INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Classification of clinical expression time series: a case study in patients with Multiple Sclerosis

André Valério Raposo Carreiro

Dissertation submitted to obtain the Master's Degree in

Biomedical Engineering

Jury

President:	Paulo Jorge Peixeiro de Freitas, PhD
Supervisor:	Sara Alexandra Cordeiro Madeira, PhD
Co-supervisor:	João André Nogueira Custódio Carriço, PhD
Members:	Cláudia Martins Antunes, PhD
	Francisco Rodrigues Pinto, PhD

November 2010

Acknowledgments

I start by thanking all my family, especially my parents Mitó and Luís, and my sister Ana Teresa, for all the support, far beyond the months I took working on my master thesis. For as long as I can remember, they constantly showed their faith in me as I was always encouraged to give my best in every situation. I would like to also address my grandparents, as I feel that, in some way, I am fulfilling one of their dreams.

Now, I also thank my girlfriend, and best friend, Catarina, with whom I shared my joys and (fortunately only a few) frustrations, always having a loving word to offer.

To all my fellow Biomedical Engineering students at IST, especially my class of 2005 (and friends), I express my appreciation for all the companionship and real brotherhood. They provided me a “family” away from home, offering me many beautiful moments I will never forget, even considering the “all-nighters” we all endured in the LTI. I also address a special note of gratitude to all my friends from my home island, Santa Maria, who helped me growing into the person I am today. To name a few (in no special order): Carlos, João, Angela, Sara, Ana, Joana, Isabel, Henrique, Marina, Linda, Aléxio and the three Pedro’s.

Furthermore, I express my gratitude to my supervisors, João Carriço and Sara Madeira, for all the precious support throughout the development of this thesis and for the confidence they showed in my effort, even when the results were not as good as we might have expected. To Sara Madeira, also for always being available to assist me in whatever I needed and for all the brainstorming sessions we set to overcome the rising challenges. In this point I include the help of Orlando in these discussions and with state of the art classifiers. Finally, I greatly appreciate the opportunity to integrate the KDBio (Knowledge Discovery and Bioinformatics) group at INESC-ID, which I found to be a very enriching experience.

Abstract

The constant drive towards a more personalized medicine in the last years led to the arrival of temporal gene expression analyses. Due to the consideration of a temporal aspect, this kind of analyses represents a great advantage to better understand disease progression and treatment results at a molecular level. Nevertheless, several problems accompany studies of this kind, with the sample size limitation being one of the most relevant. This limitation is patent in two ways: in the number of objects (in this case, patients), and in the number of measured time points. In this work, the data used were multiple gene expression time series, used to classify the response of multiple sclerosis patients to the standard treatment with Interferon-β, to which nearly half of the patients reveal a negative response. Therefore, obtaining a highly predictive model of a patient's response would definitely improve the quality of life, avoiding useless and possibly harmful therapies for the non-responder group.

In this context, several new strategies for time series classification are proposed, based on a biclustering technique. These are applied to the classification of the Interferon-β response by multiple sclerosis patients from a dataset analyzed over the last decade. Although our classification methods do not outperform the ones proposed for this same dataset, it is worth noting that some of the developed strategies reveal important potentialities that should be further explored, either with other clinical time series data, or even in other classification problems, in general.

Keywords: time series, biclustering, bioinformatics, multiple sclerosis, IFN-Beta, data mining

Resumo

O constante desenvolvimento visando uma medicina cada vez mais personalizada nos últimos anos conduziu ao aparecimento de análises temporais de expressão genética. Devido à consideração do aspecto temporal, este tipo de análises representa uma grande vantagem num melhor conhecimento da progressão e tratamento de doenças. Todavia, variados desafios acompanham este tipo de estudos, sendo o tamanho da amostra um dos mais relevantes. Esta limitação está patente em duas formas: no número de objectos (neste caso, pacientes) e no número de medições temporais. Neste trabalho, os dados usados são provenientes de múltiplas séries temporais de expressão genética, usadas para classificar a resposta de pacientes com esclerose múltipla ao tratamento com Interferão-β, ao qual aproximadamente metade dos pacientes apresenta uma resposta negativa. Assim, a obtenção de um modelo altamente predictivo da resposta destes pacientes permitiria certamente melhorias significativas na qualidade de vida, evitando terapias desnecessárias e muitas vezes acompanhadas de efeitos adversos, especialmente para o grupo de pacientes com resposta negativa.

Neste contexto, são propostas algumas estratégias de classificação de séries temporais baseadas numa técnica de *biclustering,* sendo aplicadas à classificação da resposta ao tratamento com Interferão-β para esclerose múltipla, num conjunto de dados já analisado anteriormente. Embora os resultados não tenham ultrapassado as abordagens anteriormente descritas para este problema em particular, realça-se que algumas das estratégias desenvolvidas apresentam importantes potencialidades, encorajando-se uma análise mais extensiva, abordando outros dados de séries temporais de origem clínica, ou mesmo outros problemas de classificação em geral.

Palavras-chave: séries temporais, *biclustering*, bioinformática, esclerose múltipla, IFN-Beta, *data mining*

Contents

List of Figures

List of Tables

List of Abbreviations

AUC – Area under the curve
CCC-Bicluster – Contiguous column coherent bicluster
CNS – Central nervous system
CV – Cross validation
DNA – Deoxyribonucleic acid
EDSS – Extended Disability Status Scale
EM – Expectation-Maximization
FC – Fold change (expression level)
HLA – Human leukocyte antigen
HMM – Hidden Markov Model
HUVEC – Human umbilical vein endothelial cells
IFN – Interferon
IL – Interleukin
IMSGC – International Multiple Sclerosis Genetics Consortium
kNN – k–Nearest Neighbors
LOO – Leave-One-Out
MHC – Major Histocompatibility Complex
MLP – Multilayer Perceptron
MRI – Magnetic Resonance Imaging
mRNA – Messenger ribonucleic acid
MS – Multiple sclerosis
PBMC – Peripheral blood mononuclear cells
RBF – Radial basis function
RR-MS – Relapsing-remitting multiple sclerosis
SOM – Self-organizing Maps
SVD – Singular Value Decomposition
SVM – Support Vector Machines
TF – Transcription factor
TNF – Tumor necrosis factor

1 Introduction

1.1 Context and Motivation

Microarray technology has come a long way since its development, and today it is possible to analyze expression of up to thousands of genes simultaneously. However, until recently, gene expression analysis was limited to a static framework, where the analyses were conducted on different samples, but only at a given instant in time, disregarding any temporal information. Only when the need to account for the temporal evolution of gene expression arose, time series gene expression analysis methods were developed.

One important field of research turned possible with time series analysis is the study of disease progression and treatment response. Drug therapy affects the organism at the cellular level, where the response causes specific gene transcriptional profiles, revealing differences in the regulation of a number of genes, when these profiles are compared to the ones where the drug therapy is absent. By studying the temporal evolution of the genetic material of the target cells, it can be possible to infer which genes are involved in the response mechanisms. This opens the door to the identification of different types of patient responses to the same therapy or treatment.

In the last decade, there have been several important works of time series gene expression analysis, including the study of treatment response on patients with relapsing-remitting multiple sclerosis (RR-MS). Interferon (IFN)-β is the most common form of therapy for these patients, and the investigators' interest in this clinical example is due mostly to the associated adverse side effects and to a large proportion of patients who do not respond to the treatment (defined as bad responders). Classifying RR-MS patients based on its responder type (good or bad response to the applied treatment), is a problem under intense investigation in the last years. Nevertheless, aside the unique characteristics of the disease and response to the treatment, a major challenge rises from the time series analysis: the patient specific rate has to be taken into account, since different patients, even of the same responder class, can present time-shifted expression profiles. The used data are multiple gene expression time series, resulting from microarray analysis. Essentially, a microarray consists of a set of features (in this case, DNA) to which a set of target molecules are combined (by hybridization in DNA microarrays), resulting in either quantitative or qualitative data, as gene expression and diagnostic, respectively.

1.2 Problem Formulation

The underlying goal of this thesis is the classification of clinically relevant gene expression time series for prediction of MS patient's response to IFN-β, in order to reduce useless and harmful treatments. Given multiple time series of gene expression data for a set of patients characterized by a particular clinical condition, such as a certain treatment for a disease, and a set of labels identifying

their class, we wish to build a classification strategy that is able to predict the class of a new patient with the same kind of data.

1.3 Contributions

The main contributions resulting from this work are the following:

1) a set of new biclustering-based classifiers for clinical time series designed to cope with specific properties of these type of data such as the temporal interdependencies.

2) a comprehensive study of biclustering-based and state of the art classifiers using a dataset consisting in the profiling of RR-MS patients undergoing a treatment with IFN-β.

3) interesting insights on the specificities of the classification problem under study, identification of bottlenecks in the proposed approaches, and innovative proposals to overcome them.

1.4 Thesis Outline

This dissertation begins with a brief description of the microarrays technology, followed by a characterization of multiple sclerosis, discussing the hypotheses for its origin. Several published conclusions on the MS pathogenetics are also mentioned in this section, aiming at a better understanding of the highly discussed mechanisms of the disease.

Then, we proceed with a discussion of state of the art techniques for time series expression data analysis, including some of the important challenges encountered, together with possible solutions. Then, a related work section is presented, where we compare the methods used recently in classification problems for clinical time series with MS patients, pointing out their strengths and weaknesses.

As we enter a new section, the methods proposed in this work are explained, including the developed algorithms to binary classification of clinical time series gene expression data, and the evaluation schemes used to assess the classifiers performance, when applied to the classification of a clinical dataset consisting in the expression profiling of MS patients under IFN-β treatment.

The experimental results are then presented and discussed in the respective sections. Different classifiers are put against each other, in order to analyze their potentialities. Finally, the dissertation ends with a section reserved for the main conclusions drawn from this work, and an outline of ongoing and future work to carry out in this area.

2 Background

In this section, we start by briefly describing the microarrays technology (focusing on DNA microarrays). Then, multiple sclerosis (MS) is defined and the main conclusions of several investigations are presented to support a better comprehension on the underlying mechanisms of the disease. Current therapies and strategies to treat MS are also discussed. Then, we present the state of the art in the research area of time series expression data, focusing on the classification problems and the associated challenges. The section is concluded with the presentation and discussion of related work on classification of clinical time series, with data from MS patients undergoing IFN-β treatment.

2.1 Microarrays

The evolution of microarray technology has clearly influenced genomic analysis, as well as molecular diagnostic, over the years. Actually, the reduction of the associated costs and increase of sensitivity and specificity, together with more available sequenced genomes lead to the incorporation of these techniques in more clinical applications. Their main strength relies on the capability of producing data relative to transcriptional response of possibly the whole organism's genome to a stimulus, either from genetic or environmental nature. Although there are other types of microarrays (as proteins microarrays), what we define as microarray is related to the DNA type.

Several techniques of microarray analysis are available, differing in characteristics, such as the probe and the method chosen for target linking and detection, together with the solid-surface support used [1]. The probe can be defined as the DNA sequence (its nature might vary) bound to the solid-surface support in the microarray, and the target is the "test" sequence whose expression we wish to observe. Generally, the target molecules are the transcription resulting from the genetic expression for a given sample.

The main goal is to observe (and quantify) the level of expression of target genes (for example). To achieve this, probes are synthesized (with different possible strategies discussed below) and immobilized as discrete features, called spots [1]. Note that, in each feature, one can find up to millions of identical probes. The target molecules (mRNA transcripted from the target genes) are labeled with a fluorescent label and hybridized to the probes. Whenever a successful hybridization occurs between the probe and the marked target, the fluorescence intensity increases (over a background baseline). Finally, the overall fluorescence intensity for all the hybridization events across the set of features is measured using a fluorescence scanner [1]. The probe length and synthesis method and the number of features (defined as the density of the microarray) are included in the experimental specifics.

The basic workflow for the processing of printed microarrays, the most common approach in this kind of analysis, is presented in Figure 1. Further information can be found in [1], including the

description of different microarray techniques, such as in situ-synthesized oligonucleotide microarrays, high-density bead arrays, electronic microarrays and suspension bead arrays.

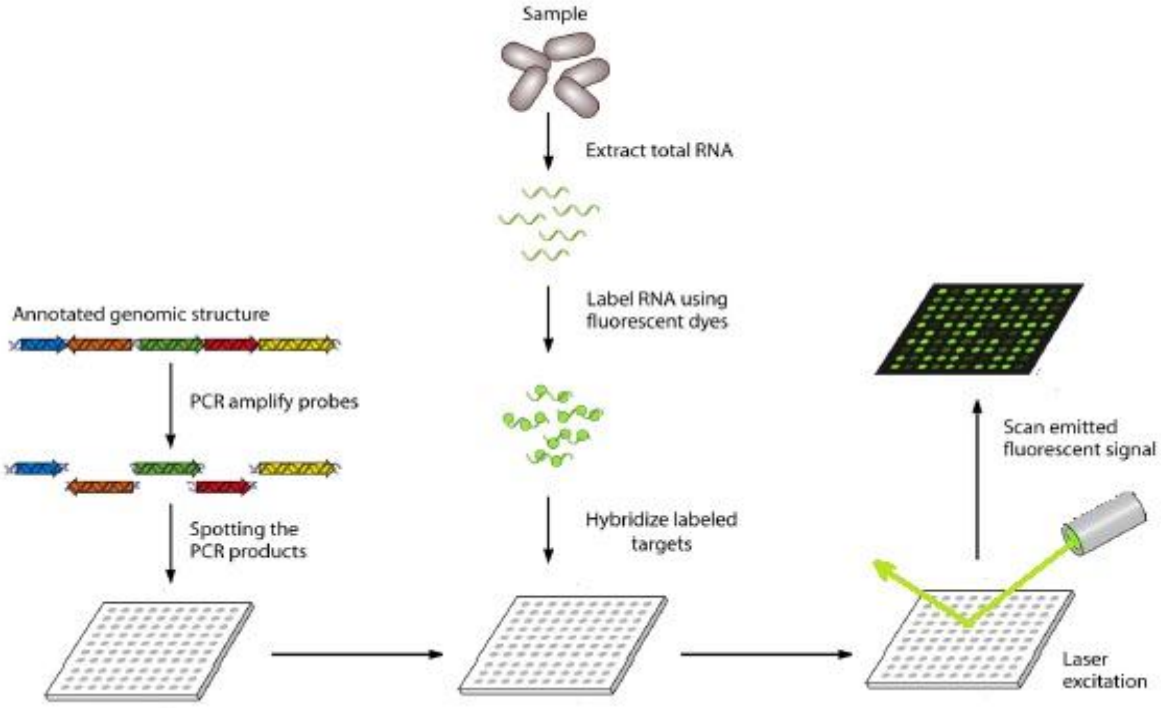


Figure 1 - Basic representation of the workflow of printed arrays (adapted from [1]).

In this situation, the solid-surface to which the probes are printed (hence the naming of the technique) is, most commonly, a microscope glass slide. The probe is printed by contact or non-contact printing. This is a critical step to which most attention must be given, due to possibility of cross-hybridization and printing inconsistency [2]. Regarding the nature of the probe, it can be either ds (double stranded)-DNA, or oligonucleotides. For the first case, the probes result of PCR amplified primers, built from known genomic sequences, shotgun library sequences or complementary (c)DNA. Note that these probes should be denatured to allow hybridization, before or after its immobilization (with electrostatic linking or UV-binding, for example [1]). In the oligonucleotides microarrays, the probes are chemically synthesized, resulting in a short sequence that can be then immobilized in the glass slide.

According to Figure 1, after the printing of the probes, RNA (transcription material) is extracted from the sample and labeled with a fluorescent dye. Then, all target mRNA molecules copied, with the help of reverse-transcriptase, into cDNA which are then hybridized to the bound probes. After a laser excitation, the fluorescence intensity is measured by a fluorescence scanner, resulting in a signal proportional to the target gene expression, since typically, (considering little or no cross-hybridization events) each feature corresponds to a target gene and a higher signal results from more hybridization between probe and target mRNA. Generally, a filtering step is applied, where only the intensities which are significantly above the background value are accepted [2].

Usually, the obtained data, after several preprocessing steps, are interpreted in a fold-change (FC) scheme, where the expression of a given gene is said to be up- or down-regulated by comparison with some kind of baseline (a possibility is a gene not considered to be expressed for the analyzed conditions: an “housekeeping” gene [2]). In the situation of a time series analysis, where the expression measurement is made for several time points, the notion of up- or down-regulation can be associated to the expression variation between time points (a gene is said to be up-regulated if its expression level increases from one previous time point to the next). However, if the data is not transformed, there are important differences for the cases when a gene is up- or down-regulated. For example, a fourfold up-regulated gene has an expression ratio of 4, whereas the corresponding down-regulated gene has an expression ratio of 0.25. A possible solution, commonly used in such analyses, is to apply the logarithm, also resulting in a simple normalization process: the previous case now returns 2 and (-2) as the values for, respectively, up- and down-regulated gene [2]. Other normalization strategies are available, and are widely discussed as a crucial step, aiming at reducing the systematic biases resulting from experimental inconsistencies.

2.2 Multiple Sclerosis

Multiple Sclerosis can be defined as a chronic inflammatory disease, characterized by a demyelinating disorder of the central nervous system (CNS), which affects mostly young female adults [3]. Because of the high degree of disability associated with this disease, and inherent significant impact at the socioeconomic level, several studies were conducted over the last decades on this disease, aiming at a better comprehension of the mechanisms involved in MS. A significant number of these studies show evidences indicating that, pathogenetically, MS is a T cell-mediated autoimmune disease [3], although other hypotheses have been formulated, as shall be seen later in this section. As the etiology remains to date far from total understanding, the interplay of both genetic and environmental factors, like the sunlight or vitamin D [4] is believed to be of crucial importance to the development of MS.

Additionally, more recent work contests previously established notions related to the disease. The idea of the brain as a special immunological zone, mostly based on the lack of an effective immune response because of the brain’s anti-inflammatory and pro-apoptotic environment, has been challenged by recent works with infectious and autoimmune models [3].

A major factor to be considered is the phenotypic heterogeneity in MS, where different pathologic patterns may indicate differences in the pathogenic mechanisms [5]. Despite this stated complexity in the disease phenotype, probably due to the several possible agents associated with the onset of MS, the knowledge of the nature of the acquired immune response in the Central Nervous System (CNS) of MS patients is certainly clearer, although the primary path giving rise to MS is still widely debated. There are three hypotheses for the disease pathogenesis, but it is generally accepted that the initial activation of the immune cells occurs in the lymphoid tissue [3]. These three hypotheses

differ in the nature of the antigens released into the periphery. The autoimmune hypothesis states that they are proteins from cross-reactive antigens, while the infection hypothesis defends a migration of some kind of brain-resident pathogen, yet to find. Finally, there is the degenerative hypothesis, according to which the responsible for the disease onset is the release of CNS proteins after primary degeneration. The pros and cons of each of the three hypotheses can be found in the Table 1 [3].

Table 1 - Multiple Sclerosis – accepted hypotheses for its pathogenesis

Hypothesis	Pros	Cons
Autoimmune	• Focused on the myelinated areas of the brain; • Response to immunosuppression and modulation.	• Immune responses to myelin antigens have not been associated with disease onset or progression.
Infectious	• Response to interferons; • Findings in infectious CNS disease models.	• No pathogen yet identified.
Degenerative	• Early neuronal loss; • Little inflammation seen in the progressive phase.	• Extent and chronicity of inflammation.

After the onset, the antigens (whatever may be their origin) reach the centers of the immune system, setting off an acquired immune response, where the dendritic cells assume special importance in the processing and presentation of T-cell antigens. They allow the priming of CD4+ and CD8+ T-cells, with the help of immune receptors like the Major Histocompatibility Complex (MHC) molecules (class I and II) [3]. Soluble proteins captured by B-cells are essential to the efficient initiation of the response of both T- and B-cells. Hence, these cells are ready to cross the blood-brain barrier and migrate to the antigen exposure site, gaining effector functions [6].

The antigens are displayed on neurons, glial cells and, particularly, in activated microglial cells, leading to a pro-inflammatory environment. When reactivated by the antigen, the primed cells initiate their effector functions. Antibodies are released from B-cells differentiation when they re-encounter the respective antigen, like the myelin basic protein and myelin oligodendroglial glycoprotein [6]. Moreover, both CD4+ and CD8+ T-cells are activated when presented with the antigen by the respective MHC molecules. They either release cytokines, or target the cell directly. Other inflammatory cells, like macrophages, are recruited to the site. These cells can phagocytize cell debris, and capture the antibody-antigen complex, presenting them on MHC class II molecules to CD4+ T-cells. After its activation, they secrete inflammation mediators, such as the Tumor Necrosis Factor Alpha (TNF-α) [3].

Finally, after both acquired and innate immune responses clear the target antigen from the lesion, the respective cells either enter in an apoptotic pathway or are redistributed to other tissues. At the same time, remyelination is also possible as a repair mechanism [3].

Despite all the investigation in this field, the search for candidate genes that could, alone, account for the disease development, is unfruitful until today. The epidemiological studies in different MS populations [7-9] showed similar results to those obtained for different autoimmune diseases. The main conclusion is that MS is genetically complex, and it is not possible to select single genes to explain the disease susceptibility. Instead, it might be a result of the interaction of several altered genes [10].

Chapman et al. [11] show that the best relation with MS susceptibility comes from the alleles of the MHC, or Human Leukocyte Antigen (HLA) in humans, on chromosome 6p21, location supported by the findings of Bomprezzi et al. [12], standing out mainly for harboring the HLA complex, histone cluster and the heat shock protein-70. More recently it was shown that almost all the HLA-DRB1 haplotypes are somehow related to the disease, retiring the idea of the antigen presentation being the sole mechanism of the MHC associated disease susceptibility [13].

The use of microarrays to analyze the gene expression profile of peripheral blood mononuclear cells (PBMC) can be found in previous studies of other autoimmune disorders, as well of MS [14, 15]. According to these studies, the differences between MS patients and control subjects were found in immune related, and cell cycle related genes, leading to the conclusion that other cellular events are of great importance, beyond the lymphocyte activation and cell-cell interaction [12].

One must notice that there is a distinction between the so-called susceptibility genes and the ones whose expression gets altered as a consequence of the disease, as a downstream effect of other genes. Bomprezzi et al. [12] also reported a differential expression of MS samples vs controls. If, for some of the genes found, the respective function in autoimmunity is easily understood, for other ones, it becomes less intuitive. Other relations can be found between the altered genes in different studies [12, 14].

In summary, there are many and significant evidences showing the expression profiling of MS consists in an overall weak signal, giving rise to the idea that there is a fine balance of a large number of factors, and even a slight deviation can take the system to a critical point, leading to the development of the disease.

Aiming at the identification of the molecular behavior behind the MS heterogeneity explained above, other authors studied the gene expression profile of peripheral blood CD3+ T-cells [16], comparing MS patients and healthy controls. They identified four subgroups of MS patients, and five gene clusters that are differentially expressed between them. Moreover, by analyzing the response to IFN-β treatment, they found that the responders (vs non-responders) were included in two of the aforementioned subgroups. Ramanathan et al. [14] have also obtained significant differences between relapsing remitting (RR)-MS patients and control. A subset of 25 genes was found to be up-regulated in the disease, and one of 9, down-regulated. To that date, only 12 genes of the obtained set had been correlated with inflammatory and/or immunological functions, possibly in the base of MS.

As a result of the International Multiple Sclerosis Genetics Consortium (IMSGC) whole genome analysis [17], 38 genes were found to be active on the MS susceptibility. Some of them were identified in [18], amongst other deregulated genes revealing some degree of specificity to RR-MS pathogenesis. The CD58 gene, as IL7Rα and IL2Rα, the two most significant genes in the IMSGC study, are believed to affect regulatory T-cell, or Treg, promoting differentiation and proliferation [17], as a result of its over-expression.

The DBC1 gene is thought to represent an anti-proliferative function, hence justifying its high level of expression observed in a remission state of the disease (immunosuppressant) [19]. The two genes ALOX5 and TGFβ1 were found to be differentially expressed between MS patients and healthy controls in the last work and the whole genome analysis performed by the IMSGC. While the first gene is up-regulated during relapse and remission, suggesting a possible allergic component for MS, the higher expression of TGFβ1 remains a hot topic of discussion in what concerns to its role in MS pathogenesis. In [18] it is proposed that this up-regulation may be a result of a proinflammatory state, promoting the Treg cell proliferation (as seen for CD58).

In this context, we are far from completely understanding the pathways leading to MS. There are many interacting components, differing across different study conditions and datasets, and thus, there is still a great need for more specific experiments in order to unravel the pathogenetics behind MS.

2.2.1 Treating Multiple Sclerosis

Taking aside some new therapies under development, the approved therapies for relapsing-remitting Multiple Sclerosis can be categorized in three different groups. Interferon (IFN)-β, with three different preparations: IFN-β-1a (Avonex®), IFN-β-1b (Betaseron®) and mitoxantrone (Novantrone®) [20].

TNF–α was shown to be involved in the pathogenetic mechanisms of MS [21]. Thus, its inhibition would benefit the patients. Contradictorily, when this idea was put to the test, the results did not follow the expectation created by the experiences in the treatment of rheumatoid arthritis, and blocking TNF-α was shown to worsen the disease [22].

Consequent to the heterogeneity of the disease, the treatment response, even for one stage of MS only (RR-MS), presents a high variability, suggesting different responses at the molecular level, leading to diverse clinical outcomes as the inhibition of CNS inflammation [23].

Since this thesis is focused on a case study of MS patients under an IFN-β treatment, this therapy is further discussed. Interferons, in general, are small proteins, with paracrine action aiming at the growth arrest and apoptosis of the infected cells, avoiding its propagation [24]. The Type I IFNs,

which include IFN-β, are a constituent of the innate immune system, and exert antiviral activities. Besides, they can provoke complex immunomodulatory effects [25, 26].

Although not entirely understood, the main idea for the mechanism of action of IFN as immunomodulator rests on modulation of adhesion molecules expression, inhibition of matrix metalloproteinases, regulation of apoptosis and induction of anti-inflammatory cytokines such as interleukin-10 (IL-10) [27]. Despite this anti-inflammatory modulation, IFN-β was shown to also up-regulate a significant number of pro-inflammatory mediators [26], thus suggesting that a possible clinical benefit comes from the balance of medium or even small alterations in numerous biological pathways [23].

The treatment of RR-MS patients has routinely been carried with the use of recombinant human interferon beta (rIFN-β) [25]. Nevertheless, up to half the patients show no benefits from this treatment, and negative side effects have to be considered [24].

The first studies carried with the goal of better understanding the mechanisms of the disease and influence of IFNs in its treatment were limited to hypothesis-based experiments, restricting the processes analyzed, especially if the heterogeneity of the disease and the complex interaction of treatment and disease are taken into account. More recently, microarrays enabled the study of expression profiles, leading to a shared notion that the pharmacological and physiological effects of IFNs involve a more complex pattern of gene regulation [28]. These experiments were first restricted to *in vitro* studies, using different cell types from those that are actually the target cells of IFN in multiple sclerosis (a human brosarcoma® cell line and human umbilical vein endothelial cells (HUVEC), respectively). Then, a clinical study was performed, with gene expression profiles of RR – MS patients in response to a treatment with IFN-β [23].

Sturzebecher et al. [23] have also analyzed the gene expression profile with *in vitro* pretreatment with IFN-β, but with the purpose of including it as a baseline responsiveness, and comparing it to the biological response *in vivo,* allowing also to investigate which genes are regulated *in vitro*, but do not meet the minimum expression threshold *in vivo*. The results suggest significant differential expression profiles across responders and non-responders for some genes of the *ex vivo*[1] data, showing regulation also in the *in vitro* environment. Moreover, even defending that the definition of a responder class should preferentially rest on a group of genes, interesting results for IL-8 are obtained, where its expression profiles suggest a significant discrimination of the response states. These results are yet to be supported by studies with a larger cohort of patients. IL-8's expression for the responder group was down-regulated, while it was up-regulated or constant for the non-responder group (including the early responders that later lost their response).

There are important differences in the responder classification across different studies. If we compare the responder labeling method used by Sturzebecher et al. [23] with the one used by Baranzini et al. [15], numerous divergent points are found. For example, for the first authors, the

[1] When the *in vivo* effects are assessed by means of arrays, the term *ex vivo* is used.

assessment of responsiveness to IFN-β was not based only on the clinical disease activity, like the Expanded Disability Status Scale (EDSS), but also with the study of MRI lesions. In the first case there are not only two groups as in [15]: good and bad responders. Instead, the bad or poor responders are divided into two subgroups. Patients who do not fully respond to the treatment from the beginning (initial non-responders - INRs), and those who start by responding well to the IFN-β therapy, but lose their response after developing high titres of neutralizing antibodies (NabNR). Patients were categorized as good responders when the disease was considered to be clinically stable.

Other studies have focused primarily on the classification criteria for categorizing RR-MS patients based on their response to IFN-β therapy [29]. These authors carried out a longitudinal study with such patients, classified using criteria such as the number of relapses during the 2-year trial (similarly to Baranzini et al. [15]), the number of new T2 lesions (MRI analysis) after the 2 years. The considered outcomes include the 2-year change in the EDSS (commonly used in this classification), Multiple Sclerosis Functional Composite and brain parenchymal fraction. This study showed a correlation of new MRI lesions during IFN-β treatment with a poor response, while baseline characteristics failed to discriminate successfully the outcomes (IFN-β vs placebo treatment). Another study analyzed the responder classification problem facing the disability progression (investigating changes in the EDSS) against the relapse rate during the course of treatment [30]. In conclusion, a criterion based on the disability progression proved to be more relevant from a clinical point of view, showing higher sensitivity, specificity and accuracy.

In this context, the main goal of a time-course profiling of the treatment response of MS patients rests, as already stated, in the possibility of accurately predicting a given patient's response, avoiding useless and possibly harmful treatments.

2.3 Temporal Gene Expression Analysis

Gene-expression experiments were, until recently, limited to a static (non-temporal) analysis, in which only a snapshot of gene expression for a set of samples was available. Currently, this static analysis is still used in various applications, as in the comparison of samples from the same tissue for a given instant, under a specific condition. However, the last years have witnessed the arrival and evolution of time-course gene expression experiments, in which one can measure a temporal process, with evidences of strong autocorrelation between sequential points in time. In this context, at any given instant, the amount of transcript mRNA for a gene, is the balance between its production and degradation [31], and through the analysis of time series expression data it is possible to determine the stable state that follows a new condition or perturbation, as well as the pathway and networks whose activation lead to this state [32].

Analyzing gene expression of such a high number of genes at the same time, allows for insights into the multidimensional dynamics of complex biological systems. It is also possible to

observe crucial temporal responses, emerging coherently from several interacting system components.

With the possibility to perform a temporal analysis of gene expression, some questions of great importance arise, such as the problem of identifying coherent responses, which can distinctly characterize the various classes of objects, conditions or patients. In other words, this problem lies in the identification of combinations of up- or down-regulated genes, or more interestingly, genes with coherent expression profiles, then used for disease classification.

Other important issue is the characterization of the actual state of evolution of a biological system. It includes the identification of differentially activated pathways, as well as interaction networks [31].

The questions with potential to be addressed by means of gene expression analysis can be divided into four categories [32]:

1. Biological systems analysis, where the temporal information is put together in order to understand the underlying dynamics. Possible examples are cell cycle [33] and circadian clock studies [34].
2. Genetic interactions and response dynamics, understood by subjecting the biological system to rigorously controlled perturbations. It comprises also knockout experiments (eliminating a given gene's expression) to identify the functions of individual genes (downstream effects). Known examples include cell cycle double knockouts [35] and knockouts under stress conditions [36].
3. Development, involving the complex sequences of cell proliferation and differentiation (stem cells [37]).
4. Disease progression studies. The analysis of a global expression allows the follow-up of the evolution of pathological characteristics. There are several different studies of time series expression analysis of human cells infected with different pathogens [38], cancer [33] and multiple sclerosis [15, 24, 39].

The fundamental reason for transcription profiling lies on the assumption that gene expression, with the expressed mRNAs, can code for specific proteins, hence producing a determined phenotypic response [31]. The monitorization of mRNA transcripts is a critical source of information. However, it exhibits some important limitations, such as the role of post translation modifications, mRNA stability, or other destabilizing factors. These complications can be seen as quite a drawback on the role of mRNA as a proxy for quantifying the active products after translation [32].

At this point, we can introduce what was named as the Guilt by Association Principle [40], which states that genes that exhibit similar responses to a given signal must be controlled by similar mechanisms of regulation. The critical step in this kind of analysis is the identification of the measured transcripts that are correlated in some level, a problem which can be included in a more general class of computational ones: characterization of multidimensional trajectories [41].

From the studies of time-course gene expression data, it is possible to reverse-engineer primarily regulatory networks. Acting at specific control points in regulatory paths holds great promise for discovering new drugs, by controlling the regulation of the respective transcription factors (TFs), by the identification of new drug targets [42] and understanding disease progression [43].

2.3.1 Methods and Computational Challenges

The computational challenges in the analysis of time series expression data can be divided into four levels [32], and are explained in the next subsections.

2.3.1.1 Experimental Design

This level consists mainly in the previous establishment of the required number of microarrays and representative probes for a determined gene sequence, aiming to minimize the possibility of cross-hybridization [1]. Regarding this issue, there are some challenges that must be taken into account, as the sample size limitations.

The datasets are normally very small in terms of time points and/or patients, although datasets with more time points are arising. These smaller datasets contain far less information than what is desired for these experiments to succeed completely, and the distinction between noise and effective signal is also a very important question to be addressed. Several different limitations, of technological or practical order, limit the sample size. The number of time points that can be measured is very restricted, and there is a great difficulty in collecting the data, especially if we are dealing with clinical trials, where blood has to be harvested for every time point.

Equally or even more significantly, the number of biological and technical replicates is also an important problem [31], and again, an important example lies on the clinical datasets, where the participating patients have to fulfill many requirements. Computationally, this issue is called as learning in almost empty spaces [44], since the classifier acts upon a small dataset in high-dimensional spaces.

Regarding the noisy data issue, unless there is some knowledge about the implicit concept generating the data, the detection and distinction of noise is a really difficult task to complete. Attempts have been made to introduce some kind of prior knowledge about the system into the expression analysis [45].

Ernst et al. [46] proposed an algorithm capable of improving the classification of short time series expression. The major advantage of this method lies on a measure of profile significance, allowing a reduction of the profile space to the significant ones. So far, the classification algorithms did not have the capability of distinguishing real response profiles from the ones appearing randomly, fruit of the overfitted data (a great number of genes for a small number of time points).

The sampling rate determination is also an important issue of experimental design. If undersampling takes place, some key events can be missed, or temporal aggregation may occur. With these coarse sampling rates, genes that reveal interdependence can actually be independent amongst themselves. On the other hand, if the sampling rate leads to an oversampling, the experiments get more expensive and time consuming. To compensate these issues, one could proceed with a shorter experiment duration, but with the real possibility of missing significant genes that act in a later stage of the process. There are biological consequences associated, since the sampling rate should depend on the transcription/degradation of mRNA [32].

The other major challenge of the experimental design is concerned with the synchronization of the cells. Generally, they are arrested so they all begin the cycle in the same phase. Nevertheless, it is possible for them to lose synch after some time [33]. Thus, the main objective is to determine if and when the cells lose their synchronization, helping in the determination of the time points that reflect most accurately the behavior of the biological system under study.

Up to date, the sampling rates depend almost exclusively on the biologist's intuition and experience. One possible solution could be the use of an online algorithm, starting from an initial sampling rate, and changing it over time, until a required confidence level is attained [32].

In what concerns to the cell synchronization, there is an example of an algorithm with Fourier analysis, which relies in the comparison between two sets of genes: the ones best explained by a periodic curve and the others best explained by a a-periodic curve [47]. Lin et al. [39] assumed a sinusoidal pattern, followed by the expression profiles, while other authors proposed the deconvolution of the data with a predetermined model from external information [48].

2.3.1.2 Data Analysis

In this level, the focus is on the gene. This can be easily stated in tasks like the study of the continuous evolution for each gene, identification of differential expression, and coherent expression patterns, from different time-course expression experiments [32].

Some important challenges and questions also rise up in this level, such as the search to overcome the sampling rate problems mentioned before and the possible missing values, which can arise more often with the extraction of the continuous representation of all genes for the whole duration of the experiment.

The solution to deal with noisy data and small number of replicates must go past the simple methods of interpolation of individual genes, which lead to inadequate estimates [32]. Another major challenge in the data analysis is the variability of the timing of the biological processes, since it differs between organisms, genetic variants and environmental circumstances.

The identification of differentially expressed genes across samples and experiments also reveals a great importance, namely after an experimental perturbation, or for comparison between normal and diseased cells.

In order to obtain a continuous representation of the time series expression data, some algorithms have been proposed in the literature. The first one is based on B-splines, which is a specific type of spline where each point is represented as a linear combination of a determined set of basis polynomials [32]. There can be some constraints in the spline coefficients, aiming to avoid the over-fitting of the data, such as the ones of co-expressed genes having the same covariance matrix. This allows the identification of differentially expressed genes, using the differences between the aligned continuous curves [32]. The parameters for this model can be determined with resource to an Expectation-Maximization (EM) algorithm, for datasets with more than 10 points. Other approaches include simple interpolation methods already mentioned or Dynamic Time Warping, a discrete method that makes use of dynamic programming [49].

Finally, to identify the differentially expressed genes between experiments, one can find several algorithms throughout the dedicated literature, as the cluster analysis [35, 36], the general singular value decomposition (SVD) [50], custom-tailored models [51] and Fourier analysis to identify significant peaks in a periodgram [52].

2.3.1.3 Pattern Recognition

Basically, this level deals with the organization and visualization of the data. A considerable number of methods is available, based on the concept of similarity or distance between samples [32, 53, 54].

1) Point-wise distance-based clustering methods (PwDbCM)

Normally, the data points are organized in a matrix, with dimensions Ng (number of genes) by Nt (number of time points). With these methods, the clustering is achieved by determining the distance between two samples, and creating clusters with samples that fall within a certain threshold. That distance, or similarity, can be measured with norm-based distances and combination of correlation metrics [31].

These methods can be divided in two types of clustering: partitioning and hierarchical. The first type comprises algorithms such as the k-means and self-organizing maps (SOM) [31], whereas the second one, as suggested by its name, consists in creating a dendrogram representing the hierarchy of the relative distances of the data points, along an one-dimensional axis.

More recent techniques for partitioning algorithms have risen, with combinations of k-means and kernel methods, with the goal of rendering the data linearly separable, by means of transforming the original data space. However, these distance-based methods neglect an important aspect of the

time-course experiments: the temporal dependencies. One can change the order of the time points arbitrarily, and the clustering results remain unaltered [31].

2) Model-based clustering methods (MbCM)

In this type of clustering methods, instead of focusing the similarity on the data, *per se*, the similarity measure is based on an unknown model built to describe the data.

The goal here is the identification of a mixture model, given by the appropriate combination of base functions, capable (most as possible) of explaining the data. Interesting variations of these methods can be found in the literature, such as the possibility of an autoregressive model, so the time delays can be accounted for [31].

To our knowledge, the currently most promising clustering method for time series expression datasets is based on Hidden Markov Models (HMMs). These allow to overcome the lack of consideration for the temporal nature of the data, leading to a more effective clustering [39], together with an interesting capability of coping with important issues, such as both cyclic and non-cyclic behavior of the temporal dependencies in the expression profiles [54]. These models can describe a sequence of events, with emission of symbols in each state of the sequence, thus corresponding to the transformed time series gene expression profile. A good introduction to HMMs and its properties can be found in [55].

Essentially, the HMM-based clustering algorithm consists in partitioning the data into K HMMs, then maximizing the likelihood of the data given the learned HMM. The iterative algorithm consists of two steps, as follows:

- First, each gene of the dataset is assigned to the most likely HMM (of the previously determined K).
- Then, the parameters of each HMM are computed, considering the new set of genes that were assigned to it [32].

It has to be noted that when using these methods, there is an implicit assumption that the actual state of the system takes into account its previous states [54].

Before this clustering method, other algorithms were already capable of some good results, in spite of lacking some already discussed considerations, as the temporal relationships within the datasets.

It was stated that unsupervised learning is not the most suitable method for analyzing gene expression data [54]. The inclusion of some relevant information about the genes leads to a more effective grouping, becoming a partially or semi-supervised learning task. The available information can be the gene function or mechanism of regulation, allowing, at least, the constraint of membership of pairs of genes in the same group. The extension of EM algorithms in learning the mixtures with

semi-supervised learning has also been used in [56], since the EM algorithm alone guarantees only the convergence to a local maximum of the likelihood function. This combination of prior knowledge of the genes leads to a higher robustness of the estimation with respect to noise, at the same time leading to a better quality estimate [54].

3) Feature-based clustering methods (FbCM)

The goal of this sort of clustering methods is the identification of prominent features from the expression profiles, analyzing local or global consistencies in transformed data [57], rather than using the aforementioned quantifiable metrics. This results in a higher flexibility, minimizing the influence of noise and uncertainties associated with the mRNA expression quantification.

Graph-based methods can be included in this category, as the nature, structure, properties and characteristics of a hierarchic graph are seen as features which can be further analyzed [31].

These clustering methods reveal important potentialities, such as the analysis of multiple conditions at the same time, since it must contribute with more significant information, when compared to the analysis resulting from a single perturbation [58].

4) Biclustering

Biclustering's main difference from the aforementioned clustering methods (k-means, SOM, hierarchical clustering, among other) is that, although it also searches for groups of similarly behaving genes, it does not require that the similarity holds for all the conditions or time points. Therefore, a bicluster can be defined as a subset of genes that display an analogous expression pattern for a subset of conditions or time points (in the time series analysis) [59]. The key advantage presented by this method is the opportunity to unravel processes which are not active during all the time points or across all conditions, but only for a smaller set.

Biclustering presents itself as a very important resource in the analysis of gene expression, by taking the concept of similarity away from the limitation of pairs of genes or conditions towards a measure of general coherence of genes/conditions.

This more recent technique has revealed some interesting advantages when compared to simple clustering, in the task of identifying groups of genes with coherent expression patterns, thus rising as a promising tool for identifying potential regulatory mechanisms [60]. Several studies made use of different biclustering techniques in the search for local expression patterns, as can be found in the survey presented by [58]. The general biclustering problem is NP-hard [61], and thus most of the approaches presented in the literature are heuristic, and therefore not guaranteed to find an optimal solution. Alternatives can be found, such as an exhaustive search (limiting the size of the biclusters to obtain acceptable runtimes) [62].

The goal of identifying coherent patterns does not depend on the exact numeric values of the matrix, leading to numerous authors using a discretized version of the matrix, but even this formulation is, generally, NP-hard [63]. The discretization methods based on the transition between time points, therefore considering the temporal dependency, show better results than the ones that do not take that aspect into account [64]. To circumvent the complexity issue, there is a restriction that renders the problem tractable [60]. The analysis can be confined to contiguous columns, corresponding to consecutive time points, where the biclusters found in this framework represent the group of coherently expressed genes over consecutive time points, also taking into account the temporal dependencies mentioned in previous sections. Its importance comes from the observation that biological processes start and finish in a contiguous but unknown way [63].

Regarding this matter, Madeira et al. [60] proposed an algorithm capable of finding maximal contiguous column coherent biclusters (CCC-Biclusters) in a time linear in the size of the expression matrix. Essentially, it processes a discretized version of the original expression matrix with effective string manipulation techniques based on suffix trees. This algorithm will be explained in Section 3.3.

After the identification of all maximal CCC-Biclusters, there is a step of statistical validation, where all of them are sorted according to the probability of being a product of random events. Finally, the algorithm searches and discards highly overlapping CCC-Biclusters (after the application of the statistical test), eliminating redundant groups at the desired minimum level.

Comparing this proposed exhaustive method with heuristic algorithms, such as [65], the results were significantly better for the former, given that the latter processed the original, instead of the discretized, expression matrix. This fact caused the algorithm to converge rapidly to a local minimum, from which would not escape, and the solution often included all the columns, meaning all time points were eventually taken into the clustering, resulting not in a bicluster, and therefore, showed no utility in the search for local patterns of coherent expression.

Madeira and Oliveira [66] introduced in their algorithm the important possibility of some degree of error when looking for local patterns, which was shown to improve the results.

As a conclusion on the pattern recognition level, it is important to state that the algorithms should be able to return the dynamics of the system, as well the different classified (clustered) objects, in order to achieve some insight on the regulation mechanisms between genes, already shown to be related.

2.3.1.4 Regulatory Networks

One can state that this is a higher level of the general time-course expression experiments, because the attention is on the genes interaction and different systems in the whole cell, trying to model them, either descriptively or predictably. The first challenge arising at this level, both for static and time series experiments, comes from the necessary combination of data from different biological

sources, which includes protein-DNA and protein-protein interactions, as well as prior knowledge on the expression data [32].

For the analysis of regulatory mechanisms and networks, it is often necessary to get data from gene knockout experiments, in different experimental perturbations. Still, generally, one simple gene knockout is not sufficient, leading to much more expensive experiments.

Ideally, the inferred model would be appropriate for different studies. Practically, this is not feasible, and the solution rests on gene modules: sets of genes assumed to share a similar function or pathways. Identifying these modules, assigning them to temporal networks, is a very important help in the initiation of the modeling tasks, heading to a significant gain in statistical confidence [32]. Regarding the incorporation of information from different biological sources, this issue has risen as a hot topic in recent research.

In conclusion, most computational problems raised by the time series gene expression data are mainly related to the experimental design step and networks description and inference analysis.

2.3.1.5 Quality of Clustering

With the genotype as the input, the phenotype is what is actually observed [31]. So, the evaluation of overall quality of a clustering algorithm must not be based on the similarity on the input space, but rather on the biological information gained by analyzing the clustered samples.

Whilst the analysis of genome-wide mRNA expression is becoming more routinely available, the gain of biological insight from the computational results is still the subject of a great deal of research [67]. When comparing clustering methods, the result is usually biased and dependent on the method used, as well on the type and nature of the data [31]. This also supports the advantage of evaluating the biological insight gained from the results of the computational analysis. Presently, it is a current practice to evaluate the clustering method based on the accordance of its results with biological reality, including functional ontologies and transcription factors (TFs) of co-expressed genes [68].

TFs are proteins, and consequently, gene products, that bind to precise sites in DNA, promoting or inhibiting a certain gene's expression [32]. This expression control can also be used as a test for the effectiveness of the clusters formed. Because transcription factor binding sites (TFBS) motifs are so short and degenerate (5 to 9 base pairs), the matches found are mostly due to chance, not being functional at all. A possible solution is to exclude the TFBS which are not in evolutionarily conserved regions [31].

With time series experiments it is possible to cluster genes into subsets of certain distinctive characteristics. An example is the common regulatory mechanisms, which leads to a situation where genes that belong to the same cluster, display similar response profiles [31].

2.4 Related Work - Classification of clinical time series

To date, there is a significant number of publications reporting studies on classification of clinical time series. The ones analyzing an important case study in multiple sclerosis patients are discussed briefly in this section. All these works share a common goal with this thesis, of building a classifier that is able to correctly predict the MS patient's response to IFN-β. Further explanations can be found in the Appendix A.

2.4.1 Comparative analysis

Table 2 shows the main similarities and differences between the three previously discussed time series classification of RR-MS patients undergoing an IFN-β treatment.

The work of Baranzini et al. [15] revealed a very important limitation, as the analysis was based on the first time point only, disregarding all the remainder temporal information. The best results for prediction accuracy were obtained when analyzing the data tridimensionally, that is, in sets of three genes, called triplets. Although the authors show an 86% prediction accuracy for the best discriminative gene triplet (see Table 2), these results were shown to be optimistically biased [39].

The best prediction results were shown in [24, 39], and these studies tried to address some of the major challenges faced when dealing with classification of time series gene expression data. The use of HMMs with fewer states than time points allows the alignment of the expression profiles, thus accounting for the patients-specific response rates [39]. On the other hand, the constrained mixture estimation introduced by Costa et al. [24], beyond supporting the role of a number of significant genes in the treatment response discrimination, allowed the identification of responder subgroups, namely in the class of good responders to IFN-β. This is a really important conclusion, because it might lead to further research in order to proceed with different therapy models.

Table 2 - Summary of main characteristics of the three clinical time series classification studies analyzed.

	Method	Features	Main results
Baranzini et al. [15]	Quadratic analysis-based integrated Bayesian inference system (IBIS)	• Triplets of Genes (3D) • Only the first time point	• Best discriminative gene triplet – Caspase 2, Caspase 10, FLIP (86% Prediction Accuracy, shown to be an optimistically biased value)
Lin et al. [39]	Hidden Markov Models (HMMs) with discriminative learning	• Recursive Feature Elimination (RFE): eliminating less discriminative genes • From 2 to 7 time points	• Best prediction accuracy of ~88%. • Identification of discriminative genes such as Caspase 2, Caspase 3 Caspase 10, Jak2, IL-4Ra, MAP3K1, RAIDD. • Identification of the patient-specific treatment response rates
Costa et al. [24]	Constrained mixture estimation of HMMs	• Feature selection based on prior distribution, with parameters computed with EM algorithm: 17 genes • All 7 time points	• Prediction accuracy > 90%. • The 17 selected genes include the ones presented for Lin et al. [39] • Identification of responder subgroups • Identification of potentially mislabeled patients

In a final note, Costa et al. [24] state that special attention must be given to a few important inherent issues, when dealing with clinical data from patient expression profiles: in MS, as in diseases with multiple molecular causes, a specific response type can reveal more than one expression signature [24]. As such, some of the profiled patients might be mislabeled, and, in fact, these authors discovered that one of the patients in the dataset was indeed misclassified, confirming this information with Baranzini et al. [15].

3 Methods

In this section we present the new biclustering-based classification strategies developed in this thesis: kNN with several different similarity measures, a meta-profiles and meta-biclusters classifiers. All the classifiers were coded in Java and, although in this thesis they are applied only on the data of the expression profiling of MS patients under treatment with IFN-β, we note that they are applicable on time series gene expression data in general, and even on other classification problems. The basic workflow of the proposed biclustering-based classification algorithms is displayed in Figure 2.

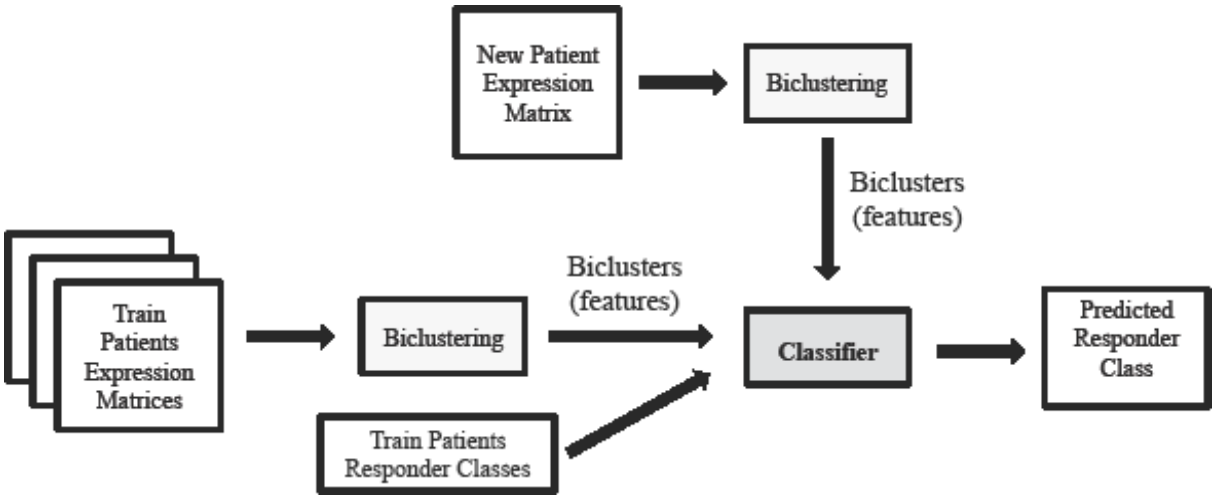


Figure 2 - Basic workflow of a biclustering-based classification method.

Before describing the classifiers, we start with the dataset description and its preprocessing procedures (including missing values handling, normalization and discretization steps). Then the CCC-Biclustering algorithm [60] is explained in more detail. Finally, we present a collection of state of the art classifiers, from the software package Weka, applied also on the MS dataset, to better evaluate our classifiers' performance.

3.1 Dataset Description

The dataset used in this work is the same collected by Baranzini et al. [15]. It consists in the profiling of MS patients subjected to the standard treatment with IFN-β. Fifty two patients with relapsing-remitting (RR) MS were followed for a minimum of two years after the treatment initiation. After that time, patients were classified according to their response to the treatment, as good or bad responders. Initially, thirty three patients were initially classified as good responders, and nineteen as bad responders. However, as mentioned in the subsection 2.4.1, it was discovered that a patient of this dataset was mislabeled. Since the minimum criterion of the Expanded Disability Status Scale (EDSS) was met (precisely the one point increase required to the classification in the group of bad responders), this patient, previously mislabeled as a good responder, was analyzed in the group of

bad responders. In result, we have thirty two good responders and twenty bad responders (the good responders' proportion is now approximately 62% while before it was slightly over 63%).

Patients were classified as good responders if there were no more relapses, and no increase in the EDSS was verified after two years. Bad responders, on the other hand, were the patients that suffered at least two relapses or a confirmed increase of one point in the score of the EDSS [15].

Seventy genes were pre-selected based on biological criteria, and their expression profiles (raw genes expression levels) were measured, using one-step kinetic reverse-transcription PCR [15] for seven time points (although there were some missing measurements for some of the patients, the seven time points correspond to the treatment initiation, and three, six, nine, twelve, eighteen and twenty-four months after the initial point).

The data can be organized in two different ways:

1. In a single matrix, with as many rows as the number of patients (N_p), and the number of columns equal to the product between the number of genes (N_g) and time points (N_t). Each line then consists of N_t blocks of N_g elements (the first N_g elements of the line represent the expression value of each gene in the first time point, and so on). Below, an exemplificative matrix is displayed.

$$\begin{array}{c} \\ patient\ 1 \\ \vdots \\ \vdots \\ patient\ N_p \end{array} \begin{array}{c} \begin{array}{cccc} G_1, TP_1 & G_2, TP_1 & \cdots & G_{N_g-1}, TP_{N_t} \quad G_{N_g}, TP_{N_t} \end{array} \\ \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,(N_g-1)\times N_t} \quad a_{1,N_g\times N_t} \\ \vdots & & \vdots\ a_{p,i}\ \vdots & \vdots \\ a_{N_p,1} & a_{N_p.2} & \cdots & a_{N_p,(N_g-1)\times N_t} \quad a_{N_p,N_g\times N_t} \end{bmatrix} \end{array}$$

where $a_{p,i}$ represents the expression level of gene g for time point t, such as $i = g \times t$, for patient p.

2. A matrix per patient, with N_g rows by N_t columns. This is the data organization used for the biclustering-based classifiers developed in this thesis. An example of such a matrix is here presented below.

$$\begin{array}{c} \\ Gene\ 1 \\ \vdots \\ \vdots \\ Gene\ N_g \end{array} \begin{array}{c} \begin{array}{ccc} TP_1 & \cdots & TP_{N_t} \end{array} \\ \begin{bmatrix} a_{1,1} & \cdots & a_{1,N_t} \\ \vdots & a_{g,t} & \vdots \\ a_{N_p,1} & \cdots & a_{N_g,N_t} \end{bmatrix} \end{array}$$

where $a_{g,t}$ represents the expression value of gene g for the time point t.

3.2 Dataset Preprocessing

3.2.1 Missing Values

Many patients missed one or more of the gene expression measurements, resulting in a missing value for that(those) time point(s). The standard classifiers cannot deal with these missing values, and so they must be filled beforehand. This can be performed using several techniques such as filling the missing value with the average value between the previous and next gene expression values (closest neighbors), or repeating the previous value if the next one does not exist. It is also possible to compute an average expression value, but now within a window of greater length (considering more neighboring time points).

The discretization step that will be explained later is capable of leading with these missing values, by jumping them and continuing the operation. However, the biclusters consequently obtained are surely different and so it is interesting to see how this aspect of the data influences the results.

3.2.2 Normalization

A step of data normalization is essential in this kind of analysis, to reduce the discrepancies between the different conditions at which the measurements are carried. This can be done in several different ways, depending on where the focus is, though usually it is accomplished to zero mean and unit standard deviation:

1) by patient – data considered as whole, disregarding the differences between the measurements of different genes and time points.

2) by gene (for each patient) – often used in gene expression analysis to turn the different genes' expression levels comparable.

3) by time point (for each patient) – turns the temporal information comparable, considering the inconsistencies introduced by gene expression measurements when collected in different instants in time.

Combinations of the two last normalization strategies can also be applied, and in this work several tests are made to take some conclusions about the importance of this pre-processing step.

3.2.3 Discretization

The classification methods developed in this work are all based on the results of a biclustering technique (CCC-Biclustering), proposed by Madeira et al. [60], already mentioned in Section 2.3.1.3.4. In CCC-Biclustering, the search for biclusters is not carried over the continuous space of the data. Instead, it is performed over a discretized version of the expression matrix. The discretization can be

achieved using several different techniques (see [60] for details), although two of them revealed to be most interesting for this particular application:

- Variation between time points - using an alphabet with three symbols: U for up-regulation, N for non-regulation and D for down-regulation. The decision of which symbol to use is made on the basis of variations of the expression levels between time points, thus resulting in a pattern of the evolution of the expression of a given gene along the time axis, with three possibilities: decrease (D), no change (N) and increase (U). A threshold is defined and the symbol U is chosen if the difference between the expression levels in consecutive time points exceeds it. If that difference is negative and lower than the symmetric of the threshold, then the symbol D is chosen for the discretized matrix. If none of the above conditions is met, the chosen symbol is N.

Note that, since we are dealing with the variations between time points, the discretized matrix will have one less column than the original expression matrix. Although the original formulation of this algorithm allowed the existence of missing values, as different patterns for different patients are in comparison, the algorithm is changed so that to consider a missing transition whenever one or two of its time points correspond to missing values (the chosen symbol is '-').

The discretized matrix A is computed in two steps: a first one consists in building the matrix of variations between time points, A'' (N_g rows by ($N_t - 1$) columns), from the normalized expression matrix, A' (N_g rows by N_t columns).

$$A''_{ij} = \begin{cases} -, & if\ A'_{ij} = missing\ or\ A'_{i(j+1)} = missing \\ -1, & if\ A'_{ij} = 0\ and\ A'_{i(j+1)} < 0 \\ 1, & if\ A'_{ij} = 0\ and\ A'_{i(j+1)} > 0 \\ 0, & if\ A'_{ij} = 0\ and\ A'_{i(j+1)} = 0 \\ \dfrac{A'_{i(j+1)} - A'_{ij}}{|A'_{ij}|}, & if\ A'_{ij} \neq 0 \end{cases} \quad (1)$$

At this point, the discretized matrix comes from the assigning each variation value to a symbol, considering a given positive threshold, *t*:

$$A_{ij} = \begin{cases} D, & if\ A''_{ij} \leq -t \\ U, & if\ A''_{ij} \geq t \\ N, otherwise \end{cases} \quad (2)$$

In order to properly use this discretization technique, the data is normalized by gene to zero mean and unit standard deviation, and t is considered to be 1.

As already stated, numerous authors defend that the methods that consider the time dependencies perform better [39, 54], and so this is the most promising discretization strategy.

- Gene mean and standard deviation - in this other technique, the same alphabet is used, but the choice of the symbol lies in the mean expression value and standard deviation for each gene for all time points, respectively $\overline{g_i}$ and σ_i for gene i: an alpha parameter is given, and its product with the standard deviation can be understood as a threshold ($\alpha \times \sigma_i$). If an expression value for a given

gene at a given time point exceeds the sum of the mean value with the mentioned threshold, the value is replaced with a symbol U. Instead, if the value is lower than the mean expression value for the gene taken the threshold, the symbol D is chosen. The values that fall in between these limits are replaced by N. Missing values are represented by '-', as above.

$$A_{ij} = \begin{cases} -, & if\ A'_{ij} = missing \\ D, & if\ A'_{ij} < (\overline{g}_i - \alpha \times \sigma_i) \\ U, & if\ A'_{ij} > (\overline{g}_i + \alpha \times \sigma_i) \\ N, & otherwise \end{cases} \quad (3)$$

Unlike the first technique, this one maintains the original number of columns in the discretized matrix, since each value is compared with the mean expression value for the gene.

3.3 Biclustering

Most of the biclustering algorithms deal with an NP-hard problem [58]. Nonetheless, there is a characteristic of the time series gene expression experiments that renders the problem tractable. It is the fact that when analyzing temporal expression data we are searching for coherent patterns with no temporal breaks. As such, the algorithm searches only for biclusters with contiguous columns, drastically reducing the problem complexity [60].

The biclustering algorithm used in this work is the CCC-Biclustering [60], which finds all maximal contiguous column coherent biclusters (CCC-Biclusters) by analyzing a discretized version of the expression matrix using a generalized suffix tree (see [69] for details on string processing techniques in general, and suffix trees in particular). First, the basic definitions are presented for strings, generalized suffix trees, CCC-Biclusters and maximal CCC-Biclusters. We use the authors' definitions to explain the algorithm, in a simple way (refer to the paper for more detailed information):

Definition 1 – String, Substring and Suffix

A string S is an ordered list of symbols over an alphabet Σ (with $|\Sigma|$ symbols) written contiguously from left to right. For any string S (with $|S|$ symbols), $S[i \dots j], (i \geq 0, j \leq |S|)$ is its (contiguous) substring starting at position i and ending at position j. $S[i \dots |S|]$ is the suffix of S that starts at position i.

Definition 2 – Suffix Tree and Generalized Suffix Tree

A suffix tree T of a string S is a rooted directed tree with exactly $|S|$ leaves, numbered 1 to $|S|$, such that 1) each internal node in T, other than the root, has at least two children, and each edge is labeled with a nonempty substring of S, 2) no two edges out of a node have edge labels starting with the same symbol, and 3) for any leaf i, the label of the path from the root to the leaf i exactly spells out the suffix of S starting at position i. A generalized suffix tree is a suffix tree built for a set of strings $\{S_i\}$.

Definition 3 – CCC-Bicluster

A CCC-Bicluster A_{IJ} is a subset of rows $I = \{i_1, \dots, i_k\}$ and a contiguous subset of columns $J = \{r, r+1, \dots, s-1, s\}$ such that $A_{ij} = A_{lj}$, for all rows $i, l \in I$ and columns $j \in J$. Each CCC-Bicluster defines a string S that is common to every row in I for the columns in J.

Definition 4 – Maximal CCC-Bicluster

A CCC-Bicluster A_{IJ} is maximal if no other CCC-Bicluster exists that properly contains it, that is, if for all other CCC-Biclusters A_{LM}, $I \subseteq L \wedge J \subseteq M \Longrightarrow I = L \wedge J = M$.

In Figure 3.a (example taken from [60]) one can observe a simple version of the generalized suffix tree obtained from the strings correspondent to the rows of the discretized matrix in Figure 3.b. To avoid confusion, this figure does not contain the leaves that represent string terminators that descend directly from the root. Here one can also see the maximal CCC-Biclusters found with more than one row.

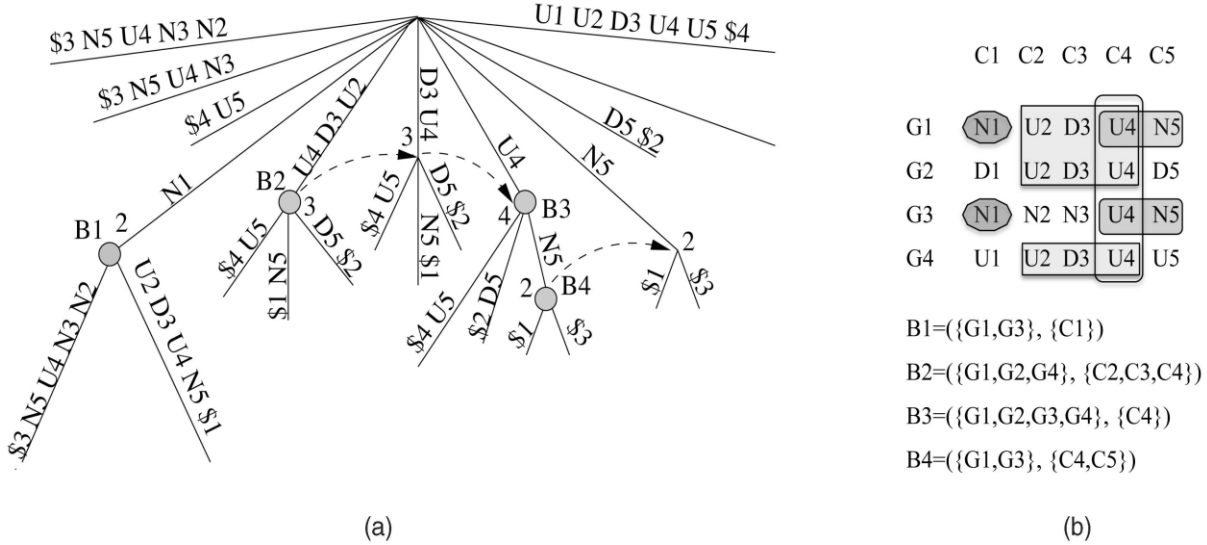


Figure 3 - a) generalized suffix tree for the discretized matrix in b). The circles, labeled B1 to B4 represent the maximal CCC-Biclusters with more than one row. In the matrix in b), the biclusters B1 to B4 and their sets of rows and columns are represented. Note that the strings of each bicluster corresponds to its expression pattern, or profile.

CCC-Biclustering is based on the definition of max node and the theorem below that relates nodes in the suffix tree with maximal CCC-Biclusters.

Definition 5 – Max Node

An internal node v of a generalized suffix tree T is called a Max Node iif it satisfies one of the following conditions:

1. It does not have incoming suffix links.
2. It has incoming suffix links only from nodes u_i such that for every node u_i, $L(u_i) < L(v)$ ($L(v)$ is the number of leaves of the node v).

Theorem 1 – Every maximal CCC-Bicluster with at least two rows corresponds to an internal node in the generalized suffix tree T that satisfies Definition 5, and each of these internal nodes defines a maximal CCC-Bicluster with at least two rows.

The algorithm that returns all the maximal CCC-Biclusters can be summarized in the following way: from the discretized gene expression matrix, get the set of strings $\{S_i, \dots, S_j\}$. Then build a generalized suffix tree T for that set of strings. Finally, analyze all the internal nodes to see if they meet the requirements of Theorem 1 to represent a maximal CCC-Bicluster.

All the time series expression data classifiers proposed in this thesis are based on CCC-Biclustering, using the resulting CCC-Biclusters as the class discriminative objects (for the sake of simplicity, these shall be simply called biclusters from now on, except if the situation requires the exact designation).

3.4 Classification

In the following subsections the developed biclustering-based classifiers are described and discussed, beginning with a k-Nearest Neighbors (kNN) method with different measures of similarity/distance between the classification objects (MS patients in this particular case study). Then, we propose a method based on meta-profiles, which represent similar expression patterns shared between patients, and a processing technique using representative sets of similar biclusters (meta-biclusters) classifying its information with state of the art algorithms. These are also used on the original dataset (and on a discretized version) in order to obtain a term of comparison for the obtained results.

3.4.1 Biclustering-based k-Nearest Neighbors (kNN)

The kNN algorithm is a very simple method in the category of supervised learning in pattern recognition. Its goal is to classify an object based on the k closest training instances. The k parameter is a positive integer, usually small, chosen empirically with the help of some cross validation schemes, for example. If k equals 1, a new object is simply classified with the class of the most similar (train) instance.

The simplest form of predicting the new class comes from finding the most frequent class in the set of the k closest training examples: each train instance has a vote for its class, and the predicted class is the one with more votes. In case both classes appear the same number of times for the k best scoring patients, the less frequent class in the original dataset is chosen (for this particular case, the bad responder class is chosen). To avoid these ties, k is usually an odd number.

Since the k best scoring neighbors do not have necessarily (and typically do not have) the same similarity with the test object, the classification algorithm should give preference to higher scoring train instances, thus resulting in a distance-weighted algorithm. The weights used can be a

function of their rank (with a weight of $\frac{1}{d}$, d being the rank or the distance to the test object), or be directly their own score (after normalization), i.e. the algorithm sums the scores for the two classes and predicts the one with the highest sum.

The matrix that represents the relationships between the test and train patients, from where the k most similar patients are selected to classify each test patient, shall be called from now on as the score matrix between patients (higher the score, the higher the degree of similarity between the patients). It has as many rows as the number of train patients, and the number of columns equal to the number of test patients.

To reduce the effects of unbalanced data, as is the case with this dataset, a penalty can be included in the computation of this matrix, reducing the actual score between the two patients, if the class of the train patient is the overrepresented one.

The algorithm of the kNN classifier for this particular problem can be summarized as follows:

Algorithm 1 – Biclustering-based k-Nearest-Neighbors
Input: Score Matrix between patients
1 **for** each test patient **do**
2 build a list with the k highest scoring train patients: $kPatients$
3 from $kPatients$, separate the scores from each class: $scores0$ and $scores1$
4 **if** $sum(scores1) > sum(scores0)$ **then**
5 $predicted\ class = 1.$
6 **else** $predicted\ class = 0.$

In the steps 4-6 the criteria based on the sum of scores of each class is presented, thus considering the scores between the test-train patients as weights. However, there are other criteria that can be used to predict the test patient's class (steps 4-6), such as:

- more frequent class in the k closest instances
- threshold comparison for the scores of a given class ($scores0 > t$ or $scores1 > t$))

The score matrix can be computed in several different ways, which will be discussed below.

3.4.1.1 Computation of the Score Matrix between patients

In this subsection, we explain the different approaches to compute the score matrix between the patients in the dataset. The possibilities include similarities based on biclusters (as a whole or considering its symbolic elements) and expression profiles.

3.4.1.1.1 Biclusters Similarities

In this method to compute the score matrix, after the biclustering step, the algorithm proceeds with the computation of a matrix that represents the similarities between the biclusters of each pair of test and train patients. The entry (i, j) of this matrix represents the degree of similarity between a bicluster B_i from the set of biclusters of a test patient, and a bicluster B_j from the set of biclusters of a

train patient. This similarity can be computed from the fraction of common elements of the two biclusters in comparison:

- in the two dimensions (genes and time points), with or without the symbolic pattern information.
- just in the genes dimension, again with or without the information of the bicluster profile, i.e. the symbols of the discretized matrix that represent the evolution of the gene along the time points.

Equation 4 represents an adaptation of the Jaccard Index used in [60] to compute the similarity between biclusters, adapted to take into account the symbols of the biclusters under comparison.

It is important to note that since the biclusters to be compared are from different patients, two biclusters sharing the same genes and time points do not necessarily have the same expression pattern.

$$J(B_1, B_2) = \frac{|B_{11P}|}{|B_{01}| + |B_{10}| + |B_{11}|} \tag{4}$$

where

$$B_{11P} = \{(i,j): (i,j) \in B_1 \wedge (i,j) \in B_2 \wedge Symbol(B_1(i,j)) = Symbol(B_2(i,j))\},$$

$$B_{11} = \{(i,j): (i,j) \in B_1 \wedge (i,j) \in B_2\},$$

$$B_{01} = \{(i,j): (i,j) \notin B_1 \wedge (i,j) \in B_2\},$$

and

$$B_{10} = \{(i,j): (i,j) \in B_1 \wedge (i,j) \notin B_2\}.$$

In Equation 4, B_{11P} is replaced by B_{11} to compare biclusters of the same patient, or across patients disregarding the expression patterns. On the other hand, if the comparison between biclusters is needed only in one of the dimensions, the equations shown above can be modified, considering only the i or j dimension (common genes/ common time points).

For each test-train pair of patients, a similarities matrix, where

$$Sim_{ij} = Sim(B_i, B_j) = J(B_i, B_j)$$

is obtained as described above.

However, it is necessary to transform this measure of similarity between two patients in a single score value, to enter the k-nearest neighbors classification, which is not possible with the previous matrix. This transformation can be carried with two techniques:

- For each test bicluster B_i, find the best similarity along the train biclusters B_j. The score that represents the relationship between the two patients $S(P_{test}P_{train})$, is the average (of the maximum) similarity along the set of test biclusters:

$$S(P_{test}, P_{train}) = \frac{\sum_{i=1}^{\#B_{test}} \max(Sim(B_i, B_j), j \in \{1, \dots, \#B_{train}\})}{\#B_{test}} \quad (5)$$

where $\#B_{test}$ and $\#B_{train}$ represent the number of biclusters of, respectively, the test and train patient.

- The score between patients is calculated using the percentage of similarity values between biclusters that are over a predefined threshold.

$$S(P_{test}, P_{train}) = \frac{Sum(BinarySim)}{\#Sim} \quad (6)$$

where $BinarySim$ is the binary matrix resulting from the decision of the similarity being or not superior than a threshold t, defined as follows.

$$BinarySim(i,j) = \begin{cases} 1, if\ Sim_{ij} \geq t \\ 0, otherwise \end{cases}$$

and $\#Sim$ is the number of elements in the similarities matrix between the biclusters for the test-train pair of patients.

The algorithm is stated in the following manner:

Algorithm 2 – Score Matrix based on bicluster similarities

1 **for** each test patient **do**

2 **for** each train patient **do**

3 compute similarities matrix between biclusters: Sim (where $Sim_{ij} = J(B_i, B_j)$)

4 compute score between patients: $S(P_{test}P_{train})$

The first approach does not consider the influence of the similarities between all biclusters, but instead, only the best similarity value is taken for each test patient. On the other hand, the second approach, while considering all the pairs of biclusters that exceed a given threshold, might be considering more “bad” pairs of biclusters. It could also be interesting to consider a weighting scheme, in order to favor the best pairs of biclusters in the calculation of the score between the two patients.

Filtering Non-Discriminative Biclusters based on Similarities

The importance of a feature selection was already pointed out in the related work (Section 2.4). With this same MS dataset, previous authors concluded that the prediction accuracy was significantly increased when the feature space was reduced to the most discriminative genes [24, 39].

Keeping this concept in mind, a new filter must be proposed to eliminate features, which are in this case profiles or biclusters, instead of only genes, that discriminate poorly the two classes, or even contribute to the confusion between them.

In order to build such a filter, the first necessary step is to define what should be understood as the discriminative power of a given bicluster. A possible definition is proposed as follows.

Definition 6 – Discriminative Bicluster

A bicluster is discriminative for a class c, and so should be maintained in the feature space, if and only if the proportion of similar biclusters (above a similarity threshold) of the class c is greater than a predefined threshold (class proportion threshold).

In other words, the similarities between a bicluster (belonging to a patient of class c) and all the other computed ones are computed. Then, the sufficiently similar ones are chosen, taking into account that the biclusters originating from the same patient are not considered. Finally, the class proportions are computed in this reduced set of biclusters, and if the class c proportion (0 if there are no similar biclusters belonging to patients of the class c, and 1 if all the similar biclusters found belong to patients of class c) is over a given threshold, the bicluster is maintained. Otherwise it is rejected, as can be seen in the brief description of the algorithm in pseudo code (Algorithm 3). Again, to avoid some problems related to an overrepresented class, it is possible to use a threshold for each class.

Algorithm 3 – Filter Non-Discriminant Biclusters based on similarities
Input: Similarities Matrix between all biclusters from all train patients
Similarity threshold for each class: $st0$ and $st1$
Class proportion threshold for each class: $ct0$ and $ct1$

```
1  for each bicluster B_j do
2      for each bicluster B_i, excluding those from the same patient do
3          if B_i belongs to a patient of class 0 then
4              if Sim(B_i, B_j) > st0 then
5                  add B_i to the list of similar biclusters to B_j
6          else if Sim(B_i, B_j) > st1 then
7              add B_i to the list of similar biclusters to B_j
8      compute class proportions for the list of similar biclusters to B_j: prop0 and prop1
9      if B_j belongs to a patient of class 0 then
10         if prop0 > ct0 then
11             add Bj to the filtered set of biclusters.
12     else if prop1 > ct1 then
13         add B_j to the filtered set of biclusters
```

3.4.1.1.2 Profile Similarities

Another strategy developed to compute the score matrix between the test patients and the training set, relies on the fact that each bicluster is represented by a pattern of symbols, that shall be called a profile from here on, result of the discretization of the normalized matrix of gene expression values, and representative of the evolution of that gene expression along the time points.

These profiles can be compared across different patients, since the whole set of profiles for each one is computed right after the biclustering step. The conditions that have to be met in order to state that a certain profile is shared between two patients, contributing to their score of similarity, can be adjusted, as the minimum number of common genes and/or time points. This means that beside representing the same expression pattern, to be considered as a shared profile, it has to represent biclusters (from the two patients) that have the minimum required number of genes and time points in common. Using this concept, the score matrix between patients is computed, in which an entry (i, j) represents the number of profiles shared between train patient i and test patient j. This algorithm is described concisely as follows:

Algorithm 4 – Score Matrix based on profile similarities
Input: List with each patient's set of biclusters and profiles
Minimum number of common genes: $minGenes$
Minimum number of common time points: $minTimePoints$
1 **for** each test patient j **do**
2 **for** each test profile p **do**
3 **for** each train patient i **do**
4 **initialize** $S(i, j) = 0$;
5 **if** the train patient i's set of profiles contains the test profile p **then**
6 compute number of common genes between the two biclusters with the profile p: ncg
7 compute number of common time points between the two biclusters with the profile p: nct
8 **if** $ncg > minGenes$ and $nct > minTimePoints$ **then**
9 increase $S(i, j)$

If there is a case where the test profile is found to also represent more than one bicluster of the train patient set of biclusters, only the first match is considered, although all the matches should be considered. Since this is a rare situation, this approach is maintained for the sake of simplicity.

Instead of the sum of shared profiles between patients, the entry (i, j) of the score matrix can be computed with a polynomial kernel (its degree is given as a parameter, but generally a quadratic kernel is used). With this measure, the patients with a greater number of biclusters are penalized,

because a higher number of profile matches could be due to random events. The score expression with kernels and its definition are stated below.

$$S(P_{test}, P_{train}) = 2K(x, y) - (K(x, x) + K(y, y)) \quad (7)$$

The Kernel function is defined as

$$K(x, y) = x\, y^t$$

where x and y are binary vectors with length equal to the number of possible profiles (computed after the biclustering step). The element in the position i is 1 if the profile i in the space of possible profiles is present for the patient, and is 0 otherwise. Hence, the number of common elements (profiles) is simply given by the inner product between the two vectors ($K(x, y)$).

The use of a metric based on a polynomial kernel is usually aimed at computing a distance. However, the symmetric version of Equation 7 is used in order to get a score instead of a distance, in which the maximum similarity is zero. Here, the introduction of a minimum number of common genes and/or time points, proves to be a rather more difficult task.

When using the whole set of computed biclusters, as previously mentioned, one can be carrying out computations with statistically insignificant biclusters [60], and consequently, profiles, that might adulterate the results. To minimize this effect and reduce complexity, a filter based on a p-value threshold can be applied, or even the selection of a given number of the best biclusters (in terms of p-value).

Filtering Non-Discriminative Biclusters by Profiles

One way to achieve a filter based on the profiles is to use a similar strategy to the one used to compute the similarities between patients based on the shared profiles. Now, a given profile is kept in the filtered set, if and only if it contributes more to the discrimination than to the confusion between classes, that is, a profile in a train patient's set of profiles is maintained if and only if is shared by more patients of the same class than of the other class. Like before, a parameter of a minimum number of genes and/or time points shared, can be included and fine-tuned. If the train patient's profile is not repeated across the remainder patients, it is maintained because it might be shared with a new (test) patient, thus contributing to the classification.

As one uses the information of the train patient's responder class to a successful filtering, this method is included in the category of supervised learning. This prevents the use of a new patient, with unknown responder class, in this step, unless the two possible filtered sets (one for each of the two possible classes of the test patient) are computed, as shall be explained further.

The initial number of selected biclusters (and thus, profiles) is also a critical aspect of this filtering step, due to the reasons already explained for the first p-value filter: with all the biclusters for all patients, a significant, and possibly discriminative bicluster of a patient, can be similar to non-significant ones (as explained in [60]) from patients of a different class, which eliminates it, when, in

fact, it should be kept in the filtered set. On the contrary, if the initial number of biclusters is too low, finding shared profiles becomes more difficult, due to the possible loss of important information, which is also the main difficulty when analyzing a small dataset. Reducing the number of patients used in this filtering step (like with k-fold cross validation), will surely reduce the discriminative power between classes. To avoid this risk, the leave-one-out cross validation (LOO CV) can be used, taking only one patient to be tested, with $N-1$ patients remaining in the training set (N equals the total number of patients in the dataset). The algorithm of the filtering step for the training set only is presented:

Algorithm 5 – Filter Non-Discriminating Biclusters by profiles
Input: Train patients' sets of profiles and biclusters
Minimum number of genes: $minGenes$
Minimum number of time points: $minTimePoints$
1 **for** each train patient tp **do**
2 **for** each profile p of train patient tp **do**
3 **initialize** $sameClassCounter = 0$;
$differentClassCounter = 0$;
4 **for** all patients except tp **do**
5 **if** other patient's set of profiles contains profile p **then**
6 compute number of common genes between the two biclusters with profile p: ncg;
7 compute number of common time points between the two biclusters with profile p: nct;
8 **if** $ncg > minGenes$ and $nct > minTimePoints$ **then**
9 **if** other patient's class = tp's class **then**
10 increase $sameClassCounter$;
11 **else** increase $differentClassCounter$;
12 **if** $sameClassCounter > differentClassCounter$ **then**
13 keep profile p in the filtered set for patient tp.

As stated before, the filter can be performed using only the information of the training patients, and testing with the remainder patients. Alternatively, and in the case of LOO CV, the filter can be realized using all the patients, by considering the two possible classes for the test patient, and originating two sets of filtered profiles/biclusters: one resulting from the unknown class being 0, and other from being 1. With the filtered sets of biclusters/profiles, the score matrix between patients can then be computed. Depending on the filtered set, the computation of the score matrix can also be carried out in different ways:

- The most simple case is when the profile filter is applied to the training set (no matter what size), and each test patient's set of profiles is faced with the filtered ones for each train patient, searching for shared ones. This number of shared profiles defines the score between the two patients, that is, for a train patient i and test patient j, it is the entry (i,j) of the score matrix.

- Applying the filter again to the training set, in this approach the test patients' profiles are then also filtered, by comparing them with the filtered train set profiles. Since the test patient's

responder class is unknown, the two possible sets of filtered profiles have to be computed (one for each possible class). Then, both score matrices between patients are computed just like in the previous case. The choice of the best score matrix is discussed below.

- In the last strategy, exclusively applicable to LOO CV, the test patient is included in the profile filter, considering the two possible classes. This also results in two score matrices, from which only one is used in the classification step. Instead, if each profile of the test patient is compared to each one of the two filtered sets, the choice is made at the score level (choosing the maximum, for example), and not between the two score vectors (not a matrix, since it is constructed with only a test patient).

In what follows, the algorithm will be succinctly explained for the case where the filter is applied only to the training set, but the process is similar to when the test patient is included, aside the fact that two score matrices are computed, as already mentioned.

Algorithm 6 – Score Matrix based on filtered profiles
Input: Filtered set of profiles for the train patients
Set of profiles for the test patients
1 **for** each test patient j **do**
2 **for** each train patient i **do**
3 compute number of shared profiles between patients = $S(i,j)$

The choice between the two score matrices computed (for the two possible unknown classes) can be performed based on several criteria: based in the choice of the score matrix (or vector) with more shared profiles (matrix/vector with highest sum, or, similarly, highest average); or, in a more sophisticated way, the choice can be based in the variance of the score vector, where the algorithm chooses the one with least variance between classes, meaning that the vector that is more class specific is chosen.

The discriminative power of a profile can also be calculated based on its correlation with a given class, instead of only considering its contribution to discrimination or confusion between classes. In order to achieve a filter of such nature, the Pearson's chi square (χ^2) is used [70], as indicated in Equation 8, for a profile P_k.

$$\chi^2(P_k) = \sum_{c \in \{C_1 \dots C_j\}} \sum_{p \in \{P_k, \overline{P_k}\}} \frac{(\Pr(p,c) - Ex(p,c))^2}{Ex(p,c)} \tag{8}$$

with $\Pr(p,c) = \frac{\#(p,c)}{N}$, $\quad \mathrm{Ex}(p,c) = \frac{\#(p)}{N} \times \frac{\#(c)}{N}$

In these equations, N is the total number of instances in the dataset used in the filter, i.e., the number of patients in the training set. Note that $\#(P_k, c)$ represents the number of instances of class c

that contain the profile P_k, and $\#(\overline{P_k}, c)$ denotes the number of instances of class c that do not contain the profile P_k. In this work, $c \in \{0,1\} = \{bad, good\}\ responder$.

3.4.1.1.3 Comparing Discretized Matrices

Three additional strategies were designed to compute the score matrix between patients, all based in the comparison between the discretized matrices (with biclusters information) of pairs of patients, but with different levels of complexity.

1) Element of Bicluster

The most simple approach lies just on the fact that a symbol in the discretized matrix is either an element of a bicluster or not. This way, for each patient, a binary matrix is computed, where (i, j) is 1 if the respective symbol in the discretized matrix belongs to a bicluster, and 0 otherwise. Then, the score between a pair of test-train patients is simply the inner product between the two binary matrices, returning the number of common elements. This algorithm can be summarized as follows:

Algorithm 7 – Score Matrix based on discretized matrices – element of bicluster
Input: Set of discretized matrices and biclusters for all patients
1 Compute binary matrices for each patient, with 1 if the element belongs to a bicluster, and 0 otherwise
2 **for** each test patient j **do**
3 **for** each train patient i **do**
4 $S(i,j) = binaryMatrix_i \cdot binaryMatrix_j$
where · represents the inner product between the two matrices.

To penalize the patients with a greater number of biclusters, an adapted version of the polynomial kernel presented in the subsection 3.4.1.1.2 (Equation 7) may be used. It is crucial to filter the biclusters for statistical significance, as mentioned also for the other methods, since when all biclusters are used in this calculation, all (or nearly all) elements of the discretized matrix, belong at least to one bicluster, resulting in a matrix with all elements equal to 1. Note that the bicluster filter can also be applied here, potentially leading to an improvement in the results of classification.

We further note that, besides its simplicity, this is a really rough strategy, mainly due to the fact that no consideration is made regarding the nature of the symbols of the biclusters. It only matters if a given position (i, j) belongs to a bicluster or not. This means that a perfect match between the symbols has the same importance as a mismatch, and thus one can already expect that this method will not lead to satisfactory results.

2) Symbol pairing

To overcome the previous situation, we devised a new strategy that considers the nature of the comparison between symbols belonging to biclusters. For that purpose, each element of the

discretized matrix for one patient (with all or only a given number of biclusters, and considering that 'X' is the symbol used to represent the elements that do not belong to any of the selected biclusters) is compared to the corresponding element of the other patient's discretized matrix, considering the following symbol comparison possibilities: $X - X$, $X - \{U, N, D\}$, $\{U, N, D\}$ mismatch, $\{U, N, D\}$ match.

The division of the different match possibilities has the goal of allowing a certain flexibility in the comparison results, since one pairing might have a medium importance, like an $X - \{U, N, D\}$ pair, while the most important one remains the perfect match, exception made to the $X - X$ pair.

Algorithm 8 – Score Matrix based on discretized matrices – symbol pairing
Input: Discretized matrices of the biclusters for all patients
Array of weights to use in the calculation of the score between patients, w
1 **for** each test patient j **do**
2 **for** each train patient i **do**
3 compare the discretized matrices of patients j and i, and build a vector aux, as follows:
3.1 $aux[0] \leftarrow$ number of pairs $X - X$
3.2 $aux[1] \leftarrow$ number of pairs $X - \{U, N, D\}$
3.3 $aux[2] \leftarrow$ number of perfect matches $\{U, N, D\}$
3.4 $aux[3] \leftarrow$ number of mismatches $\{U, N, D\}$
4 $S(i, j) = w \cdot aux$

It is important to notice that if only the perfect match is to be considered, there is a simple method of comparison returning 1 if the elements are equal and 0 otherwise. The sum of this binary matrix defines the score between the two patients.

3) Symbol pairing with Time-Lags

The last of these three strategies introduces a new consideration: the possibility of time-lags in the gene expression. As one might expect, even when the same genes are involved in some mechanism for different patients, the expression evolution pattern for one patient might be delayed when faced to the other's. This possibility should be taken into consideration, as it is a consequence of the patient-specific expression (in the particular case of treatment response, it is a patient-specific response rate), not considered in the methods discussed so far, and shown to be of particular importance in previous time-series expression studies [39].

In this approach, all the biclusters (or filtered ones) of the test patient are analyzed. A parameter for a maximum time-lag (instants to consider in the delay) is defined, and then, for each of the test biclusters, the comparison of the elements of the genes and time points that form the bicluster is made with the ones from the discretized matrix of a train patient, resulting in a score defined by the sum of 1's and 0's whether we have a perfect match of symbols or not. The comparison is then also performed considering translations in the time points, from 0 (the original position) to the maximum time-lag, and its symmetric, allowing translations in both directions along the time axis. The time-lag

that returns the highest score is chosen, and the binary submatrix resulting from that specific comparison is written in the final matrix. The sum of this final matrix represents the score between the two patients, the entry (i, j) of the score matrix for the whole set of patients.

Algorithm 9 – Score Matrix based on discretized matrices – symbol pairing with Time-Lags
Input: Discretized matrices of the biclusters for all patients
Maximum time-lag: $maxT$
1 **for** each test patient j **do**
2 **for** each train patient i **do**
3 **for** each test bicluster B_j **do**
3 **for** each $timeTranslation$ in $\{-maxT, -maxT+1, \ldots, maxT-1, maxT\}$ **do**
4 compute number of matches between the discretized matrices on the elements (with $timeTranslation$) of B_j;
5 compute the submatrix for the translation with highest number of matches and write its elements on the corresponding elements of a matrix M;
6 $S(i,j) = sum(M)$

3.4.2 Meta-Profiles Classification

Having explored different strategies to combine biclustering and the kNN classification algorithm, we now present a new classification approach following the biclusters computation. It is based on the mentioned fact that each bicluster has a pattern of temporal evolution in terms of gene expression, which is represented by a profile. The same expression profile can be shared across different patients, a characteristic used before in the Profile Similarities strategy to compute the score matrix between patients (section 3.4.1.1.2). In this approach, what is interesting is to evaluate the class proportions of each profile. In other words, we wish to analyze if a given profile is shared between more patients of one of the classes. For example, if a train profile is shared only between good responders, then if a test patient shows an equivalent expression profile, the probability of this patient being a good responder increases. One can consider the profiles in the original space as representatives of all the equivalent biclusters' expression profiles, hence the naming of meta-profiles. This algorithm can be described as follows:

Algorithm 10 – Meta-Profiles Classification
Input: Meta-profiles space (vector with all the possible profiles found in the biclustering step).

1 **for** each meta-profile m **do**
2 **for** each train patient tp **do**
3 **if** the set of profiles of tp contains the meta-profile m **then**
4 add tp to a list: $TrainIndexes$
// with the class information from the train patients that share the meta-profile, each class proportion can then be computed ($X\%$ shared by good responders and $(100 - X)\%$ shared by bad responders):
5 compute the meta-profile m classes proportions: $Proportion0$ and $Proportion1$
6 **for** each test patient i **do**
7 **for** each test profile p **do**
8 **if** test profile p belongs to the meta-profiles space **then**
9 associate the meta-profiles class proportions to the test profile p
//use a criterion based on the class proportions to classify the patient: maximum, average, sum. The comparison between the average proportions is shown here:
10 **if** $averageProportion0 > averageProportion1$ **then**
11 $predicted\ class$ = 0
12 **else** $predicted\ class$ = 1

The steps 10 to 12 in the algorithm described above can change based on the criterion that is used for the classification. A test patient has a list of proportions for the class 0 and another for the class 1, and one can compute their sum, average value, maximum, etc., and compare the two values between them, or compare only one of them with a predefined threshold to predict the test patient's class.

Because of the discussed difference in the class distributions (there are more good responder patients, in a proportion of approximately 62% to 38% of bad responders), a penalty can also be introduced here to soften the binary classification, namely in the adjustment of the threshold or penalizing a given score.

It is worth noting that in this classification strategy, we do not consider the similarities at the level of genes and time points shared. This means that it is possible to have contributions of biclusters that share the profile but not a single gene or time point. Nonetheless, disregarding the shared time points, allows for the different patient specific response rates to be aligned (in terms of time delays), which could prove to be an advantage to the classification.

3.4.3 Meta-Biclusters Classification

All the strategies discussed until now were set upon simple machine learning techniques like the kNN algorithm and a simple meta-profiles representation to classify the test patients in a fashion similar to a voting scheme, with the profiles' classes proportions. Now, we shall use biclustering as a preprocessing step, and build a binary matrix from the biclusters, which will then be classified with more complex classification algorithms, like Support Vector Machines (SVM), Decision Trees, or other state of the art machine learning techniques.

The principle is based on meta-biclusters, which represent a set of similar biclusters. These are obtained by performing a hierarchical clustering in the bicluster space, for all patients. The number of meta-biclusters is a user-defined parameter, an integer higher than 1. Previously, the distances matrix has to be computed, and this is achieved by computing a similarities matrix between all biclusters (Equation 4), and turning this measure into a distance:

$$Distance_{ij} = 1 - Similarity_{ij} \tag{9}$$

Defining the steps of this algorithm:

Algorithm 11 – Meta-Biclusters
Input: distances matrix for all biclusters.
1 Apply Hierarchical Clustering to the distances matrix, with K meta-biclusters
// Build a binary matrix M with as many rows as patients and columns as meta-biclusters
2 **for** each patient p **do**
3 **for** each meta-bicluster m **do**
4 **if** patient p has at least one bicluster present in the meta-bicluster m **then**
5 $M(p,m) = 1.$
6 **else** $M(p,m) = 0.$

Finally, add each patient's responder class to a new column of this matrix (becomes the last column). This binary matrix can then be introduced as the dataset into various state of the art classification algorithms.

3.4.4 State of the Art Classifiers

In order to obtain a term of comparison for the obtained results from the classification methods proposed in this thesis, the dataset was classified using state of the art classifiers, either from the original dataset, either from the result of the meta-biclusters method aforementioned. This was carried using the software package Weka (Waikato Environment for Knowledge Analysis), available in www.cs.waikato.ac.nz/ml/weka. It is an open source software, issued under the GNU General Public License, and is a collection of machine learning algorithms for data mining tasks, from which we used: Decision Tree, k – Nearest Neighbors (kNN), Support Vector Machines (SVM), Logistic Regression,

Radial Basis Function (RBF) Network and Multilayer Perceptron (MLP). A brief description of these classifiers is presented in Appendix B.

3.5 Evaluation

Since we are dealing with a classification task, the evaluation of any method must be based on the algorithm capability of predicting the responder class of a given MS patient. If equal misclassification costs are considered, the accuracy of a classifier C is the probability of correctly predicting the class of a randomly selected instance [71]. The meaning of a single instance's prediction accuracy is not significant. Instead, the prediction accuracy is computed for a finite dataset, and accompanied by a confidence interval.

$$Prediction\ Accuracy = \frac{\sum_{i=1}^{\#instances} \delta(C(i), class(i))}{\#instances} \tag{10}$$

where

$$\delta(i,j) = \begin{cases} 1, & if\ i = j \\ 0, & otherwise \end{cases}$$

and $C(i)$ is the predicted class for instance i and $class(i)$ is its actual class.

Nevertheless, prediction accuracy comes accompanied with some problems, such as the assumption of equal misclassification costs and approximately uniform class distribution, which do not happen in most of the clinical classification problems (for example, 95% of healthy individuals in a cancer case study against only 5% of disease instances).

3.5.1 Confusion Matrix

The confusion matrix for a classifier can provide more information about its performance (see Table 3) [72].

Table 3 - Representation of a confusion matrix for binary prediction.

		Predicted	
		negative	**positive**
Real	**negative**	**a** TN – true negatives	**b** FP – false positives
	positive	**c** FN – false negatives	**d** TP – true positives

From this type of matrix, one can retrieve several measures of a classifier's performance, such as:

$$Accuracy = \frac{(a+d)}{(a+b+c+d)} = \frac{TN+TP}{Total} \tag{11}$$

$$sensitivity = TP\ rate = \frac{d}{(c+d)} = \frac{TP}{real\ positives} \quad (12)$$

$$specificity = TN\ rate = \frac{a}{(a+b)} = \frac{TN}{real\ negatives} \quad (13)$$

$$precision = predicted\ positive\ value = \frac{d}{(b+d)} = \frac{TP}{predicted\ positives} \quad (14)$$

$$FP\ rate = \frac{b}{(a+b)} = 1 - specificity \quad (15)$$

$$FN\ rate = \frac{c}{(c+d)} = 1 - sensitivity \quad (16)$$

3.5.2 Receiver Operating Characteristics curve

Another possibility to analyze a classifier's performance is to visualize the two most informative values: True Positive (TP) rate and False Positive (FP) rate, using a curve, such as the Receiver Operating Characteristics (ROC) curve. This curve provides information about a classifier's performance for all misclassification costs, and all possible class ratios or proportions. Finally, it allows the investigator to see under which conditions a classifier C1 outperforms a classifier C2 [72].

A ROC curve is a plot that relates the false positive rate on the X-axis, with the true positive rate on the Y-axis. The point (0,100) represents the perfect classifier, in which all the cases/instances are correctly predicted. On the opposite end, the point (100,0) represents the worst case, incorrectly predicting all the instances. The point (0,0) means that all the instances are classified negatively, while the point (100,100) identifies the case where all instances are positively classified.

Generally, the classifier has an adjustable parameter that balances the TP and FP rates (increasing one at the cost of the other), allowing the construction of the ROC curve with different (FP,TP) pairs. In this work we use only a few (FP,TP) pairs obtained for different set of parameters, thus originating an approximate ROC curve. Figure 4 represents a fictional example of a ROC curve for two classifiers, C1 and C2 [72].

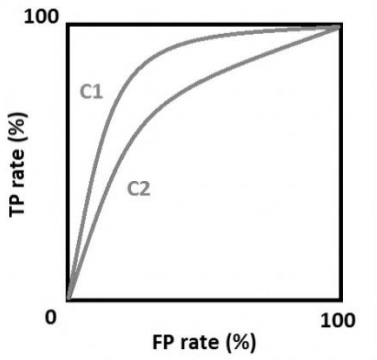


Figure 4 - Fictional example of a ROC chart with two ROC curves, corresponding to the (FP,TP) pairs for two classifiers: C1 and C2 (adapted from [72]).

The measure of the area under the curve (AUC) is often used as a statistic for model comparison. Nonetheless, it has been recently challenged, based on evidences showing AUC is a quite noisy classification measure [73]. Essentially, this measure can be interpreted as the probability of assigning a higher score to a positive example than to a negative one, when one positive and one negative example are randomly picked [72].

3.5.3 Cross Validation

In order to evaluate a classifier's performance in a given dataset, there are a few different methods available, some of which are used in the course of this work.

It is a known fact that no accuracy estimation is correct all the time [74]. Nevertheless, there have been several studies to investigate the pros and cons of various methods of accuracy estimation, although a trade-off is normally necessary between bias and variance.

The most usually chosen strategy is the k-fold cross validation (CV) scheme, also known as rotation estimation, where the dataset D is randomly partitioned into k mutually exclusive subsets, of equal (or approximate) size: $D_1, \ldots D_k$. Then, for each fold (1 to k), the classifier is trained with the $k-1$ remaining subsets, and tested with the subset D_k. This is repeated k times, and the overall prediction accuracy estimate is the mean prediction accuracy for all k folds. For each fold, the prediction accuracy is the number of instances correctly classified divided by the total number of instances in the test subset, $|D_k|$.

A special case of the k-fold cross validation occurs when k equals the number of instances in the dataset. It is then called Leave-one-out (LOO) cross validation (CV), also known as jackknife, because one instance is left out to test the classifier, while all the others constitute the training set. Since all the combinations of k splits of the dataset are run through, LOO cross validation is an example of a complete cross validation. Finally, if the class proportions in the original dataset are maintained in the subsets or folds, it is called a stratified cross validation [71].

Another category of these accuracy estimation methods is the bootstrap method [75]. Here, from a dataset with N instances, a bootstrap sample is built by taking N samples random and uniformly from the original dataset. The main difference is that, in this case, the collected samples for the bootstrap can be repeated, since the sampling is done with reposition. However, this method fails to return the expected value when the classifier is a perfect memorizer (e.g., one nearest neighbor classifier) and the dataset has a binary class random distribution [71]. For these reasons, this method is not used or mentioned in the other studies previously discussed for MS classification.

Only two evaluation methods are used in this work: LOO cross validation, and 5 repetitions of 4-fold cross validation. Although the second strategy is the one used in previous related works [24, 39], especially because its statistical significance is more meaningful due to more possible random

splits, the LOO cross validation presents the advantage of using almost all the data available, thus enriching the training of the classifier. This is particularly important when the dataset under study is limited in the number of instances, as is the case, although it increases the possibility of the occurrence of overfitting related issues [71].

There are several studies that compare the accuracy estimation methods, and, focusing on the k-fold cross validation, the general consensus seems to be that a 10-fold stratified cross validation is the best choice in model selection [71, 76]. However, when comparing LOO cross validation to k-fold cross validation with k small (2 to 5), LOO seems to present a smaller bias and RMS (root mean square) error [76]. Actually, some results suggest that LOO cross validation leads to nearly unbiased estimates, but with higher variability, increasing the risk of unreliable estimates, especially when the dataset number of instances is small [71].

3.5.4 Parity Tests

All the previously discussed evaluation methods allow for a grasp of a given classifier's performance. Nonetheless, even if they provide measures that suggest that one classification technique performs better than other, we would be interested in a more quantitative measure revealing if the differences between the two models under comparison are statistical significant, with a certain confidence degree of C%.

When both classifiers are applied on the same model set, the sought measure can be achieved with the use of parity tests, or paired t-tests since they resort to the t-Student distribution to assess the statistical significance of the differences between the different classifiers' results. Parity tests have been widely used in this kind of model comparison, and a more comprehensive description can be found in [77].

Providentially, there is a large collection of methods that perform these tests. In this thesis, we used the paired t-test built-in function in the Microsoft's Excel ® to compute the p-value associated to the differences between the prediction accuracies obtained for different classification techniques.

The p-value is associated to the probability of the null hypothesis being true (in this case, the null hypothesis is that there are no differences between the two classifiers' performance). This means that a lower p-value corresponds to more significant differences between the classifiers' results. The null hypothesis is usually rejected whenever the p-value is less than 0.05, corresponding to a 95% confidence degree that the classification methods are statistically different [77].

4 Experimental Results

In this section we present the main experimental results obtained by the application of the proposed biclustering-based classification methods, explained in detail in Section 3.4, whose workflow we recall from Figure 2 (Section 3). These classifiers were applied on the dataset resulting from gene expression time series collected from MS patients under IFN-β treatment, as described in Section 3.1. We start by displaying some examples of the biclusters resulting from the CCC-Biclustering method used as features in the classification methods proposed. Then, the most relevant results, such as confusion matrices and values of prediction accuracy, as well as approximate ROC curves, are presented for the developed classification strategies, and for some of the state of the art machine learning techniques, thus allowing a term of comparison for our classifiers' performance.

4.1 Biclustering

For a better understanding of the bicluster concept, in this section, the best two biclusters (in terms of statistical significance) are displayed as an example for two patients: a good and a bad responder, respectively in Figure 5 and Figure 6. The images are obtained with BiGGEsTS (Biclustering Gene Expression Time Series), an integrated environment for biclustering analysis of time series expression data [78], which uses the CCC-Biclustering algorithm already discussed in Section 3.3 [60]. Note that the computed biclusters with respective expression pattern in terms of temporal evolution, are the features used in all the presented biclustering-based classifiers.

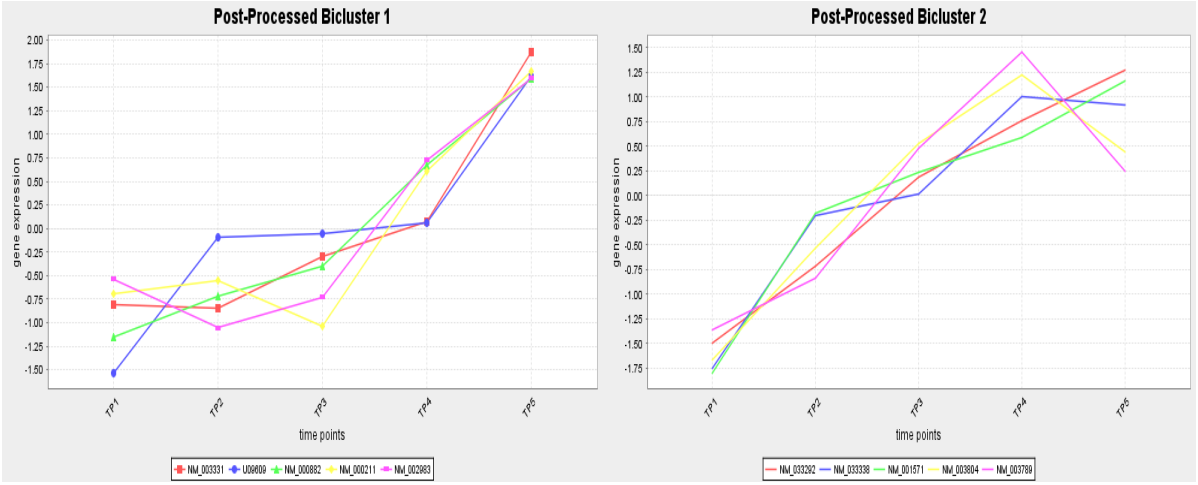


Figure 5 - Graphical representation of the two most significant biclusters computed for patient 1, a good responder. The x-axis represents the time points, and in the y-axis represents the gene expression. Each line corresponds to a different gene (identified in the chart legend).

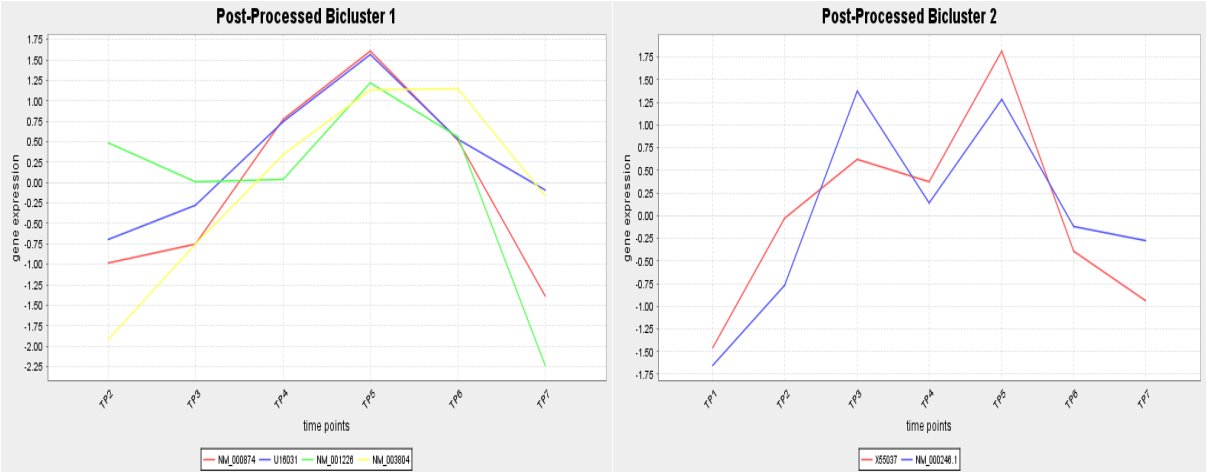


Figure 6 - Graphical representation of the two most significant biclusters computed for patient 52, a bad responder. The x-axis represents the time points, and in the y-axis represents the gene expression. Each line corresponds to a different gene (identified in the graph legend).

4.2 Biclustering-based k – Nearest Neighbors classifier

As explained in Section 3.4.1, this classification method lies in the similarity between patients to predict the class of a new (unlabeled) patient. Its classification is performed based on the k most similar train patients, and the choice of parameter k can be achieved empirically, or performing several tests, returning the k value associated to the best prediction accuracy. The different possibilities of computing the similarities between patients (used to build the score matrix) were tested on the MS dataset, and the main results are displayed in the following subsections.

4.2.1 Score Matrix computed from Biclusters Similarities

The parameters for this algorithm are the number of nearest neighbors (k) to consider and the penalty (weight) value, introduced to account for the different class ratios in the dataset. In order to evaluate the parameter sensitivity, several tests were performed using different sets of parameters. These results can be used to construct the approximate ROC curve (Section 3.5.2), measuring the performance of the classifier along the possible adjustable parameters. In Figure 7 are represented the confusion matrices obtained with Leave-One-Out (LOO) cross validation (CV), for the 75% best biclusters (in terms of p-value) and for all the computed set of biclusters.

The confusion matrix represented in Figure 7.a is the result of the classification performed with k = 1 for the best 75% of biclusters and a penalty/weight of 1.02, meaning that the actual score value between the patients is multiplied by 1.02 if the patient is a good responder. Using all the computed biclusters, the best results are as represented in Figure 7.b, which we use as a baseline for the biclustering-based classifiers.

		Classified as	
		0	1
Real Class	0	7 (35.00%)	13 (65.00%)
	1	6 (18.75%)	26 (81.25%)

a)

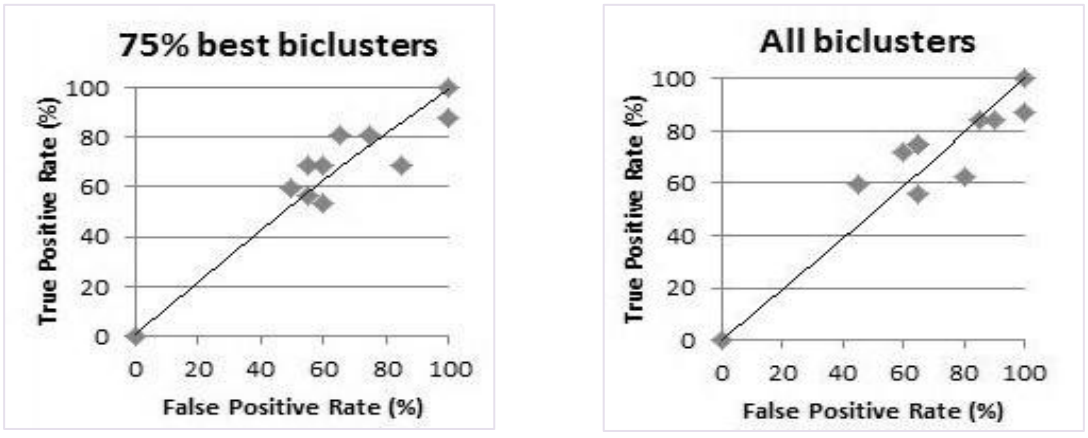
		Classified as	
		0	1
Real Class	0	7 (35.00%)	13 (65.00%)
	1	8 (25.00%)	24 (75.00%)

b)

	a) 75% best biclusters	b) all biclusters
Prediction Accuracy (LOO) (%)	63.46	59.62
Prediction Accuracy (5 x 4-fold) (%)	63.92	63.08

Figure 7 - Confusion matrices obtained for the biclustering-based kNN method based on biclusters similarities, for: a) 75% best biclusters (in terms of p-value) and b) all computed biclusters, together with the respective prediction accuracies (k = 1, no penalty). The patients' class 0 and 1 correspond, respectively, to bad and good responders.

Although several sets of parameters had the same overall prediction accuracy, we chose to present the results for the same set of parameters (k = 1 and penalty = 1.02). Now, both the approximate ROC curves built for the two experiments (75% and all of the biclusters) are represented below (Figure 8). The curve points ((FP,TP) pairs) are obtained with $k \in \{1,3\}$ and $penalty \in \{0.98,1.00\ (no\ penalty), 1.02\}$.



a) b)

Figure 8 - Representation of the approximate ROC curve associated to the kNN classifier with score based on biclusters similarities, a) with the 75% best biclusters (in terms of p-value), and b) wih all biclusters. In the x-axis is represented the False Positive rate, and the y-axis corresponds to the True Positive rate.

Analyzing both the ROC curves, we can conclude that there are no significant differences in the performance of the classifier when using the top 75% of the biclusters, or all of them, even if the

overall prediction accuracy is slightly higher for the first situation, or the higher TP rates are achieved for lower FP rates (for the pairs (FP,TP) tested).

With the objective of improving the prediction ability, a filter was applied to eliminate the least discriminant genes. As described in Algorithm 3 (see Section 3.4.1.1.1), the main parameters are the penalty on the score between patients, the number of neighbors k together with the similarity and class thresholds (that may differ between classes to account for unbalanced data). The obtained prediction accuracy was 59.62% with a LOO CV evaluation, and 48.08% with a 5 x 4-fold CV scheme. The respective confusion matrix and approximate ROC curve can be found in Figure 17 in Appendix C. We can conclude that the application of such a filter does not improve the results.

4.2.2 Score Matrix computed from Profile Similarities

Another strategy used to compute the score matrix between patients is based on the similarities between the expression profiles of the patients' biclusters. As previously described in Section 3.4.1.1.2, the score between patients can be obtained either directly from the number of shared profiles between the two patients in question, or with the application of a polynomial kernel (symmetric of Equation 7), that penalizes the patients with more profiles (without any penalty, these patients would, statistically, have more shared profiles).

4.2.2.1 Absolut number of shared profiles

If only the absolut number of shared profiles between patients is considered, the best prediction accuracy obtained with LOO cross validation was 50.00%, and 52.69% with a 5 x 4-fold CV scheme, resulting in a non-important classification (see Figure 18 in Appendix C for confusion matrix and approximate ROC curve). This low prediction accuracy values are most probably due to the presence of many random, non-significant, expression profiles among the set of computed biclusters for all patients. Also, there are certainly profiles that are common to a majority of individuals, even when they belong to different classes. Although these profiles may be significant in terms of representing a given cellular process, they might be associated to functions that are not class discriminative, that is, they can represent base functions of normal cell cycle, or of normal disease mechanisms, without differences at the IFN-β response level.

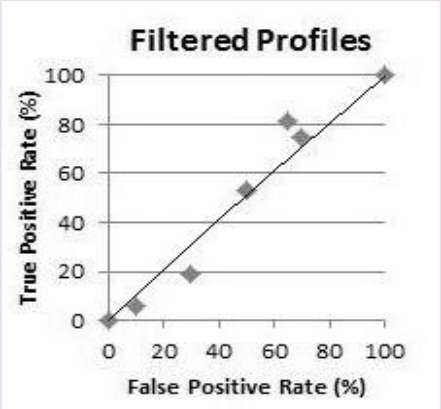
For these reasons, it was important to reduce the influence of the aforementioned non discriminative profiles/biclusters. The main results of the application of such a filter (Algorithm 5 described in Section 3.4.1.1.2, filtering only the training set), with k = 3 and penalty = 1.02, are summarized in Figure 9.

The confusion matrix in Figure 9.a shows that, comparing to the baseline in Figure 7.b (Section 4.2.1), 2 more good responders were correctly predicted. The approximate ROC curve can be analyzed in Figure 9.b.

The prediction accuracy of 63.46% obtained with a LOO CV suggests a positive influence of this filter, whereas the value of 46.54% obtained for the prediction accuracy with a 5 x 4-fold CV challenges this premise. Note, however, since the number of patients is very reduced (only fifty two individuals), a 4-fold CV scheme means that, in a given fold, thirteen patients are drawn to the test set, not affecting the training step. Given that the number of shared profiles is also very low, important information might be lost in the different test-train splits, degrading the prediction accuracy.

		Classified as	
		0	1
Real Class	0	7 (35.00%)	13 (65.00%)
	1	6 (18.75%)	26 (81.25%)

a)



b)

	Filtered profiles similarities
Prediction Accuracy (LOO) (%)	63.46
Prediction Accuracy (5 x 4-fold) (%)	46.54

Figure 9 - a) Confusion matrix obtained for the biclustering-based kNN method based on filtered profiles similarities (k = 3, penalty = 1.02), together with the respective prediction accuracies and b) the approximate ROC curve. The patients' class 0 and 1 correspond, respectively, to bad and good responders.

4.2.2.2 Polynomial Kernel

With the goal of reducing the effect of patients with higher number of profiles, we now present the results obtained with the application of a polynomial kernel (symmetric version of Equation 7). Here, the maximum score between patients is 0, since the distance function where the kernel is normally applied was turned into a similarity measure by changing its signal. The prediction accuracy values obtained were of 46.15% with LOO CV and 46.54% with 5 x 4-fold CV. The obtained confusion matrix and approximate ROC curve are represented in Figure 19 in Appendix C.

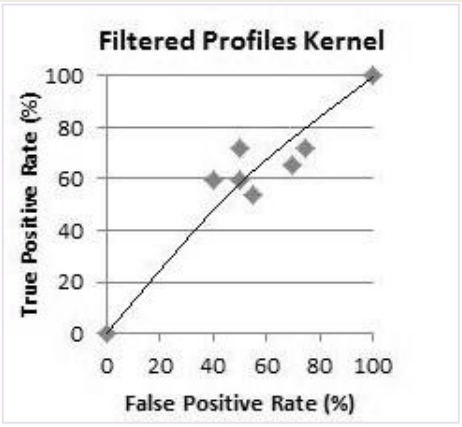
The reasons for these lower values may include the justification used before, based on the non-discriminative or non-significant profiles, taking aside the data characteristics that shall be addressed in a later section. In this context, the same profile filter used before (Section 4.2.2.1) is applied also in this situation, and the main results are presented in Figure 10, for k = 3 and no penalties on the scores between patients.

Making the comparison between the confusion matrix in Figure 10.a and the baseline in Figure 7.b (Section 4.2.1) we can observe that 3 more bad responders were correctly predicted, against only 1 less good responder.

Observing the approximate ROC curve plotted in Figure 10.b (test parameters were $k \in \{1,3\}$, $penalty \in \{0.98, 1.00, 1.02\}$), we can see that, although it is far from a well performing classifier, the behavior reflects a better performance when compared to the previous situation when no filter was applied (see Figure 19.b in Appendix C). In fact, the same TP rate is achieved in the filtered case, for lower FP rates, suggesting a better quality classifier. It is possible to assess if the filter introduces a significant advantage to the classification problem, performing a paired t-test, as explained in the Section 3.5.4, where the missing rates (100 – Prediction Accuracy) for the two confronted models are compared. The statistical significance of their differences is determined using the T-student distribution. The p-value obtained using such a paired t-test is 0.016, which supports the idea that the filter introduces a statistical significant advantage in the classification, with a confidence higher than 98%.

		Classified as	
		0	1
Real Class	0	10 (50.00%)	10 (50.00%)
	1	9 (28.12%)	23 (71.88%)

a)



b)

	Filtered profiles similarities (kernel)
Prediction Accuracy (LOO) (%)	63.46
Prediction Accuracy (5 x 4-fold) (%)	57.31

Figure 10 - a) Confusion matrix obtained for the biclustering-based kNN method based on filtered profiles similarities computed with a quadratic kernel (k = 3, no penalty), together with the respective prediction accuracies and b) the approximate ROC curve. The patients' class 0 and 1 correspond, respectively, to bad and good responders.

Instead of carrying out the filtering step with only the training set, the two possible classes of the test patient can be used to include more information in the filter, as explained in Section 3.4.1.1.2. However, since this strategy requires the computation of two score matrices (one for each possible class to which a test patient can belong), an additional selection step is included, in order to choose the score matrix that leads to better results. Unfortunately, for these data, the tested criteria to select one of the two resulting score matrices (or vectors, in the LOO CV situation) did not return any interesting result: for the highest sum criteria, the classifier always chose the score matrix/vector that lead to the classification as a good responder (the overrepresented class). On the other hand, if the lowest variance (more class specific) criterion was chosen, the classifier always predicted the patients

as being bad responders. This suggests that the class information strongly affected the filtered profiles, as both possibilities for the test patient's responder class lead to such different and class specific results.

4.2.3 Score Matrix computed from Discretized Matrices

As stated in Section 3.4.1.1.3, this similarity computation is based on the temporal expression patterns, resulting from the discretization process applied on the data. In this subsection we show the results obtained for three different strategies with increasing complexity, using the discretization step based on the genes' expression variation between consecutive time points (see Section 3.2.3).

4.2.3.1 Element of Bicluster

As explained before in Section 3.4.1.1.3.1, this is the simplest method of comparing the discretized matrices to compute the score between patients. It consists only in creating a binary matrix with the same dimensions as the discretized matrices, where the elements which belong to a bicluster are 1, and the others are 0. The score is simply the inner product between the two binary matrices. As antecipated, this method did not return any satisfactory results. In fact, if all the computed biclusters were used, all elements (or nearly all) belong at least to one bicluster. This means that the score will be maximum (some might be slightly lower) between all patients, and as such, the classifier resorts always to the same k nearest neighbors. In this dataset, since the first patients compared are good responders, the classifier predicts the class 1 for all test patients, and thus the prediction accuracy corresponds exactly to the good responders class ratio – 61.54%. Even the application of the profile/bicluster filter does not change the classification outcome, mainly because the bad responders have less biclusters after the filter (the good responders retain more biclusters, since these are represented in more patients of the same class, contributing to the classification, and the same biclusters, if found in bad responders, are eliminated).

4.2.3.2 Symbol Pairing

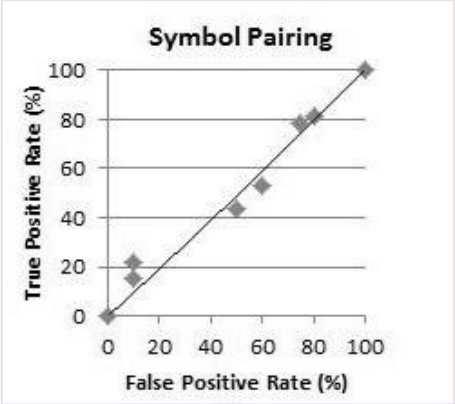
This method introduces the information of the bicluster expression profile, comparing the discretized symbols so that the score between patients can be computed. Again, the obtained results with this strategy are not consistent with a well performing classifier, although now the symbols themselves are compared. Nevertheless, this is not sufficiently discriminative, and the approximate ROC curve obtained with different sets of number of neighbors and penalties ($k \in \{1,3\}$, $penalty \in \{0.98, 1.00, 1.02\}$) can be analyzed in Figure 11, as well as the confusion matrix associated to the highest prediction accuracy, obtained with k = 3 and penalty = 1.02.

The comparison between this confusion matrix (Figure 11.a) and the baseline (Figure 7.b in Section 4.2.1) allows us to see that 5 more patients were classified as good responders, but only 2 of them were correctly predicted. The approximate ROC curve (Figure 11.b) confirms the earlier

discussion. In fact, the curve almost aligns perfectly with the FP = TP line, leading to the conclusion that this classifier is no better than a random one.

		Classified as	
		0	1
Real Class	0	4 (20.00%)	16 (80.00%)
	1	6 (18.75%)	26 (81.25%)

a)



b)

	Symbol Pairing
Prediction Accuracy (LOO) (%)	57.69
Prediction Accuracy (5 x 4-fold) (%)	60.77

Figure 11 - a) Confusion matrix obtained for the biclustering-based kNN method based on discretized symbols pairing (k = 3, penalty = 1.02), together with the respective prediction accuracies and b) the approximate ROC curve. The patients' class 0 and 1 correspond, respectively, to bad and good responders.

Finally, for similar reasons than the ones mentioned for the previous situation, the application of the profile/bicluster filter takes the classifier to predict all instances as good responders, since these are the ones that maintain a larger number of biclusters, and even penalizing the higher number of biclusters for the class 1, the discrimination remains unaltered and biased to 1's.

4.2.3.3 Symbol Pairing with Time-Lags

As explained in Section 3.4.1.1.3.3, one of the major challenges of the clinical expression time series was not considered in the previous approaches: the patient-specific response rate. To overcome this fact, this method includes the possibility of time-lags in the comparison between the discretized matrices. Although this accounts for different starting points between patients, it does not contemplate the possibility of a patient's gene expression remaining in a given state for more time points than other patients, a consideration made possible with HMMs in [39].

The confusion matrices corresponding to the best results of prediction accuracy for each of the maximum time-lags considered are displayed in Figure 12. The used parameters were k = 1 (for maximum time-lag = 1), k = 3 (for maximum time-lag = 2) and penalty = 1.02.

		Classified as	
		0	1
Real Class	0	5 (25.00%)	15 (75.00%)
	1	1 (3.12%)	31 (96.88%)

a)

		Classified as	
		0	1
Real Class	0	2 (10.00%)	18 (90.00%)
	1	2 (6.25%)	30 (93.75%)

b)

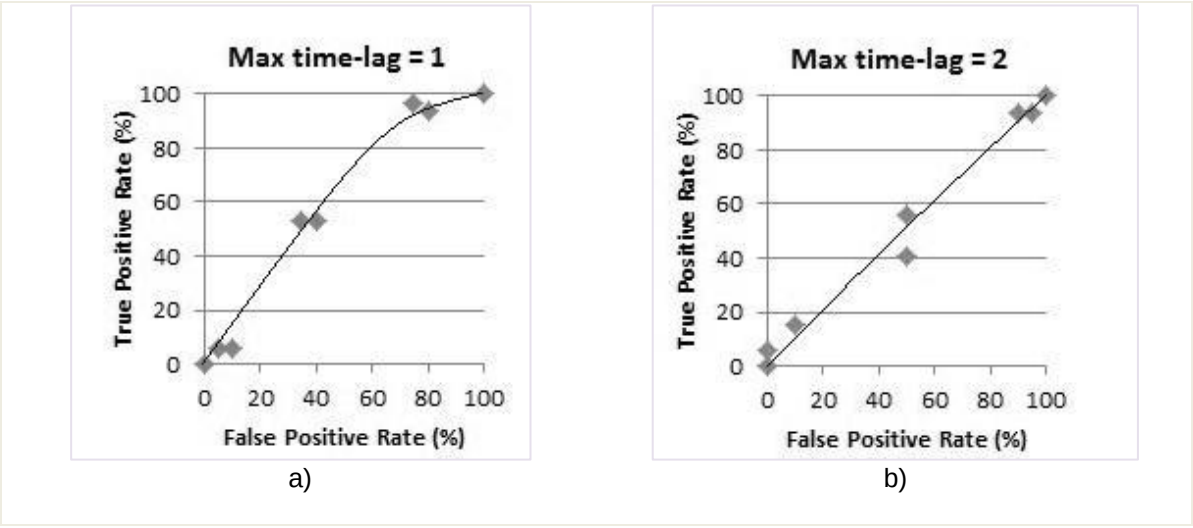
	Maximum time-lag = 1	Maximum time-lag = 2
Prediction Accuracy (LOO) (%)	69.23	61.54
Prediction Accuracy (5 x 4-fold) (%)	68.08	60.38

Figure 12 - Confusion matrices obtained for the biclustering-based kNN method based on discretized symbols pairing with time-lag: a) maximum time-lag = 1 (k = 1, penalty = 1.02), and b) maximum time-lag = 2 (k = 3, penalty = 1.02). The patients' class 0 and 1 correspond, respectively, to bad and good responders.

Comparatively to the baseline results (Figure 7.b in Section 4.2.1), in Figure 12.a we have more 9 patients classified as good responders (5 of them correctly predicted), and in Figure 12.b, we see that nearly all the patients are classified in the good responders class (only 4 predicted bad responders: 2 correctly classified).

From the observation of the approximate ROC curves in Figure 13, the main conclusion is that, although it is still not a classifier capable of dealing successfully with these data, the case where the maximum time-lag is 1 time point reveals an advantage when compared with the previous classifier (Figure 11 in Section 4.2.3.2), which did not consider any aspect of the patient-specific response rate. When this parameter is used (maximum time-lag = 1), the approximate ROC curve has an initial higher slope, indicating a more significant classifier for these data. On the contrary, for a value of 2 time points for the maximum time-lag, the ROC curve is very close to the expected from a random classifier. In fact, a maximum time-lag of 1 time point seems to correspond to a more realistic situation, since each time point is separated of its neighbor by three months.

These results indicate that the incorporation of the possibility of time delays when comparing the profiles' symbols is, in fact, of great importance to clinical time series classification. The statistical significance of the differences between the results obtained with and without the consideration of time delays were analyzed using a paired t-test. The obtained p-value was approximately 0.0025, indicating positive evidences that these differences are in fact statistical significant, with a confidence of approximately 99.75%.



a) b)

Figure 13 - Approximate ROC curves obtained for the biclustering-based kNN method based on discretized symbols pairing with time-lags: a) maximum time-lag = 1 and b) maximum time-lag = 2. The patients' class 0 and 1 correspond, respectively, to bad and good responders.

4.3 Meta – Profiles Classification

The first results obtained with this method were obtained with the sum criterion (Section 3.4.2), and, as we already suspected, considering the nature of the data, the classifier predicted all the test patients as good responders. This may be explained with the fact that all evidences point to a particular characteristic of this MS dataset (and eventually, other clinical time series expression data): the good responders share between themselves a large number of similar biclusters, and consequently, expression profiles. On the other hand, the bad responders seem to share a great proportion of similar biclusters with the good responders, but not with each other. This fact renders the problem much more difficult and helps explaining much of the challenges faced in the discussed methods.

Amongst all the tests performed with this classification approach, however, we achieved the best classification results with criteria almost contrary to the ones presented in Section 3.4.2. When introducing a penalty to reduce the sum of the proportions for the class 1 (good responders), we observed for some values of the penalty, that if the contrary criterion is used, the classifier presents a good prediction accuracy:

New criterion based on sum of proportions:

If $sumProportion1 \times penalty < sumProportion0$

$predictedClass = 1;$

else $predictedClass = 0;$

With extensive tests, we also concluded that a criterion based on the average proportion could be used to classify the patients as well. Nevertheless, it was also contrary to the anticipated idea.

Here, the classification is based on comparing the average proportion (for class 0, for example) with a determined threshold.

New criterion based on average proportion:

If $averageProportion0 > threshold$

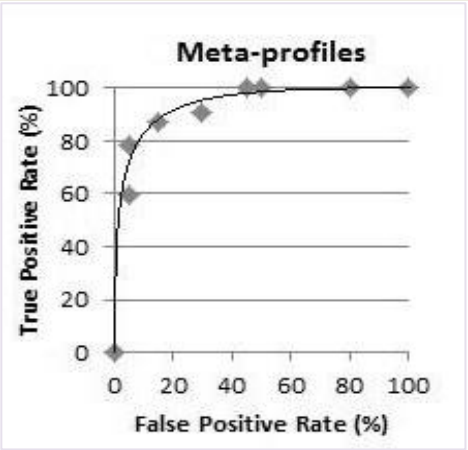
$predictedClass = 1;$

else $predictedClass = 0;$

An example of an obtained confusion matrix for the sum criterion is presented in Figure 14.a, with a penalty = 0.62. The chosen penalty is the one that returns the higher prediction accuracy, in a repetition of tests, similarly to the choice of the k parameter in kNN classification. This confusion matrix is equal to the one obtained from the average criterion for a threshold of 0.38, and comparing them to the baseline (Figure 7.b), we see that 13 more patients are accurately predicted (11 bad responders and 2 good responders). The approximate ROC curve built from testing with $penalty \in [0.5, 0.64]$, is displayed in Figure 14.b, where we can observe a behavior much more consistent with a well performing classifier, achieving higher TP rates for low values of FP rates. In fact, when testing this classifier with the more significant biclusters (90%, for example), we achieve a prediction accuracy value over 94% with LOO CV. However, for comparison purposes, we present all the results obtained with the analysis for all the computed biclusters.

		Classified as	
		0	1
Real Class	0	17 (85.00%)	3 (15.00%)
	1	4 (12.50%)	28 (87.50%)

a)



b)

	Meta-profiles
Prediction Accuracy (LOO) (%)	86.54
Prediction Accuracy (5 x 4-fold) (%)	68.08

Figure 14 - a) Confusion matrix obtained for the biclustering-based classification method based on meta-profiles (sum criterion with a penalty = 0.62), together with the respective prediction accuracies, and b) the approximate ROC curve. The patients' class 0 and 1 correspond, respectively, to bad and good responders.

A possible justification for these unanticipated criteria rises when we approach the problem from a different point of view. If we think of the class proportions in a matter of class balance, the main conclusion drawn from the obtained results is that the good responders have a more balanced distribution of their profiles between classes, an evidence consistent with the aforementioned idea that the good responders share a great number of biclusters/profiles with all other patients, while the bad

responders share profiles with the good responders, but a lower number with the rest of the patients of the same class.

Using a 5 x 4 fold CV scheme for the performance evaluation, the results drop significantly (prediction accuracy under 70%), mainly due to an issue already discussed in Section 4.2.2.2: in this case, when compared to LOO CV, the training set is reduced (the initial number of instances is already small), leading to the loss of important information, thus reducing the prediction ability.

One of the consequences of the reduced number of time points for a much larger number of genes is the data overfitting. This is particularly important in this case, and might contribute also to the reduced prediction accuracies obtained with the 5 x 4 fold CV.

Another problem related to a possible data overfitting is the specificity of the classifier. This property can be analyzed by performing a permutation test, where the training instances' labels are shuffled before the classification. The average prediction accuracy obtained by shuffling the labels 1000 times was 77.93% (LOO CV), which is in fact a lower value than the ones observed for the true labeled set (86.54%). Nonetheless, we can conclude that this classifiers' specificity is not very high, since the difference does not represent a significant drop, consistent with a high specificity classifier.

4.4 Meta – Biclusters Classification

As described in Section 3.4.3, in this method the biclusters were grouped in terms of their similarities using a hierarchical clustering technique, thus creating a determined number of meta-biclusters representatives of different sets of similar biclusters. The binary matrix resultant from the comparison of each patient's bicluster and the meta-biclusters space is then classified with standard classifiers using the software package Weka, such as Decision Trees, k – Nearest Neighbors (kNN), Support Vector Machines (SVM), Logistic Regression, Radial Basis Function (RBF) Network and Multilayer Perceptron (MLP). Table 4 summarizes the prediction accuracy values btained for 500, 750 and 1000 meta-biclusters. The respective confusion matrices can be found in the Appendix C (Figure 20 to Figure 25).

Observing the prediction accuracy values in Table 4, we observe that (exception made to the MLP classifier, which classifies all patients as good responders) using a higher number meta-biclusters, we obtain a higher prediction accuracy, although with numbers of meta-biclusters higher than 1000 the results became again biased to 1's (good responders). This fact is statistically significant, as the p-value obtained with a paired t-test (Section 3.5.4) for the differences between the prediction accuracies for 1000, 750 and 500 meta-biclusters was always lower than 0.05.

Table 4 – Summary of the prediction accuracy values (LOO CV/ 5 x 4-fold CV) obtained with the meta-biclusters classification method for 1000, 750 and 500 meta-biclusters.

	Prediction Accuracy (%): LOO CV / 5 x 4-fold CV					
Number of meta-biclusters	Decision Tree	kNN	SVM	Logistic Regression	RBF Network	MLP
1000	63.46 / 61.92	67.31 / 65.39	63.46 / 64.61	63.46 / 64.61	63.46 / 64.61	61.54 / 61.54
750	55.77 / 60.77	59.62 / 60.39	55.77 / 58.85	57.69 / 58.85	55.77 / 58.85	61.54 / 61.54
500	61.54 / 61.54	50.00 / 55.00	44.23 / 51.92	44.23 / 51.92	44.23 / 51.92	61.54 / 61.54

Furthermore, although there is an equilibrium between most of the used classifiers, the best prediction accuracy and confusion matrix, are obtained with the k – Nearest Neighbors classifier, an outcome justifiable with the simplicity of the data (binary matrix assigning 1 to the meta-bicluster containing at least one of the patient's biclusters). Besides the results, this approach should be further explored, since the use of meta-biclusters in classification was not exhaustively studied, although it is promising in theory.

4.5 State of the Art Classifiers

Standard classifiers, using the software package Weka, were also used on the MS dataset, in order to get a better perception of how well the newly developed biclustering-based classification methods perform when compared to several state of the art classifiers. In order to evaluate the importance (or influence) of the discretization process, we also apply the classifiers to a discretized version of the data.

The confusion matrices obtained for each of the standard classifiers used can be found in Appendix C, in Figure 26 for the application on the original, numeric dataset, and **Error! Reference source not found.** for the application on a discretized version of the dataset. In Section 5 (Discussion) we also present a summary table of the prediction accuracy values obtained for the different classifiers, when applied either on the numeric or discretized data.

5 Discussion

In this section, we discuss the main results obtained for the different proposed methods of classification, together with the pros and cons of each strategy.

5.1 Biclustering-based classifiers

The best values of prediction accuracy obtained for the previously described classification methods are summarized in Table 5:

Table 5 – Summary of the main results for the developed biclustering-based classification methods.

	kNN – Bicluster Similarities	**kNN – Profile Similarities**		**kNN –Profile Similarities (Kernel)**		**kNN – Element of Bicluster**
		Not Filtered	Filtered	Not Filtered	Filtered	
Prediction Accuracy (LOO) (%)	59.62	50.00	63.46	46.15	63.46	61.54[2]
Prediction Accuracy (5 x 4 fold) (%)	61.92	52.69	46.54	46.54	57.31	61.54
	kNN – Symbol Pairing	**kNN – Symbol Pairing with Time – Lags (= 1)**		**Meta - Profiles**		**Meta – Biclusters (1000; KNN)**
Prediction Accuracy (LOO) (%)	57.69	69.23		86.54		67.31
Prediction Accuracy (5 x 4 fold) (%)	60.77	68.08		68.08		65.39

Several conclusions can be drawn from these results and the respective confusion matrices (Section 4), in terms of analyzing the different approaches for this classification problem. It is important to emphasize the particular characteristics of this dataset, where beside a class unbalance, we find significant differences between what is shared between the patients of the two responder classes: good responders share a large number of biclusters/profiles between patients of the two classes. However, bad responders do not share sufficient information (biclusters/profiles) between patients of the same class, decreasing the prediction accuracy for most of the developed strategies. However, with the exception of the unanticipated criteria used in the meta-profiles classification, all the other approaches are data independent.

[2] This prediction value of 61.54% represents in fact a classification as being good responders

Nevertheless, some of the proposed classification methods reveal some potentialities, which can be further explored. These include the computation of the score matrix between patients based on the comparison of discretized matrices considering time-lags, to use in the biclustering-based kNN classifier. The patient-specific response rate is partially taken into account, given that possible delays are considered. A method that incorporates the possibility of a patient remaining in a given response state for less or more time points than others, would provide better results, since the simpler consideration of time delays already improves significantly the prediction accuracy (see Section 4.2.3.3).

Furthermore, when the score matrix between patients is built based on profile similarities using a quadratic kernel, the filtering step significantly improves the prediction accuracy (Section 4.2.2.2), even if this value is not entirely acceptable.

Finally, the most problem-specific strategy, which yielded the best results, was the meta-profiles classification. In the end, this was, by far, the best predicting method, even if the used criteria were opposed to the initially expected (Sections 3.4.2 and 4.3). As discussed before, this situation might be a result of the data particular characteristics, where the patients of each class share different amounts of biclusters/profiles between the patients of the same class. This strategy should be further explored, especially since its specificity is not very high.

5.2 State of the Art classifiers applied on the original dataset

A summary of the results obtained from these standard classifiers can be found in Table 6.

Table 6 – Summary of the main results (prediction accuracy) obtained from standard classifiers using the software package Weka, with the numeric dataset.

	Decision Tree	**kNN**	**SVM**	**Logistic Regression**	**RBF Network**	**MLP**
Prediction Accuracy (LOO) (%)	71.15	86.54	92.31	80.77	88.46	86.54
Prediction Accuracy (5 x 4 fold) (%)	70.77	82.31	85.00	80.38	83.85	86.15

In general, the standard classifiers tested on this dataset outperformed most of the previous described classification methods based on biclustering. Nonetheless, if we compare to these results the ones obtained for the meta-profiles method (Section 4.3), its prediction accuracy obtained with LOO CV is in the same range, although it presents a lower specificity. It is possible to observe that the classifier based on a decision tree has the lowest prediction ability, and it is not significantly higher (p-value $\cong$ 0.17) than the kNN with a score matrix based on symbol pairing with time-lags (Section

4.2.3.3), which presents a prediction accuracy of 69.23% (LOO) and 68.08% (5 x 4 fold) for a maximum time-lag of 1 time point in both directions.

5.3 State of the Art classifiers applied on a discretized version of the dataset

The prediction accuracies of the standard classifiers presented in this section can be arranged in the summary Table 7:

Table 7 - Summary of the main results obtained from standard classifiers using the software package Weka, with a discretized version of the original dataset.

	Decision Tree	**kNN**	**SVM**	**Logistic Regression**	**RBF Network**	**MLP**
Prediction Accuracy (LOO) (%)	51.92	55.77	59.62	40.38	57.69	46.15
Prediction Accuracy (5 x 4 fold) (%)	54.61	49.62	53.08	45.77	56.15	57.95

Recalling the prediction ability for the situation when the dataset was the original, numeric one (Table 6), one can see that the fact that we use a discretized version of the data lowers the classifier performance significantly (p-value < 0.05 for all the used classifiers), not even reaching 60% of prediction accuracy. These evidences suggest this kind of classifiers cannot deal well with discretized data of this type (especially with the particular characteristics discussed previously (Section 5.1). In fact, for example, the biclustering-based kNN classifier based on the similarities between discretized matrices considering time-lags (Section 4.2.3.3) outperforms significantly all these standard classifiers (p-value < 0.05) when acting upon discretized data.

6 Conclusion and Future Work

When confronted with a static gene expression analysis, time series expression data analysis brings some new important perspectives, which prove to be essential to overcome some of the challenges that a static point of view simply could not. Analyzing a temporal expression evolution allows for a better insight on gene regulation networks over time, in a normal or conditioned situation, such as a treatment response.

Classification problems based on time series expression data have been discussed in the last decade, where several machine learning techniques were developed and applied, taking into account the particular features of a time series analysis [15, 24, 39]. The main goal of a classification problem of this type is its application in real life situations, especially in the fields of biology and medicine. A known example is the classification of clinical time series expression data from patients with relapsing-remitting multiple sclerosis (RR-MS), under a typical IFN-β treatment. Since the response is not the desired for many patients (up to 50%) and there are negative side effects to consider, the ability to predict a response outcome would definitely change the paradigm of MS treatment, avoiding an expensive and potentially harmful treatment when it does not improve the patient's condition.

This was the starting point to this thesis, which, based on prior work on this same MS time series gene expression dataset [15, 24, 39], presents a brand new classification approach, based on a recent biclustering technique (CCC-Biclustering), to our knowledge not used before for (clinical) time series classification purposes. Here, we developed several biclustering-based classifiers, such as a kNN classifier based on different similarity measures (Section 3.4.1), a meta-profiles classifier (Section 3.42) and a meta-biclusters classification method. Then, these classifiers were tested on a clinical dataset, resulting from the expression profiling (microarray analysis with reverse-transcription PCR) of seventy preselected genes for fifty two RR-MS patients under IFN-β treatment for two years (seven time points were measured) [15].

Most of the results pointed to a singular characteristic of this dataset: good responders have a significant number of similar biclusters in common with other good responders, but also with the bad responders. These shared similar biclusters might include normal cell cycle and disease expression signatures, common to all RR-MS patients, and this shall be further investigated in future work. On the contrary, bad responders show evidences of having few similar biclusters in common, beside the ones also shared with the good responders group. This fact suggests that there are different expression signatures associated to a poor response to IFN-β treatment, a probable result of differences in the fragile balance of several pathways associated to the disease and/or treatment response.

Even disregarding this problem-specific issue, there are still other important challenges that might explain some of the lower classification results obtained, starting with unbalanced data, with respect to the classes distribution: thirty two patients are labeled as good responders and twenty as

bad responders. This difference, associated to the aforementioned characteristic, has also to be taken into account (by introducing weights/penalties, for example) in order to minimize its negative influence on the classification outcome.

Furthermore, the reduced number of time points of the data, when compared to its number of genes often results in an overfitting phenomenon. To minimize this effect, a method was proposed in [46]. It consists in reducing the expression profiles space, eliminating the ones originated randomly as an effect of the overfitted data, maintaining only the real, significant, expression profiles. Although this method was not applied in this work, we include it in our plan for future work, since we believe that this will improve the classification results.

Another possible solution to decrease the negative effects of the data overfitting, and reduce the complexity of the problem simultaneously, is a feature selection step, applied before the biclustering process itself. In this case, instead of filtering the computed biclusters based on similarities or significance (which slightly improves the results for most of the presented situations), we would filter the genes, keeping the most discriminative ones (regarding the class). Fortunately, this type of feature selection is widely discussed in the literature relative to classification problems [79], and is indeed used by all classification methods applied to this dataset and described in the related work (Section 2.4), as [15, 24, 39], and thus we intend to explore this approach in further studies.

The reduced number of instances of the dataset (only fifty two patients) is also a source of uncertainties associated to the classification results, especially in a k – fold cross validation scheme, since, for a small k, a great deal of important information is not included in the training process, thus leading to poorer results.

The main results obtained with the developed biclustering-based classifiers, are not as good as expected (though they might have been negatively influenced by some of the above discussed data issues). Except the use of the meta-profiles classifier, the best quality classifier is the kNN with the score matrix computed from the symbolic comparison, including the possibility of time-lags (kNN was already referred by other authors as superior in time series classification [80]). In fact, when compared to the symbolic comparison without the time delays consideration, the first method returns a significantly higher prediction accuracy, up to 69.23% against 57.69% (p-value = 0.0025). Additionally, it shows no significant differences from a standard classifier, the decision tree, tested with the original numeric dataset (p-value $\cong$ 0.17). This suggests that the consideration of the patient-specific response rate, even partially, improves significantly the classifier's prediction ability. It is not entirely taken into account, as opposed to [39], because the algorithm only considers the possibility of temporal differences in the expression profile as a whole, that is, possible time delays in the response profile. This approach lacks the consideration of differences in the response state duration, since a patient can remain (in terms of gene expression signatures) in a given state for more or less time points than other patients. This consideration, however, has proven to be a major challenge when dealing with the biclustering results, and we consider that overcoming it will improve the classifiers' prediction ability.

The meta-biclusters classification method was based on performing hierarchical clustering of the computed biclusters, using the resulting meta-biclusters (representative of a set of similar biclusters) as features on which a small collection of standard classifiers was applied. As before, the obtained prediction accuracy values were not agreeable with a well performing classifier. Actually, the best results were returned by the kNN algorithm, still remaining below the 70% prediction accuracy. This approach should, however, be further explored on other clinical gene expression time series or other classification problems in general.

In order to better evaluate the performance of the developed classification strategies, the same collection of standard classifiers used in the meta-biclusters method was used to classify the clinical dataset: the original, numeric one, and a discretized version to study the influence of the discretization step in the classification. Unfortunately, it was not possible to reproduce the results of the previous work on the MS dataset [24], mostly due to serious difficulties in obtaining and running their classification algorithms. Even after contacting the authors, the test/train sets used in their work were also not available, and thus reproducing their results was impossible. This lead to the choice of comparing our results only with some state of the art classifiers, even though it is clear that, with the exception of the meta-profiles method, the obtained results are lower than the ones obtained in [15, 24, 39]. We recall, however, that all these approaches used feature selection by first selecting a small set of genes, and this preprocessing step was not applied in the biclustering-based classifiers proposed in this thesis, as explained above.

The results obtained with the standard classifiers for the original dataset (numeric values) are considerably higher than the ones obtained with the biclustering-based methods, excluding the meta-profiles classifier and the standard decision tree classifier (non-significant differences from the kNN classifier based on the symbolic comparison with time-lags, p-value $\cong 0.17$). However, even the best prediction accuracy values did not surpass the results of the prior works on the same dataset (even though the cross validation test-train splits are surely different).

For the case where the standard classifiers were applied on a discretized version of the dataset, we witnessed a substantial drop on the respective prediction accuracies. For this situation, the prediction accuracy values were all under 60%, and, for example, the kNN classifier based on the symbolic comparison with time-lags outperformed significantly all of the tested standard classification algorithms (p-value < 0.05), not to mention the meta-profiles classifier (prediction accuracy of 86.54%) suggesting that the biclustering process can improve the classification of discretized versions of time series expression data.

Until this point, all classifiers' designs and criteria were problem-independent. Nevertheless, when testing the meta-profiles classification method for different criteria, we concluded that good prediction accuracy values were achieved. However, an unanticipated criterion was found for this specific dataset: the patients sharing more similar profiles with both classes (higher equilibrium in the shared class proportions), are classified as good responders, when the first choice relied on the bad responder class, since a higher equilibrium in the shared class proportions means that the test patient

shares a lower percentage of profiles with good responders. This result may be due to the discussed characteristics of this clinical dataset, where the bad patients have less information in common, and thus they have more shared expression profiles with only the good patients (normal cell cycle and disease gene expression signatures, for example). Furthermore, this method presents a low specificity. This conclusion was attained by performing a permutation test, shuffling the class labels 1000 times, which returned an average prediction accuracy of approximately 78%. This is not a decrease consistent with a highly specific classifier, even if it might be a result of the data overfitting already discussed. Nonetheless, further investigation on this classification method should be carried to better understand the underlying reasons behind these data particular characteristics.

Besides all these justifications for the lower prediction abilities, we note that, currently, the standard treatment for MS is the IFN-β therapy. If a classifier is able to reduce the number of bad responders (even if slightly) that receive such a treatment, and continues to predict correctly the good responders, then it presents a significant advantage in this classification problem. In this case, as the proportion of good responders is 61.54%, we can consider a prediction accuracy of approximately 70% as acceptable, since we gain a few more correctly predicted patients, comparing to the actual case when all patients are treated with IFN-β, regardless the responder type. However, we must separate two situations: the false positives (bad responders that are classified as good responders, thus receiving the treatment) and the false negatives (good responders classified as bad responders, missing the treatment they should receive). This question should be further explored although, in our point of view, given the associated side effects and the arising of alternative therapies, the classification should favor the bad responder classification. This means that we should minimize the false positive rate, thus avoiding useless and possibly harmful treatments, allowing for the patients to look for more reliable choices of treatment for their particular situation.

Besides the aforementioned feature selection (gene filtering) before biclustering, the significant profile “filter” [46] and including a full consideration for the patient-specific response rate, several other experiments and directions may be taken. Recently, the concept of triclustering associated to multiple time series analyses was proposed by Gonçalves et al. [81]. Essentially, this is an extension to the presented CCC-Biclustering algorithm [60], where the biclusters of genes and consecutive time points are now grouped through a third dimension: the training instances, in the case of multiple time series classification problems. This means that a tricluster is formed by a repeated bicluster across patients (in a clinical example), just as a bicluster is formed by a repeated profile (set of symbols on consecutive time points) across genes. In this context, each tricluster would have an associated class distribution, since this is a supervised learning task, and, with a committee of classifiers (weighted voting) the classification of new patients could be performed by comparing each of the test patient’s bicluster with the ones representing each of the training triclusters. This idea is on the basis of our main future work direction.

Another aspect that was not possible to explore further in this thesis, although we intend to study in future work, is the analysis of the genes/time points involved in the most class-discriminative biclusters. As displayed in Section 4.1, it is possible to visualize the biclusters’ genes expression

temporal evolution, thus possibly finding which genes are differentially expressed between the two classes, and for which time points. This would lead to a better comprehension on the mechanisms of the MS patients' response to IFN-β, providing valuable information on discriminating which patients should or should not be treated.

As a final conclusion, it can be stated that even if the obtained results do not support the idea of a time series classifier based on biclustering, some of the adopted strategies revealed important potentialities that shall be further analyzed and explored, with either more instances and/or time points, or even in different classification problems: other clinical case studies or different data mining problems, like collaborative filtering [82].

References

1. Miller, M.B. and Y.W. Tang, *Basic concepts of microarrays and potential applications in clinical microbiology.* Clin Microbiol Rev, 2009. **22**(4): p. 611-33.
2. Ehrenreich, A., *DNA microarray technology for the microbiologist: an overview.* Appl Microbiol Biotechnol, 2006. **73**(2): p. 255-73.
3. Hemmer, B., J.J. Archelos, and H.P. Hartung, *New concepts in the immunopathogenesis of multiple sclerosis.* Nat Rev Neurosci, 2002. **3**(4): p. 291-301.
4. Ramagopalan, S.V., et al., *Expression of the multiple sclerosis-associated MHC class II Allele HLA-DRB1*1501 is regulated by vitamin D.* PLoS Genet, 2009. **5**(2): p. e1000369.
5. Olerup, O. and J. Hillert, *HLA class II-associated genetic susceptibility in multiple sclerosis: a critical evaluation.* Tissue Antigens, 1991. **38**(1): p. 1-15.
6. Knopf, P.M., et al., *Antigen-dependent intrathecal antibody synthesis in the normal rat brain: tissue entry and local retention of antigen-specific B cells.* J Immunol, 1998. **161**(2): p. 692-701.
7. Miller, D.H., et al., *The role of magnetic resonance techniques in understanding and managing multiple sclerosis.* Brain, 1998. **121 (Pt 1)**: p. 3-24.
8. Fox, N.C., et al., *Progressive cerebral atrophy in MS: a serial study using registered, volumetric MRI.* Neurology, 2000. **54**(4): p. 807-12.
9. Compston, A., *Genetic epidemiology of multiple sclerosis.* J Neurol Neurosurg Psychiatry, 1997. **62**(6): p. 553-61.
10. Ebers, G.C. and D.A. Dyment, *Genetics of multiple sclerosis.* Semin Neurol, 1998. **18**(3): p. 295-9.
11. Chapman, J., et al., *APOE genotype is a major predictor of long-term progression of disability in MS.* Neurology, 2001. **56**(3): p. 312-6.
12. Bomprezzi, R., et al., *Gene expression profile in multiple sclerosis patients and healthy controls: identifying pathways relevant to disease.* Hum Mol Genet, 2003. **12**(17): p. 2191-9.
13. Ramagopalan, S.V. and G.C. Ebers, *Multiple sclerosis: major histocompatibility complexity and antigen presentation.* Genome Med, 2009b. **1**(11): p. 105.
14. Ramanathan, M., et al., *In vivo gene expression revealed by cDNA arrays: the pattern in relapsing-remitting multiple sclerosis patients compared with normal subjects.* J Neuroimmunol, 2001. **116**(2): p. 213-9.
15. Baranzini, S.E., et al., *Transcription-based prediction of response to IFNbeta using supervised computational methods.* PLoS Biol, 2005. **3**(1): p. e2.
16. Satoh, J., et al., *T cell gene expression profiling identifies distinct subgroups of Japanese multiple sclerosis patients.* J Neuroimmunol, 2006. **174**(1-2): p. 108-18.
17. Hafler, D.A., et al., *Risk alleles for multiple sclerosis identified by a genomewide study.* N Engl J Med, 2007. **357**(9): p. 851-62.
18. Arthur, A.T., et al., *Genes implicated in multiple sclerosis pathogenesis from consilience of genotyping and expression profiles in relapse and remission.* BMC Med Genet, 2008. **9**: p. 17.
19. Trauernicht, A.M., et al., *Modulation of estrogen receptor alpha protein level and survival function by DBC-1.* Mol Endocrinol, 2007. **21**(7): p. 1526-36.
20. Bielekova, B. and R. Martin, *Multiple Sclerosis: Immunotherapy.* Curr Treat Options Neurol, 1999. **1**(3): p. 201-220.
21. Sharief, M.K. and R. Hentges, *Association between tumor necrosis factor-alpha and disease progression in patients with multiple sclerosis.* N Engl J Med, 1991. **325**(7): p. 467-72.
22. Group, L.M.S.S. and T.U.o.B.C.M.M.A. Group, *TNF neutralization in MS: results of a randomized, placebo-controlled multicenter study.* Neurology, 1999. **53**: p. 457-65.
23. Sturzebecher, S., et al., *Expression profiling identifies responder and non-responder phenotypes to interferon-beta in multiple sclerosis.* Brain, 2003. **126**(Pt 6): p. 1419-29.
24. Costa, I.G., et al., *Constrained mixture estimation for analysis and robust classification of clinical time series.* Bioinformatics, 2009. **25**(12): p. i6-14.
25. Arnason, B.G., *Interferon beta in multiple sclerosis.* Clin Immunol Immunopathol, 1996. **81**(1): p. 1-11.
26. Wandinger, K.P., et al., *Complex immunomodulatory effects of interferon-beta in multiple sclerosis include the upregulation of T helper 1-associated marker genes.* Ann Neurol, 2001. **50**(3): p. 349-57.

27. Wang, X., et al., *IFN-beta-1b inhibits IL-12 production in peripheral blood mononuclear cells in an IL-10-dependent mechanism: relevance to IFN-beta-1b therapeutic effects in multiple sclerosis.* J Immunol, 2000. **165**(1): p. 548-57.
28. da Silva, A.J., et al., *Comparison of gene expression patterns induced by treatment of human umbilical vein endothelial cells with IFN-alpha 2b vs. IFN-beta 1a: understanding the functional relationship between distinct type I interferons that act through a common receptor.* J Interferon Cytokine Res, 2002. **22**(2): p. 173-88.
29. Rudick, R.A., et al., *Defining interferon beta response status in multiple sclerosis patients.* Ann Neurol, 2004. **56**(4): p. 548-55.
30. Rio, J., et al., *Defining the response to interferon-beta in relapsing-remitting multiple sclerosis patients.* Ann Neurol, 2006. **59**(2): p. 344-52.
31. Androulakis, I.P., E. Yang, and R.R. Almon, *Analysis of time-series gene expression data: methods, challenges, and opportunities.* Annu Rev Biomed Eng, 2007. **9**: p. 205-28.
32. Bar-Joseph, Z., *Analyzing time series gene expression data.* Bioinformatics, 2004. **20**(16): p. 2493-503.
33. Whitfield, M.L., et al., *Identification of genes periodically expressed in the human cell cycle and their expression in tumors.* Mol Biol Cell, 2002. **13**(6): p. 1977-2000.
34. Panda, S., et al., *Coordinated transcription of key pathways in the mouse by the circadian clock.* Cell, 2002. **109**(3): p. 307-20.
35. Zhu, G., et al., *Two yeast forkhead genes regulate the cell cycle and pseudohyphal growth.* Nature, 2000. **406**(6791): p. 90-4.
36. Gasch, A.P., et al., *Genomic expression programs in the response of yeast cells to environmental changes.* Mol Biol Cell, 2000. **11**(12): p. 4241-57.
37. Ivanova, N.B., et al., *A stem cell molecular signature.* Science, 2002. **298**(5593): p. 601-4.
38. Nau, G.J., et al., *Human macrophage activation programs induced by bacterial pathogens.* Proc Natl Acad Sci U S A, 2002. **99**(3): p. 1503-8.
39. Lin, T.H., N. Kaminski, and Z. Bar-Joseph, *Alignment and classification of time series gene expression in clinical studies.* Bioinformatics, 2008. **24**(13): p. i147-55.
40. Allocco, D.J., I.S. Kohane, and A.J. Butte, *Quantifying the relationship between co-expression, co-regulation and gene function.* BMC Bioinformatics, 2004. **5**: p. 18.
41. Vlachos, M., et al., *Indexing multidimensional time-series.* VLDB, 2006. **15**(1): p. 1-21.
42. Levy, D.E. and J.E. Darnell, Jr., *Stats: transcriptional control and biological impact.* Nat Rev Mol Cell Biol, 2002. **3**(9): p. 651-62.
43. Ruminy, P., et al., *Gene transcription in hepatocytes during the acute phase of a systemic inflammation: from transcription factors to target genes.* Inflamm Res, 2001. **50**(8): p. 383-90.
44. Duin, R.P.W. *Classifiers in almost empty spaces.* in *ICPR15 Proc. 15th Int. Conf. Pattern Recognit.* 2000. Barcelona, Spain: Los Alamitos: IEEE Comput. Soc. Press.
45. Huang, D., P. Wei, and W. Pan, *Combining gene annotations and gene expression data in model-based clustering: weighted method.* OMICS, 2006. **10**(1): p. 28-39.
46. Ernst, J., G.J. Nau, and Z. Bar-Joseph, *Clustering short time series gene expression data.* Bioinformatics, 2005. **21 Suppl 1**: p. i159-68.
47. Shedden, K. and S. Cooper, *Analysis of cell-cycle gene expression in Saccharomyces cerevisiae using microarrays and multiple synchronization methods.* Nucleic Acids Res., 2002. **30**: p. 2920-29.
48. Bar-Joseph, Z., et al., *Deconvolving cell cycle expression data with complementary information.* Bioinformatics, 2004. **20 Suppl 1**: p. i23-30.
49. Aach, J. and G.M. Church, *Aligning gene expression time series with time warping algorithms.* Bioinformatics, 2001. **17**(6): p. 495-508.
50. Alter, O., P.O. Brown, and D. Botstein, *Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms.* Proc Natl Acad Sci U S A, 2003. **100**(6): p. 3351-6.
51. Xu, X.L., J.M. Olson, and L.P. Zhao, *A regression-based method to identify differentially expressed genes in microarray time course studies and its application in an inducible Huntington's disease transgenic model.* Hum Mol Genet, 2002. **11**(17): p. 1977-85.
52. Wichert, S., K. Fokianos, and K. Strimmer, *Identifying periodically expressed transcripts in microarray time series data.* Bioinformatics, 2004. **20**(1): p. 5-20.
53. Jiang, D.X., C. Tang, and A.D. Zhang, *Cluster analysis for gene expression data: a survey.* IEEE Trans. Knowl. Data Eng., 2004. **16**(11): p. 1370-86.
54. Schliep, A., et al., *Analyzing gene expression time-courses.* IEEE/ACM Trans Comput Biol Bioinform, 2005. **2**(3): p. 179-93.

55. Mukherjee, S. and S. Mitra, *Hidden Markov Models, grammars, and biology: a tutorial.* J Bioinform Comput Biol, 2005. **3**(2): p. 491-526.
56. Nigam, K., A.K. McCallum, and T. Mitchell, *Semi-supervised Text Classification using EM.* Semi-Supervised Learning, 2006: p. 33-56.
57. Balasubramaniyan, R., et al., *Clustering of gene expression data using a local shape-based similarity measure.* Bioinformatics, 2005. **21**(7): p. 1069-77.
58. Madeira, S.C. and A.L. Oliveira, *Biclustering algorithms for biological data analysis: a survey.* IEEE/ACM Trans Comput Biol Bioinform, 2004. **1**(1): p. 24-45.
59. Prelic, A., et al., *A systematic comparison and evaluation of biclustering methods for gene expression data.* Bioinformatics, 2006. **22**(9): p. 1122-9.
60. Madeira, S.C., et al., *Identification of regulatory modules in time series gene expression data using a linear time biclustering algorithm.* IEEE/ACM Trans Comput Biol Bioinform, 2010. **7**(1): p. 153-65.
61. Peeters, R., *The Maximum Edge Biclique Problem Is NPComplete.* Discrete Applied Math, 2003. **131**(3): p. 651-5.
62. Tanay, A., R. Sharan, and R. Shamir, *Discovering statistically significant biclusters in gene expression data.* Bioinformatics, 2002. **18 Suppl 1**: p. S136-44.
63. Ji, L. and K.L. Tan, *Identifying time-lagged gene clusters using gene expression data.* Bioinformatics, 2005. **21**(4): p. 509-16.
64. Costa, I.G., A. Schonhuth, and A. Schliep, *The Graphical Query Language: A Tool for Analysis of Gene Expression Time-Courses.* Bioinformatics, 2004. **21**(10): p. 2544-46.
65. Zhang, Y., H. Zha, and C.H. Chu. *A Time-Series Biclustering Algorithm for Revealing Co-Regulated Genes*. in *Proc. Fifth IEEE Int'l Conf. Information Technology: Coding and Computing (ITCC '05),*. 2005.
66. Madeira, S.C. and A.L. Oliveira, *A polynomial time biclustering algorithm for finding approximate expression patterns in gene expression time series.* Algorithms Mol Biol, 2009. **4**: p. 8.
67. Hirano, S. and S. Tsumoto, *Empirical evaluation of dissimilarity measures for time-series multiscale matching.* Found. Intell. Syst., 2003. **2871**: p. 454-462.
68. Handl, J., J. Knowles, and D.B. Kell, *Computational cluster validation in post-genomic data analysis.* Bioinformatics, 2005. **21**(15): p. 3201-12.
69. Gusfield, D., ed. *Algorithms on Strings, Trees and Sequences*. 1997, University of California, Davis.
70. Batal, I., et al., *A temporal abstraction framework for classifying clinical temporal data.* AMIA Annu Symp Proc, 2009. **2009**: p. 29-33.
71. Kohavi, R. *A study of cross-validation and bootstrap for accuracy estimation and model selection.* in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*. 1995. San Francisco, CA.
72. Kohavi, R. and F. Provost, *Glossary of Terms.* Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process, Machine Learning, 1998. **30**(2/3): p. 271-274.
73. Hanczar, B., et al., *Small-sample precision of ROC-related estimates.* Bioinformatics, 2010. **26**(6): p. 822-30.
74. Wolpert, D.H., *The relationship between PAC, the statistical physical framework, the Bayesian framework, and the VC framework. Technical Report*. 1994, The Santa Fe Institute: Santa Fe, NM.
75. Efron, B. and R. Tibshirani, eds. *An introduction to the bootstrap.* 1993, Chapman & Hall.
76. Breiman, L. and P. Spector, *Submodel selection and evaluation in regression. The x-random case.* International Statistical Review 1992. **60**(3): p. 291-319.
77. Mitchell, T., ed. *Machine Learning*. 1997, McGraw-Hill.
78. Goncalves, J.P., S.C. Madeira, and A.L. Oliveira, *BiGGEsTS: integrated environment for biclustering analysis of time series gene expression data.* BMC Res Notes, 2009. **2**: p. 124.
79. Li, T., C. Zhang, and M. Ogihara, *A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression.* Bioinformatics, 2004. **20**(15): p. 2429-37.
80. Ye, L. and E. Keogh, *Time series shapelets: a new primitive for data mining.*, in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2009: Paris, France.

81. Gonçalves, J.P., Y. Moreau, and S.C. Madeira, *Time coherent three-dimensional clustering: unraveling local transcriptional patterns in multiple gene expression time series (Abstract and poster)*, in *Ninth European Conference on Computational Biology (ECCB 2010)*. 2010.
82. Symeonidis, P., et al., *Nearest-biclusters collaborative filtering based on constant and coherent values.* Information Retrieval, 2008. **11**(1): p. 51-75.
83. Borgwardt, K.M., S.V. Vishwanathan, and H.P. Kriegel, *Class prediction from time series gene expression profiles using dynamical systems kernels.* Pac Symp Biocomput, 2006: p. 547-58.
84. Normandin, Y., *High-performance connected digit recognition using maximum mutual information estimation.* IEEE Trans. Speech Audio Process., 1994. **2**: p. 299-311.
85. Kaminski, N. and Z. Bar-Joseph, *A patient-gene model for temporal expression profiles in clinical studies.* J Comput Biol, 2007. **14**(3): p. 324-38.
86. Sterrenburg, E., et al., *Large-scale gene expression analysis of human skeletal myoblast differentiation.* Neuromuscul Disord, 2004. **14**(8-9): p. 507-18.
87. Weinstock-Guttman, B., et al., *Genomic effects of IFN-beta in multiple sclerosis patients.* J Immunol, 2003. **171**(5): p. 2694-702.
88. Fraley, C. and A.E. Raftery, *How many clusters? which clustering method? answers via model-based cluster analysis.* Comput. J.,, 1998. **41**: p. 578-588.
89. Chapelle, O., ed. *Semi-supervised Learning.* 2006, MIT Press: Cambridge, MA.
90. Nelms, K., et al., *The IL-4 receptor: signaling mechanisms and biologic functions.* Annu Rev Immunol, 1999. **17**: p. 701-38.
91. Geurts, P., A. Irrthum, and L. Wehenkel, *Supervised learning with decision tree-based methods in computational and systems biology.* Mol Biosyst, 2009. **5**(12): p. 1593-605.
92. Li, L., et al., *Gene assessment and sample classification for gene expression data using a genetic algorithm/k-nearest neighbor method.* Comb Chem High Throughput Screen, 2001. **4**(8): p. 727-39.
93. Boser, B.E., I.E. Guyon, and V.M. Vapnik. *A training algorithm for optimal margin classifiers*. in *5th Annual ACM Workshop on COLT*. 1992. Pittsburgh, PA: ACM Press.
94. Han, J. and M. Kamber, eds. *Data Mining: Concepts and Techniques*. 2nd ed. The Morgan Kaufmann Series in Data Management Systems, ed. J. Gray. 2006, Morgan Kaufmann Publishers.
95. Haykin, S., ed. *Neural Networks: A Comprehensive Foundation*. 2nd ed. 1999, Prentice Hall: Upper Saddle River, NJ.
96. Peduzzi, P., et al., *A simulation study of the number of events per variable in logistic regression analysis.* J Clin Epidemiol 1996. **49**(12): p. 1373-79.

Appendices

A. Related Work

A.1 Baranzini et al. [15]

The dataset used in this thesis was the same collected by these authors, and described in Section 3.1. The training set was drawn from 75% of the whole original data, and the remainder 25% was used as a test set. Both sets preserved the proportion of good/bad responders in approximately 63%/37%. In this case, the expression matrix takes an asymmetric form, with more genes than samples, thus leading to an overfitting of the data .

The first method used in this work is a linear discriminant analysis-based Integrated Bayesian Inference System (IBIS) [15], which is a data-mining algorithm that searches for a gene (or group of genes) capable of predicting the outcome class, based on a voting scheme, with a committee of classifiers. The search for a significant expression signature with respect to therapeutical response was limited to local clusters, making use of conventional similarity measures, neglecting, to a certain level, the small differences between the two groups of responders.

A quadratic version of the discriminant analysis-based IBIS was implemented for triplets of genes. Given that the number of genes represents the dimension of the analysis, this was a 3D analysis, and it showed the best results, when compared to the 1D and 2D experiments.

With a cross-validation scheme, the nine highest scoring gene triplets were selected with a mixed threshold on low mean squared error (MSE) and high accuracy rates, only for the training set, the same used for building the classifier. Then, 100 random splits were made within the samples in order to minimize possible negative effects of data splitting.

This method was shown to have significant prediction ability with respect to the responder class, under a null hypothesis, and a reasonable robustness to Gaussian white noise, with a drop of less than 10%. To control the false discovery rate, the authors shuffled the class labels 1000 times, and, as expected for a high specificity algorithm, the respective prediction ability dropped significantly.

Since then, several works proposed different classification techniques to analyze this clinical dataset, as is the case of the use of support vector machines (SVM) based on dynamic system kernels to achieve a predictive model [83], and HMMs [24, 39].

A.2 Lin et al. [39]

Instead of the methods used in the previous work, these authors relied on Hidden Markov Models (HMMs) to build a predictive model upon the same clinical dataset for MS patients, aiming at a discriminative learning. The choice of the parameters is based on the maximization of the discrimination between the classes, with models learned simultaneously, where the parameters of one model are affected by the estimates of the other [84]. The training of the models is here made with both positive and negative samples, based on a Maximum Mutual Information Estimate (MMIE), an extended version of the Baum-Welch algorithm [84]. This idea is opposed to the one of generative learning, using Maximum Likelihood Estimation (MLE), in which the training is carried with only positive examples of the classes. The use of these HMMs revealed an important advantage for the analysis, as they are capable of unveiling the patient-specific treatment response rates, a major challenge in the time-series expression experiments, in which a first attempt to overcome this difficulty rested on a less appropriate method, using Support Vector Machines (SVM) with default kernels [85-87]. The results elevated the prediction accuracy obtained [15] to approximately 88%.

In order to reduce the complexity of the model, the authors performed a feature selection, selecting the most significant genes to the classification. Regarding the feature selection, there are two possible techniques: the filter approach, which consists in filtering out the most irrelevant genes, disregarding the underlying model. The alternative, wrapper approach, is slower than the first, but reveals the best results, evaluating the classifier on different gene/feature subsets.

The method of gene selection proposed by Lin et al. [39] is the HMM-RFE algorithm (where RFE stands for Recursive Feature Elimination), which begins by training the classifier, then eliminates the least contributing genes to the discrimination, and executes these two steps iteratively until a stopping criterion is achieved.

The classification of time series expression data from new patients was accomplished by using two models, one for each class of responders, based on the likelihood of the data given the models. Here, a hidden state in the model corresponds to a phase in the treatment response, and the patients can enter the states in different instants in time, and may remain in them for various time points. Moreover, the covariance matrix was made diagonal, to avoid the overfitting of the data, and the emission distribution of the HMMs states was a multivariate n-Gaussian, with n being the number of genes under study.

In order to further analyze the capabilities of their algorithm, the authors built a simulated dataset, composed by the expression profile constructed for one hundred subjects with a 50-50 distribution of good/bad responders. For each patient, one hundred genes were analyzed in a maximum of eight time points. The response profile was simulated by the random selection of a 1.5π length segment of a sine wave (in the interval 0 - 4π) [39]. Ten out of the one hundred genes were established as differentially expressed. In order to take into account the patient-specific response rate, a random scaling value (0.5 - 1.5) was considered. Finally, Gaussian noise was added to the profiles.

The main conclusion drawn by these authors was that the HMMs showed a better performance than the classifiers that do not take the temporal dependencies into account. Finally, they acknowledge the limitation of the diagonal covariance matrix, neglecting the interactions between the genes. However, with more available data, better models can be built, with more covariance terms.

A.3 Costa et al. [24]

To overcome some of the problems identified in the previous works [15, 39], these authors used the idea of constrained mixture estimation, using HMMs. Their work showed a significant improvement in the prediction accuracy (to higher than 90%), as opposed to the ones found in the previously discussed clinical analysis [15, 39].

The main contribution of this work was the application of constraints in the mixture estimation, falling in the semi-supervised category, since the constraints make use of the known class labels for the training set, while considering the new instances (with unknown class) at the same time, including them in the computation of the models' parameters. The constraints are built around pairs of patients to restrict or penalize certain solutions. Nonetheless, some of these solutions may be possible, if the penalty in the objective function is small enough, rendering the constraint softer. The constraints can be either positive, when the rule forces a pair of patients to be included in the same group, or negative, in the opposite case, forcing them to be clustered in different sets.

The mixture components of the constrained mixture estimation algorithm are usually linear HMMs, as seen in [39, 54]. Using only negative constraints, inferred from the patients' labels, allows the patients from a given responder class to be included in more than one group, thus originating a classification in subgroups [24].

A gaussian noise component was incorporated into the emissions probability density functions (pdfs), with median equal to the one of the dataset, and high variance [88].

It is also important to refer that, in this work, the authors suggested a new criterion for the feature selection, consisting in discarding the uninformative genes. Essentially, the method rests on the search for the best discriminative genes, by comparing the positive and negative prior distributions, computed with EM algorithms.

Follows a brief description of the method. More details can be found in the original paper of Costa et al. [24]. The classification algorithm consists of three steps:

1 - Estimation of a constrained mixture:

- only with training data, with known labels – supervised learning.
- with the whole dataset (labels known only for the training instances), remarking that the new instances (test patients) also contribute to the computation of the models parameters, based on their likelihood to each model – semi-supervised learning.

2 - Each of the mixture components is assigned to one of the classes (good, or bad responders). This decision is based on the contribution of the labeled (training) data points to the classes, in terms of their posteriors.

3 - Finally, the unlabeled data is assigned to the mixture components, also based in the maximum of the posterior distribution (available in the case of semi-supervised training). In the supervised learning situation, the posteriors have to be computed upon the termination of mixture estimation. The labels of data points are given through the class labels of the mixture components.

This classifier was applied on the mentioned datasets, either clinical from RR-MS patients undergoing a IFN-β treatment, and a simulated one, created in the previously discussed study [39].

The more interesting obtained results can be summarized as follows. For the simulated dataset, the addition of a noise component did not brought the prediction accuracy to values lower than 90%, revealing the method's ability to handle noisy data. The inclusion of mislabeled patients leads to a higher prediction accuracy for the constrained HMMs than for the generative HMMs (without constraints applied). The best values of prediction accuracy obtained for the MS dataset were obtained using the constrained HMMs with two or three mixing components, for both purely and semi-supervised learning, although the latter showed a slight superiority, a result supported by [89].

Feature selection showed to be crucial, since the performance was significantly worse when analyzing the whole set of seventy genes, as opposed to the one obtained for the seventeen selected genes.

The main result is the emergence of subgroups or subclasses of patients, as the two subclasses of good responders. In fact, patients belonging to the second class of good responders, show markers for both good and bad responders, like the IL-4Ra, involved in the immune response regulation (B cell mediated) [90].

B. State of the Art classifiers

B.1 Decision Trees

A Decision Tree is a predictive machine-learning model that predicts the class of a new instance based on the available attributes of the learnt data. These different attributes are denoted as the internal nodes of the decision tree, the branches inform about the possible values each attribute can take, and the terminal node represents the target value, or predicted class for that setting of attributes [91]. The algorithm starts with the choice of the attribute that is found to best discriminate the train instances (it is said to have the highest information gain). Then, if one value of the possible ones for this chosen attribute presents no ambiguity, i.e., all of the instances that fall within this value share the same class, then this class, or target value, is assigned to the attribute value. For the values for which this does not happen, the algorithm searches, among the remaining attributes, the one, inside the first attribute value that discriminates the respective instances. This process is repeated until no ambiguities still exist, or the algorithm runs out of attributes before attaining a perfect discrimination. In this case, the most frequent class under the branch is assigned to it [91].

B.2 k – Nearest Neighbors (kNN)

This method was already presented in Section 3.4.1, since an adapted distance-weighted version using biclustering is proposed in this thesis.

kNN is included in the instance-based learning, and is also referred to as a lazy learner. This is due to the fact that, here, no actual learning takes place. The training instances are only stored, and from this set, for a given test instance, the k closest train objects are retrieved, and the majority class is assigned to the test instance. The crucial aspects of this algorithm rest on the definition of the k parameter and the distance function. As already discussed, k must not be too large, to avoid the overfitting phenomenon. There are several different distance functions that can be used, and these are domain-dependent [92].

Although the kNN classifier presents the advantage of having no cost from learning and it still outperforms other algorithms of higher complexity, it has the setback of being limited to a local model, thus lacking power of generalization. Also, test cost increases linearly with the input instances. More information on the use of the kNN algorithm for gene expression data classification is found in [92].

B.3 Support Vector Machines

Support Vector Machines (SVM) are included in the group of supervised learning methods of classification and regression. Intuitively, we can think of an SVM model as a representation of the data points in a space, mapped in a manner that the instances of each class are separated from the other class with a gap as large as possible. A new instance with unknown class is mapped onto that space, and the prediction is based on which side of the gap it is. If the problem is linearly separable, for p-

dimensional data points, it means there is at least a (p-1)-dimensional hyperplane that separates the points. From all the possible hyperplanes of separation, the choice is usually made based on the hyperplane the largest margin between classes, maximizing the distance of the nearest training instances (the support vectors) in both sides to the hyperplane [93].

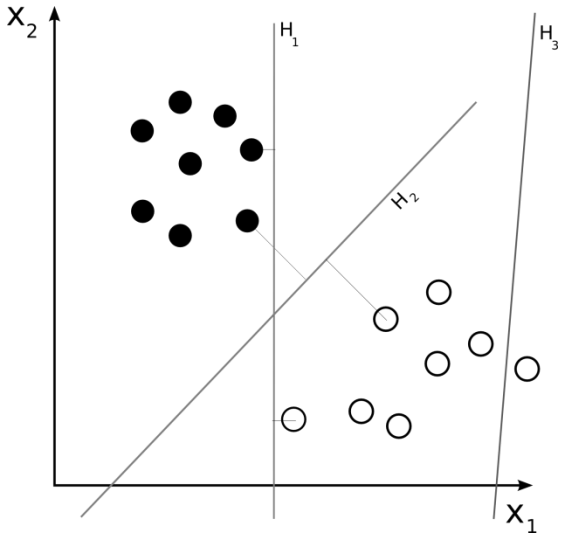


Figure 15 - SVM model with representation of 3 hyperplanes: H3 does not separate the two classes. Both H1 and H2 do, but H2 has the maximum margin (figure based on [94]).

If, on the contrary, the data are non-linearly separable, there is the possibility of still achieving a class separation with a hyperplane. However, it is necessary to transform the input space to a higher-dimensional space, the feature space, where the data can be linearly separable. This transformation is done with the application of the kernel trick. SVM have proved to have a significant importance in classification and regression tasks, for different domains, outperforming most of the other supervised learning methods in many problems. All formalization and more information on SVM can be found in [93].

B.4 Multilayer Perceptron

A Multilayer Perceptron (MLP) is a feedforward artificial neural network (ANN) model. It allows a mapping of the input data points onto a proper output. The main difference from the linear perceptron is the existence of hidden layers between the input and output nodes. However, the activation functions of those hidden layers (or neurons) are non-linear, otherwise it can be easily proven that all the layers can be reduced to the simple input-output model. Each node is connected, with a weight w_{ij}, to all of the nodes on the next node, or neuron [95].

The most used MLP activation functions are the hyperbolic tangent, $\Phi(y_i) = \tanh(v_i)$, ranging from -1 to 1 and the logistic function, $\Phi(y_i) = (1 + e^{-v_i})^{-1}$, similar in shape but ranging from 0 to 1. Another, more complex, activation function is the radial basis function, that shall be discussed in a later subsection. Here, y_i represents the output of the node i, and v_i is the weighted sum of the input

connections. Learning is processed in a supervised fashion, and each piece of training data leads to a change in the neuron's connections weights, based on the error produced by the output when faced to the expected result of that particular training input. More formalisms and information is present in [95].

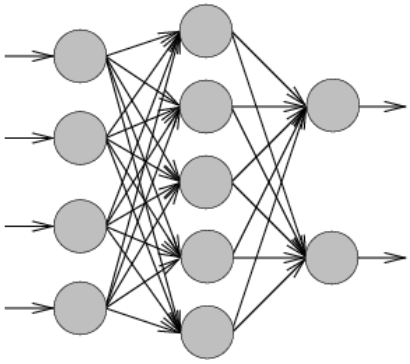


Figure 16 - Representation of a Multilayer Perceptron: in the left, there are the input nodes, and in the right, the output nodes. The intermediate nodes form a hidden layer. This layer's arrows represent the possible connections between the nodes of a layer and the ones of the next layer. Each connection has a weight wij (figure designed based on [94]).

B.5 Radial Basis Function (RBF) Network

RBF Network, similarly to the MLP, is an artificial neural network, but as it was mentioned for the MLP, the activation function is not a sigmoid function, but instead, a radial basis function. The output is given by a combination of these RBF's:

$$\varphi(x) = \sum_{i=1}^{N} a_i \, \rho(||x - c_i||) \tag{17}$$

with usually, $$\rho\big(||x - c_i||\big) = \exp(-\beta ||x - c_i||^2)$$

Here, N is the number of nodes in the hidden layer, c_i is the center vector for the node i, and the set of a_i's are the weights of the linear output node.

The learning consists in determining the weights a_i, center vectors c_i and β, in order to optimize the fit of the output φ to the data. The process is done, typically, in two stages: first, the radial basis functions are determined by means of unsupervised techniques. Then, the weights are computed with linear supervised methods, with a fast convergence. This is an important advantage of RBF networks when compared to the MLP, because there the learning is done in a single, computationally heavier step. Whenever the application requires a continuous learning (temporal prediction, online applications) the best choice should be the RBF. However, the power of generalization is, normally, higher for the MLP [95].

B.6 Logistic Regression

The most important function for this predictor is the logistic function, already mentioned as an activation function for MLP:

$$f(z) = \frac{e^z}{e^z + 1} = (1 + e^{-z})^{-1} \tag{18}$$

This is in fact a very interesting function to use in a classifier, because it can deal with input of any real value, and its output is limited to the interval]0,1[. This allows the consideration of the output $f(z)$ as the probability of a certain outcome, given the effect of a set of independent variables. The input z, also called as $logit$, represents the combined effect of that set of explanatory variables of the problem:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

$\beta_i, i \in \{1, \dots, k\}$ is the regression coefficient for the variable x_i, and β_0 is the "baseline" value (the probability of an outcome when all the variables have 0 contribution). If a coefficient is positive, it increases the probability of the specific outcome, decreasing it otherwise. On the other hand, if the coefficient absolute value is large, it strongly affects that probability. Mathematical formalisms and further analysis is presented in [96].

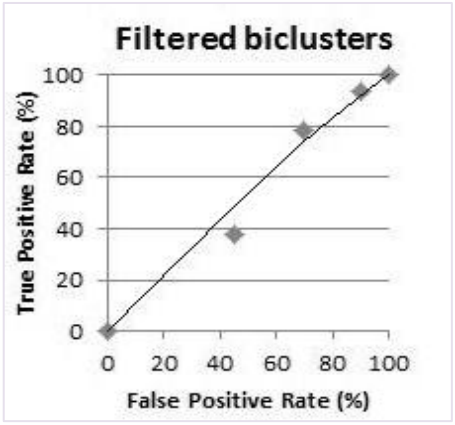
C. Experimental Results

C.1 Biclustering-based kNN classifier based on filtered biclusters similarities

The confusion matrix associated with the highest prediction accuracy is shown in Figure 17.a for k = 3 and no penalty in the score matrix. The similarity threshold used was 5% for class 0 (bad responders) and 10% for class 1 (good responders), and the class proportion threshold is 40% for class 0 and 60% for class 1. Figure 17.b shows the obtained approximation to the ROC curve for different sets of parameters ($k \in \{1,3\}$,$penalty \in \{0.98, 1.00, 1.02\}$,$similarity\ threshold \in [0.05,0.3]$ and $class\ proportion\ threshold \in [0,0.6]$).

		Classified as	
		0	1
Real Class	0	7 (35.00%)	13 (65.00%)
	1	8 (25.00%)	24 (75.00%)

a)



b)

	Filtered biclusters
Prediction Accuracy (LOO) (%)	59.62
Prediction Accuracy (5 x 4-fold) (%)	48.08

Figure 17 - a) Confusion matrix obtained for the biclustering-based kNN method based on filtered biclusters similarities (k = 3, no penalty), together with the respective prediction accuracies and b) the approximate ROC curve. The patients' class 0 and 1 correspond, respectively, to bad and good responders.

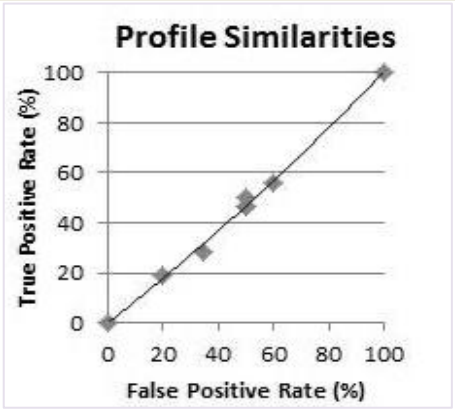
Comparing the confusion matrix in Figure 17.a with the baseline result (Figure 7.b), we see that they are identical. From the observation of the approximate ROC curve obtained with the filtered biclusters (Figure 17.b), we see that it is similar to the FP = TP line, associated to a random classifier. Thus, we can conclude that the application of this kind of filter does not improve the results.

2. Biclustering-based kNN classifier based on profile similarities (absolut number of shared profiles)

The confusion matrix for k = 3 and no penalties, together with the approximate ROC curve obtained for the experimented sets of parameters are displayed in Figure 18 ($k \in \{1,3\}$, $penalty \in \{0.98, 1.00, 1.02\}$). Comparatively to the baseline confusion matrix (Figure 7.b in Section 4.2.1), this method classified more 11 patients as bad responders (of these, only 3 were correctly predicted). In this case, the approximate ROC curve (Figure 18.b) nearly reproduces the FP = TP line, suggesting similar results than those expected for a random classifier.

		Classified as	
		0	1
Real Class	0	10 (50.00%)	10 (50.00%)
	1	16 (50.00%)	16 (50.00%)

a)



b)

	Profiles similarities
Prediction Accuracy (LOO) (%)	50.00
Prediction Accuracy (5 x 4-fold) (%)	52.69

Figure 18 - a) Confusion matrix obtained for the biclustering-based kNN method based on profiles similarities (k = 3, no penalty), together with the respective prediction accuracies and b) the approximate ROC curve. The patients' class 0 and 1 correspond, respectively, to bad and good responders.

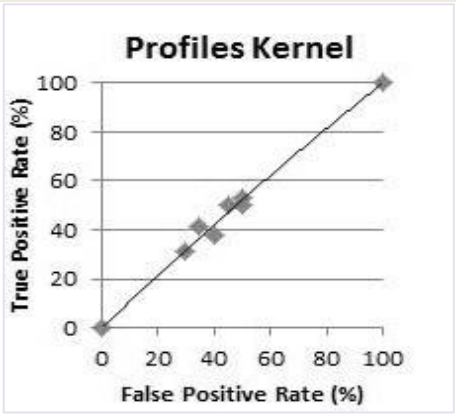
3. Biclustering-based kNN classifier based on profiles similarities (quadratic kernel)

Figure 19 shows the confusion matrix obtained with parameters k = 3, no penalties on the scores and a quadratic polynomial kernel, for a LOO CV evaluation, together with the approximate ROC curve built from several tests with different sets of parameters ($k \in \{1,3\}$, $penalty \in \{0.98, 1.00, 1.02\}$).

Facing the rates of the confusion matrix in Figure 19.a with the baseline (Figure 7.b in Section 4.2.1) we see that in the first case, 17 more patients are classified as bad responders, with only 5 of them being accurately predicted. The approximate ROC curve obtained (Figure 19.b) shows that this classifier's behavior resembles the FP = TP line, which is typical of a random classifier.

		Classified as	
		0	1
Real Class	0	12 (60.00%)	8 (40.00%)
	1	20 (62.50%)	12 (37.50%)

a)



b)

	Profiles similarities (kernel)
Prediction Accuracy (LOO) (%)	46.15
Prediction Accuracy (5 x 4-fold) (%)	46.54

Figure 19 - a) Confusion matrix obtained for the biclustering-based kNN method based on profiles similarities computed with a quadratic kernel (k = 3, no penalty), together with the respective prediction accuracies and b) the approximate ROC curve. The patients' class 0 and 1 correspond, respectively, to bad and good responders.

4. Meta-Biclusters classifier

Decision Tree

With a LOO CV evaluation scheme, the obtained results can be found in Figure 20.

		Classified as	
		0	1
Real Class	0	1(5.00%)	19 (95.00%)
	1	0 (0.00%)	32 (100.00%)

a)

		Classified as	
		0	1
Real Class	0	0(0.00%)	20 (100.00%)
	1	3 (9.40%)	29 (90.60%)

b)

		Classified as	
		0	1
Real Class	0	0(0.00%)	20 (100.00%)
	1	0 (0.00%)	32 (100.00%)

c)

	1000 meta-biclusters	750 meta-biclusters	500 meta-biclusters
Prediction Accuracy (LOO) (%)	63.46	55.77	61.54
Prediction Accuracy (5 x 4-fold) (%)	61.92	60.77	61.54

Figure 20 - Confusion matrices obtained for the decision tree classifier (software package Weka) for three numbers of meta-biclusters: a) 1000, b) 750 and c) 500, together with the respective prediction accuracies. The patients' class 0 and 1 correspond, respectively, to bad and good responders.

k – Nearest Neighbors

The confusion matrices obtained for a LOO CV evaluation with 1000, 750 and 500 meta-biclusters are shown in Figure 21.

		Classified as	
		0	1
Real Class	0	5 (25.00%)	15 (75.00%)
	1	2 (6.20%)	30 (93.80%)

a)

		Classified as	
		0	1
Real Class	0	3 (15.00%)	17 (85.00%)
	1	4 (12.50%)	28 (87.50%)

b)

		Classified as	
		0	1
Real Class	0	0(0.00%)	20 (100.00%)
	1	6 (18.70%)	26 (81.30%)

c)

	1000 meta-biclusters	750 meta-biclusters	500 meta-biclusters
Prediction Accuracy (LOO) (%)	67.31	59.62	50.00
Prediction Accuracy (5 x 4-fold) (%)	65.39	60.39	55.00

Figure 21 - Confusion matrices obtained for the standard kNN classifier (software package Weka) for three numbers of meta-biclusters: a) 1000, b) 750 and c) 500, together with the respective prediction accuracies. The patients' class 0 and 1 correspond, respectively, to bad and good responders.

Support Vector Machines

For the SVM classifier, the results come as follows, in Figure 22.

		Classified as	
		0	1
Real Class	0	5 (25.00%)	15 (75.00%)
	1	4 (12.50%)	28 (87.50%)

a)

		Classified as	
		0	1
Real Class	0	3 (15.00%)	17 (85.00%)
	1	6 (18.70%)	26 (81.30%)

b)

		Classified as	
		0	1
Real Class	0	0(0.00%)	20 (100.00%)
	1	9 (28.10%)	23 (71.90%)

c)

	1000 meta-biclusters	750 meta-biclusters	500 meta-biclusters
Prediction Accuracy (LOO) (%)	63.46	55.77	44.23
Prediction Accuracy (5 x 4-fold) (%)	64.61	58.85	51.92

Figure 22 - Confusion matrices obtained for the SVM classifier (software package Weka) for three numbers of meta-biclusters: a) 1000, b) 750 and c) 500, together with the respective prediction accuracies. The patients' class 0 and 1 correspond, respectively, to bad and good responders.

Logistic Regression

With the application of a logistic regression classifier, we obtained the results in Figure 23:

		Classified as	
		0	1
Real Class	0	5 (25.00%)	15 (75.00%)
	1	4 (12.50%)	28 (87.50%)

a)

		Classified as	
		0	1
Real Class	0	3 (15.00%)	17 (85.00%)
	1	5 (15.60%)	27 (84.40%)

b)

		Classified as	
		0	1
Real Class	0	0(0.00%)	20 (100.00%)
	1	9 (28.10%)	23 (71.90%)

c)

	1000 meta-biclusters	750 meta-biclusters	500 meta-biclusters
Prediction Accuracy (LOO) (%)	63.46	57.69	44.23
Prediction Accuracy (5 x 4-fold) (%)	64.61	58.85	51.92

Figure 23 - Confusion matrices obtained for the Logistic Regression classifier (software package Weka) for three numbers of meta-biclusters: a) 1000, b) 750 and c) 500, together with the respective prediction accuracies. The patients' class 0 and 1 correspond, respectively, to bad and good responders.

RBF Network

The results of applying a RBF Network with the different numbers of meta-biclusters are displayed in Figure 24.

		Classified as	
		0	1
Real Class	0	5 (25.00%)	15 (75.00%)
	1	4 (12.50%)	28 (87.50%)

a)

		Classified as	
		0	1
Real Class	0	3 (15.00%)	17 (85.00%)
	1	6 (18.70%)	26 (81.30%)

b)

		Classified as	
		0	1
Real Class	0	0(0.00%)	20 (100.00%)
	1	9 (28.10%)	23 (71.90%)

c)

	1000 meta-biclusters	750 meta-biclusters	500 meta-biclusters
Prediction Accuracy (LOO) (%)	63.46	55.77	44.23
Prediction Accuracy (5 x 4-fold) (%)	64.61	58.85	51.92

Figure 24 - Confusion matrices obtained for the RBF Network classifier (software package Weka) for three numbers of meta-biclusters: a) 1000, b) 750 and c) 500, together with the respective prediction accuracies. The patients' class 0 and 1 correspond, respectively, to bad and good responders.

Multilayer Perceptron

Finally, the results for the MLP classifier are presented in Figure 25. In this case, the confusion matrix and consequent prediction accuracy are equal, independent of the number of meta-biclusters used:

		Classified as	
		0	1
Real Class	0	0 (0.00%)	20 (100.00%)
	1	0 (0.00%)	32 (100.00%)

	1000, 750 and 500 meta-biclusters
Prediction Accuracy (LOO) (%)	61.54
Prediction Accuracy (5 x 4-fold) (%)	61.54

Figure 25 - Confusion matrix obtained for the MLP classifier (software package Weka) for all three numbers of meta-biclusters used: 1000, 750 and 500, together with the respective prediction accuracies. The patients' class 0 and 1 correspond, respectively, to bad and good responders.

5. Summary of statistics for biclustering-based classifiers

Table 8 - Summary of evaluation statistics obtained from the confusion matrices for the developed biclustering-based classifiers.

	Prediction Accuracy	Sensitivity 1 / Specificity 0	Specificity 1 / Sensitivity 0	Precision 1	Precision 0	False Positive rate	False Negative rate
kNN based on biclusters similarities	59.62	75.00	35.00	64.86	46.67	65.00	25.00
kNN based on filtered biclusters similarities	59.62	75.00	35.00	64.86	46.67	65.00	25.00
kNN based on profiles similarities	50.00	50.00	50.00	61.54	38.46	50.00	50.00
kNN based on filtered profiles similarities	63.46	81.25	35.00	66.67	53.85	65.00	18.75
kNN based on profiles similarities (kernel)	46.15	37.50	60.00	60.00	37.50	40.00	62.50
kNN based on filtered profiles similarities (kernel)	63.46	71.88	50.00	69.70	52.63	50.00	28.13
kNN based on symbol pairing	57.69	81.25	20.00	61.90	40.00	80.00	18.75
kNN based on symbol pairing (max time-lag = 1)	69.23	96.88	25.00	67.39	83.33	75.00	3.13
Meta-profiles	86.54	87.50	85.00	90.32	80.95	15.00	12.50
1000 Meta-biclusters (kNN)	67.31	93.75	25.00	66.67	71.43	75.00	6.25

6. State of the Art classifiers

6.1.1 Numeric Values

Figure 26 shows the confusion matrices built from the results of the collection of standard classifiers from the software package Weka, when applied to the original dataset with numeric values, together with the respective prediction accuracies.

		Classified as	
		0	1
Real Class	0	10 (50.00%)	10 (50.00%)
	1	5 (15.60%)	27 (84.40%)

a)

		Classified as	
		0	1
Real Class	0	17 (85.00%)	3 (15.00%)
	1	4 (12.50%)	28 (87.50%)

b)

		Classified as	
		0	1
Real Class	0	19(95.00%)	1 (5.00%)
	1	3 (9.40%)	29 (90.60%)

c)

	a) Decision Tree	b) kNN	c) SVM
Prediction Accuracy (LOO) (%)	71.15	86.54	92.31
Prediction Accuracy (5 x 4-fold) (%)	70.77	82.31	85.00

		Classified as	
		0	1
Real Class	0	15 (75.00%)	5 (25.00%)
	1	5 (15.60%)	27 (84.40%)

d)

		Classified as	
		0	1
Real Class	0	16 (80.00%)	4 (20.00%)
	1	2 (6.20%)	30 (93.80%)

e)

		Classified as	
		0	1
Real Class	0	18 (90.00%)	2 (10.00%)
	1	5 (15.60%)	27 (84.40%)

f)

	d) Logistic regression	e) RBF Network	f) MLP
Prediction Accuracy (LOO) (%)	80.77	88.46	86.54
Prediction Accuracy (5 x 4-fold) (%)	80.38	83.85	86.15

Figure 26 - Confusion matrices, and respective prediction accuracies, for the collection of classifiers used from the software package Weka, applied on the original, numeric, expression data: a) Decision tree, b) k-Nearest Neighbors (kNN), c) Support Vector Machines (SVM), d) Logistic regression, e) Radial Basis Function (RBF) Network and f) Multilayer Perceptron (MLP). The patients' class 0 and 1 correspond, respectively, to bad and good responders.

Discretized Values

The confusion matrices obtained when applying the collection of state of the art classifiers using the software package Weka, on a discretized version of the data, is represented in Figure 27.

.

		Classified as	
		0	1
Real Class	0	8 (40.00%)	12 (60.00%)
	1	13 (40.60%)	19 (59.40%)

a)

		Classified as	
		0	1
Real Class	0	14 (70.00%)	6 (30.00%)
	1	17 (53.10%)	15 (46.90%)

b)

		Classified as	
		0	1
Real Class	0	5(25.00%)	15 (75.00%)
	1	6 (18.70%)	26 (81.30%)

c)

	a) Decision Tree	b) kNN	c) SVM
Prediction Accuracy (LOO) (%)	51.92	55.77	59.62
Prediction Accuracy (5 x 4-fold) (%)	54.61	49.62	53.08

		Classified as	
		0	1
Real Class	0	9 (45.00%)	11 (55.00%)
	1	20 (62.50%)	12 (37.50%)

d)

		Classified as	
		0	1
Real Class	0	9 (45.00%)	11 (55.00%)
	1	11 (34.40%)	21 (65.60%)

e)

		Classified as	
		0	1
Real Class	0	7 (35.00%)	13 (65.00%)
	1	15 (46.90%)	17 (53.10%)

f)

	d) Logistic regression	e) RBF Network	f) MLP
Prediction Accuracy (LOO) (%)	40.38	57.69	46.15
Prediction Accuracy (5 x 4-fold) (%)	45.77	56.15	57.95

Figure 27 - Confusion matrices, and respective prediction accuracies, for the collection of classifiers used from the software package Weka, applied on a discretized version of the expression data: a) Decision tree, b) k-Nearest Neighbors (kNN), c) Support Vector Machines (SVM), d) Logistic regression, e) Radial Basis Function (RBF) Network and f) Multilayer Perceptron (MLP). The patients' class 0 and 1 correspond, respectively, to bad and good responders.