Multiple instance learning for sequence data : Application on bacterial ionizing radiation resistance prediction

Manel Zoghlami

UNIVERSITY CLERMONT AUVERGNE

LIMOS - CNRS UMR 6158

UNIVERSITY OF TUNIS EL MANAR

LIPAH - LR11ES14

PHD THESIS

To obtain the degree of Doctor of Philosophy

in Computer Science

Defended by

Manel ZOGHLAMI

Multiple instance learning for sequence data: Application on bacterial ionizing radiation resistance prediction

publicly defended on: December 20, 2019

Committee:

Reviewers:	
Dr. Marie-Dominique DEVIGNES	LORIA, France
Dr. Faten CHAIEB	University of Carthage, Tunisia
Examiners:	
Dr. Jean SALLANTIN	LIRMM, France
Dr. Khedija AROUR	University of Carthage, Tunisia
Advisors:	
Pr. Engelbert Mephu Nguifo	University Clermont Auvergne, France
Dr. Amel Borgi	University of Tunis El Manar, Tunisia
Guests:	
Pr. Mondher Maddouri	University of Jeddah, KSA
Dr. Sabeur ARIDHI	University of Lorraine, France

Acknowledgments

I would like to express my gratitude to my advisors Pr. Engelbert Mephu Nguifo and Dr. Amel Borgi for their help, support and patience. My thanks go as well to my co-advisors Pr. Mondher Maddouri and Dr. Sabeur Aridhi for their advices and valuable guidance. I would like also to thank Pr. Haithem Sghaier for his help and advices in biology and bioinformatics.

Special thanks are directed to Dr. Marie-Dominique Devignes and Dr. Faten Chaieb for reviewing my thesis manuscript and for their kind efforts to comply with the administrative constraints. My thanks go as well to the examiners Dr. Jean SALLANTIN and Dr. Khedija AROUR.

I would also like to thank my family who is always supporting me and encouraging me.

Clermont-Ferrand, 2019
Manel Zoghlami

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 motivations of this work. Then, we state the addressed issues in this thesis.

1.1 Context and motivations

In a traditional setting of supervised learning task, the training set is composed of feature vectors (instances), where each feature vector has a label. In MIL task, we learn a classifier based on a training set of bags, where each bag contains multiple feature vectors and it is the bag that carries a label. We do not know the labels of the instances inside the bags.

This work was originally proposed to solve the problem of ionizing radiation resistance (IRR) prediction in bacteria [Zoghlami et al., 2019a,b, 2018a,b] [Aridhi et al., 2016]. Ionizing-radiation-resistant bacteria (IRRB) are important in biotechnology. In fact, they could be used for the treatment of radioactive wastes as well as the therapeutic industry [Brim et al., 2003] [Gabani and Singh, 2013]. Several *in vitro* works studied the causes of the high resistance of IRRB to ionizing radiation to determine peculiar features in their genomes and improve the treatment of radioactive wastes. Predicting if a bacterium belongs to IRRB using *in vitro* experiments is not an easy task, it requires a big effort and a time consuming lab work. In this thesis, we aim to use machine learning in order to perform the bacterial IRR prediction task . As far as we know, there is no bioinformatics tool that performs a such task in the literature. We propose an MIL formalization of the problem since each bacterium is represented by a set of protein sequences. Bacteria represent the bags and protein sequences represent the instances. In particular, each protein sequence may differ from a bacterium to another, e.g., each bag contains the protein named *Endonuclease III*, but it is expressed differently from one bag to another: these are called orthologous proteins [Fang et al., 2010].

To learn the label of an unknown bacterium, comparing a random couple of sequences makes no sense, it is rather better to compare the protein sequences that have a functional relationship/dependency: the orthologous proteins. Hence, this work deals with the MIL problem that has the following three criteria:

- **The instances inside the bags are sequences:** to deal with sequences, we have to deal with data representation. A widely used technique to represent MIL sequence data is to apply a preprocessing step which extracts features/motifs to represent the sequences [Sutskever et al., 2014] [Lesh et al., 1999] [She et al., 2003]. Other works keep data in their original format and use sequence comparison techniques such as defining a distance function to measure the similarity between pairs of sequences [Aridhi et al., 2016] [Saigo et al., 2004] [Xing et al., 2010].

- **All the instances inside a bag contribute to define the bag's label:** the standard MIL assumption states that every positive bag contains at least one positive instance while in every negative bag all of the instances are negative. Some methods following this assumption try to identify positive instances which are relevant to learn the label of a bag [Faria et al., 2017] [Li et al., 2014]. However, the collective assumption [Amores, 2013] considers that all the instances contribute to the bag' s label. This suits the problem of bacterial IRR prediction since all the protein sequences have to contribute to the final decision.

- **The instances may have dependencies across the bags:** one major assumption of most existing MIL methods is that each bag contains a set of instances that are independently distributed. Nevertheless, in many applications, the dependencies between instances naturally exist and if incorporated in the classification process, they can potentially improve the prediction performance significantly [Zhang et al., 2011]. Many real world applications such as bioinformatics, web mining, and text mining have to deal with sequence data. When the tackled problem can be formulated as an MIL problem, each instance of each bag may have structural and/or temporal relation with other instances in other bags. This is the case of the IRR prediction problem in which the bags contain orthologous protein sequences.

Considering this issue, the problem we want to solve in this work is the MIL problem in sequence data that have dependencies between instances of different bags.

1.2 Contributions

In this work, we present two novel MIL approaches for sequence data classification named *ABClass* (which stands for **A**cross **B**ag sequences **Class**ification) and *ABSim* (which stands for **A**cross **B**ag sequences **Sim**ilarity). ABClass is a motif-based approach while ABSim uses a similarity measure between related sequences. We applied both approaches to solve the problem of IRR prediction. The experimental results were satisfactory.

1.2.1 First axis: Motif-based MIL approach for sequence data with across-bag dependencies

As a first contribution, we propose a motif-based approach, named *ABClass*, which takes into account the across-bag relations between the sequences of different bags in the classification process. In a motif-based classification for sequential data, a sequence is transformed into a feature/motif vector. The feature extraction step is very important in the classification process. Many parameters have an impact in the classification results such as the motifs frequency and length, and the matching type between motifs. Feature-based approaches are widely adopted for genomic sequence classification. In ABClass, a preprocessing step is performed in order to extract motifs from each set of related sequences. These motifs are then used as attributes to construct a vector representation for each set of sequences. In order to compute partial prediction results, a discriminative classifier is applied to each sequence of the unknown bag and its correspondent related sequences in the learning dataset. Finally, an aggregation method is applied to generate the final result.

We created a multiple instance dataset composed of real sequence data used to test the approach. It consists of a set of bacteria where each bacterium is represented using a set of primary structures of proteins implicated in basal DNA repair in IRRB. Bacteria represent the bags and protein sequences represent the instances. The used across-bag relation is the orthology. Orthologous proteins are assumed to have the same biological functions in different species. The dataset is publicly available at http://homepages.loria.fr/SAridhi/software/MIL/ .

1.2.2 Second axis: Similarity-based MIL approach for sequence data with across-bag dependencies

As a second contribution, we propose the *ABSim* algorithm. It does not use motifs to represent data and no encoding step is needed. We use a similarity measure between each sequence of the unknown bag and the corresponding sequences in the learning bags in order to create a similarity score matrix. An aggregation method is applied and the unknown bag is labeled according to the bag that presents more similar sequences. We define two aggregation methods: Sum of Maximum Scores (SMS) and Weighted Average of Maximum Scores (WAMS). In the experimental study, we used the local alignment score to measure the similarity between two protein sequences.

1.3 Outline

The remainder of this document is organized as follows. Chapter 2 presents the bioinformatics field and gives a background about the processed data and the alignment of biological sequences. It also provides a description of the bacterial IRR prediction problem. Chapter 3 provides a background about MIL fundamental notions and gives an overview of some related works in MIL. It also gives a formalization of the problem of MIL in sequence data. In Chapter 4, we present an MIL naive approach for sequence data followed by a description of the AB-Class algorithm. We provide a simple use case that serves as a running example throughout the chapters 4 and 5. Then we describe our experimental environment and we discuss the obtained results. Chapter 5 describes the ABSim approach and the two proposed aggregation methods. Concluding points and a presentation of future work make the body of Chapter 6.

Part I

Background and related works

CHAPTER 2

Bioinformatics and Sequence classification

Contents

Goals In this chapter, we will present basic notions of a main search field in this thesis: bioinformatics. We present mainly the specificity of the biological data and we introduce the investigated IRR prediction problem. We present also the particularity of sequence classification in the data mining field and we focused on the alignment of biological sequences.

2.1 Bioinformatics background

2.1.1 Bioinformatics

Bioinformatics in an interdisciplinary field which can be simply defined by the use of computer science to deal with biological data. Developing software programs to produce meaningful biological information involves the use of algorithms from different disciplines such as data mining, graph theory, statistics, artificial intelligence and image processing.

The aims of bioinformatics involve mainely the collection and storage of data in a way that allows to access them efficiently and the development of algorithms and tools that deal with the analysis, prediction and interpretation of the data.

To date, the genomic databases indicate the presence of thousands of genome projects. It is not feasible to analyze the amount of collected data manually without using tools that make the task easier. It is impossible to experimentally annotate every biological molecule identified by sequencing projects. Bioinformatics has then evolved in the past few years in order to provide software applications that need minutes or even seconds to accomplish tasks that used to require a big effort and weeks of lab work. Computational approaches could be used to provide initial prediction results related to the function of a biological molecule and help to predict the usefullness of an experimental study scenario. Examples of bioinformatics research fields include the sequencing of genomes, the 3-D visualisation of molecules, the construction of evolutionary trees, the analyses of protein functions and the ionizing radiation resistance prediction (See Section 2.2).

Table 2.1: The 20 amino acids in a protein sequence.

Letter	Amino acid	Letter	Amino acid
A	Alanine	L	Leucine
R	Arginine	K	Lysine
N	Asparagine	M	Methionine
D	Aspartic acid	F	Phenylalanine
C	Cysteine	P	Proline
Q	Glutamine	S	Serine
E	Glutamic acid	T	Threonine
G	Glycine	W	Tryptophan
H	Histidine	Y	Tyrosine
I	Isoleucine	V	Valine

2.1.2 Biological data

Mainly, bioinformatics deals with three biological macromolecules named protein, DNA and RNA. The last two macromolecules are called nucleic acids.

- **Proteins** They are macromolecules responsible of a variety of functions within organisms such as DNA replication, and transporting molecules from one location to another. They are complex chains of molecules known as *amino acids* so they can be viewed as strings of an alphabet of the 20 amino acids provided in Table 2.1.

- **Nucleic acids.** Nucleic acids include DNA and RNA macromolecules.

  - **DNA** Deoxyribonucleic acid (shortly DNA) is known to be the molecule that carries the genetic instructions of organisms. It has a double helical twisted structure. Each side is made of four *bases* which are represented by the four letters A (adenine), C (cytosine), G (guanine) and T (thymine). A DNA could then be represented by a sequence of the alphabet $\{A,C,G,T\}$.

  - **RNA** Ribonucleic acid (shortly RNA) is a molecule very similar to DNA but has some chemical differences. It play various roles in coding, decoding, and expression of genes. The four bases are the same as in

DNA with thymine (T) replaced by uracyl (U). Then, an RNA molecule could be represented by a sequence of the alphabet $\{A, C, G, U\}$.

2.1.3 Proteins

2.1.3.1 Protein structures

There are four levels of protein structures as described in Figure 2.1.

- Primary structure: A primary structure represents a protein as a sequence of amino acids which attach to each other in long chains. The terms *protein* or *polypeptide* refers to sequences longer than 50 amino acids while sequences with fewer amino acids are called *peptides*.

- Secondary structure: The chain of amino acids can fold to form a three-dimensional structure. Two main types of secondary structure are the α-helixes and β-sheets.

- Tertiary structure: The secondary structures are folded to form the over-all shape of a protein, also known as the protein 3-D structure or the tertiary structure.

- Quaternary structure: Several proteins are composed of more than one sequence of amino acids. The combination of these sequences conform the quaternary structure.

2.1.3.2 Protein sequence data databases

With the evolution of sequencing technologies, the amount of biological sequence data has exponentially increased. Some publicly available databases offer to users the possibility to search and download protein sequence data.

- **GOLD database** The Genomes OnLine Database (GOLD) [Mukherjee et al., 2016] provides a comprehensive information regarding genome and metagenome sequencing projects with their associated metadata. Data are

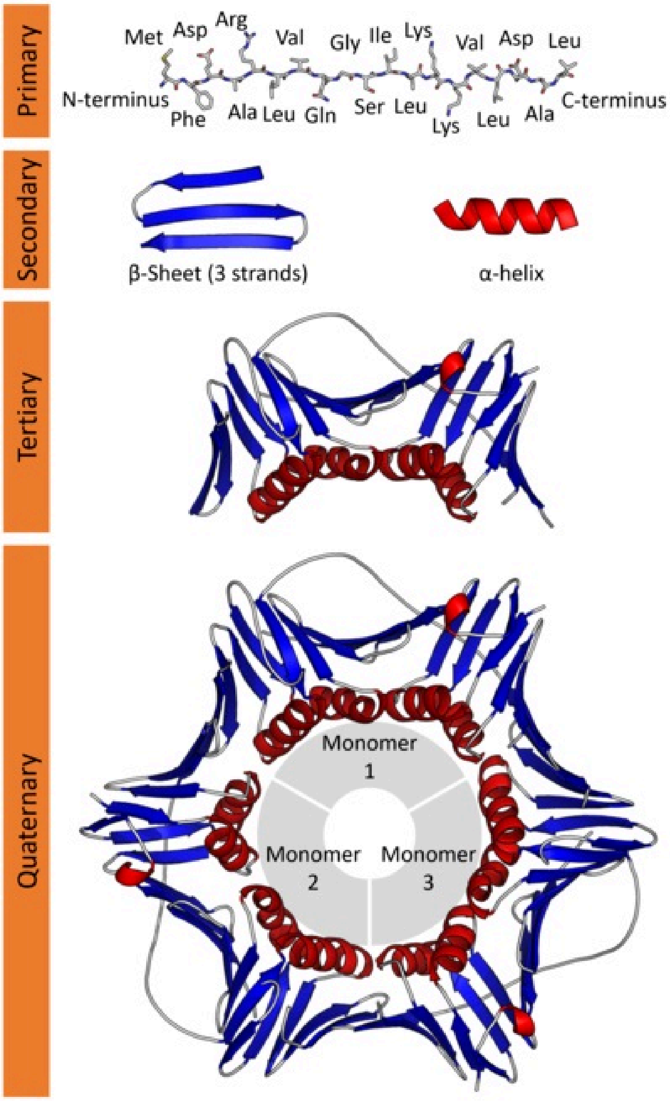


Figure 2.1: The four levels of the protein structure [1].

imported from three main sources: (1) projects deposited by users which are regularly monitored for data accuracy and consistency, (2) projects imported from public resources like BioProject database [Federhen et al.,

[1] https://en.wikipedia.org/wiki/File:Protein_structure_(full).png, November 2019.

2014] and (3) projects sequenced at the Joint Genome Institute (JGI) [2]. The latest publication reported 97 212 Sequencing Projects. GOLD is available at https://gold.jgi.doe.gov/.

- **UniProt** The **Uni**versal **Prot**ein resource (UniProt) is a biological repository of protein sequences and their functional information [Apweiler et al., 2004]. It contains four databases: Swiss-Prot and TrEMBL which are subparts of UniProtKB, UniParc and UniRef.

  SwissProt contains non-redundant, manually annotated protein sequences [Boutet et al., 2016]. In order to perform the annotations, information extracted from biological literature are combined with computational analysis evaluated by biocurator. The goal is to provide relevant known information related to proteins available in the database. Figure 2.2 shows the increasing size of SwissProt database over thirty years. The amount of available protein sequences was doubled during three years from 2007 to 2010. **TrEMBL** is a database that contains automatically annotated protein sequences [Gane et al., 2014]. In fact, the large amount of data generated by genome projects could not be manually analysed and annotated according to the process of UniProtKB/SwissProt. Thus, data are automatically processed and added to the TrEMBL database. **UniParc** (for UniProt Archive) [Leinonen et al., 2004] contains non-redundant protein sequences from the main publicly available databases. **UniRef** (for UniProt Reference Clusters) [Suzek et al., 2007] contains clustered protein sequences from SwissProt, TrEMBL and selected UniParc entries.

- **GenBank and RefSeq** The National Centre for Biotechnology Information (NCBI) [3] hosts two sequence databases named GenBank [Benson et al., 2012] and RefSeq [Pruitt et al., 2011]. GenBank and RefSeq provide an annotated collection of publicly available nucleotide and protein sequences, while UniProt contains only protein sequence data, Un-

[2] https://jgi.doe.gov

[3] https://www.ncbi.nlm.nih.gov

like GenBank sequences, RefSeq ones are non-redundant, curated and limited to some organisms for which sufficient data are available. GenBank contains sequences for any submitted organism. Refseq is available at https://www.ncbi.nlm.nih.gov/refseq/ and GenBank is available at https://www.ncbi.nlm.nih.gov/genbank/.

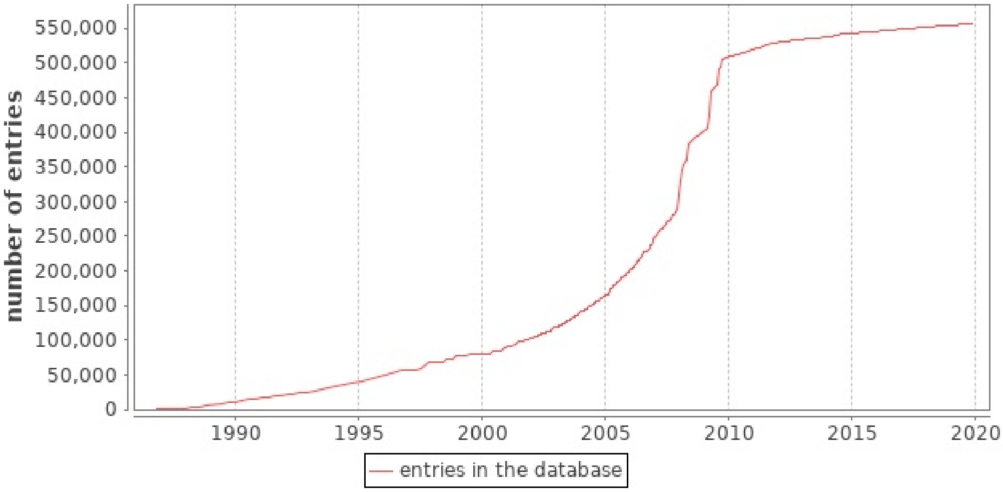


Figure 2.2: Number of entries of SwissProt database over time [4].

2.1.3.3 Protein signatures

Protein signatures consist of models which describe protein families, domains or sites. A protein family is a group of proteins that share the same evolutionary origin. Proteins in a same family have similar sequences/structures and biological functions. Families are usually hierarchically organized. A domain is a part of a protein which is able to evolve, function, and exist independently of the rest of the protein sequence/structure. From sequence perspective, a protein domain is a subsequence of amino acids. Domains vary in length from about 25 amino acids to 500 amino acids. They also vary in biological functions. The average size of protein domains is 150 amino acids. The concept of protein domains

[4] https://www.uniprot.org/statistics/Swiss-Prot, November 2019

and families are applicable to both sequences and structural proteins. Several proteins are multi-domain. Figure 2.3 shows a visualization of the three domains of the protein *Pyruvate kinase*, each domain has a different color. The ordered arrangement of domains in a protein, called the protein domain organization or the protein domain architecture, is important to maintain the function and the structure of the protein.

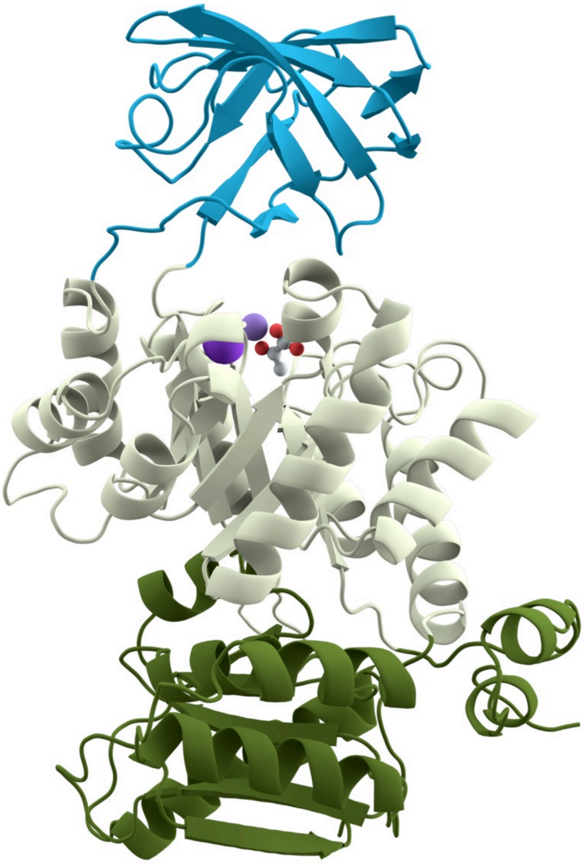


Figure 2.3: A visualization of the three domains of the protein Pyruvate kinase [6].

Signature could be simple such as patterns or more complex such as Hidden Markov Models (HMMs). Signature methods are divided into patterns, profiles, fingerprints and HMMs. Conserved subsequences, also known as **motifs**, are

[6] https://commons.wikimedia.org/wiki/File:Pyruvate_kinase_protein_domains.png, November 2019

extracted and then used to build regular expressions that serve as patterns. Profiles are computed by converting multiple sequence alignments into position-specific scoring systems (PSSMs), i.e., assigning a score to amino acids at each position according to the frequency with which they occur in the alignment. Fingerprints are created using multiple profiles generated using multiple alignment techniques. The main advantage of fingerprints is in identifying the differences in protein sequences at four levels of clan, superfamily, family and subfamily which helps to make a more accurate functional predictions for unknown sequences. HMMs are statistical models that, like profiles, convert multiple sequence alignments into PSSMs and represent amino acid insertions and deletions. Its can model the entire alignment, including divergent regions.

Figure 2.4 shows a list of well known protein domain databases grouped based on the used protein signatures. Domain databases are described below.

- Prosite provides entries that describe protein domains and families, and related patterns and profiles used to identify them. It contains documentation about signatures and the structure and function of proteins. Figure 2.4 differentiates between Prosite entries based on patterns (in orange) and those based on profiles (in green). The database is available at http://prosite.expasy.org/.

- Prints is a database of fingerprints [Attwood et al., 2003] which contains an annotation list for protein families and a diagnostic tool for newly discovered protein sequences. The database is accessible at http://www.bioinf.man.ac.uk/dbbrowser/PRINTS/.

- CDD [Marchler-Bauer et al., 2005] [Marchler-Bauer et al., 2014] is the Conserved Domain Database for the functional annotation of proteins. It includes manually curated domain models from NCBI (National Center for Biotechnology Information in) and other domain models imported from a set of external databases such as Pfam, and TIGRFAMs. In order to generate NCBI-curated domains, 3D-structure information is used to characterize domains and relationship between into sequences and related structure and function. CDD is accessible at http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml.

- Pfam is a database of protein domains and families represented by multiple sequence alignments and hidden Markov models (HMMs) [Bateman et al., 2004] [Finn et al., 2015]. It has a large coverage of proteins and a realistic way of naming domains. It provides two subsets data depending on the quality of the families: Pfam-A and Pfam-B. Pfam-A provides manually curated families with high quality alignments and well-characterized protein domains. Pfam-B contains a lower quality data where families are automatically generated.

- TIGRFAMs [Haft et al., 2003] [Haft et al., 2012] is a database of protein families that supports manual and automated curated genome annotation. It includes multiple sequence alignments and a corresponding HMM generated from the alignment. If the score of a sequence exceeds a defined threshold of a given TIGRFAMs HMM, the protein sequence is assigned to the related protein family. TIGRFAMs is available at `http://www.jcvi.org/cgi-bin/tigrfams/index.cgi`.

- Panther (for Protein ANalysis THrough Evolutionary Relationships) [Thomas et al., 2003] [Mi et al., 2016] is a large collection of protein families manually subdivided into functionally related subfamilies. A phylogenetic tree is built for each family and could be used in order to classify an uncharacterized protein sequence. Each node in the tree is annotated with heritable attributes that are propagated to a decedent node. A protein is then annotated according to its ancestor in the phylogenetic tree. Panther database is available via `http://pantherdb.org/`.

- SMART (Simple Modular Architecture Research Tool) [Schultz et al., 1998] [Letunic et al., 2011] is a database that provides the identification of domains and the analysis of their architectures. It uses HMMs built from multiple sequence alignments in order to identify protein domains. SMART data was used to create the CDD database.

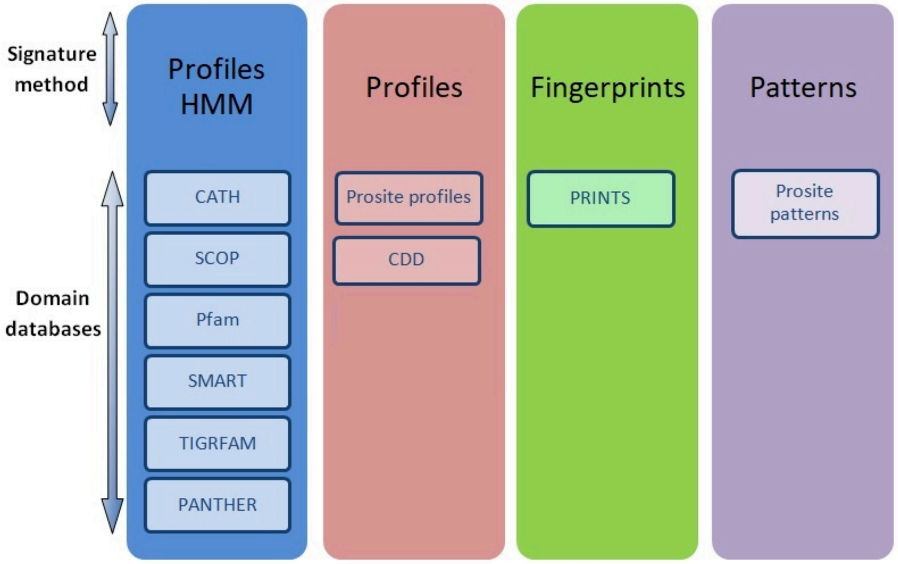


Figure 2.4: An overview of protein domains databases [Alborzi, 2018].

- CATH [Orengo et al., 1997] [Pearl et al., 2003] is a database of curated classification of protein domain structures [Orengo et al., 1997, Pearl et al., 2003]. In order to perform this classification, a combination of multiple procedures is used including literature review, expert analysis, computational algorithms and statistical analysis. It shares many features with the SCOP resource, however they may differ greatly in detailed classification. CATH database is available at `http://www.cathdb.info/`.

- SCOP (Structural Classification of Proteins) database [Murzin et al., 1995] is a classification of structural domains of the proteins based on their evolutionary and structural relationships. The goal is to provide a comprehensive and detailed description of the relationships between all proteins having known 3D structures. SCOP database is available at `http://scop.mrc-lmb.cam.ac.uk/scop/`. It stopped updating in 2010 and a successor named SCOP2 [Andreeva et al., 2013] has been proposed. SCOP2 is available at `http://scop2.mrc-lmb.cam.ac.uk/`.

InterPro All domains classifications in Figure 2.4 are integrated into the InterPro database [Apweiler et al., 2001] [Finn et al., 2016]. In fact, InterPro is a composite database combining the information of many databases of protein domains. The goal is to rationalise protein sequence analysis by combining information from different resources in a consistent manner, removing redundancy, and adding rich annotation about the proteins and their signatures. Features found in known proteins are applied to unknown ones (such as new sequenced proteins) in order to characterise their functions. It contains signatures and the proteins that they significantly match. InterproScan is a tool used to search a query against the diverse databases of protein domains, motifs, signatures and families. The disadvantage is the runtime since the Interproscan webservice can be very slow if we need to analyse thousands of proteins. A solution is to download and install the whole suite locally.

2.2 The bacterial ionizing radiation resistance problem

Bacteria are small single-cell organisms. Most bacteria are helpful for mankind, but some are harmful. Few species cause disease. In particular, ionizing-radiation-resistant bacteria (IRRB) are important in biotechnology. They could be used for the treatment of mixed radioactive wastes by developing a strain to detoxify both mercury and toluene [Brim et al., 2000]. These organisms are also being engineered for *in situ* bioremediation of radioactive wastes[Brim et al., 2003]. In [Gabani and Singh, 2013], the authors discuss the potential uses of radiation-resistant extremophiles (e.g. micro-organisms with the ability to survive in extreme environmental conditions) in biotechnology and the therapeutic industry.

Several *in vitro* and *in silico* works studied the causes of the high resistance of IRRB to ionizing radiation to determine peculiar features in their genomes and improve the treatment of radioactive wastes. However, limited computational works are provided for the prediction of bacterial IRR [Aridhi et al., 2016] [Sghaier et al., 2008][Makarova et al., 2007]. In this thesis, we aim to develop a machine learning algorithm which predicts whether an unlabelled bacterium belongs to

IRRB or IRSB. Each bacterium is represented using a set of protein sequences implicated in basal DNA repair (see Figure 2.5).

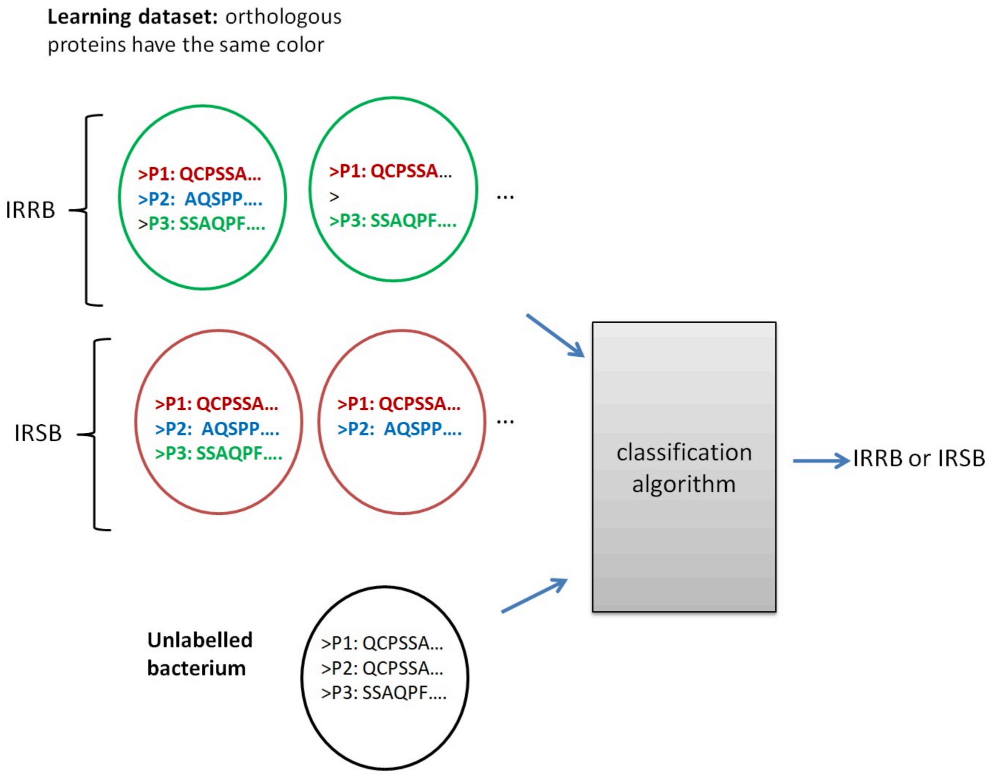


Figure 2.5: An illustration off the IRR prediction problem.

2.3 Sequence Classification

2.3.1 Definition of a sequence

A **sequence** is an ordered list of events. An event can be represented as a symbolic value, a numerical value, a vector of values or a complex data type [Xing et al., 2010]. There are many types of sequences including symbolic sequences, simple time series and multivariate time series [Xing et al., 2010]. In our work, we are interested in symbolic sequences since the protein sequences are described using symbols (amino acids). We denote Σ an *alphabet* defined as a finite set of characters or symbols. A simple symbolic sequence is defined as an ordered list of symbols from Σ.

2.3.2 Sequence classification approaches in machine learning

Existing sequence classification approaches can be divided into three large categories [Xing et al., 2010]: feature-based classification, distance-based classification and model-based classification.

In feature-based classification, a sequence is transformed into a feature vector. This representation scheme could lead to very high-dimensional feature spaces. The feature extraction step is very important since it would impact the classification results. This step should deal with many parameters such as the criteria used for selecting features (e.g. frequency and length) and the matching type (i.e. exact or inexact with gaps). After adapting the input data format, a conventional classification method is applied. Feature-based approaches are widely adopted for genomic sequence classification [Blekas et al., 2005] [She et al., 2003] [Chuzhanova et al., 1998].

In distance-based classification, a similarity function should be defined to measure the similarity between a pair of sequences. Then an existing classification method could be used such as the Support Vector Machine (SVM) or the K-Nearest Neighbors (KNN) algorithm. The similarity function determines the quality of the classification significantly. In bioinformatics, alignment based distances are popularly adopted to deal with sequences such as protein sequences and DNA sequences. Section 2.4 provides an overview on biological sequences alignment.

Model-based classification methods define a classification model based on the probability distribution of the sequences over the different classes. This model is then used to classify unknown sequences. Naive Bayes is a simple model-based classifier that makes the assumption that the features of the sequences are independent. In [Cheng et al., 2005], the authors apply Decision Tree and Naïve Bayes classifiers on a protein classification problem. Markov Model and

Hidden Markov Model (HMM) could be used in order to model the dependencies among sequences. In [Yakhnenko et al., 2005], a k-order Markov model is used to classify protein sequences and text data. HMM and alignment scores are used in [Srivastava et al., 2007] in order to make a genomic sequences classification. A protocol named HMM-ModE is defined in order to generate family specific HMMs.

Hierarchical clustering is also commonly used in genomic sequences/organisms classification [Ni et al., 2018] [Pagnuco et al., 2017] [Lukjancenko et al., 2010]. It groups the samples into groups called clusters. In the clustering process, inter-cluster distances should be maximized and intra-cluster distances should be minimized. Hierarchical clustering produces a nested series of clusters which may be represented in a tree structure, called a dendrogram, which may facilitate the interpretation of the classification results. In order to create the clusters, the genomic sequences are compared. Although the sequence alignment score is commonly used to make the comparison, some hierarchical clustering algorithms use alignment-free comparison methods [Ni et al., 2018] [Wei et al., 2012].

2.4 Aligning biological sequences: basic notions

2.4.1 What is the alignment of biological sequences

The sequence alignment problem is one of the cornerstones of computational biology. Sequence alignment is a way of arranging sequences in order to identify regions of similarity. This similarity could provide a structural, functional or evolutionary significance. The majority of biological sequence comparison methods rely on first aligning sequences and computing a score for the alignment [Vinga and Almeida, 2003].

As stated, the goal is to line up two (or more) sequences in order to maximise their degree of similarity. Identical bases are matched In the case of DNA and RNA. For proteins, amino acids are matched if they are identical. An amino acid could be replaced by another one on the basis of a substitution matrix.

Some genomic sequences comparison problems are not simply resolved using one or two alignment tool. In [Gracy and Argos, 1998], local similarity search is coupled to multiple sequence alignment in order to classify an entire protein sequence database. Additional contextual information could be integrated in order to improve the genomic sequences comparison. Domain co-occurrence is a powerful feature of proteins which can be used in this context [Menichelli et al., 2018].

2.4.1.1 Gaps

When the sequences do not align well with each other, a *gap* could be inserted into any of the sequences by pushing a letter one index. The goal is to obtain a better alignment. A gap is marked by the symbol ´-´. The biological interpretation of using a gap is that a mutation (a deletion or an insertion) occurred during the evolution of a sequence.

Example of an alignment using the two sequences TACCAGT and CCCG-TAA

$$\begin{array}{ccccccc|ccccccccc}
\multicolumn{7}{c|}{\text{No gaps}} & \multicolumn{9}{c}{\text{Gaps}} \\
T & A & C & C & A & G & T & T & A & C & C & A & G & T & - & - \\
C & C & C & G & T & A & A & C & - & C & C & - & G & T & A & A
\end{array}$$

We note that other alignments are possible, an option is listed below.

$$\begin{array}{ccccccccc}
T & A & C & C & A & G & T & - & - \\
- & C & C & C & G & T & A & A & -
\end{array}$$

2.4.1.2 Alignment scoring

As different alignments are possible, we can use a scoring function in order to select the best alignment. Gap penalty functions are used in order to compute an alignment score based on the number and length of gaps. The idea is that inserting too many gaps can lead to a meaningless alignment, so we need to minimize the number of gaps. Some gap penalty functions are listed below.

- **Constant gap penalty** It is a simple scoring function. A fixed negative cost is assigned to every gap, regardless of its length.
- **Linear gap penalty** A fixed negative score is assigned to every inserted or deleted symbol. The penalty is then directly proportional to the length of the gap.
- **Affine gap penalty.** It is a widely used scoring function. Different scores are assigned to the extension of a current gap and the starting of a new one.

If we perform an alignment of protein sequences, substitution matrices could be used in the scoring alignment instead of using fixed scores. In fact, some amino acids have similar structures and can be substituted in nature. Mutations of amino acids are quantified in the substitution matrices Two well-known matrices are PAM [Dayhoff et al., 1978] and BLOSUM [Henikoff and Henikoff, 1992].

2.4.2 Global alignment and local alignment

In pairwise alignment, only two sequences are involved in the alignment process, otherwise, it is a multiple sequence alignment. Alignment technics could be divided into two types based on the completeness:

- **global alignment** which attempts to match the sequences to each other from end to end. It is suitable for similar and equal length sequences.
- **local alignment** which searches for highly similar regions of the two sequences. It is more suitable for sequences which are partially similar and/or have different length. It is then useful for comparing sequences that share a common conserved pattern (motif) but differ elsewhere.

Several sequence alignment approaches have been proposed. Some algorithms use dynamic programming and provide optimal alignments such as the Needleman-Wunsch algorithm [Needleman and Wunsch, 1970] and The Smith-Waterman [Waterman, 1981] algorithm. Other alignment methods are based on heuristics such as BLAST, the widely used alignment tool in bioinformatics.

2.4.2.1 Dynamic programming based alignment

Dynamic programming is originally used in the field of mathematical optimization [Sniedovich, 2010]. In computer science, dynamic programming is the approach based on dividing a problem into smaller subproblems. Each of the subproblems is divided further into subproblems until some basic case is reached. Needleman-Wunsch algorithm [Needleman and Wunsch, 1970] and Smith-Waterman algorithm are based on Dynamic Programming. The first one is a classical global alignment algorithm while the second one performs a local alignment. Both approaches produce an optimal alignment based on a scoring matrix. A gap penalty could be used during the alignment process.

2.4.2.2 Heuristic based alignment

Heuristic approaches are much faster than dynamic programming ones, but they may overlook optimal alignments. They are widely used in large-scale database searches. BLAST [Altschul et al., 1990] (stands for Basic Local Alignment Search Tool) is a well-known alignment tool. It performs local alignment, i.e., it does not enforce the alignments on full length to measure the similarity between two sequences. BLAST requires a query sequence to search for, and a target sequence to search against or a sequence database containing multiple target sequences. The algorithm splits the query sequence into small subsequences and scans the database for word matches. All matches are then extended in both directions as far as possible in order to seek high-scoring alignments. Many extensions of BLAST have been proposed such as PSI-BLAST [Altschul et al., 1997] and BLAT [Kent, 2002] [Bhagwat et al., 2012]. The main idea of BLAST-like methods is to identify short common subsequences between the sequences, and then expand the matching regions.

2.5 Conclusion

In this chapter, we introduced basic notions the bioinformatics research field. We presented the biologial data sequences and we introduced the bacterial IRR prediction problem that we aim to investigate in this work. We focused on the alignment of biological sequences.

Chapter 3

Multiple instance learning

Contents

Goals This chapter introduces the MIL and its paradigms. It is mainly dedicated to present, in a simplified way, the basic notions related to MIL. We mainly focus on presenting MIL paradigms and describing some approaches.

3.1 Multiple instance learning

3.1.1 Multiple instance learning VS standard supervised learning

The standard supervised learning task deals with data that consist of a set of objects/examples, where each object is associated with a label. The learning dataset contains n labeled object $DB = \{(x_i, y_i), i = 1, \ldots, n\}$ where x_i is a seen example and y_i is the label that indicates the category that the object x_i belongs to (see Figure 3.1) . An MIL task deals with data that consist of a set of n **bags** where each bag is an unordered set of examples (see Figure 3.1). In an MIL context, each example is called an **instance**. MIL can be seen as a variant of supervised learning. However, labels are assigned to bags rather than individual instances. This category of learning is considered as *weakly supervised* since we do not know the label of each instance inside the bag, and only bags carry the labels. In this thesis, we only consider two-class classification problems, so the label of each bag is either 1 for a positive bag or -1 for a negative one.

3.1.2 Problem formulation

Let DB be a learning database that contains a set of n labeled bags $DB = \{(B_i, Y_i), i = 1, 2 \ldots, n\}$ where $Y_i = \{-1, 1\}$ is the label of the bag B_i. Instances in B_i are denoted by B_{ij}. Formally $B_i = \{B_{ij}, j = 1, 2 \ldots, m_{Bi}\}$, where m_{Bi} is the total number of instances in the bag B_i. We note that the bags do not contain the same number of instances. The goal is to learn a multiple instance classifier from DB. Given a query bag $Q = \{Q_k, k = 1, 2 \ldots, q\}$, where q is the total number of instances in Q, the classifier should use data in this bag and in each bag of DB in order to predict the label of Q.

3.1.3 Applications

MIL has many real word applications including the drug activity problem, the image categorization and the text categorization.

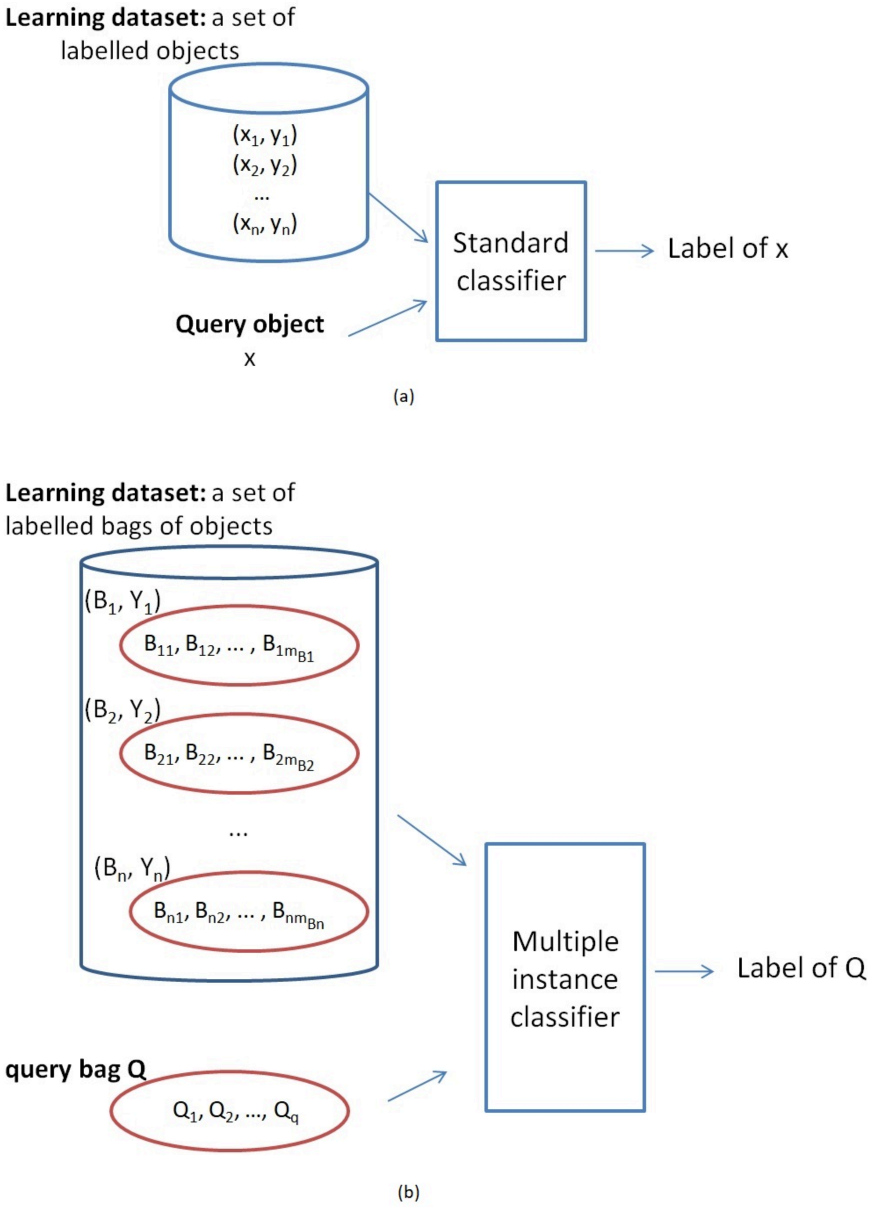

Learning dataset: a set of labelled objects
(x_1, y_1)
(x_2, y_2)
...
(x_n, y_n)
Standard classifier
Label of x
Query object
x
(a)
Learning dataset: a set of labelled bags of objects
(B_1, Y_1)
$B_{11}, B_{12}, \dots, B_{1m_{B1}}$
(B_2, Y_2)
$B_{21}, B_{22}, \dots, B_{2m_{B2}}$
...
(B_n, Y_n)
$B_{n1}, B_{n2}, \dots, B_{nm_{Bn}}$
Multiple instance classifier
Label of Q
query bag Q
$Q_1, Q_2, \dots, Q_q$
(b)

Figure 3.1: Standard supervised classification (a) vs multiple instance classification (b).

- **Drug activity** The original application for MIL is the drug activity prediction problem described in [Dietterich et al., 1997]. It deals with the first MI dataset known as the *musk dataset* which contains molecules occurring in different conformations. One of the conformations determines if a molecule belongs to either "musk" class or "non-musk" one. In fact, if a molecule is able to bind strongly to a binding site on the target molecule, it is classified as a good drug. The molecule is a bag and its conformations are the instances inside this bag. The musky smell is the positive label. We do not know which conformations bind well on a target molecule so we have no idea which instances are positive.

- **Image categorization** When applying MIL to the image categorization problem, an image is considered as a bag and its subimages are considered as instances that conform the bag. A processed image is then affiliated into one class or another. Several works use MIL in image categorization. In [Maron and Ratan, 1998], authors treat the natural scene images as bags. A bag is classified as a scene of waterfall if at least one of its subimages is a waterfall. In [Andrews et al., 2003] , the positive images show an animal (a fox, a tiger or an elephant), the negative images are selected randomly from other classes (the classes represent more than these three animals). An other image categorization problem defines a bag as an eye fundus image and an instance as a patch [Kandemir and Hamprecht, 2015]. The goal is to predict whether an image is of a subject with diabetes (positive) or a healthy subject (negative).

- **Text categorization** When dealing with a document categorization problem using an MIL setting, a document is considered as a bag, and its paragraphes are considered as instances. In [Ray and Craven, 2005], authors study a problem of biomedical text categorization. The goal was to predict whether a text should be annotated as relevant for a particular protein. A bag is a biomedical text and instances are paragraphs in the document. The newsgroup dataset [Zhou et al., 2009] is a popular text categorization MI dataset. The goal is to categorize collections of posts from different newsgroups corpus. A bag is a collection of posts (instances). A positive bag for a category contains 3% of posts about a topic while negative bags contain only posts about other topics.

3.2 Background

3.2.1 MIL assumptions

The standard MIL assumption states that a bag is positive if and only if one or more of its instances are positive while in every negative bag all of the instances are negative. This assumption is used in many MIL problems such as traditional problem of *musk* drug activity described in Section 3.1.3. A molecule is classified according to its conformations. If one on more conformations bind well to the target site, then the molecule belongs to the positive class.

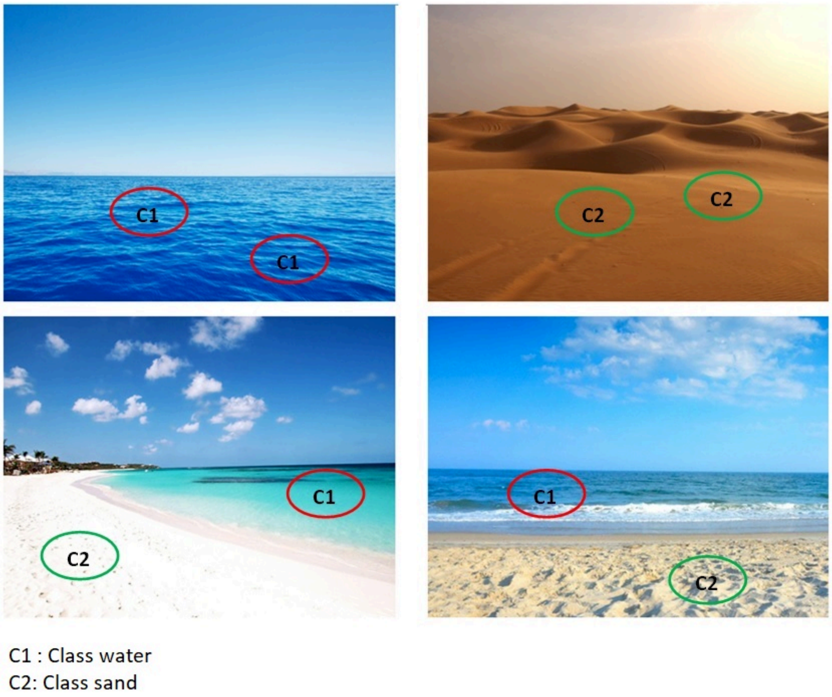


Figure 3.2: A classification problem of images into "beach" (bottom) and "non-beach" (top).

The standard assumption is not suitable for some MI problems. For example,

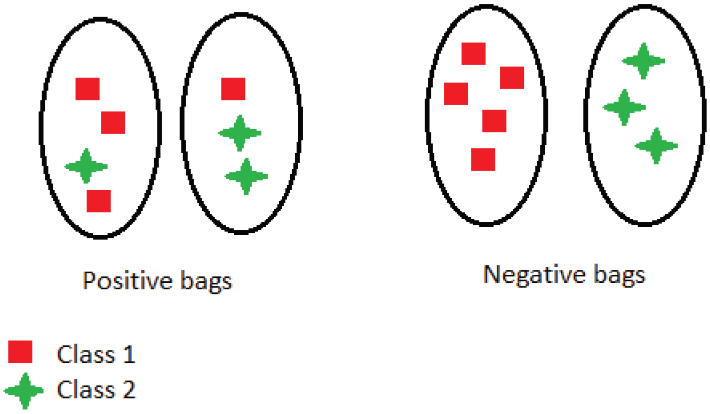


Figure 3.3: Illustrative example of an MIL problem where standard assumption should not be adopted.

let us consider the image classification problem provided by Figure 3.2. An image (bag) is segmented into small patches (instances). The goal is to label an image as "beach" (positive label) or "no beach" (negative label) image. Some images contain only sand and other ones contain only water (sea). These are negative bags. In order to obtain a beach image, we need to have both a sand segment and a sea segment. The figure 3.3 represents an illustrative example of an MIL problem where standard assumption should not be adopted. To make analogy with the above image recognition problem, *class 1* could be the sand and *class 2* could be the sea. Since the standard assumption is not guaranteed to hold in some domains, it can be relaxed to address problems where positive bags cannot be identified by a single instance. Other characteristics could be used to perform the classification such as the distribution of the instances that conform the bag or the interaction between them. Alternative assumptions are then considered [Foulds and Frank, 2010] [Amores, 2013] such as the presence-based MI Assumption, the collective assumption and the weighted collective MI assumption.

- The presence-based MI assumption states that a bag is positive if and only if there exist one or more instances in the bag that that belong to a set of required instance-level concepts. This is suitable for the previously presented image categorization problem. If there is just one required concept, we will have the standard MI assumption witch is a special case of the presence-based MI assumption.

- The threshold-based MI Assumption requires that, in order to consider a bag as positive, a certain number of instances in the bag have to belong to each of the required concepts.
- The count-based MI Assumption is close to the previous assumption but it requires that a maximum and a minimum number of instances have to belong to each of the required concepts.
- The collective assumption supposes that all instances in a bag contribute equally to the bag's label [Foulds and Frank, 2010]. All instances are considered in the learning process.
- The weighted collective MI assumption is an extension of the previous assumption that uses different weights for each instance.

We note that many MI approaches do not use the standard assumption but it is not always stated which new assumption is adopted instead.

3.2.2 Instance-level and bag-level learning

MIL methods could be categorized according to how the information contained in the MIL data is exploited. In [Amores, 2013], the author proposed to differentiate between the Instance-Space (IS) paradigm and the Bag-Space (BS) paradigm. A third category of MIL approaches based on the Embedded-Space (ES) paradigm was proposed. In this section, a lower-case notation will be adopted to refer instances (x) and instance-level classifiers (f), an upper-case notation is used to denote bags (X) and bag-level classifiers (F).

- **Instance level** The IS paradigm is based on local instance-level information since we consider the characteristics of individual instances in the learning process without looking at more global characteristics of the whole bag. Figure 3.4 illustrates the IS paradigm. A discriminative instance level classifier $f(x)$ is trained on the instances in order to separate instances of positive bags ($f(x)=1$) from instances in negative bags ($f(x)=0$). A bag level classifier $F(X)$ is then obtained by applying an aggregation on instance level results. Diverse Density and MISVM are two examples of algorithms which use the IS paradigm (see Section 3.3).

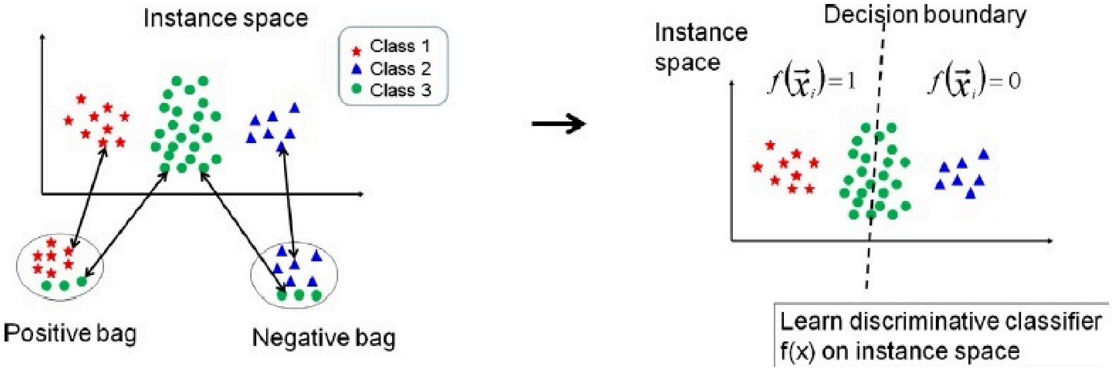


Figure 3.4: Illustrative example using the IS paradigm [Amores, 2013].

- **Bag level** In the BS paradigm each bag is treated as a whole entity. Instead of aggregating instance-level decisions, a global bag-level information is used to make the discriminative decision. Figure 3.5 provides an illustrative example using the BS paradigm. In the training step, a distance function is defined to compare two bags. Then, a learning algorithm is applied to create a model. In order to predict its label, a new bag is compared to other bags of the training set using the bag level distance function. A classifier F uses the computed distances, the model and the learned parameters Θ to make the prediction. Citation-Knn is an example of algorithms which use the BS paradigm.

- **Embedded level** In the ES paradigm, the relevant information about each bag is summarized in a single feature vector. The difference between BS and ES paradigms lies in the way this bag-level information is extracted: it is done implicitly in the BS paradigm and explicitly in the ES one through the definition of a mapping function. An illustration of using ES learning is

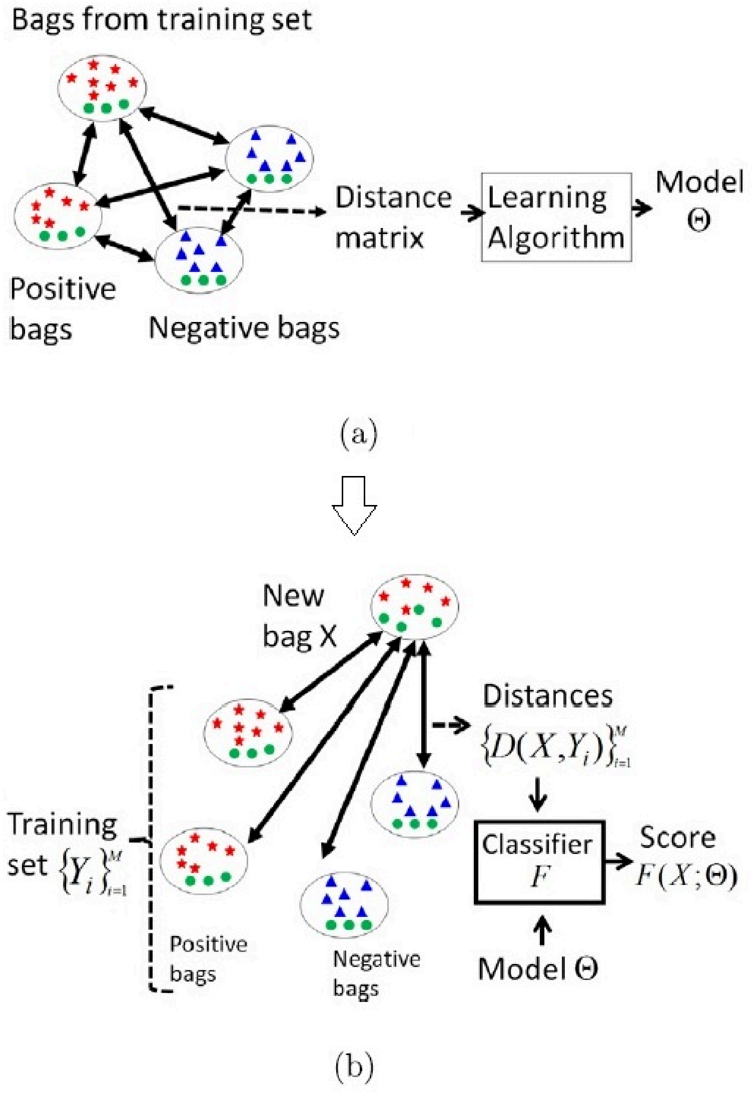
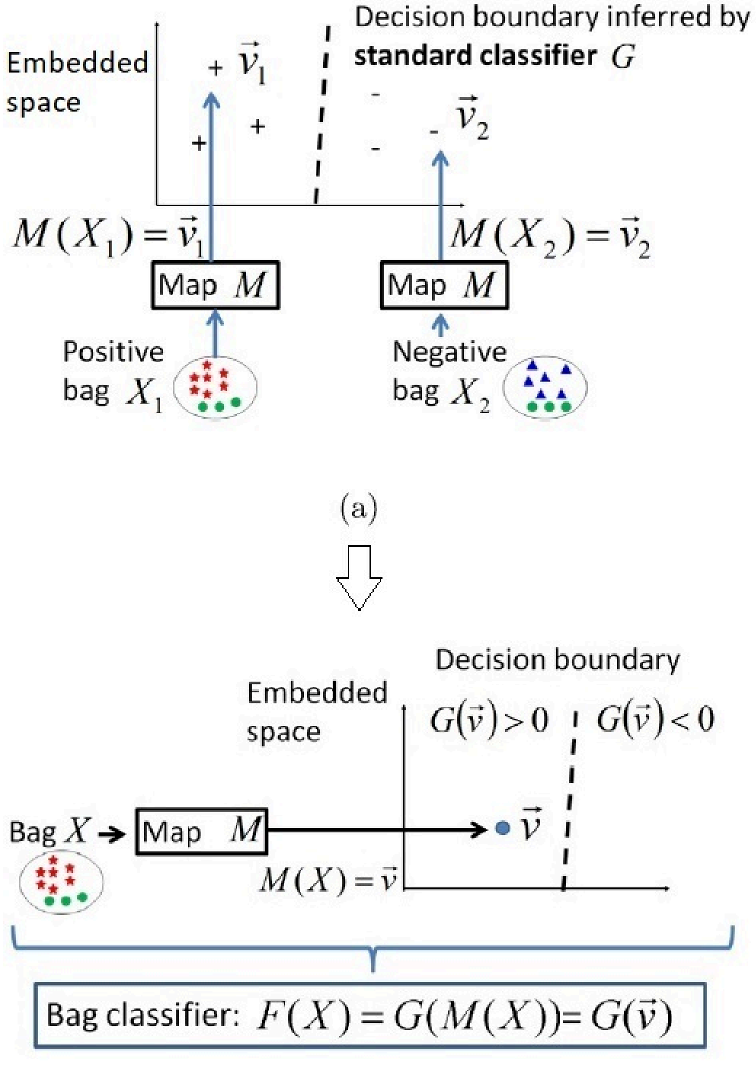


Figure 3.5: Illustrative example using the BS paradigm [Amores, 2013]: training (a) and test (b)

provided by the Figure 3.6. In the training step, the original training space is mapped to a vectorial embedded space by defining a mapping function M which associates a feature vector to each bag. A standard discriminant classifier G is then learned. In order to predict the class of a new bag X, the mapping M is used to generate the correspondent feature vector $\vec{v}$. The bag classifier $F(x)$ is the obtained using the discriminant classifier G and the new vector v. It can be expressed as $F(X) = G(\vec{v})$. A simple algorithm that uses the ES paradigm is SimpleMI described is Section 3.3.



(a)

(b)

Figure 3.6: Illustrative example using the ES paradigm [Amores, 2013]: training (a) and test (b)

3.3 An overview of MIL methods

The original work that introduces the MIL problem proposes the axis-parallel hyper-rectangle (APR) approach [Dietterich et al., 1997]. It tries to identify an hyper-rectangle that includes at least one instance of every positive bag and does not include any instances from negative bags. Many MIL approaches are then proposed. Diverse Density (DD) [Maron and Lozano-Pérez, 1998] is one of the popular MIL algorithms. It was proposed as a general framework for solving MIL problems. Several MIL approaches have been proposed. Some algorithms deal with the MIL problem directly in either instance level such as mi-SVM [Andrews et al., 2003] and MILKDE [Faria et al., 2017] or in bag level such as MI-SVM mi-SVM [Andrews et al., 2003] and MIGraph [Zhou et al., 2009]. Other algorithms try to shift the MIL problems into instance space via embedding such as MILDE [Amores, 2015], Submil [Yuan et al., 2016] and miVLAD [Wei et al., 2016]. Several regular supervised classifiers are extended to work in the MIL setting such as MI-SVM and Citation-kNN which extend respectively the SVM and the k-nearest neighbours approaches. Methods which are based on instance selection try to identify representative instances of the bags [Faria et al., 2017] [Chen et al., 2006]. In [Zhou et al., 2009] and [Zhang et al., 2011], authors try to identify the relations which exists between bags/instances and use them to improve the classification results. Some algorithms focus on defining dissimilarities between bags/instances, one example is MInD [Cheplygina et al., 2015] that uses a bag dissimilarity approach. A review of MIL approaches and a comparative study can be found in [Amores, 2013], [Alpaydın et al., 2015] and [Herrera et al., 2016]. A description of some algorithms is provided below.

DD [Maron and Lozano-Pérez, 1998] attempts to find the concept points in the feature space that are close to at least one instance from every positive bag and far from instances in negative bags. The optimum concept point is determined by maximizing the diversity density score, which is a measure of how positive a point is (i.e. positive bags have instances near the point and how far the negative instances are away from it.) An unknown bag is classified as positive if at least one of its instances is sufficiently close to the concept point, otherwise it is classified as negative. Some MIL methods proposed later are based on the DD algorithm such as EM-DD [Zhang and Goldman, 2002] which uses a set of hidden variables in order to identify which instance determines the label of a bag. These hidden variables are estimated using an expectation maximization approach.

MI-SVM and mi-SVM are two algorithms which extend a regular supervised learning approach. They are two extensions of support vector machines (SVM) where margin maximization is redefined in order to consider the MIL settings. MI-SVM deals with the problem at bag level, whereas mi-SVM deals with instance level. In regular SVMs for supervised learning, the labels of each instance in the training set are known. However, this is not the case in MIL where only the labels of the bags are known. Considering the standard MIL assumption, the labels of the negative bags instances are known to be negative. The margin could be then defined as in a regular SVM. However, the problem with the labels of positive bags instances is that they are unknown and therefore defining the margin is a complicated task. Then, mi-SVM propose to treat the instance labels as unknown integer variables. It uses a maximum instance margin formulation which tries to recover the instance labels of the positive bags. The goal is to find both the optimal labeling and the optimal hyperplane. On the other hand, MI-SVM algorithm generalizes the notion of a margin to bags. The goal is to recover the *key positive instances* which are instances used to represent positive bags. In fact, the margin of a positive bag is defined by the margin of the most positive instance, while the margin of a negative bag is defined by the least negative instance. The negative instances in the positive bags are ignored. The algorithm introduces witness variables which represent the selected instances to represent positive bags. A main difference between the mi-SVM and MI-SVM margin formulation is that in mi-SVM the margin of every instance in a positive bag matters and we can define their labels in order to maximize the margin, however, in MI-SVM only one instance in the positive bag matters to define the margin of the bag.

MIRSVM [Melki et al., 2018] is a an algorithm which uses a bag-representative selector and trains an SVM based on a bag-level information. The idea is to select representative instances from both positive and negative bags and use them in order to find an optimal unbiased separating hyperplane. Iteratively, the algorithm chooses an instance used to represent each bag, then a new hyperplane is defined according to the selected representatives until they converge. During the training process, MIRSVM gives preference to negative bags because all instances inside these bags are guaranteed to be negative according to the standard MI assumption, whereas the distribution of the instance labels in positive bags is unknown. A main difference between MIRSVM and MI-SVM algorithms is that the first one uses representatives from positive and negative bags, while the second one only optimizes over representatives from positive bags. Another difference is that MIRSVM allows for balanced selection of bag representatives, i.e. one representative is allowed for each bag regardless of its label, while MI-SVM uses one representative for positive bags and multiple representatives for negative ones.

In [Wang and Zucker, 2000], the authors present two extensions of the kNN algorithm called Bayesian-KNN and Citation-KNN. In order to transform the measure between instances (such as in standard kNN) in a measure between bags, authors propose to use the Hausdorff distance: two sets A and B are within Hausdorff distance d of each other if every point of A is within distance d of at least one point of B, and every point of B is within distance d of at least one point of A. In order to classify an unknown bag, the Bayesian method computes the posterior probabilities of its label based on the labels of its neighbors. Citation-kNN suggests the notion of *citation*. The idea is to take into account not only the neighbors of a bag B (according to the Hausdorff distance) but also its citers which are the bags that count B as their neighbor.

Some MIL approaches focus on selecting positive instances. One example is MILKDE which tries to find the most representative instances in each positive bag based on a likelihood computation. The idea is to select positive instances having the common characteristics considering all positive bags. The Kernel Density Estimation (KDE) [Parzen, 1962] is used in order to compute the maximum likelihood between those instances. The algorithm starts by looking for the most positive instance considering all instances in all positive bags, i.e. the one presenting the higher likelihood value. Given a positive bag, the algorithm computes the Euclidean distance of all instances to the previously defined MP instance. The instance which presents the shortest distance is defined as a representative of the processed bag. The resulting set of the selected positive instances as well as all negative ones represent the data used to construct the classifier. MILES [Chen et al., 2006] is another algorithm based on positive instance selection, but it does not make the instance selection in the beginning. It uses all instances in the bags as a vocabulary and defines a similarity between bags and instances in embedding space. SVM is applied to the new space and an instance selection is then done.

MIGraph and miGraph [Zhou et al., 2009] are two algorithms that use a graph representation of the processed data. The key idea is to treat the instances as non independently and identically distributed samples. Figure 3.7 gives an illustrative example which shows how taking into account the relation among instances could impact the classification decision of three sample bags. In Figure 3.7 (a), if we do not take into account the relations between the instances inside the same bag, the three bags could be considered as similar since they have identical number of similar instances. Whereas in Figure 3.7 (b), the first two bags are more similar than the third one if we take into account the relations between the instances. MI-Graph works at a bag level. It maps every bag to an undirected graph and designs a graph kernel for distinguishing the positive and negative bags. miGraph constructs graphs implicitly. Similar instances in a bag are then grouped in cliques and a graph kernel is computed based on the clique information.

In [Zhang et al., 2011], an optimization algorithm that deals with multiple instance learning on structured data (MILSD) is proposed. The idea is to use the rich dependency/structure information between instances/bags in order to improve the performance of existing MIL algorithms. This additional information is represented using a graph that depicts the structure between either bags or instances. The proposed formulation deals with two sets of constraints caused by learning on instances within individual bags and learning on structured data and has a non-convex optimization problem. To solve this problem, authors present an iterative method based on constrained concave-convex procedure (CCCP). It is an optimization method that deals with the concave convex objective function with concave convex constraints [Smola et al., 2005]. However, in many real world applications, the number of the labeled bags as well as the number of links between bags are huge. To solve the problem efficiently, an adaptation of the

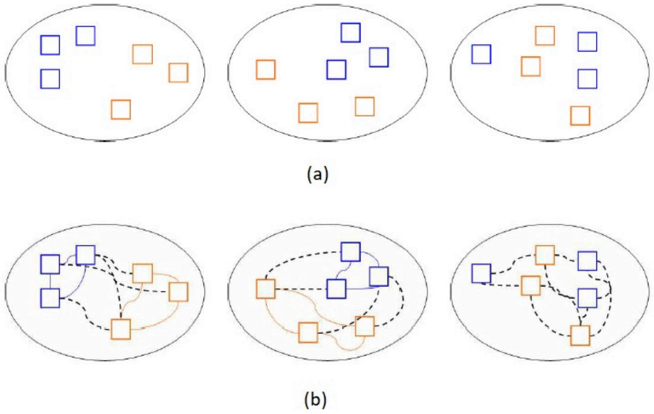


Figure 3.7: Illustrative example showing the impact of treating the instances as non independently and identically distributed samples [Zhou et al., 2009]. See text.

cutting plane method [Kelley, 1960] is proposed. The goal is to find two small subsets of constraints from a larger constraint set.

MInD (Multiple Instance Dissimilarity) algorithm [Cheplygina et al., 2015] focuses in defining dissimilarities between bags. The MIL problem is converted to a standard supervised learning problem by representing each bag by its dissimilarities to other bags. Authors discuss different ways to define a dissimilarity between two bags: viewing a bag as a set of points, as a distribution instance space and as an attributed graph. Many other algorithms convert the MIL problem to a supervised learning one such as SimpleMI [Dong, 2006] which maps each bag to the average of the instances inside. It simply aggregates statistics about the instances without making a difference between them. It is efficient when the average of positive and negative bags is different.

3.4 MIL for sequence data

3.4.1 Related works using sequence data

When the processed instances inside bags are sequences, we have an MIL problem for sequence data. Using the attribute-value format in order to encode the input data is widely used when applying MIL algorithms on sequence data.

When MIL is applied in order to deal with the document categorization problem, documents are considered as bags and some sentences represent the instances [Wang et al., 2016] [Liu et al., 2012] [Andrews et al., 2003]. An extremely sparse and high dimensional attribute-value representation of the data is generated when terms are simply used to present the text. In [Wang et al., 2016], authors we use a convolution neural network model to learn sentence representations by combining both local (at sentence/instance level) and global (at document/bag level) information.

Some works use MIL when dealing with the problem of transcription factor binding sites (TFBS) identification [Zhang et al., 2019] [Hu et al., 2019] [Gao and Ruan, 2013]. Transcription factors (TF) play important roles in the regulation of gene expression. They can modulate gene expression by binding to specific DNA regions, which are known as TFBS. It is commonly assumed that a DNA sequence that can be bound by a TF should contain one or more TFBS (a positive bag), while a DNA sequence that cannot be bound by the TF should have no TFBS (a negative bag). A sliding window is applied to check the substrings of each sequence and use them as instances mapped to feature vectors. Structural DNA properties [Bauer et al., 2010] are commonly used to generate a feature vector representation of the instances.

The identification of thioredoxin-fold (Trx-fold) proteins is another challenging problem in bioinformatics where an MIL-based problem formulation could be applied on sequence data. The Trx-fold is a characteristic protein structural motif that has been found in five distinct classes of proteins. In [Tao et al., 2004] and [Zhang et al., 2011], a dataset of protein sequences is used in the empirical evaluation: each protein sequence is considered as a bag and some of its subsequences are considered as instances. These subsequences are aligned and mapped to an 8-dimensional feature space: 7 numeric properties [Kim et al., 2000] and an 8^{th} feature that represents the residue's position. So we obtain an attribute-value format description of the dataset. In [Zhang et al., 2011], the alignment score is used in order to identify the bag-level relations between proteins. If the score between a pair of proteins exceed 25, then authors consider that there exists a link between them. We note that these works do not deal with the across-bag relations that may exist between the instances.

3.4.2 Problem Formulation

We extend the problem formulation detailed in Section 3.1.2 to deal with sequence data instances. Instances B_{ij} of a bag B_i are sequences . We note that there is an *equivalence relation* $\mathfrak{R}$ between instances of different bags denoted *the across-bag relation* which is defined according to the application domain. An equivalence relation is a binary relation that is reflexive, symmetric and transitive. To represent $\mathfrak{R}$, we opt for an index representation. We note that this notation does not mean that instances are ordered. In fact, a preprocessing step assigns an index number to the instances inside each bag according to the following notation: each instance B_{ij} of a bag B_i is related by $\mathfrak{R}$ to the instance B_{hj} of another bag B_h in DB. An instance may not have any corresponding related instance in some bags, i.e., a sequence is related to zero or one sequence per bag. We do not have necessarily the same number of instances in each bag.

$$\mathfrak{R} : DB \to DB$$
$$\mathfrak{R}(B_{ij}) = B_{hj}$$

where i and h $\in \{1,\dots,n\}$ and $j \in \{1,\dots,m\}$

$\mathfrak{R}$ is defined according to the application domain. The relation $\mathfrak{R}$ could be generalised to deal with problems where each instance has more than one target related instance in each bag. The index notation as described previously will not be suitable in this case.

3.4.3 Delimitation of the problem

The goal of this thesis is to deal with the MIL problem that has the following three criteria:

- **The instances inside the bags are sequences:** To deal with sequences, we have to deal with data representation.

- **All the instances inside a bag contribute to define the bag's label:** In the problem of bacterial IRR prediction, all the protein sequences contribute to the final decision. The standard MIL assumption is not suitable to our investigated problem, we adopt instead the collective assumption.

- **The instances may have dependencies across the bags:** The bags contain orthologous protein sequences. The across bag relation between instances could be used in the learning process.

  Considering this issue, the problem we want to solve in this work is the MIL problem in sequence data that have dependencies between instances of different bags.

3.5 Conclusion

In this chapter, we presented the MIL and some of its applications. We presented the MIL assumptions and the different levels of learning (i.e. bag level and instance level). Then we provided an overview of some MIL algorithms. Finally, we explained the particularity of the investigated problem of MIL for sequence data with across-bag dependencies and provide a formalization of the problem.

Part II

Contributions

Chapter 4

Motif-based MIL approach for sequence data with across-bag dependencies

Contents

Goals In this chapter we introduce the naive MIL approach for sequence data. Then, we present our proposed approach named ABClass. We describe the algorithm and we present the experimental study.

4.1 Naive approach

4.1.1 The algorithm

The simplest way to solve the problem of MIL for sequence data is to use standard MIL classifiers. The naive approach contains two steps (see Fig. 4.1). We first make a preprocessing step that transforms the set of sequences to an attribute-value matrix where each row corresponds to a bag of sequences and attributes conform the columns. The second step consists in applying an existing MIL classifier. In the case of sequence data, the most used technique to transform data to an attribute-value format is to extract motifs that serve as attributes/features. We note that finding a uniform description of all instances using a set of motifs is not always an easy task. Since our naive approach takes into account the across bag relations between instances, the preprocessing step extracts motifs from each set of related instances. The union of these extracted motifs is then used as features to construct an attribute-value matrix where each row corresponds to a bag. The presence or the absence of an attribute in a sequence is respectively denoted by 1 or 0. Using this approach, we obtain an attribute-value matrix that contains a large number of motifs. It is worthwhile to mention that only a subset of the used attributes is representative for each processed sequence. Therefore, we may have a big sparse matrix when trying to present the whole sequence data using an attribute value format.

4.1.2 Running example

In order to illustrate our proposed approach, we rely on the following running example. Let $\Sigma = \{A, B, \dots, Z\}$ be an alphabet. Let $DB = \{(B_1, +1), (B_2, +1), (B_3, -1), (B_4, -1), (B_5, -1)\}$ a learning database that contains 5 bags (B_1 and B_2 are positive bags, B_3, B_4 and B_5 are negative bags). Initially, the bags contain the following sequences:

B_1 = {ABMSCD, EFNOGH, RUVR}
B_2 ={CCGHDDEF, EABZQCD}
B_3 = {GHWMY, ACDXYZ}

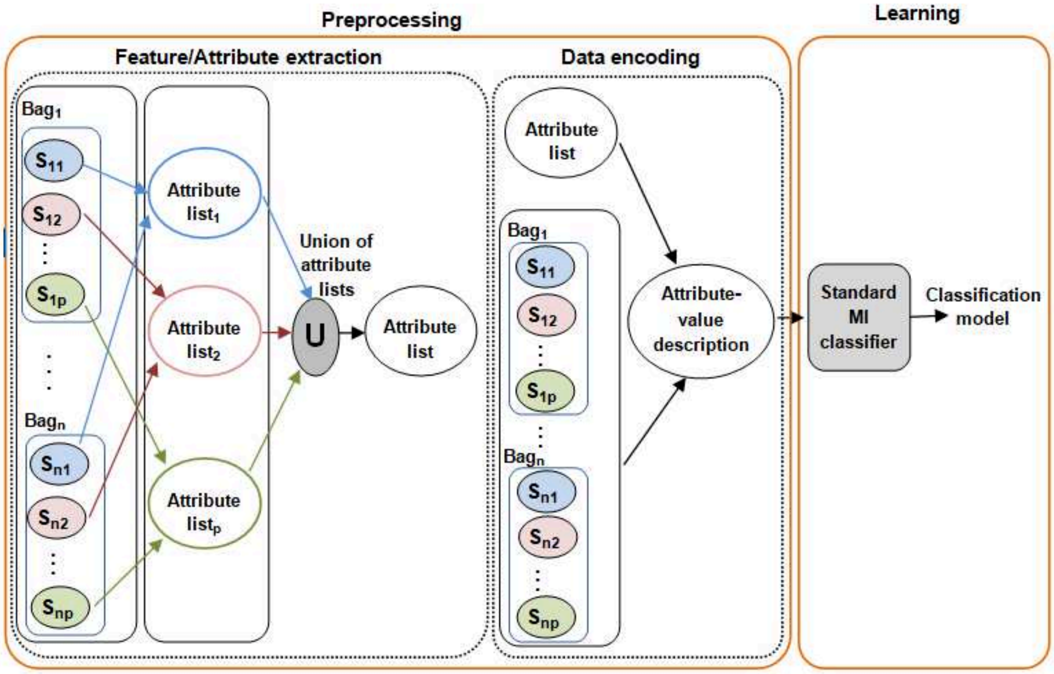


Figure 4.1: System overview of the naive approach for MIL in sequence data

B_4 ={ABIJYZ, KLSSO, EFYRTAB}
B_5 ={EFFVGH, KLSNAB}
We first use the across bag relation $\mathfrak{R}$ to represent the related instances using the index notation as described previously.

$$B_1 = \begin{cases} B_{11} = \mathbf{AB}MS\mathbf{CD} \\ B_{12} = \mathbf{EF}NO\mathbf{GH} \\ B_{13} = RUVR \end{cases} \quad B_2 = \begin{cases} B_{21} = E\mathbf{AB}ZQ\mathbf{CD} \\ B_{22} = CC\mathbf{GH}DD\mathbf{EF} \end{cases} \quad B_3 = \begin{cases} B_{31} = A\mathbf{CD}X\mathbf{YZ} \\ B_{32} = \mathbf{GH}WMY \end{cases}$$

$$B_4 = \begin{cases} B_{41} = \mathbf{AB}IJ\mathbf{YZ} \\ B_{42} = \mathbf{EF}YRT\mathbf{AB} \\ B_{43} = \mathbf{KL}SSO \end{cases} \quad B_5 = \begin{cases} B_{52} = \mathbf{EF}FV\mathbf{GH} \\ B_{53} = \mathbf{KL}SN\mathbf{AB} \end{cases}$$

The goal here is to predict the class label of an unknown bag $Q = \{Q_1, Q_2, Q_3\}$ where:

$$Q = \begin{cases} Q_1 = \textsf{AB}WX\textsf{CD} \\ Q_2 = \textsf{EF}XY\textsf{GH}N \\ Q_3 = \textsf{KL}OF \end{cases}$$

We apply the naive approach to our running example. We suppose that attributes are subsequences (minimum length = 2) that occur at least in 2 instances. Let $AttributeList_1 = \{AB, CD, YZ\}$ be the list of features extracted from the instances $\{B_{i1}, i = 1, \ldots, 4\}$. $AttributeList_2 = \{EF, GH\}$ is the list of features extracted from the instances $\{B_{i2}, i = 1, \ldots, 5\}$ and $AttributeList_3 = \{KL\}$ is the list of features extracted from the instances $\{B_{i3}, i \in \{1, 4, 5\}\}$. The union of $AttributeList_1$, $AttributeList_2$ and $AttributeList_3$ produces the list $AttributeList = \{AB, CD, YZ, EF, GH, KL\}$. In order to encode the learning sequence data, we generate the following attribute-value matrix denoted M. A missing value is denoted by "-".

$$\begin{array}{rcc} & \begin{array}{ccc} \text{instance 1} & \qquad\qquad\text{instance 2} & \qquad\qquad\text{instance 3} \end{array} & \\ \mathbf{M} = & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & | & 0 & 0 & 0 & 1 & 1 & 0 & | & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & | & 0 & 0 & 0 & 1 & 1 & 0 & | & - & - & - & - & - & - \\ 0 & 1 & 1 & 0 & 0 & 0 & | & 0 & 0 & 0 & 0 & 1 & 0 & | & - & - & - & - & - & - \\ 1 & 0 & 1 & 0 & 0 & 0 & | & 1 & 0 & 0 & 1 & 0 & 0 & | & 0 & 0 & 0 & 0 & 0 & 1 \\ - & - & - & - & - & - & | & 0 & 0 & 0 & 1 & 1 & 0 & | & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} & \begin{array}{c} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{array} \end{array}$$

The sparsity percentage of M is 77.2%. If we have a big learning database, M could result to a huge and sparse matrix since only a subset of the used subsequences is representative for each processed sequence.

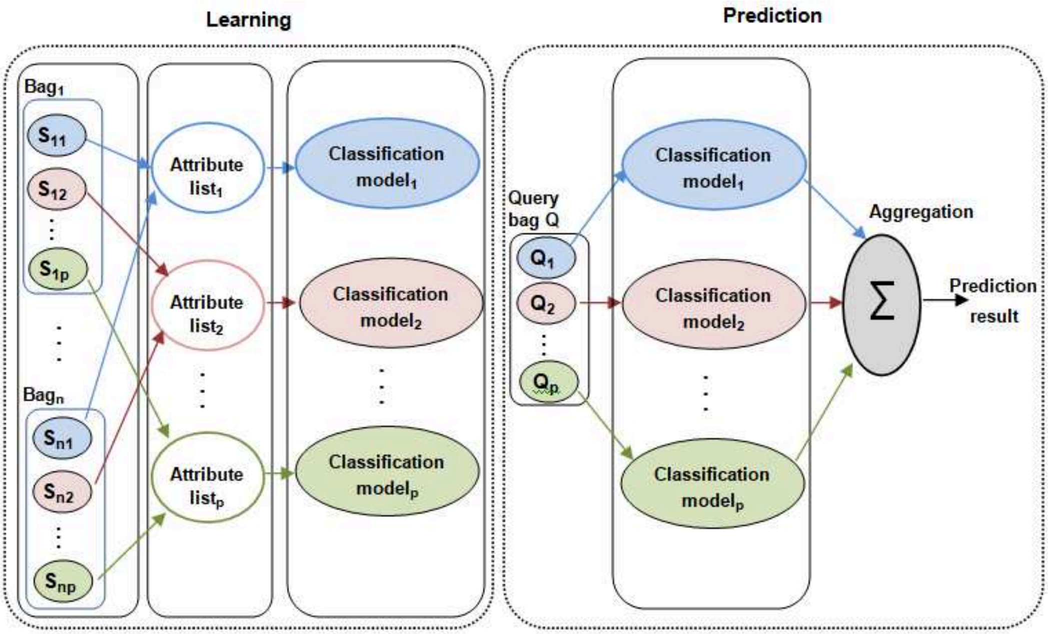


Figure 4.2: System overview of the ABClass approach

4.2 ABClass: Across-Bag sequences Classification approach

4.2.1 The approach

ABClass takes advantage of the across-bag relationship between sequences in order to reduce the number of attributes that are not representative for each processed sequence during the encoding step. Fig. 4.2 represents the system overview of ABClass. Each set of related instances will be presented by its own motifs vector. This relationship is also used during the learning step when generating partial models. Every vector of motifs will be used to produce a partial prediction result. These results will be then aggregated to compute the final result. Based on the formalization, the algorithm discriminates bags by applying a classification model to each instance of the query bag.

ABClass is described in Algorithm 1. The acrossBagSeq function groups the related instances among bags into a list. During the execution of the algorithm, we will use the following variables:

- A matrix M to store the encoded data of the learning database.
- A vector QV to store the encoded data of the query bag.
- A vector PV to store the partial prediction results.

Informally, the main steps of the *ABClass* algorithm are:

1. For each instance sequence Q_k in the query bag Q, the related instances among bags of the learning database are grouped into a list (lines 1 and 2).
2. The algorithm extracts motifs from the list of grouped instances. These motifs are used to encode instances in order to create a discriminative model (lines 3 to 5).
3. *ABClass* uses the extracted motifs to represent the instance Q_k of the unknown bag into a vector QV_k, then it compares it with the corresponding model. The comparison result is stored in the kth element of a vector PV (lines 6 and 7).
4. An aggregation method is applied to PV in order to compute the final prediction result P (line 9), which consists in a positive or a negative class label.

4.2.2 Running example

We apply the *ABClass* approach to our running example. Since the query bag contains 3 instances Q_1, Q_2 and Q_3, we need 3 iterations followed by an aggregation step.

Iteration 1: The algorithm groups the set of instances that are related across bags and extracts the corresponding motifs.

$$AcrossBagsList_1 = \{B_{11}, B_{21}, B_{31}, B_{41}\}$$
$$MotifList_1 = \{AB, CD, YZ\}$$

Algorithm 1 *ABClass* algorithm

```
Input: Learning database $DB = \{(B_i, Y_i) | i = 1, 2, \ldots, n\}$ , Query bag $Q = \{Q_k | k = 1, 2, \ldots, q\}$
Output: Prediction result $P$
for all $Q_k \in Q$ do
    $AcrossBagSeqList_k \leftarrow AcrossBagSeq(k, DB)$
    $MotifList_k \leftarrow MotifExtractor(AcrossBagsList_k)$
    $M_k \leftarrow EncodeData(MotifList_k, AcrossBagsList_k)$
    $Model_k \leftarrow GenerateModel(M_k)$
    $QV_k \leftarrow EncodeData(MotifList_k, Q_k)$
    $PV_k \leftarrow ApplyModel(QV_k, Model_k)$
end for
$P \leftarrow Aggregate(PV)$
return $P$
```

Then, it generates the attribute-value matrix M_1 describing the sequences related to Q_1.

$$M_1 = \begin{array}{c c} \begin{array}{ccc} AB & CD & YZ \end{array} & \\ \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} & \begin{array}{c} B_{11} \\ B_{21} \\ B_{31} \\ B_{41} \end{array} \end{array}$$

The sparsity percentage of the produced matrix M_1 is reduced to 33% because there is no need to use the motifs extracted from instances $\{B_{i2}, i = 1, .., 5\}$ and $\{B_{i3}, i \in 1, 4, 5\}$ to describe instances $\{B_{i1}, i = 1, .., 4\}$. A model is then created using the encoded data and a vector QV_1 is generated to describe Q_1.

$$QV_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

By applying the model to the vector QV_1, we obtain the first partial prediction result and we store it into the vector PV.

$$PV_1 \leftarrow ApplyModel(QV_1, Model_1)$$

Iteration 2: The second iteration concerns the second instance Q_2 of the query bag. We do the same instructions described in the first iteration.

$$AcrossBagsList_2 = \{B_{21}, B_{22}, B_{32}, B_{42}, B_{52}\}$$
$$MotifList_2 = \{EF, GH\}$$

$$M_2 = \begin{array}{cc} EF & GH \\ \left(\begin{array}{cc} 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{array}\right) \end{array} \begin{array}{c} \\ B_{12} \\ B_{22} \\ B_{32} \\ B_{42} \\ B_{52} \end{array}$$

$$QV_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$PV_2 \leftarrow ApplyModel(QV_2, Model_2)$$

Iteration 3: Only B_1, B_4 and B_5 have related instances to Q_3.

$$AcrossBagsList_3 = \{B_{13}, B_{43}, B_{53}\}$$
$$MotifList_3 = \{KL\}$$

$$M_3 = \begin{array}{c} KL \\ \left(\begin{array}{c} 0 \\ 1 \\ 1 \end{array}\right) \end{array} \begin{array}{c} \\ B_{13} \\ B_{43} \\ B_{53} \end{array}$$

$$QV_3 = \begin{pmatrix} 1 \end{pmatrix}$$

$$PV_3 \leftarrow ApplyModel(QV_3, Model_3)$$

The aggregation step is finally used to generate the final prediction decision using the partial prediction results. We opt for the majority vote.

4.3 Creating the bacterial IRR database

We created a dataset composed of bags of real sequence data. Table 4.1 shows the 28 bacteria (the bags): 14 IRRB (B1 to B14) and 14 IRSB (B15 to B28). Each bacterium contains 25 to 31 primary structures of proteins implicated in basal DNA repair in IRRB. Table 4.2 contains the used proteins. More details about the number of proteins in each bacterium and the number of protein sequences in each positive bag (IRRB) and negative one (IRSB) are provided in Appendix A. Bacteria represent the bags and protein sequences represent the instances. The used across-bag relation is the orthology. Orthologous genes are assumed to have the same biological functions in different species.

Information on complete and ongoing IRRB genome sequencing projects was obtained from the GOLD database [Liolios et al., 2008]. We initiated our analyses by retrieving orthologous proteins implicated in basal DNA repair in IRRB and IRSB with sequenced genomes. Proteins of the bacterium *Deinococcus radiodurans* (B7) were downloaded from the UniProt website. In the preprocessing step, we used the perfectBLAST [Santiago-Sotelo and Ramirez-Prado, 2012] tool in order to identify orthologous proteins. Proteomes of other bacteria were downloaded from the NCBI FTP website. We note that some proteins do not have any ortholog in some bags. We do not have the same number of instances in each bag. The dataset is publicly available in the following link: `https://homepages.loria.fr/SAridhi/software/MIL/#downloads` .

4.4 Experimental study

We applied the naive approach and ABClass to solve the problem of IRR prediction in bacteria. The proposed MIL-based prediction systems aim to affiliate an unknown bacterium to either IRRB or IRSB.

4.4.1 Experimental environment

For our tests, we used the dataset described in Section 4.3. We used WEKA [Hall et al., 2009] data mining tool in order to apply existing well known classifiers to test the proposed approaches. When running ABClass experiments, we used the following classifiers: SVM, SMO, IBk (a K-nearest neighbor implementation), J48 (an implementation of C4.5 decision tree algorithm) and Logistic (a logistic regression based classifier). In order to test the naive approach, the following classifiers of WEKA were used: MISVM (implementation of the instance based mi-SVM algorithm), MISMO (uses the SMO algorithm [Platt, 1998] for SVM learning in conjunction with a multiple instance kernel), citationKNN (multiple instance extension of K-nearest neighbor algorithm), MILR (multiple instance adaptation of the logistic regression classification), MITI (a decision tree algorithm adapted to multiple instance settings) and QuickDDIterative (an iterative faster version of the basic DD algorithm).

4.4.2 Experimental protocol

In order to evaluate the naive approach and the ABClass approach, we first encode the protein sequences of each bag using a set of features/motifs generated by an existing motif extraction method. Then, we apply an existing classifier to the encoded data. We used the Leave-One-Out (LOO) evaluation technique. In our tests, we used DMS [Maddouri and Elloumi, 2004] as a motif extraction method. DMS allows building motifs that can discriminate a family of proteins from other ones. It first identifies motifs in the protein sequences. Then, the extracted motifs are filtered in order to keep only the discriminative and minimal ones. A substring is considered to be discriminative between the family F and the other families if it appears in F significantly more than in the other families. DMS extracts discriminative motifs according to α and β thresholds where α is the minimum rate of motif occurrences in the sequences of a family F and β is the maximum rate of motif occurrences in all sequences except those of the family F. In the following, we present the used motif extraction settings according to the values of α and β:

Table 4.1: IRRB and IRSB learning set.

ID	Bacterium	Phylogenetic group	D$_{10}$ (kGy)[a]
B1	*Chroococcidiopsis thermalis* PCC 7203	Cyanobacteria	4[b] [Billi et al., 2002]
B2	*Deinococcus deserti* VCD115	*Deinococcus-Thermus*	>7.5 [Slade and Radman, 2011]
B3	*Deinococcus geothermalis* DSM 11300	*Deinococcus-Thermus*	10-16 [Slade and Radman, 2011]
B4	*Deinococcus gobiensis* I 0	*Deinococcus-Thermus*	12.7 [Slade and Radman, 2011]
B5	*Deinococcus maricopensis* DSM 21211	*Deinococcus-Thermus*	$\sim$11 [Rainey et al., 2005]
B6	*Deinococcus proteolyticus* MRP	*Deinococcus-Thermus*	>15 [Brooks and Murray, 1981]
B7	*Deinococcus radiodurans* R1	*Deinococcus-Thermus*	10 [Ito et al., 1983]
B8	*Geodermatophilus obscurus* DSM 43160	Actinobacteria	9 [Gtari et al., 2012]
B9	*Kineococcus radiotolerans* SRS30216	Actinobacteria	2 [Phillips et al., 2002]
B10	*Kocuria rhizophila* DC2201	Actinobacteria	2[c] [Rainey et al., 1997] [Brooks and Murray, 1981]
B11	*Methylobacterium radiotolerans* JCM 2831	Proteobacteria	1 [Green and Bousfield, 1983] [Ito and Iizuka, 1971]
B12	*Modestobacter marinus*	Actinobacteria	6 [Gtari et al., 2012]
B13	*Rubrobacter xylanophilus* DSM 9941	Actinobacteria	5.5 [Ferreira et al., 1999]
B14	*Truepera radiovictrix* DSM 17093	*Deinococcus-Thermus*	>5 [Albuquerque et al., 2005]
B15	*Brucella abortus* S19	Proteobacteria	0.34 [Federighi and Tholozan, 2001]
B16	*Escherichia coli* B REL606	Proteobacteria	0.7 [Daly et al., 2004]
B17	*Escherichia coli* str. K-12 substr. DH10B	Proteobacteria	0.7 [Daly et al., 2004]
B18	*Neisseria gonorrhoeae* FA 1090	Proteobacteria	0.07-0.125 [Daly et al., 2004]
B19	*Neisseria gonorrhoeae* TCDC NG08107	Proteobacteria	0.07-0.125 [Daly et al., 2004]
B20	*Pseudomonas putida* S16	Proteobacteria	0.25 [Daly et al., 2004]
B21	*Shewanella oneidensis* MR-1	Proteobacteria	0.07 [Daly et al., 2004]
B22	*Shigella dysenteriae*1617	Proteobacteria	0.22 [Federighi and Tholozan, 2001]
B23	*Thermus thermophilus* HB27	*Deinococcus-Thermus*	0.8 [Federighi and Tholozan, 2001]
B24	*Thermus thermophilus* HB8	*Deinococcus-Thermus*	0.8[d][Federighi and Tholozan, 2001]
B25	*Thermus thermophilus* JL-18	*Deinococcus-Thermus*	0.8[d] [Federighi and Tholozan, 2001]
B26	*Thermus thermophilus* SG0.5JP17-16	*Deinococcus-Thermus*	0.8[d] [Federighi and Tholozan, 2001]
B27	*Vibrio parahaemolyticus* RIMD 2210633	Proteobacteria	0.03-0.06 [Federighi and Tholozan, 2001]
B28	*Yersinia enterocolitica* 8081	Proteobacteria	0.1-0.21 [Federighi and Tholozan, 2001]

a. D$_{10}$: Dose for 90% reduction in Colony Forming Units (CFUs); for IRRB, it is greater than 1 kGy.

b. for *Chroococcidiopsis* spp.

c. for *Kocuria rosea*.

d. for *T. thermophilus* HB27.

Table 4.2: Replication, repair and recombination proteins.

ID	Protein	Function
P1	Hypothetical DNA polymerase	DNA polymerase
P2	DNA polymerase III, α subunit	
P3	DNA-directed DNA polymerase	
P4	DNA polymerase III, τ/γ subunit	
P5	Single-stranded DNA-binding protein	Replication complex
P6	Replicative DNA helicase	
P7	DNA primase	
P8	DNA gyrase, subunit B	
P9	DNA topoisomerase I	
P10	DNA gyrase, subunit A	
P11	Smf proteins	Other DNA-associated proteins
P12	Endonuclease III	
P13	Holliday junction resolvase	
P14	Formamidopyrimidine-DNA glycosylase	
P15	Holliday junction DNA helicase	
P16	RecF protein	
P17	DNA repair protein radA	
P18	Holliday junction binding protein	
P19	Excinuclease ABC, subunit C	
P20	DNA repair protein RecN	
P21	Transcription-repair coupling factor	
P22	Excinuclease ABC, subunit A	
P23	DNA helicase II	
P24	DNA helicase RecG	
P25	Exonuclease SbcD, putative	
P26	Exonuclease SbcC	
P27	Ribonuclease HII	
P28	Excinuclease ABC, subunit B	
P29	A/G-specific adenine glycosylase	
P30	RecA protein	
P31	DNA-3-methyladenine glycosidase II, putative	

- **S1** ($\alpha = 1$ and $\beta = 0.5$): used to extract frequent motifs with medium discrimination.

- **S2** ($\alpha = 1$ and $\beta = 1$): used to extract frequent motifs without discrimination.

- **S3** ($\alpha = 0.5$ and $\beta = 1$): used to extract motifs having medium frequencies without discrimination.

- **S4**: ($\alpha = 0$ and $\beta = 1$): used to extract infrequent and non discriminative motifs.

- **S5**: ($\alpha = 1$ and $\beta = 0$): used to extract frequent and strictly discriminative motifs.

We calculated the accuracy, specificity and sensitivity results of the used approaches. It is helpful at this point to introduce the confusion matrix which could be presented as:

		Real class	
		Positive (IRRB)	Negative (IRSB)
Predicted class	Positive (IRRB)	True prositive (TP)	False prositive (FP)
	Negative (IRSB)	False negative (FN)	True Negative (TN)

The accuracy measures the proportion of true results (both true positives and true negatives) among the total number of classified bags. The specificity rate measures the proportion of actual negatives which are correctly identified as such. The sensitivity rate measures the proportion of actual positives which are correctly identified as such. In terms of the above confusion matrix, the accuracy, specificity and sensitivity are defined as:

$$\text{accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{FP} + \text{FN} + \text{TN}).$$
$$\text{sensitivity} = \text{TP} /(\text{TP} + \text{FN})$$
$$\text{specificity} = \text{TN}/(\text{FP} + \text{TN}).$$

Table 4.3: Sparsity of the attribute-value matrix used in the naive approach.

Motif extraction setting	Total number of motifs	Sparsity (%)
S1	519	84.3
S2	1141	84
S3	4167	89.6
S4	7670	93.5

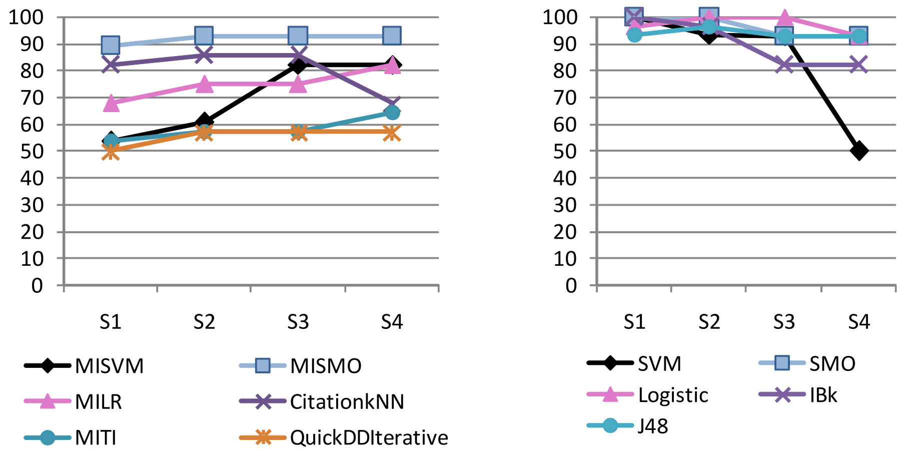
4.4.3 Experimental results

In order to use standard multiple instance classifiers, we apply a preprocessing technique that consists in extracting motifs from each set of protein sequences using the DMS method. Table 4.4 presents for each extraction setting the number of extracted motifs from each set of orthologous protein sequences. For the setting S5 ($\alpha = 1$ and $\beta = 0$), there is no frequent and strictly discriminative motifs for most proteins. This is why we will not use these values of α and β for our next experiments. We note that the number of extracted motifs increases for high values of β and low values of α. As presented in Table 4.3, the number of infrequent and non discriminative motifs is very high. In order to encode data in the naive approach, the union of the extracted motifs from each protein is used as attributes. Consequently, the attribute-value matrix representing the data becomes large and sparse since only a small subset of the used motifs is representative for each protein. We show in Table 4.3 the sparsity of the matrix which measures the fraction of zero elements over the total number of elements. The sparsity is generally proportional to the number of used motifs. For example, it goes from 84% with 1141 motifs to 93.5% with 7670 motifs.

ABClass provides good overall accuracy, specificity and sensitivity results (see Figures 4.3, 4.4 and 4.5) compared to those obtained using the naive approach. This shows that the proposed approach is efficient. The best result is reached using ABClass approach and the motif extraction settings S1, S2 and S3. Using these three settings, a minimum threshold of frequency and/or discrimination should be reached when extrcating motifs. The figures 4.3 (a) and (b) show the

Table 4.4: Number of extracted motifs for each set of orthologous protein sequences using a minimum motif length = 3.

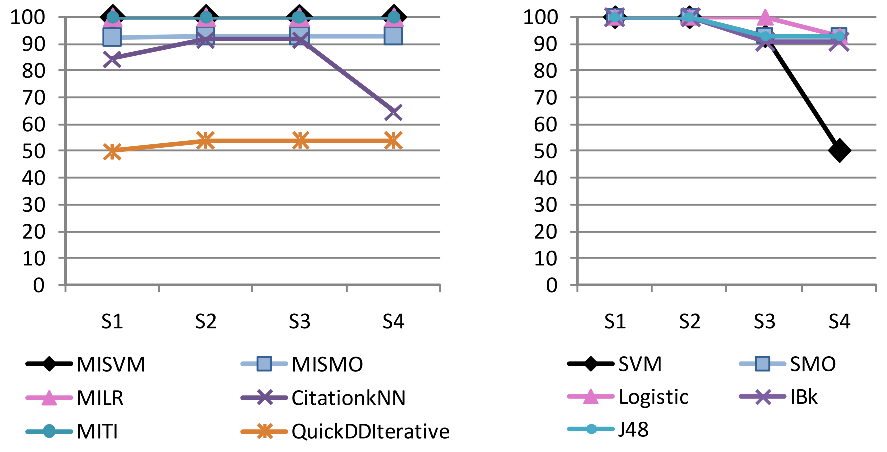
Protein ID	Motif extraction setting				
	S1	S2	S3	S4	S5
P1	348	352	612	2226	229
P2	15	76	1139	5152	0
P3	6	41	681	4361	0
P4	2	21	446	3751	0
P5	1	1	119	1698	0
P6	11	29	349	3379	0
P7	5	18	371	3907	1
P8	3	62	484	3910	0
P9	7	42	780	4211	0
P10	25	90	719	3830	0
P11	3	7	200	2769	0
P12	4	17	144	1871	0
P13	0	1	111	1544	0
P14	2	12	133	2444	0
P15	3	50	303	2071	0
P16	0	1	187	2659	0
P17	3	27	349	2712	0
P18	0	1	81	1752	0
P19	7	14	427	3800	0
P20	2	20	343	3218	0
P21	21	79	882	4581	1
P22	18	173	785	3910	1
P23	5	43	524	4152	0
P24	5	48	520	3861	0
P25	1	5	264	2563	0
P26	22	72	778	3355	2
P27	5	9	162	1667	0
P28	16	111	572	3308	1
P29	2	11	189	2729	0
P30	9	66	281	1852	0
P31	0	0	92	2061	0
Total	551	1499	13072	95304	235



(a) Naive approach

(b) ABClass approach

Figure 4.3: Accuracy results of the naive approach and ABClass



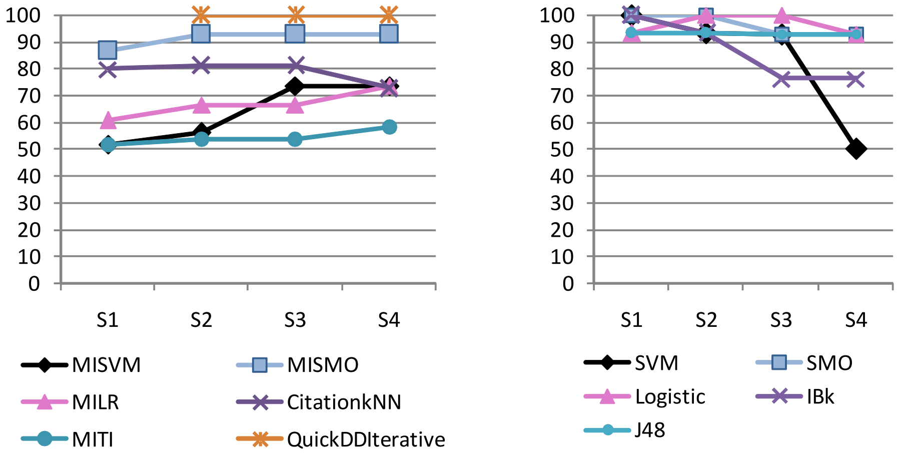
(a) Naive approach

(b) ABClass approach

Figure 4.4: Sensitivity results of the naive approach and ABClass

Table 4.5: Rate of successful classification models for each bacterium using AB-Class approach and LOO evaluation method

Bacterium ID	S1 motif extraction setting					S4 motif extraction setting				
	SVM	SMO	Logistic	IBk	J48	SVM	SMO	Logistic	IBk	J48
B1	86.3	86.3	90.9	90.9	81.8	**24**	68	80	**44**	60
B2	96.2	96.2	96.2	96.2	96.2	61.2	100	100	100	96.7
B3	92.5	92.5	92.5	92.5	92.5	61.2	100	100	100	90.3
B4	96.1	96.1	96.1	96.1	92.3	66.6	100	100	100	93.3
B5	100	100	100	100	92.3	**53.3**	100	100	100	86.6
B6	100	100	100	100	92.3	**46.6**	100	100	100	86.6
B7	88.8	88.8	92.5	92.5	88.8	58	100	100	100	93.5
B8	92	92	92	92	92	**41.3**	100	100	96.5	93.1
B9	95.6	92	91.3	91.3	86.9	**36**	100	100	96	84
B10	88	100	88	88	84	**32.1**	100	100	92.8	82.1
B11	**54.1**	62.5	**45.8**	**45.8**	**41.6**	**14.2**	**17.8**	**46.4**	**10.7**	**46.4**
B12	91.6	91.6	91.6	91.6	91.6	**42.8**	100	100	100	92.8
B13	95.6	95.6	95.6	95.6	82.6	**25.9**	92.5	96.2	18.5	66.6
B14	84	80.7	84.6	84.6	61.5	**33.3**	96.6	96.2	43.3	70
B15	83.3	83.3	87.5	87.5	79.1	**17.8**	**10.7**	**3.5**	**10.7**	**28.5**
B16	100	100	100	100	100	80	100	100	100	100
B17	95.8	95.8	95.8	95.8	95.8	81.4	96.2	96.2	100	96.2
B18	100	100	100	100	100	68	100	100	100	100
B19	100	100	100	100	100	65.3	100	100	100	100
B20	88	96	92	92	88	**51.7**	86.2	89.6	93.1	**58.6**
B21	100	100	100	100	100	**55.1**	93.1	93.1	93.1	82.7
B22	96.1	96.1	96.1	96.1	96.1	80	100	100	100	100
B23	88.8	92.5	96.2	96.2	92.5	**48.3**	100	100	100	96.7
B24	88.8	92.5	96.2	96.2	92.5	**48.3**	100	100	100	100
B25	88.8	92.5	96.2	96.2	92.5	**48.3**	100	100	100	96.7
B26	88.8	92.5	96.2	96.2	92.5	**48.3**	100	100	100	100
B27	100	100	100	100	100	62.9	100	100	100	96.2
B28	96.1	96.1	96.1	96.1	96.2	66.6	96.6	100	96.6	96.6



(a) Naive approach

(b) ABClass approach

Figure 4.5: Specificity results of the naive approach and ABClass

impact of the motif extraction settings on the prediction results using the naive approach and ABClass. For example, using MISVM classifier, the accuracy varies from 53.5% using S1 to 82.1% using S3. Although the motifs extracted using S1 are discriminative, the naive approach does not provide good accuracy results for most multiple instance classifiers. For some classifiers, the results using S1 are the lowest comparing with the other motif extraction settings. However, using this setting, ABClass provides good results since it reaches 100% of accuracy using SVM, SMO and IBk classifiers, 96.4% using Logistic and 93.3% using J48. This could be explained by the fact that the naive approach looses the advantage of representing the instances using discriminative motifs when it uses the union of all motifs in the data encoding step. Using S4, ABClass does not reach 100% of accuracy although it succeeds to reach it with some classifiers using the other three settings S1, S2 and S3. No constraints related to frequency ($\alpha = 0$) or discrimination ($\beta = 1$) were required when extracting motifs using S4.

We compute the rate of classification models that contribute to predict the true class of each bacterium using ABClass approach (see Table 4.5). In each LOO iteration, this rate is calculated for each bag as the quotient of the number of models (already generated for each set of related sequences) which successfully predict the class of that bag by the total number of sequences which belong to that bag. We present this rate for the motif extraction setting that provides the best accuracy rates i.e., S1 and the setting that provides low accuracy pourcentages, i.e., S4. The rate of successful classification models that does not exceed 60% are marked with bold text. The two bacteria B11 and B15 often generate low rates.

Biological explanation

The results illustrated in Table 4.5 may help to understand some characteristics of the studied bacteria. In particular, the IRRB *M. radiotolerans* (B11) and the IRSB *B. abortus* (B15) present a high rate of failed predictions. Although B11 is sometimes successfully classified, its higher successful classification rate does not exceed 62.5%. The rate of B15 does not reach 30% using S4. *M. radiotolerans* is often predicted as IRSB and *B. abortus* is predicted as IRRB; the former is an intracellular parasite [Halling et al., 2005] and the latter is an endosymbiont of most plant species [Fedorov et al., 2013]. We provided a possible biological explanation in [Aridhi et al., 2016] and [Zoghlami et al., 2018b]. The explanation could be the increased rate of sequence evolution in endosymbiotic bacteria [Woolfit and Bromham, 2003]. As our training set is composed mainly of members of the phylum *Deinococcus-Thermus*; expectedly, the *Deinococcus* bacteria (B2-B7) present a very low rate of failed predictions.

4.5 Conclusion

In this chapter, we presented the naive MIL approach for sequence data. We described our novel approach for MIL in sequence data with across-bag relations. We applied it to the problem of prediction of IRR in bacteria. By running experiments, we have shown that the proposed approach is efficient.

Chapter 5

Similarity-based MIL approach for sequence data with across-bag dependencies

Contents

Goals This chapter introduces the ABSim approach for MIL in sequence data with across-bag dependencies. we provide a description of the algorithm and the two used aggregation methods. We apply ABSim to the illustrative running example used in the previous chapter. Finally, we present an experimental study by applying ABSim to solve the bacterial IRR prediction problem.

5.1 ABSim: Across-Bag sequences Similarity approach

We propose an algorithm, named ABSim, that focuses on discriminating bags based on a similarity measure which could be defined according to the specificity of the processed instances. ABSim was originally presented in [Aridhi et al., 2016] as an algorithm used for IRR prediction. When applied on genomic sequences, ABSim uses the alignment score as similarity measure to compare protein sequences.

5.1.1 The approach

According to the specificity of the processed data, a similarity measure can be defined and used to discriminate instances. In order to discriminate the bags, ABSim measures the similarity between each sequence in the query bag and its corresponding related sequences in the different bags of the learning database.

Let M be a matrix used to store similarity measurement score vectors during the execution of the algorithm. The *ABSim* algorithm works as follows (see Algorithm 2).

```
Algorithm 2 SET AcrossBagSequencesSimilarity($DB$, $Q$)
Input: Learning database $DB = \{(B_i, Y_i) | i = 1, 2 \dots, n\}$ , Query bag $Q = \{Q_k | k = 1, 2, \dots, p\}$
Output: Prediction result $P$
1: for all $Q_k \in Q$ do
2:     for all $B_i \in DB$ do
3:         $M_{ik} \leftarrow similarityMeasure(Q_k, B_{ik})$ {$B_{ik}$ is the instance number $k$ in the bag $B_i$}
4:     end for
5: end for
6: $P \leftarrow Aggregate(M)$
7: return $P$
```

Informally, the algorithm is described as follows:

1. For each instance sequence Q_k in the query bag Q, it computes the corresponding similarity scores (line 1 to 4). The similarity scores of all instances

of the query bag are grouped into a matrix M (line 3). The element M_{ik} corresponds to the similarity score between the instance Q_k of Q and the instance B_{ik} of the bag B_i.

2. An aggregation method is applied to M in order to compute the final prediction result P (line 6). According to the aggregation result, a class label is associated to the query Bag.

5.1.2 Aggregation methods: SMS and WAMS

In our work, we define two aggregation methods: Sum of Maximum Scores (SMS) and Weighted Average of Maximum Scores (WAMS). Algorithms 3 and 4 illustrate the SMS and WAMS aggregation methods.

For each sequence in the query bacterium, we scan the corresponding line of M, which contains the obtained scores against all the other bags of the training database. The SMS method selects the maximum score among the similarity scores against bags that belong to the positive class label (which we call max_P) and the maximum score among the similarity scores against bags that belong to the negative class label (which we call max_N). These scores are then compared. If max_P is greater than max_N, it adds max_P to the total score of the positive class label (which we denote $total_P(M)$). Otherwise, it adds max_N to the total score of the negative class label (which we denote $total_N(M)$). When all selected sequences were processed, the SMS method compares total scores of positive class label and negative class label. If $total_P(M)$ is greater than $total_N(M)$, the prediction output is the positive class label. Otherwise, the prediction output is the negative class label.

Using the $WAMS$ method, each sequence Q_i has a given weight w_i. For each sequence in the query bag, we scan the corresponding line of M, which contains the obtained scores against all other bags of the training database. The $WAMS$ method selects the maximum score among the similarity scores against bags that belong to positive class label (which we denote $max_P(M)$) and the maximum score among the similarity scores against bags that belong to the negative class label (which we denote $max_N(M)$). It then compares these scores. If the $max_P(M)$ is

Algorithm 3 SMS(M)

```
Input: Similarity matrix $M = \{M_{ij} | i = 1, 2 \ldots, n$ and $j = 1, 2 \ldots, p\}$
Output: A prediction result $P$
$total_P \leftarrow 0$
$total_N \leftarrow 0$
for $i \in [1; n]$ do
    $max_P \leftarrow 0$
    $max_N \leftarrow 0$
    for $j \in [1; p]$ do
        if $Y_j = +1$ and $M_{ij} \geq max_P$ then
            $max_P \leftarrow M_{ij}$
        else if $Y_j = -1$ and $M_{ij} \geq max_N$ then
            $max_N \leftarrow M_{ij}$
        end if
    end for
    if $max_P \geq max_N$ then
        $total_P \leftarrow total_P + max_P$
    else
        $total_N \leftarrow total_N + max_N$
    end if
end for
if $total_P \geq total_N$ then
    $P \leftarrow +1$
else
    $P \leftarrow -1$
end if
return $P$
```

greater than $max_N(M)$, it adds $max_P(M)$ multiplied by the weight of the sequence to the total score of the positive class label and it increments the number of positive bags having a max score. Otherwise, it adds $max_N(M)$ multiplied by the weight of the sequence to the total score of the negative class label and it increments the number of negative bags having a max score. When all the selected sequences were processed, we compare the average of total scores of positive class labels (which we denote $avg_P(M)$) and the average of total scores of negative class labels (which we denote $avg_N(M)$). If $avg_P(M)$ is greater than $avg_N(M)$, the prediction output is the positive class label. Otherwise, the prediction output

Algorithm 4 WAMS(M, W)

```
Input: Similarity matrix $M = \{M_{ij} | i = 1,2\ldots,n$ and $j = 1,2\ldots,p\}$, Weight vector
    $W = \{w_i | i = 1,2\ldots,p\}$
Output: A prediction result $P$
$total_P \leftarrow 0$
$total_N \leftarrow 0$
$nb_P \leftarrow 0$
$nb_N \leftarrow 0$
for $i \in [1;p]$ do
    $max_P \leftarrow 0$
    $max_N \leftarrow 0$
    for $j \in [1;n]$ do
        if $Y_j = +1$ and $M_{ij} \geq max_P$ then
            $max_P \leftarrow M_{ij}$
        else if $Y_j = -1$ and $M_{ij} \geq max_N$ then
            $max_N \leftarrow M_{ij}$
        end if
    end for
    if $max_P \geq max_N$ then
        $total_P \leftarrow total_P + (max_P \cdot w_i)$
        $nb_P \leftarrow nb_P + 1$
    else
        $total_N \leftarrow total_N + (max_N \cdot w_i)$
        $nb_N \leftarrow nb_N + 1$
    end if
end for
$avg_P(M) \leftarrow total_P / nb_P$
$avg_N(M) \leftarrow total_N / nb_N$
if $avg_P(M) \geq avg_N(M)$ then
    $P \leftarrow +1$
else
    $P \leftarrow -1$
end if
return $P$
```

is the negative class label.

5.1.3 Running example

In order to apply the *ABSim* approach to our running example, we use a simple similarity measure that consists in the number of common symbols between the sequences. The first iteration computes the common symbols between the instance Q_1 of the query bag and the four related instances B_{11}, B_{21}, B_{31} and B_{41} (there is no related instance in the bag B_5). The results are stored in the first column of the matrix M.

$$M = \begin{array}{c} \begin{pmatrix} 4 & - & - \\ 4 & - & - \\ 4 & - & - \\ 2 & - & - \\ - & - & - \end{pmatrix} \end{array} \begin{array}{c} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{array} \tag{5.1}$$

The second iteration computes the similarity score between the instance Q_2 and its five related sequences. The results are stored in the second column of M.

$$M = \begin{pmatrix} 4 & 5 & - \\ 4 & 4 & - \\ 3 & 3 & - \\ 2 & 3 & - \\ - & 5 & - \end{pmatrix} \begin{array}{c} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{array} \tag{5.2}$$

The last iteration computes the third column of the matrix M.

$$M = \begin{pmatrix} 4 & 5 & 0 \\ 4 & 4 & - \\ 3 & 3 & - \\ 2 & 3 & 3 \\ - & 5 & 2 \end{pmatrix} \begin{array}{c} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \end{array} \tag{5.3}$$

Using the *SMS* aggregation method, we have the following results:

$$total_P(M) = 9$$
$$total_N(M) = 0$$

The query bag Q is finally classified as positive. In order to use the *WAMS* aggregation method, we need to specify a weight value for each instance. We suppose that all sequences are equally weighted, then we have the following results:

$$avg_P(M) = 4.5$$
$$avg_V(M) = 0$$

The query bag Q is finally classified as positive.

5.2 Experimental study

5.2.1 Experimental environment

We used the dataset described in the previous chapter in Section 4.3. The similarity measure used when applying the ABSim approach is the local alignment bit-score computed using the BLAST alignment tool. We downloaded the standalone executable of BLAST+ [1] and integrated it into our pipeline using the command-line. In each run, the alignment used two related sequences (a query and a subject). Appendix B shows two examples of two sequence alignment results (an alignment using two IRRB and another one using one bacterium IRRB and one bacterium IRSB).

5.2.2 Results

In order to study the importance of considering the problem of predicting bacterial IRR as a multiple instance learning problem, we present in Table 5.1 the experimental results using a set of proteins to represent the studied bacteria. For each

[1] https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=Download

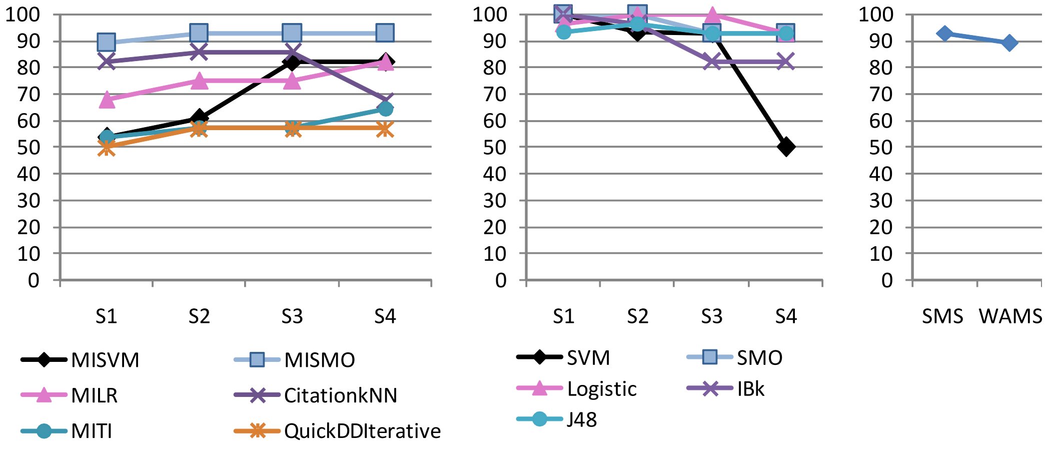
set of proteins and for each aggregation method, we present the accuracy, the sensitivity and the specificity percentages. The *WAMS* aggregation method was used with equally weighted proteins. We notice that the use of the whole set of proteins to represent the studied bacteria allows good accuracy accompanied by high values of sensitivity and specificity. This can be explained by the pertinent choice of basal DNA repair proteins to predict the phenotype of IRR. The high values of specificity presented by ABSim indicate the ability of this algorithm to identify negative bags (IRSB). Using all proteins, we have 92.8% of accuracy and specificity. As shown in Table 5.1, the SMS aggregation method allows better results than the WAMS aggregation method using the whole set of proteins to represent the studied bacteria. Using the other subsets of proteins (DNA polymerase, replication complex and other DNA-associated proteins) to represent the bacteria, SMS and WAMS present the same results.

Table 5.2 presents for each bacterium in the learning database the number of runs that succeed to classify the bacterium. More than 89% of tested bacteria show successful predictions of 100%. This means that we succeed to correctly predict the IRR phenotype of those bacteria. On the other hand, the results illustrated in Table 5.2 may help to understand some characteristics of the studied bacteria. In particular, the IRRB *M. radiotolerans* (B11) and the IRSB *B. abortus* (B15) present a high rate of failed predictions. We note that results are similar to those found using ABClass. A possible biological explanation is provided at the end of the Section 4.4.3.

Figures 4.3, 5.2 and 5.3 show that both ABClass and ABSim approaches provide good overall results compared to those obtained using the naive approach. A better result could be provided either by ABClass or by ABSim according to the used settings. The highest accuracy pourcentage was reached using ABClass and the motif extraction settings S1, S2 and S3 (see Section 4.4.3). The results provided by ABSim using the SMS aggregation method are slightly better than those obtained using WAMS. ABSim does not use motifs to represent data since no encoding step is needed. The local alignment score is used to perform the prediction. This makes ABSim faster and easier to use than ABClass unless we already have the representative motifs for each set of orthologous proteins or if

Table 5.1: Experimental results of ABSim with LOO-based evaluation technique.

Used proteins	Aggregation method	Accuracy (%)	Sensitivity (%)	Specificity (%)
All proteins	SMS	**92.8**	**92.8**	**92.8**
	WAMS	89.2	92.3	86.6
DNA polymerase proteins	SMS	89.2	92.3	86.6
	WAMS	89.2	92.3	86.6
Replication complex proteins	SMS	**92.8**	**92.8**	**92.8**
	WAMS	**92.8**	**92.8**	**92.8**
Other DNA-associated proteins	SMS	**92.8**	**92.8**	**92.8**
	WAMS	**92.8**	**92.8**	**92.8**



(a) Naive approach (b) ABClass (c) ABSim

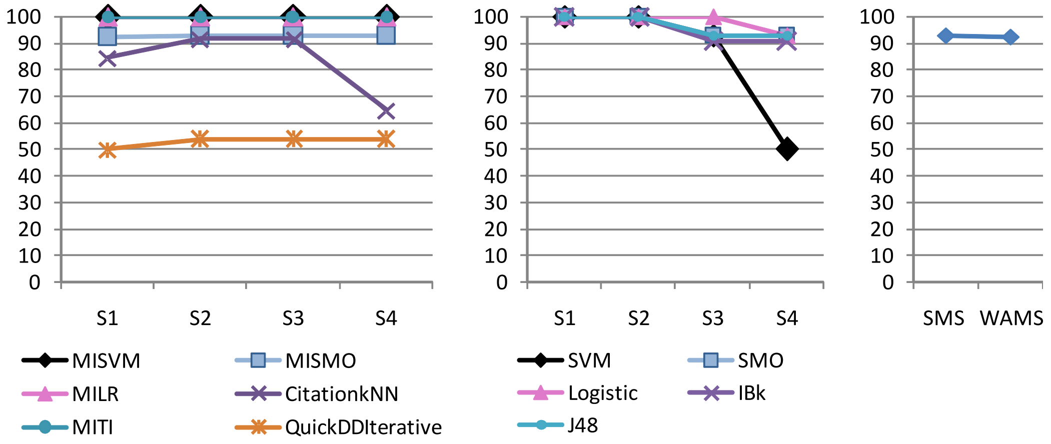
Figure 5.1: Accuracy results of the naive approach, ABClass and ABSim.

we think that the extraction of motifs will not be an expensive task (according to the data size, the used motifs extractor and the extraction settings e.g. required motifs length).

Table 5.2: Number of successful predictions (for 8 runs): The following 4 settings were used with SMS and WAMS aggregation methods: (1) all proteins (2) DNA polymerase proteins (3) replication complex proteins and (4) other DNA-associated proteins.

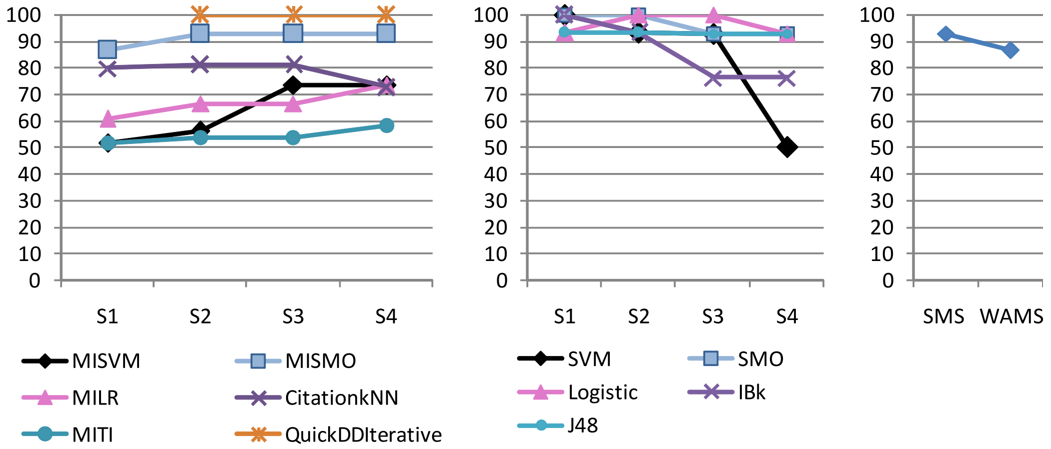
Phenotype	Bacterium ID	Successful predictions
IRRB	B1	8
	B2	8
	B3	8
	B4	8
	B5	8
	B6	8
	B7	8
	B8	8
	B9	8
	B10	8
	B11	**0**
	B12	8
	B13	8
	B14	5[a]
IRSB	B15	**0**
	B16	8
	B17	8
	B18	8
	B19	8
	B20	8
	B21	8
	B22	8
	B23	8
	B24	8
	B25	8
	B26	8
	B27	8
	B28	8

a. Successfully classified bacterium using 5 settings: (1) all proteins with SMS aggregation method (2) replication complex proteins with SMS and WAMS aggregation methods and (3) other DNA-associated proteins with SMS and WAMS aggregation methods.



(a) Naive approach (b) ABClass (c) ABSim

Figure 5.2: Sensitivity results of the naive approach, ABClass and ABSim.



(a) Naive approach (b) ABClass (c) ABSim

Figure 5.3: Specificity results of the naive approach, ABClass and ABSim.

5.3 Conclusion

In this chapter, we described a novel approach called *ABSim* for MIL in sequence data with across-bag dependencies. It uses a matrix to store similarity measurement score vectors to discriminate the related instances. Then it applies an aggregation step in order to generate the final classification result. We applied ABSim and ABClass presented in Chapter 4 to solve the problem of IRR prediction in bacteria. By running experiments, we have shown that the proposed approaches are efficient. A better accuracy result could be provided by *ABClass* according to the used settings.

Chapter 6

Conclusion and prospects

Contents

Goals In this chapter, we conclude the thesis by summarizing our contributions. Then, we highlight the ongoing works we are conducting in extension to this thesis.

6.1 Summary of the contributions

6.1.1 ABClass: a motif-based MIL approach for sequence data with across-bag dependencies

We addressed the issue of MIL in the case of sequence data. We focused on data that present relationships between instances of different bags. The first contribution of this thesis consists of an MIL approach that provides a prediction about the bacterial IRR. We developed a motif-based MIL tool for bacterial IRR prediction. We proposed an MIL formalization of the problem: each bacterium represent a bag and protein sequences represent the instances inside this bag. Some instances are related across the bags: the orthologous proteins. ABClass takes into account this relations in the learning process. Each sequence is represented by one vector of attributes extracted from the set of related instances. For each sequence of the unknown bag, a discriminative classifier is applied in order to compute a partial classification result. Then, an aggregation method is applied in order to generate the final result. We applied ABClass to solve the problem of bacterial Ionizing Radiation Resistance (IRR) prediction. We manually construct the dataset. The experimental results were satisfactory.

6.1.2 ABSim: a similarity-based MIL approach for sequence data with across-bag dependencies

The second contribution of this thesis consists of an MIL approach that uses a similarity measure to compare sequences instead of extracting motifs from related instances and use them to represent the sequences of the bags and then apply a classical classifier to make the prediction. ABSim discriminates bags by measuring the similarity between each sequence in the query bag and its corresponding related sequences in the different bags of the learning database. When applied on protein sequences, ABSim uses the alignment score as similarity measure. ABSim and ABClass were used to solve the problem of IRR prediction in bacteria. By running experiments, we have shown that the proposed approaches are efficient. A better

Table 6.1: Tools related to protein signatures identification.

Tool	Description
InterProScan 5 [Jones et al., 2014]	scans sequences against InterPro signatures.
PfamScan [Mistry et al., 2007] [Li et al., 2015]	searches sequences against a collection of Pfam HMMs.
HMMER-hmmbuild[a]	constructs profiles from multiple sequence alignments
HMMER-hmmscan[a]	searches sequence(s) against a profile database
Pratt [Jonassen, 1997] [Li et al., 2015]	Searches for patterns conserved in sets of unaligned protein sequences.

[a] http://www.hmmer.org/

accuracy result could be provided by ABClass according to the used settings.

6.2 Future work and prospects

In this section, we present the main axes of our future works.

6.2.1 Short-term perspective

ABClass is based on motifs extracted from across-bag related instances. We aim to extend our work by including different protein signatures including patterns and domains in the learning process. We started by exploring the usefulness of using protein domains to solve the bacterial IRR prediction problem. Table 6.1 presents a short description of the tools which could be used to determine protein signatures.

Motifs vs domains

Both motifs and domains are parts of the protein chain. But there are differences between them.

A protein domain could be seen as an **independent** unit which has a **function**. A motif is a particular arrangement of amino acids that can be found in other proteins, it does not necessary depict a functional role. A domain is always a functional unit of the protein. An other main difference is that domains are independent units. If they are cleaved off the protein chain, motifs will loose their functions while domains will be still able to perform their functions.

Using domains in our approach

Using protein domain annotation could be an alternative to sequence similarity searches [Bouchot et al., 2014]. We are exploring the possibility of using protein domains in the classification step. Domain databases were presented in the section 2.1.3.3. We propose to start by using Interproscan tool in order to identify protein domains of each instance of each bag and use them to encode the sequences. Interproscan is a software that allows sequences to be scanned against InterPro's signatures. It is available at `https://www.ebi.ac.uk/interpro/interproscan.html/`. The diagnostic uses protein signatures from multiple databases including Pfam, PROSITE, PRINTS, SMART, SUPERFAMILY, TIGRFAMs and PANTHER. Figure 6.1 shows the InterProScan analysis of the protein *DNA polymerase III subunit alpha* of the bacterium *Deinococcus radiodurans* R1. The provided annotations concern families and domains from different source databases.

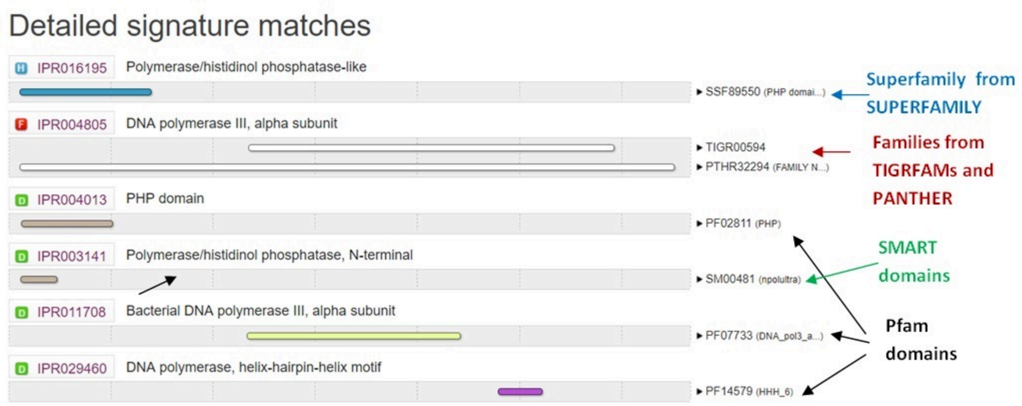
6.2.2 Long-term perspectives

6.2.2.1 Multi-criteria learning

We aim to introduce other criteria in the step of the data representation: using some bio-chemical criteria to represent the sequences instead of using motifs to represent the data. Some criteria could be the *protein domain* and some numeric properties such as hydrophobicity, aromaticity, isoelectric point(pI), instability Index (II), alpha-helix, coil and beta sheet.

6.2.2.2 Defining weights of the protein sequences

We will study how to use the *a priori* knowledge in order to improve the efficiency of our algorithm. In fact, some proteins may have more impact in making a bacterium resistant to ionising radiation than other proteins. We specifically want to define weights for sequences using *a priori* knowledge in the learning phase.



Detailed signature matches

H IPR016195 Polymerase/histidinol phosphatase-like
▸ SSF89550 (PHP domai...)

F IPR004805 DNA polymerase III, alpha subunit
▸ TIGR00594
▸ PTHR32294 (FAMILY N...)

D IPR004013 PHP domain
▸ PF02811 (PHP)

D IPR003141 Polymerase/histidinol phosphatase, N-terminal
▸ SM00481 (npolultra)

D IPR011708 Bacterial DNA polymerase III, alpha subunit
▸ PF07733 (DNA_pol3_a...)

D IPR029460 DNA polymerase, helix-hairpin-helix motif
▸ PF14579 (HHH_6)

Superfamily from SUPERFAMILY
Families from TIGRFAMs and PANTHER
SMART domains
Pfam domains

Figure 6.1: Graphical representation of the InterProScan analysis of the protein P2 of the bacterium B7

6.2.2.3 Extend the dataset

We encountered difficulty in defining the baterial IRR dataset used in this thesis: bags that contain sequences with across-bag dependencies. In the future work, we aim to define a larger dataset in order to study the computational complexity. One possible solution could be to construct a dataset containing genomic sequences of other extremophiles.

Bibliography

Alborzi, S. Z. (2018). *Automatic Discovery of Hidden Associations Using Vector Similarity : Application to Biological Annotation Prediction*. PhD thesis, Université de Lorraine.

Albuquerque, L., Simoes, C., Nobre, M. F., Pino, N. M., Battista, J. R., Silva, M. T., Rainey, F. A., and de Costa, M. S. (2005). Truepera radiovictrix gen. nov., sp. nov., a new radiation resistant species and the proposal of trueperaceae fam. nov. *FEMS microbiology letters*, 247(2):161–169.

Alpaydın, E., Cheplygina, V., Loog, M., and Tax, D. M. (2015). Single-vs. multiple-instance classification. *Pattern Recognition*, 48(9):2831–2838.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402.

Amores, J. (2013). Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 201:81–105.

Amores, J. (2015). MILDE: multiple instance learning by discriminative embedding. *Knowledge and Information Systems*, 42(2):381–407.

Andreeva, A., Howorth, D., Chothia, C., Kulesha, E., and Murzin, A. G. (2013). SCOP2 prototype: a new approach to protein structure mining. *Nucleic acids research*, 42(1):310–314.

Andrews, S., Tsochantaridis, I., and Hofmann, T. (2003). Support Vector Machines for Multiple-Instance Learning. In Thrun and Obermayer, K., editors, *Advances in Neural Information Processing Systems*, pages 561–568, Cambridge, MA. MIT Press.

Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Birney, E., Biswas, M., Bucher, P., Cerutti, L., Corpet, F., Croning, M. D., et al. (2001). The InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic acids research*, 29(1):37–40.

Apweiler, R., Bairoch, A., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., et al. (2004). UniProt: the universal protein knowledgebase. *Nucleic acids research*, 32(suppl 1):115–119.

Aridhi, S., Sghaier, H., Zoghlami, M., Maddouri, M., and Nguifo, E. M. (2016). Prediction of ionizing radiation resistance in bacteria using a multiple instance learning model. *Journal of Computational Biology*, 23(1):10–20.

Attwood, T. K., Bradley, P., Flower, D. R., Gaulton, A., Maudling, N., Mitchell, A. L., Moulton, G., Nordle, A., Paine, K., Taylor, P., et al. (2003). Prints and its automatic supplement, preprints. *Nucleic acids research*, 31(1):400–402.

Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E. L., et al. (2004). The Pfam protein families database. *Nucleic acids research*, 32(suppl 1):138–141.

Bauer, A. L., Hlavacek, W. S., Unkefer, P. J., and Mu, F. (2010). Using sequence-specific chemical and structural properties of DNA to predict transcription factor binding sites. *PLoS computational biology*, 6(11):e1001007.

Benson, D. A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., and Sayers, E. W. (2012). Genbank. *Nucleic acids research*, 41(1):36–42.

Bhagwat, M., Young, L., and Robison, R. R. (2012). Using BLAT to find sequence similarity in closely related genomes. *Current protocols in bioinformatics*, 37(1):10–8.

Billi, D., Friedmann, E., Hofer, K., Caiola, M., and Ocampo-Friedmann, R. (2002). Ionizing-radiation resistance in the desiccation-tolerant cyanobacterium chroococcidiopsis. *Applied and Environmental Microbiology*, 66:1489–1492.

Blekas, K., Fotiadis, D. I., and Likas, A. (2005). Motif-based protein sequence classification using neural networks. *Journal of Computational Biology*, 12(1):64–82.

Bouchot, J. L., Trimble, W. L., Ditzler, G., Lan, Y., Essinger, S., and Rosen, G. (2014). *Advances in Machine Learning for Processing and Comparison of Metagenomic Data*, chapter 14, pages 295 – 329. Academic Press, Oxford, second edition.

Boutet, E., Lieberherr, D., Tognolli, M., Schneider, M., Bansal, P., Bridge, A. J., Poux, S., Bougueleret, L., and Xenarios, I. (2016). UniProtKB/Swiss-Prot, the manually annotated section of the UniProt KnowledgeBase: how to use the entry view. In *Plant Bioinformatics*, pages 23–54. Springer.

Brim, H., McFarlan, S. C., Fredrickson, J. K., Minton, K. W., Zhai, M., Wackett, L. P., and Daly, M. J. (2000). Engineering deinococcus radiodurans for metal remediation in radioactive mixed waste environments. *Nature biotechnology*, 18(1):85–90.

Brim, H., Venkateswaran, A., Kostandarithes, H. M., Fredrickson, J. K., and Daly, M. J. (2003). Engineering deinococcus geothermalis for bioremediation of high-temperature radioactive waste environments. *Applied and environmental microbiology*, 69(8):4575–4582.

Brooks, B. and Murray, R. (1981). Nomenclature for "micrococcus radiodurans" and other radiation-resistant cocci: Deinococcaceae fam. nov. and deinococcus gen. nov., including five species. *Journal of Systematic Bacteriology*, 31:353–360.

Chen, Y., Bi, J., and Wang, J. Z. (2006). MILES: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1931–1947.

Cheng, B. Y. M., Carbonell, J. G., and Klein-Seetharaman, J. (2005). Protein classification based on text document classification techniques. *Proteins: Structure, Function, and Bioinformatics*, 58(4):955–970.

Cheplygina, V., Tax, D. M., and Loog, M. (2015). Multiple instance learning with bag dissimilarities. *Pattern Recognition*, 48(1):264–275.

Chuzhanova, N. A., Jones, A. J., and Margetts, S. (1998). Feature selection for genetic sequence classification. *Bioinformatics (Oxford, England)*, 14(2):139–143.

Daly, M. J., Gaidamakova, E. K., Matrosova, V. Y., Vasilenko, A., Zhai, M., Venkateswaran, A., Hess, M., Omelchenko, M. V., Kostandarithes, H. M., Makarova, K. S., Wackett, L. P., Fredrickson, J. K., and Ghosal, D. (2004). Accumulation of Mn(II) in deinococcus radiodurans facilitates gamma-radiation resistance. *Science*, 306:1025–1028.

Dayhoff, M., Schwartz, R., and Orcutt, B. (1978). A model of evolutionary change in proteins. In *Atlas of protein sequence and structure*, volume 5, pages 345–352.

Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71.

Dong, L. (2006). *A comparison of multi-instance learning algorithms*. PhD thesis, The University of Waikato, Hamilton, New Zealand.

Fang, G., Bhardwaj, N., Robilotto, R., and Gerstein, M. B. (2010). Getting started in gene orthology and functional analysis. *PLoS computational biology*, 6(3):e1000703.

Faria, A. W., Coelho, F. G. F., Silva, A., Rocha, H., Almeida, G., Lemos, A. P., and Braga, A. P. (2017). MILKDE: A new approach for multiple instance learning based on positive instance selection and kernel density estimation. *Engineering Applications of Artificial Intelligence*, 59:196–204.

Federhen, S., Clark, K., Barrett, T., Parkinson, H., Ostell, J., Kodama, Y., Mashima, J., Nakamura, Y., Cochrane, G., and Karsch-Mizrachi, I. (2014).

Toward richer metadata for microbial sequences: replacing strain-level ncbi taxonomy taxids with bioproject, biosample and assembly records. *Standards in genomic sciences*, 9(3):1275.

Federighi, M. and Tholozan, J.-L. (2001). *Traitements ionisants et hautes pressions des aliments*, page 161–169. Economica.

Fedorov, D. N., Ekimova, G. A., Doronina, N. V., and Trotsenko, Y. A. (2013). 1-Aminocyclopropane-1-carboxylate (ACC) deaminases from Methylobacterium radiotolerans and Methylobacterium nodulans with higher specificity for ACC. *FEMS Microbiol Lett*, 343(1):70–76.

Ferreira, A. C., Nobre, M. F., Moore, E., Rainey, F. A., Battista, J. R., and da Costa, M. S. (1999). Characterization and radiation resistance of new isolates of rubrobacter radiotolerans and rubrobacter xylanophilus. *Extremophiles*, 3(4):235–238.

Finn, R. D., Attwood, T. K., Babbitt, P. C., Bateman, A., Bork, P., Bridge, A. J., Chang, H.-Y., Dosztányi, Z., El-Gebali, S., Fraser, M., et al. (2016). InterPro in 2017- beyond protein family and domain annotations. *Nucleic acids research*, 45(1):190–199.

Finn, R. D., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Mistry, J., Mitchell, A. L., Potter, S. C., Punta, M., Qureshi, M., Sangrador-Vegas, A., et al. (2015). The Pfam protein families database: towards a more sustainable future. *Nucleic acids research*, 44(1):279–285.

Foulds, J. and Frank, E. (2010). A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25(1):1–25.

Gabani, P. and Singh, O. V. (2013). Radiation-resistant extremophiles and their potential in biotechnology and therapeutics. *Applied microbiology and biotechnology*, 97(3):993–1004.

Gane, P., Bateman, A., Mj, M., O'Donovan, C., Magrane, M., Apweiler, R., Alpi, E., Antunes, R., Arganiska, J., Bely, B., Bingley, M., Bonilla, C., Britto, R., Bursteinas, B., Chavali, G., Cibrián-Uhalte, E., Ad, S., De Giorgi, M., Dogan, T., and Zhang, J. (2014). Uniprot: A hub for protein information. *Nucleic Acids Research*, 43:204–212.

Gao, Z. and Ruan, J. (2013). A structure-based multiple-instance learning approach to predicting in vitro transcription factor-DNA interaction. *BMC genomics*, 16:9–9.

Gracy, J. and Argos, P. (1998). Automated protein sequence database classification. I. Integration of compositional similarity search, local similarity search, and multiple sequence alignment. *Bioinformatics*, 14(2):164–173.

Green, P. and Bousfield, I. (1983). Emendation of methylobacterium; methylobacterium rhodinum comb. nov. corrig.; methylobacterium radiotolerans comb. nov. corrig.; and methylobacterium mesophilicum comb. nov. *International Journal of Bacteriology*, 33:875–877.

Gtari, M., Essoussi, I., Maaoui, R., Sghaier, H., Boujmil, R., Gury, J., Pujic, P., Brusetti, L., Chouaia, B., Crotti, E., Daffonchio, D., Boudabous, A., and Normand, P. (2012). Contrasted resistance of stone-dwelling geodermatophilaceae species to stresses known to give rise to reactive oxygen species. *FEMS Microbiology Ecology*, 80(3):566–577.

Haft, D. H., Selengut, J. D., Richter, R. A., Harkins, D., Basu, M. K., and Beck, E. (2012). Tigrfams and genome properties in 2013. *Nucleic acids research*, 41(1):387–395.

Haft, D. H., Selengut, J. D., and White, O. (2003). The tigrfams database of protein families. *Nucleic acids research*, 31(1):371–373.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Halling, S. M., Peterson-Burch, B. D., Bricker, B. J., Zuerner, R. L., Qing, Z., Li, L.-L., Kapur, V., Alt, D. P., and Olsen, S. C. (2005). Completion of the genome sequence of brucella abortus and comparison to the highly similar genomes of brucella melitensis and brucella suis. *Journal of Bacteriology*, 187(8):2715–2726.

Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919.

Herrera, F., Ventura, S., Bello, R., Cornelis, C., Zafra, A., Sánchez-Tarragó, D., and Vluymans, S. (2016). *Multiple instance learning: foundations and algorithms*. Springer.

Hu, J., Wang, J., Lin, J., Liu, T., Zhong, Y., Liu, J., Zheng, Y., Gao, Y., He, J., and Shang, X. (2019). MD-SVM: a novel SVM-based algorithm for the motif discovery of transcription factor binding sites. *BMC bioinformatics*, 20(7):200.

Ito, H. and Iizuka, H. (1971). Taxonomic studies on a radio-resistant pseudomonas. Part XII. studies on the microorganisms of cereal grain. *Agricultural and Biological Chemistry*, 35:1566–1571.

Ito, H., W. H., Takeshia, M., and Iizuka, H. (1983). Isolation and identification of radiation-resistant cocci belonging to the genus deinococcus from sewage sludges and animal feeds. *Agricultural and Biological Chemistry*, 47:1239–1247.

Jonassen, I. (1997). Efficient discovery of conserved patterns using a pattern graph. *Bioinformatics*, 13(5):509–522.

Jones, P., Binns, D., Chang, H.-Y., Fraser, M., Li, W., McAnulla, C., McWilliam, H., Maslen, J., Mitchell, A., Nuka, G., et al. (2014). InterProScan 5: genome-scale protein function classification. *Bioinformatics*, 30(9):1236–1240.

Kandemir, M. and Hamprecht, F. A. (2015). Computer-aided diagnosis from weak supervision: a benchmarking study. *Computerized medical imaging and graphics*, 42:44–50.

Kelley, J. (1960). The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, pages 703–712.

Kent, W. J. (2002). BLAT - the BLAST-like alignment tool. *Genome research*, 12(4):656–664.

Kim, J., Moriyama, E. N., Warr, C. G., Clyne, P. J., and Carlson, J. R. (2000). Identification of novel multi-transmembrane proteins from genomic databases using quasi-periodic structural properties. *Bioinformatics*, 16(9):767–775.

Leinonen, R., Diez, F. G., Binns, D., Fleischmann, W., Lopez, R., and Apweiler, R. (2004). UniProt archive. *Bioinformatics*, 20(17):3236–3237.

Lesh, N., Zaki, M. J., and Ogihara, M. (1999). Mining features for sequence classification. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 342–346. ACM.

Letunic, I., Doerks, T., and Bork, P. (2011). SMART 7: recent updates to the protein domain annotation resource. *Nucleic acids research*, 40(1):302–305.

Li, W., Cowley, A., Uludag, M., Gur, T., McWilliam, H., Squizzato, S., Park, Y. M., Buso, N., and Lopez, R. (2015). The EMBL-EBI bioinformatics web and programmatic tools framework. *Nucleic acids research*, 43(1):580–584.

Li, Z., Geng, G.-H., Feng, J., Peng, J.-y., Wen, C., and Liang, J.-l. (2014). Multiple instance learning based on positive instance selection and bag structure construction. *Pattern Recognition Letters*, 40:19–26.

Liolios, K., Mavromatis, K., Tavernarakis, N., and Kyrpides, N. C. (2008). The Genomes On Line Database (GOLD) in 2007: status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Research*, 36(suppl 1):475–479.

Liu, G., Wu, J., and Zhou, Z.-H. (2012). Key instance detection in multi-instance learning. *Journal of Machine Learning Research*, 25:253–268.

Lukjancenko, O., Wassenaar, T. M., and Ussery, D. W. (2010). Comparison of 61 sequenced Escherichia coli genomes. *Microbial ecology*, 60(4):708–720.

Maddouri, M. and Elloumi, M. (2004). Encoding of primary structures of biological macromolecules within a data mining perspective. *Journal of Computer Science and Technology*, 19(1):78–88.

Makarova, K. S., Omelchenko, M. V., Gaidamakova, E. K., Matrosova, V. Y., Vasilenko, A., Zhai, M., Lapidus, A., Copeland, A., Kim, E., Land, M., et al. (2007). Deinococcus geothermalis: the pool of extreme radiation resistance genes shrinks. *PLoS One*, 2(9):e955.

Marchler-Bauer, A., Anderson, J. B., Cherukuri, P. F., DeWeese-Scott, C., Geer, L. Y., Gwadz, M., He, S., Hurwitz, D. I., Jackson, J. D., Ke, Z., et al. (2005). CDD: a conserved domain database for protein classification. *Nucleic acids research*, 33(suppl 1):192–196.

Marchler-Bauer, A., Derbyshire, M. K., Gonzales, N. R., Lu, S., Chitsaz, F., Geer, L. Y., Geer, R. C., He, J., Gwadz, M., Hurwitz, D. I., et al. (2014). CDD: NCBI's conserved domain database. *Nucleic acids research*, 43(1):222–226.

Maron, O. and Lozano-Pérez, T. (1998). A framework for multiple-instance learning. In *Proceedings of the 1997 conference on Advances in neural information processing systems 10*, NIPS '97, pages 570–576, Cambridge, MA, USA. MIT Press.

Maron, O. and Ratan, A. L. (1998). Multiple-instance learning for natural scene classification. In *ICML*, volume 98, pages 341–349.

Melki, G., Cano, A., and Ventura, S. (2018). MIRSVM: multi-instance support vector machine with bag representatives. *Pattern Recognition*, 79:228–241.

Menichelli, C., Gascuel, O., and Bréhélin, L. (2018). Improving pairwise comparison of protein sequences with domain co-occurrence. *PLoS computational biology*, 14(1):e1005889.

Mi, H., Huang, X., Muruganujan, A., Tang, H., Mills, C., Kang, D., and Thomas, P. D. (2016). PANTHER version 11: expanded annotation data from gene ontology and reactome pathways, and data analysis tool enhancements. *Nucleic acids research*, 45(1):183–189.

Mistry, J., Bateman, A., and Finn, R. D. (2007). Predicting active site residue annotations in the Pfam database. *BMC bioinformatics*, 8(1):298.

Mukherjee, S., Stamatis, D., Bertsch, J., Ovchinnikova, G., Verezemska, O., Isbandi, M., Thomas, A. D., Ali, R., Sharma, K., Kyrpides, N. C., et al. (2016). Genomes OnLine Database (GOLD) v. 6: data updates and feature enhancements. *Nucleic acids research*, page gkw992.

Murzin, A. G., Brenner, S. E., Hubbard, T., and Chothia, C. (1995). SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

Ni, H. M., Qi, D. W., and Mu, H. (2018). Applying MSSIM combined chaos game representation to genome sequences analysis. *Genomics*, 110(3):180–190.

Orengo, C. A., Michie, A., Jones, S., Jones, D. T., Swindells, M., and Thornton, J. M. (1997). CATH - a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109.

Pagnuco, I. A., Pastore, J. I., Abras, G., Brun, M., and Ballarin, V. L. (2017). Analysis of genetic association using hierarchical clustering and cluster validation indices. *Genomics*, 109(5-6):438–445.

Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076.

Pearl, F. M. G., Bennett, C., Bray, J. E., Harrison, A. P., Martin, N., Shepherd, A., Sillitoe, I., Thornton, J., and Orengo, C. A. (2003). The CATH database: an extended protein family resource for structural and functional genomics. *Nucleic acids research*, 31(1):452–455.

Phillips, R. W., Wiegel, J., Berry, C. J., Fliermans, C., Peacock, A. D., White, D. C., and Shimkets, L. J. (2002). Kineococcus radiotolerans sp. nov., a radiation-resistant, gram-positive bacterium. *International journal of systematic and evolutionary microbiology*, 52(3):933–938.

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. MIT Press.

Pruitt, K. D., Tatusova, T., Brown, G. R., and Maglott, D. R. (2011). NCBI Reference Sequences (RefSeq): current status, new features and genome annotation policy. *Nucleic acids research*, 40(1):130–135.

Rainey, F., Nobre, M.F., S. P. S. E., and da Costa, M. (1997). Phylogenetic diversity of the deinococci as determined by 16s ribosomal DNA sequence comparison. *International Journal of Bacteriology*, 47:510–514.

Rainey, F. A., Ray, K., Ferreira, M., Gatz, B. Z., Nobre, M. F., Bagaley, D., Rash, B. A., Park, M.-J., Earl, A. M., Shank, N. C., et al. (2005). Extensive diversity of ionizing-radiation-resistant bacteria recovered from sonoran desert soil and description of nine new species of the genus deinococcus obtained from a single soil sample. *Applied and environmental microbiology*, 71(9):5225–5235.

Ray, S. and Craven, M. (2005). Learning statistical models for annotating proteins with function information using biomedical text. *BMC bioinformatics*, 6(1):S18.

Saigo, H., Vert, J.-P., Ueda, N., and Akutsu, T. (2004). Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682–1689.

Santiago-Sotelo, P. and Ramirez-Prado, J. H. (2012). prfectBLAST: a platform-independent portable front end for the command terminal BLAST+ stand-alone suite. *BioTechniques*, 53(5):299–300.

Schultz, J., Milpetz, F., Bork, P., and Ponting, C. P. (1998). SMART, a simple modular architecture research tool: identification of signaling domains. *Proceedings of the National Academy of Sciences*, 95(11):5857–5864.

Sghaier, H., Ghedira, K., Benkahla, A., and Barkallah, I. (2008). Basal dna repair machinery is subject to positive selection in ionizing-radiation-resistant bacteria. *BMC genomics*, 9(1):297.

She, R., Chen, F., Wang, K., Ester, M., Gardy, J. L., and Brinkman, F. S. (2003). Frequent-subsequence-based prediction of outer membrane proteins. In *Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 436–445. ACM.

Slade, D. and Radman, M. (2011). Oxidative stress resistance in deinococcus radiodurans. *Microbiology and Molecular Biology Reviews*, 75:133–191.

Smola, A. J., Vishwanathan, S., and Hofmann, T. (2005). Kernel methods for missing variables. In *Proceedings of International Workshop on Artificial Intelligence and Statistics*, pages 325–332.

Sniedovich, M. (2010). *Dynamic programming: foundations and principles*. CRC press.

Srivastava, P. K., Desai, D. K., Nandi, S., and Lynn, A. M. (2007). HMM-ModE-Improved classification using profile hidden Markov models by optimising the discrimination threshold and modifying emission probabilities with negative training sequences. *BMC bioinformatics*, 8(1):104.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Suzek, B. E., Huang, H., McGarvey, P., Mazumder, R., and Wu, C. H. (2007). UniRef: comprehensive and non-redundant UniProt reference clusters. *Bioinformatics*, 23(10):1282–1288.

Tao, Q., Scott, S., Vinodchandran, N., and Osugi, T. T. (2004). SVM-based generalized multiple-instance learning via approximate box counting. In *Proceedings of the twenty-first international conference on Machine learning*, pages 799–806.

Thomas, P. D., Kejariwal, A., Campbell, M. J., Mi, H., Diemer, K., Guo, N., Ladunga, I., Ulitsky-Lazareva, B., Muruganujan, A., Rabkin, S., et al. (2003). PANTHER: a browsable database of gene products organized by biological function, using curated protein family and subfamily classification. *Nucleic acids research*, 31(1):334–341.

Vinga, S. and Almeida, J. (2003). Alignment-free sequence comparison - a review. *Bioinformatics*, 19(4):513–523.

Wang, J. and Zucker, J.-D. (2000). Solving the multiple-instance problem: A lazy learning approach. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 1119–1126, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Wang, W., Ning, Y., Rangwala, H., and Ramakrishnan, N. (2016). A multiple instance learning framework for identifying key sentences and detecting events. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 509–518. ACM.

Waterman, M. S. (1981). Identification of common molecular subsequence. *Journal of Molecular Biology*, 147:195–197.

Wei, D., Jiang, Q., Wei, Y., and Wang, S. (2012). A novel hierarchical clustering algorithm for gene sequences. *BMC bioinformatics*, 13(1):174.

Wei, X.-S., Wu, J., and Zhou, Z.-H. (2016). Scalable algorithms for multi-instance learning. *IEEE transactions on neural networks and learning systems*, 28(4):975–987.

Woolfit, M. and Bromham, L. (2003). Increased rates of sequence evolution in endosymbiotic bacteria and fungi with small effective population sizes. *Molecular Biology and Evolution*, 20(9):1545–1555.

Xing, Z., Pei, J., and Keogh, E. (2010). A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, 12(1):40–48.

Yakhnenko, O., Silvescu, A., and Honavar, V. (2005). Discriminatively trained markov model for sequence classification. In *Proceedings of the fifth IEEE International Conference on Data Mining*, pages 498–505.

Yuan, J., Huang, X., Liu, H., Li, B., and Xiong, W. (2016). SubMIL: discriminative subspaces for multi-instance learning. *Neurocomputing*, 173:1768–1774.

Zhang, D., Liu, Y., Si, L., Zhang, J., and Lawrence, R. D. (2011). Multiple instance learning on structured data. In *Advances in Neural Information Processing Systems*, pages 145–153.

Zhang, Q. and Goldman, S. A. (2002). EM-DD: an improved multiple-instance learning technique. In *Advances in neural information processing systems*, pages 1073–1080.

Zhang, Q., Shen, Z., and Huang, D.-S. (2019). Modeling in-vivo protein-DNA binding by combining multiple-instance learning with a hybrid deep neural network. *Scientific reports*, 9(1):8484.

Zhou, Z.-H., Sun, Y.-Y., and Li, Y.-F. (2009). Multi-instance learning by treating instances as non-iid samples. In *Proceedings of the 26th annual international conference on machine learning*, pages 1249–1256. ACM.

Zoghlami, M., Aridhi, S., Maddouri, M., and Nguifo, E. M. (2018a). ABClass: une approche d'apprentissage multi-instances pour les séquences(ABClass: A multiple instance learning approach for sequence data). In *Actes de la Conférence Nationale d'Intelligence Artificielle et Rencontres des Jeunes Chercheurs en Intelligence Artificielle (CNIA+RJCIA 2018), Nancy, France*, pages 10–18.

Zoghlami, M., Aridhi, S., Maddouri, M., and Nguifo, E. M. (2018b). An overview of in silico methods for the prediction of ionizing radiation resistance in bacteria. In *Ionizing Radiation: Advances in Research and Applications*, Physics Research and Technology Series. Nova Science Publishers Inc.

Zoghlami, M., Aridhi, S., Maddouri, M., and Nguifo, E. M. (2019a). Multiple instance learning for sequence data with across bag dependencies. *International*

Journal of Machine Learning and Cybernetics, DOI:10.1007/s13042-019-01021-5.

Zoghlami, M., Aridhi, S., Maddouri, M., and Nguifo, E. M. (2019b). A structure based multiple instance learning approach for bacterial ionizing radiation resistance prediction. *23rd International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, Budapest, Hungary*.

Appendix A

Further details about the dataset

This appendix gives further details about the bags and instances of the used dataset. Table A.1 contains the number of proteins for each bacterium and Table A.2 contains the number of occurrences of each type of protein sequence in the positive bags (IRRB) and in the negative bags (IRSB).

Table A.1: Number of protein sequences for each bacterium.

IRRB ID	Number of proteins	IRSB ID	Number of proteins
B1	25	B15	28
B2	31	B16	30
B3	31	B17	27
B4	30	B18	25
B5	30	B19	26
B6	30	B20	29
B7	31	B21	29
B8	29	B22	30
B9	25	B23	31
B10	28	B24	31
B11	28	B25	31
B12	28	B26	31
B13	27	B27	27
B14	30	B28	30
Total for IRRB	403	Total for IRSB	405

Table A.2: Number of occurrences of each type of protein sequence in the positive and negative bags.

Protein ID	Positive bags	Negative bags
P1	11	4
P2	14	14
P3	14	13
P4	13	14
P5	13	14
P6	13	14
P7	14	14
P8	11	14
P9	14	14
P10	13	14
P11	14	14
P12	13	14
P13	12	14
P14	14	14
P15	14	14
P16	13	12
P17	14	14
P18	14	14
P19	13	14
P20	14	14
P21	14	13
P22	14	13
P23	13	14
P24	13	14
P25	11	10
P26	10	10
P27	11	14
P28	13	14
P29	12	14
P30	14	13
P31	13	9
Total	403	405

Appendix B
Examples of sequence alignment

In this appendix, we provide two sequence alignment results provided by BLAST.

- **Alignment of the two protein sequences P4 of the two bacteria B6 and B7 (Two IRRB)**.

```
BLASTP 2.2.26+
Query= tr|Q9RRS5|Q9RRS5_DEIRA DNA polymerase III, tau/gamma subunit
OS=Deinococcus radiodurans (strain ATCC 13939 / DSM 20539 / JCM
16871 / LMG 4051 / NBRC 15346 / NCIMB 9279 / R1 / VKM B-1422)
GN=DR_2410 PE=4 SV=1
Length=615
Subject= gi|325283277|ref|YP_004255818.1| DNA polymerase III, subunits gamma
and tau [Deinococcus proteolyticus MRP]
Length=810
 Score =  655 bits (1691),  Expect = 0.0, Method: Compositional matrix adjust.
 Identities = 368/523 (70%), Positives = 417/523 (80%), Gaps = 25/523 (5%)
Query  1    MSAIYQRARPIRWEDVVGQEHVKDVLRTALEQGRIGHAYLFSGPRGVGKTTTARLIAMTA  60
            MSAIYQRARPI W++VVGQEH+K VL+TALEQGR+GHAYLFSGPRGVGKTTTARLIAMTA
Sbjct  1    MSAIYQRARPIHWDEVVGQEHIKGVLKTALEQGRVGHAYLFSGPRGVGKTTTARLIAMTA  60
Query  61   NCTGPAPKPCGECESCLAVRAGSHPDVMEIDAASNNSVDDVRDLREKVGLAAMRGGKKIY  120
            NCTGP PKPCGECE+C AVRAGSHPDV+EIDAASNNSV+DVR+LREKVGLA MRGGKKIY
Sbjct  61   NCTGPQPKPCGECENCRAVRAGSHPDVLEIDAASNNSVEDVRELREKVGLAPMRGGKKIY  120
Query  121  ILDEAHMMSRAAFNALLKTLEEPPEHVIFILATTEPEKIIPTILSRCQHYRFRRLTSEEI  180
            ILDEAHMMSRAAFNALLKTLEEPPEHVIFILATTEPEKIIPTILSRCQHYRFRRLT+EEI
Sbjct  121  ILDEAHMMSRAAFNALLKTLEEPPEHVIFILATTEPEKIIPTILSRCQHYRFRRLTAEEI  180
Query  181  AGKLAGLVTLEGASADPDALNLIGRLADGAMRDGESLLERMLAAGTAVTRPAVEEALGLP  240
            AGKLAGL   EG SA+P+AL LIGRLADGAMRDGESLLERMLAAGTAVTR +VEEALGLP
Sbjct  181  AGKLAGLAEGEGVSAEPEALGLIGRLADGAMRDGESLLERMLAAGTAVTRRSVEEALGLP  240
Query  241  PGERVRGVASALLVGDAGEAISGAAQLYRDGFAARTVVEGLVAAFGAALHAELGL-----  295
            PGE++R +A AL  GDAG A+S A +LYR GFAARTVVEGLV A   A+HAELG+
Sbjct  241  PGEQMRALAGALAQGDAGPALSSAGELYRAGFAARTVVEGLVEALSQAIHAELGVLEGAE  300
Query  296  GEEGRLEGAEVPRLLKLQAALDEQEARFARSADQQS----LELALTHALLAADGGTGGGA  351
             +  RL+GA+VPRLL+LQAALDEQEARF+R+AD  S    L  AL  A   ADG  GGGA
Sbjct  301  AQAARLDGADVPRLLRLQAALDEQEARFSRAADLLSLELALTHALLAADGGADGSAGGGA  360
Query  352  PSLGSAATSAPAQVPGDLLQRLNRLEKELSTLRSAPRAAAPASAVPAAPA--------EK  403
             +  +AA +A   V  DL  RL+RLE+EL+ LR+   A APA+A PA PA         +
Sbjct  361  AAARAAAPAASPAVSSDLAARLSRLERELAALRAGESAVAPAAAAPAGPAVDDFDPGQRR  420
Query  404  RGPAPAREAVREAAASIAP-AAAPTQGSWADVMAQTTMQMRAFLKPARMHAQDGYVSLTY  462
```

```
            R PAP         A  AP  AAP  G+WADV+   +MQ RAFLKPARMHA+ GYVSL+Y
Sbjct  421  RTPAP-------VGARPAPQVAAPANGTWADVLGMVSMQTRAFLKPARMHAEAGYVSLSY  473
Query  463  EDRSSFHAKQVAGKFDELAALVERVFGPITFELIAPEGLGRKR  505
            + + SFHA+Q+  K DEL  L+ERVFGP+T ELI  +G G ++
Sbjct  474  DAKGSFHARQIMTKLDELTPLLERVFGPVTLELITADGSGGRK  516
Lambda     K      H
   0.315    0.130    0.375
Gapped
Lambda     K      H
   0.267   0.0410    0.140
Effective search space used: 444096
Matrix: BLOSUM62
Gap Penalties: Existence: 11, Extension: 1
Neighboring words threshold: 11
Window for multiple hits: 40
```

- Alignment of the two protein sequences P4 of the two bacteria B7 (IRRB) and B16 (IRSB)

```
BLASTP 2.2.26+
Query= tr|Q9RRS5|Q9RRS5_DEIRA DNA polymerase III, tau/gamma subunit
OS=Deinococcus radiodurans (strain ATCC 13939 / DSM 20539 / JCM
16871 / LMG 4051 / NBRC 15346 / NCIMB 9279 / R1 / VKM B-1422)
GN=DR_2410 PE=4 SV=1
Length=615
Subject= gi|254160539|ref|YP_003043647.1| DNA polymerase III subunits gamma
and tau [Escherichia coli B str. REL606]
Length=643
 Score =  230 bits (586),  Expect = 3e-070, Method: Compositional matrix adjust.
 Identities = 118/250 (47%), Positives = 161/250 (64%), Gaps = 2/250 (1%)
Query  6    QRARPIRWEDVVGQEHVKDVLRTALEQGRIGHAYLFSGPRGVGKTTTARLIAMTANC-TG  64
            ++ RP  + DVVGQEHV   L   L  GRI HAYLFSG RGVGKT+ ARL+A   NC TG
Sbjct  8    RKWRPQTFADVVGQEHVLTALANGLSLGRIHHAYLFSGTRGVGKTSIARLLAKGLNCETG  67
Query  65   PAPKPCGECESCLAVRAGSHPDVMEIDAASNNSVDDVRDLREKVGLAAMRGGKKIYILDE  124
                PCG C++C  +  G   D++EIDAAS   V+D RDL + V  A  RG  K+Y++DE
Sbjct  68   ITATPCGVCDNCREIEQGRFVDLIEIDAASRTKVEDTRDLLDNVQYAPARGRFKVYLIDE  127
Query  125  AHMMSRAAFNALLKTLEEPPEHVIFILATTEPEKIIPTILSRCQHYRFRRLTSEEIAGKL  184
             HM+SR +FNALLKTLEEPPEHV F+LATT+P+K+  TILSRC  +  + L  E+I  +L
Sbjct  128  VHMLSRHSFNALLKTLEEPPEHVKFLLATTDPQKLPVTILSRCLQFHLKALDVEQIRHQL  187
Query  185  AGLVTLEGASADPDALNLIGRLADGAMRDGESLLERMLAAGT-AVTRPAVEEALGLPPGE  243
              ++  E  + +P AL L+ R A+G++RD  SL ++ +A+G   V+  AV   LG    +
Sbjct  188  EHILNEEHIAHEPRALQLLARAAEGSLRDALSLTDQAIASGDGQVSTQAVSAMLGTLDDD  247
Query  244  RVRGVASALL  253
            +   +  A++
Sbjct  248  QALSLVEAMV  257
 Score = 18.5 bits (36),  Expect = 0.84, Method: Compositional matrix adjust.
 Identities = 10/24 (42%), Positives = 13/24 (54%), Gaps = 0/24 (0%)
Query  404  RGPAPAREAVREAAASIAPAAAPT  427
```

```
            R P P  E  R++ A +AP A  T
Sbjct  362  RMPLPEPEVPRQSFAPVAPTAVMT  385
Lambda     K      H
   0.315    0.130    0.375
Gapped
Lambda     K      H
   0.267   0.0410    0.140
Effective search space used: 349085
Matrix: BLOSUM62
Gap Penalties: Existence: 11, Extension: 1
Neighboring words threshold: 11
Window for multiple hits: 40
```

Multiple instance learning for sequence data: Application on bacterial ionizing radiation resistance prediction

Abstract:

In Multiple Instance Learning (MIL) problem for sequence data, the instances inside the bags are sequences. In some real world applications such as bioinformatics, comparing a random couple of sequences makes no sense. In fact, each instance may have structural and/or functional relationship with instances of other bags. Thus, the classification task should take into account this across-bag relationship. In this thesis, we present two novel MIL approaches for sequence data classification named *ABClass* and *ABSim*. *ABClass* extracts motifs from related instances and use them to encode sequences. A discriminative classifier is then applied to compute a partial classification result for each set of related sequences. *ABSim* uses a similarity measure to discriminate the related instances and to compute a scores matrix. For both approaches, an aggregation method is applied in order to generate the final classification result. We applied both approaches to the problem of bacterial ionizing radiation resistance prediction. The experimental results were satisfactory.

Keywords: multiple instance learning, sequence data classification, prediction of bacterial ionizing radiation resistance.

Apprentissage multi-instance des données de séquences: Application à la prédiction de la radio-résistance chez les bactéries.

Resumé:

Dans l'apprentissage multi-instances (MI) pour les séquences, les données d'apprentissage consistent en un ensemble de sacs où chaque sac contient un ensemble d'instances/séquences. Dans certaines applications du monde réel, comme la bioinformatique, comparer un couple aléatoire de séquences n'a aucun sens. En fait, chaque instance de chaque sac peut avoir une relation structurelle et/ou fonctionnelle avec d'autres instances dans d'autres sacs. Ainsi, la tâche de classification doit prendre en compte la relation entre les instances sémantiquement liées à travers les sacs. Dans cette thèse, nous présentons deux approches de classification MI des séquences nommées *ABClass* et *ABSim*. ABClass extrait les motifs à partir des instances reliées et les utilise pour encoder les séquences. Un classifieur discriminant est ensuite appliqué pour calculer un résultat de classification partiel pour chaque ensemble de séquences liées. ABSim utilise une mesure de similarité pour discriminer les instances reliées et calcule une matrice de scores. Pour les deux approches, une méthode d'agrégation est appliquée afin de générer le résultat final de la classification. Nous appliquons les deux approches au problème de prédiction de la résistance aux rayonnements ionisants chez les bactéries. Les résultats expérimentaux sont satisfaisants.

Mots-clés: apprentissage multi-instances, classification des séquences , prédiction de la résistance aux rayonnements ionisants chez les bactéries.