد مِمَّا يُشَكِّل مَجَال عَلَائِقِي يُتِيح استِخدَام المُصَنِّفَات المَعرُوفَة. إِلَّا أَنَّ تَصمِيم مَجَال عَلَائِقِي مُلَائِم يُعتَبَر عَمَلِيَّة غَير هَيِّنَة.

نَتَعَامَل مَعَ نَوعَين مِن البرُوتِينَات، اَلسَلَاسِل و البُنيَات ثُلَاثِيَّة الأَبعَاد. فِي المِحوَر الأَوَّل المُتَعَلِّق بِالسَلَاسِل، نَقتَرِح طَرِيقَة جَدِيدَة لِاشتِقَاق المُمَيِّزَات التَسَلسُلِيَّة و اَلتَرمِيز قَائِمَة عَلَى استِعمَال مَصفُوفَات الاستِبدَال لِلاحمَاضِ النَوَوِيَّة قَصد تَحدِيد مَفهُوم الاستِبدَال بِالنِسبَة لِلمُمَيِّزَات. و قَد قُمنَا بِتبيَان نَجَاعَة مُقَارَبَتِنَا مِن خِلَال مُقَارَنَتِهَا بِمُقَارَبَات أُخرَى بِاستِخدَام بَعض المُصَنِّفَات. كَمَا نَقتَرِح مقَايِيس جَدِيدَة لِقِيَاس مَدِي مَتَانَة بَعض طُرُق اشتِقَاق المُمَيِّزَات فِي حَالَة تَشوِيش المُعطيَات المُدخَلَة. هَذِه المَقَايِيس تُمَكِن مِن اختِبَار مَدَى قُدرَة الطَرِيقَة عَلَى التَأَثُّر بِأَيِّ تَغيِيرَات طَارِئَة عَلَى المُعطِيَات المُدخَلَة و كَذَلِك قُدرَتِهَا عَلَى استِهدَاف المُمَيِّزَاتِ المُهِمَّة. اَلمِحوَر الثَانِي مُكَرَّس لِلبُنَى البرُوتِينِيَّة الَّتِي وَقَعَ فِي الآونَة الأَخِرَة تَمثِيلُهَا بِشَبَكَات لِلأَحمَاض الأَمِينِيَّة. نَقُوم بِإِجرَاء دِرَاسَة استِقصَائِيَّة مُوجَزَة حَول الطُرُقَ الأَكثَر استِعمَالًا لِتَمثِيل البرُوتِينَات فِي شَكل شَبَكَات و نَقتَرِح طَرِيقَة بَسِيطَة لِلمُسَاعَدَة عَلَى هَذَا الأَمر. كَمَا نُبَيِّن أَنَّ بَعض الطُرُق وَاسِعَة الانتِشَار تنطَوِي عَلَى نِقَاط ضَعف هَامَّةٍ و لَا تَعكِس بِالضَرُورَة التَشَكُّل المَجَالِي الحَقِيقِي لِلبرُوتِينَات. إِلَى جَانِب ذَالِك، نَهتَمّ بِاكتِشَاف الأَجزَاء المُتَكَرِّرَة فِي البرُوتِينَات الَّتِي قَد تُعطِي إِضَاءَات هَيكَلِيَّة و وَظِيفِيَّة هَامَّة. إِذ نَقتَرِح خَوَارِزمِيَّة جَدِيدَة لِاشتِقَاق اَلمُمَيِّزَات المَجَالِيَّة مِن البُنَى اَلبرُوتِينِيَّة. هَذِه المُمَيِّزَاتِ تَخضَع لِقَالَب مُحَدَّد وِفق تَعلِيل بِيُولُوجِي. مِن ثَمَّة، نُقَارِن مَعَ مُمَيِّزَات تَسلسُلِيَّة و مَجَالِيَّة لِأَعمَال أُخرَى مُشَابِهَة.

بِالنِسبَة لِمُختَلَف إِضَافَاتِنَا، تُؤَكِّد نَتَائِج الاختِبَارَات نَجَاعَة مُقتَرَحَاتِنَا لِتَمثِيل السَلَاسِل و البُنَى البرُوتِينِيَّة فِي عَمَلِيَّات التَصنِيف. اَلبَرمَجِيَّات المُصَمَّمَة خِلَال هَذَا العَمَل مُتَوَفِّرَة عَلَى صَفحَتِي الافتِرَاضِيَّة.

اَلكَلِمَات المَفَاتِيح: مُعَالجَة أَوَّلِيَة، إِشتِقَاق المُمَيِّزَات، تَصنِيف البرُوتِينَات، بُنيَة برُوتِينِيَّة، مُمَيِّزَة تَسلسُلِيَّة، مُمَيِّزَة مَجَالِيَّة.