Universitat Politécnica de Catalunya

Hierarchical information representation and efficient classification of gene expression microarray data

PhD Thesis

Student:
Mattia Bosio

Thesis advisors:
Philippe Salembier
Albert Oliveras Vergés

2014

A tutta la mia famiglia.

"I have always thirsted for knowledge, I have always been full of questions"

Herman Hesse, *Siddartha*

Summary

In the field of computational biology, microarryas are used to measure the activity of thousands of genes at once and create a global picture of cellular function. Microarrays allow scientists to analyze expression of many genes in a single experiment quickly and efficiently. Even if microarrays are a consolidated research technology nowadays and the trends in high-throughput data analysis are shifting towards new technologies like Next Generation Sequencing (NGS), an optimum method for sample classification has not been found yet.

Microarray classification is a complicated task, not only due to the high dimensionality of the feature set, but also to an apparent lack of data structure. This characteristic limits the applicability of processing techniques, such as wavelet filtering or other filtering techniques that take advantage of known structural relation. On the other hand, it is well known that genes are not expressed independently from other each other: genes have a high interdependence related to the involved regulating biological process.

This thesis aims to improve the current state of the art in microarray classification and to contribute to understand how signal processing techniques can be developed and applied to analyze microarray data. The goal of building a classification framework needs an exploratory work in which algorithms are constantly tried and adapted to the analyzed data. The developed algorithms and classification frameworks in this thesis tackle the problem with two essential building blocks. The first one deals with the lack of a priori structure by inferring a data-driven structure with unsupervised hierarchical clustering tools. The second key element is a proper feature selection tool to produce a precise classifier as an output and to reduce the overfitting risk.

The main focus in this thesis is the binary data classification, field in which we obtained relevant improvements to the state of the art. The first key element is the data-driven structure, obtained by modifying hierarchical clustering algorithms derived from the Treelets algorithm from the literature. Several alternatives to the original reference algorithm have been tested, changing either the similarity metric to merge the feature or the way two feature are merged. Moreover, the possibility to include external sources of information from publicly available biological knowledge and ontologies to improve the structure generation has been studied too. About the feature selection, two alternative approaches have been studied: the first one is a modification of the IFFS algorithm as a wrapper feature selection, while the second approach involved an ensemble learning focus. To obtain good results, the IFFS algorithm has been adapted to the data characteristics by

introducing new elements to the selection process like a reliability measure and a scoring system to better select the best feature at each iteration. The second feature selection approach is based on Ensemble learning, taking advantage of the microarryas feature abundance to implement a different selection scheme. New algorithms have been studied in this field, improving state of the art algorithms to the microarray data characteristic of small sample and high feature numbers.

In addition to the binary classification problem, the multiclass case has been addressed too. A new algorithm combining multiple binary classifiers has been evaluated, exploiting the redundancy offered by multiple classifiers to obtain better predictions.

All the studied algorithm throughout this thesis have been evaluated using high quality publicly available data, following established testing protocols from the literature to offer a proper benchmarking with the state of the art. Whenever possible, multiple Monte Carlo simulations have been performed to increase the robustness of the obtained results.

Resumen

En el campo de la biología computacional, los microarrays son utilizados para medir la actividad de miles de genes a la vez y producir una representación global de la función celular. Los microarrays permiten analizar la expresión de muchos genes en un solo experimento, rápidamente y eficazmente. Aunque los microarrays sean una tecnología de investigación consolidada hoy en día y la tendencia es en utilizar nuevas tecnologías como Next Generation Sequencing (NGS), aun no se ha encontrado un método óptimo para la clasificación de muestras.

La clasificación de muestras de microarray es una tarea complicada, debido al alto número de variables y a la falta de estructura entre los datos. Esta característica impide la aplicación de técnicas de procesado que se basan en relaciones estructurales, como el filtrado con wavelet u otras técnicas de filtrado. Por otro lado, los genes no se expresen independientemente unos de otros: los genes están inter-relacionados según el proceso biológico que les regula.

El objetivo de esta tesis es mejorar el estado del arte en la clasificación de microarrays y contribuir a entender como se pueden diseñar y aplicar técnicas de procesado de señal para analizar microarrays. El objetivo de construir un algoritmo de clasificación, necesita un estudio de comprobaciones y adaptaciones de algoritmos existentes a los datos analizados. Los algoritmos desarrollados en esta tesis encaran el problema con dos bloques esenciales. El primero ataca la falta de estructura, derivando un árbol binario usando herramientas de clustering no supervisado. El segundo elemento fundamental para obtener clasificadores precisos reduciendo el riesgo de overfitting es un elemento de selección de variables.

La principal tarea en esta tesis es la clasificación de datos binarios en la cual hemos obtenido mejoras relevantes al estado del arte. El primer paso es la generación de una estructura, para eso se ha utilizado el algoritmo Treelets disponible en la literatura. Múltiples alternativas a este algoritmo original han sido propuestas y evaluadas, cambiando las métricas de similitud o las reglas de fusión durante el proceso. Además, se ha estudiado la posibilidad de usar fuentes de información externas, como ontologías de información biológica, para mejorar la inferencia de la estructura. Se han estudiado dos enfoques diferentes para la selección de variables: el primero es una modificación del algoritmo IFFS y el segundo utiliza un esquema de aprendizaje con " ensembles". El algoritmo IFFS ha sido adaptado a las características de microarrays para obtener mejores resultados, añadiendo elementos como la medida de fiabilidad y un sistema de evaluación para seleccionar la mejor variable en cada iteración. El método que utiliza "ensembles" aprovecha la abundancia de features de los microarrays para implementar una selección diferente. En este

campo se han estudiado diferentes algoritmos, mejorando alternativas ya existentes al escaso número de muestras y al alto número de variables, típicos de los microarrays.

El problema de clasificación con más de dos clases ha sido también tratado al estudiar un nuevo algoritmo que combina múltiples clasificadores binarios. El algoritmo propuesto aprovecha la redundancia ofrecida por múltiples clasificadores para obtener predicciones más fiables.

Todos los algoritmos propuestos en esta tesis han sido evaluados con datos públicos y de alta calidad, siguiendo protocolos establecidos en la literatura para poder ofrecer una comparación fiable con el estado del arte. Cuando ha sido posible, se han aplicado simulaciones Monte Carlo para mejorar la robustez de los resultados.

Acknowledgments

I would like to thank my thesis advisors, Philippe Salembier and Albert Oliveras Vergés, who accompanied me through these years with encouragements, advices and guidance in my growth as a researcher. I want also to thank all the guys from the GPI group, thanks to whom I felt among friends. A special thanks is for Pau Bellot, with whom we shared several meetings and gave relevant feedback for the research process.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The developed work in this thesis lies in the field of automatic microarray data analysis-analysis and fits well the National Institute of Health, NIH, definition of bioinformatics

> Bioinformatics: Research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral or health data, including those to acquire, store, organize, archive, analyze, or visualize such data.

More in detail, this research work consists in developing a novel, global approach, with which high-throughput data like microarrays can be classified. To this end, signal processing techniques have been developed, applied and evaluated to improve the current results within the microarray analysis field. In[110], the usefulness of signal processing techniques in the bioinformatics field is well described:

> The recent development of high-throughput molecular genetics technologies has brought a major impact to bioinformatics and systems biology. These technologies have made possible the measurement of the expression profiles of genes and proteins in a highly parallel and integrated fashion. The examination of the huge amounts of genomic and proteomic data holds the promise for understanding the complex interactions between genes and proteins, the functional processes of a cell, and the impact of various factors on a cell, and ultimately, for enabling the design of new technologies for intelligent management of diseases. ... The importance of signal processing techniques is due

> to their important role in extracting, processing, and interpreting the information contained in genomic and proteomic data. It is our hope that signal processing methods will lead to new advances and insights in uncovering the structure, functioning and evolution of biological systems.

Signal processing techniques are key in the analysis process, since the problems to solve in the microarray data analysis are similar problems already faced in the telecommunication-related signal processing field (e.g. analysis and compression of large data, noise cancellation, pattern detection, feature selection and classification). Moreover, a vast literature already exists, in which a whole plethora of algorithms from the signal processing world are taken, modified and adapted for the analysis of high-throughput data such as microarrays. This thesis work aims to further improve the application of signal processing techniques to the analysis of a widely adopted tool like microarrays.

The main tasks treated in this thesis are the classification of incoming samples (e.g. to determine whether a microarray sample represents a person with a certain disease type or not), the relevant feature extraction of a microarray set (e.g. to identify the most discriminating genes between two classes) and the improvement of results interpretability from a biological point of view.

The developed techniques and tools focus on building a hierarchical data representation for the gene expression data able to produce useful features for classification, either using only the numerical information from microarray, or by including previous biological knowledge to ease the results interpretation and to increase the biological coherence of the generated structure. Algorithms have been developed for the binary classification problem, which is by far the most studied task in classification. In this area, properly tuned feature selection algorithms have been developed and tested to take into account the microarray data characteristics. The multiclass classification has also been considered by developing a novel ensemble classification technique combining multiple binary classifier to obtain a more robust sample classification.

Microarrays are an important and well established technology in the biomedical research field, developed to allow researchers to gather a very large number of gene expressions simultaneously. By measuring the mRNA level, the state of a cell can be determined

and inferences about phenomena inside the cell can be made [138]. In each microarray experiment, a large number of gene expressions are measured, typically tens of thousands, with a relatively small sample number. Microarrays are an extreme example of sample scarcity, or high-dimensionality of the feature set and this is a critical issue during the data analysis step.

The first publication using microarrays for cancer classification is from Golub in 1999 [52], where a gene subset with large mean value difference between classes and small variance within each class has been selected from the initial dataset and used as a predictor classifier. Since then, a wide variety of learning approaches have been proposed for microarray data analysis, like for example data normalization and correction, classification or regulatory network identification.

1.1 Microarray Data

In the field of computational biology, microarrays are used for gene expression profiling, which is the measurement of the activity (the expression) of thousands of genes at once, to create a global picture of cellular function.

Microarrays allow scientists to analyze expression of many genes in a single experiment quickly and efficiently. They represent a major methodological advance and are a powerful research tool, used by scientists to try to understand fundamental aspects of growth and development as well as to explore the underlying genetic causes of many human diseases.

Microarrays data are usually visualized with the help of a heat map, like the example shown in Figure 1-1, in which genes are arranged as columns, while each row represents a sample. In figure 1-1, the samples are sorted by their classes: the first 32 rows are from a class while the last 30 are from another. In the adopted color scheme, red values indicate high gene expression level, while blue values indicate low gene expression level. The heat map gives a visual summary of the collected genetic information and, at the same time, well visualizes the problem to be faced: there is too much information without an associated knowledge to easily discriminate between classes.

Microarray classification is a complicated task, not only due to the high dimensionality of the feature set, but also to an apparent lack of data structure. Even if data are presented

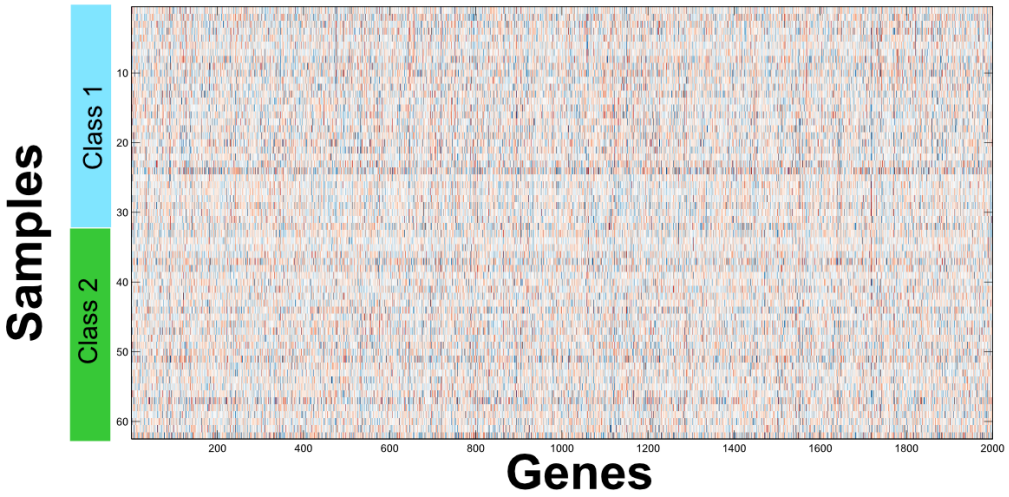

Figure 1-1: Microarray data visualization with heat map. Each columns represents a single gene, while each row represents a sample and it visualizes the lack of apparent regularity in a microarray dataset.

as a matrix, no a priori relation exists from the geometrical proximity, see for example in Figure 1-1 where there is no local uniformity across the columns. This characteristic limits the applicability of processing techniques, such as wavelet filtering or other filtering techniques that take advantage of known structural relation. On the other hand, it is well known that genes are not expressed independently from each other [50]: genes have a high interdependence depending on the involved regulating biological process. Therefore, even if gene expressions have no geometrical structure in the microarray data, the measured values themselves do have an unknown structure, which could be used to process the data.

An additional issue when analyzing microarray data is the measurement noise. In microarray experiments, fluorescent intensities related to gene expression levels are measured with sophisticated algorithms of image processing. Even so, an issue many researches find compelling to solve is how to effectively discern the actual values from experimental noise [68]. This is an issue even if in recent studies like [112] it is stated how the actual technical noise is low. It still is not zero and data suffer from random gene expression fluctuation which can alter the real expression value. To address the main noise effect due to some systematic error, various normalization and batch effect correction techniques have been developed throughout the literature [50, 107]. With some differences all of them manage to obtain comparable data across various microarray samples (even if not noise free). To

address the residual noise fluctuation, benefits would be obtained if the underlying data structure for the gene expression was found.

1.2 Problem statement

As anticipated in Section 1.1, microarray data characteristics can add complexity to the classification task:

- High feature set dimension with respect to the sample number also known as curse of dimensionality [11];
- Lack of a priori known data structural relations;
- Residual measurement noise even after applying normalization techniques.

The main problem to be solved is how to develop an algorithm able to output a precise and reliable classifier with repeatable result, considering the microarray data characteristics. Even if microarrays are a consolidated research technology nowadays and the trends in high-throughput data analysis are shifting towards new technologies like Next Generation Sequencing (NGS) [102], an optimum method for sample classification has not yet been found.

In recent studies from the microarray quality control study consortium, MAQC, [112], an extensive evaluation of classification algorithms has been performed. From MAQC results in [112], no individual method resulted to be always the best in all datasets. Furthermore, from the published results in [112], it can be observed how there is still a lot of room for improvement for the classification predictive properties. Moreover, the research for a better microarray classification algorithm is interesting for the current and future sequencing techniques like NGS. With NGS, the output data are basically affected by the same problems as microarrays, with the added inconvenience of not having neither consolidated data normalization and correction techniques, nor a wide availability of data or previous works to compare with. On the other hand, a new algorithm analyzing microarray data can be compared with a large amount of preexisting literature. Moreover, there is the possibility to analyze many public datasets from for example Gene Expression

Omnibus, GEO, [41], thus it is possible to focus on the algorithmic aspect without being too conditioned by the data quality control like with the current state of NGS data analysis. In this way, algorithms can be developed for microarrays, compared with the best alternatives and later straightforwardly adapted to the next high-throughput sequencing technology with good chance of maintaining the performances.

In the literature a plethora of microarray classification methods have been developed and a review of the most popular alternatives is presented in Chapter 2. In almost every case, feature selection algorithms have been applied to reduce the impact of the feature number. The aim of the feature selection task is to choose a subset of relevant features for building robust learning models. By removing the most irrelevant and redundant features from the data, the feature selection helps to improve the predictive performance. In this way, the generalization capability and the model interpretability are enhanced.

The lack of structure affects the possibility to apply a whole set of learning techniques based on some proximity measure, being it spatial, spectral or functional. The lack of structure is also an issue for noise reduction techniques based on low-pass filtering: the lack of knowledge about features that are supposed to have a similar behavior limits the applicability of low pass operators. In order to extract a structure from the numerical data, unsupervised learning techniques have also been proposed in the literature, among which an important subset are the clustering techniques. The clustering operation defines sets of related genes by some similarity measure. A whole universe of alternatives exists, and a review of them is included in Chapter 2.

Finally, a determinant role in the classification process is played by the classification rule itself. A review of existing classification techniques applied to microarray data analysis is included in Section 2.3, but more complete information can be found in [58, 38, 50]. The panorama of classification techniques is extremely diverse, from very simple classification rules to highly complex network systems. This thesis focus is to produce a general purpose classification methodology for microarray data. Different classification rules have been studied and compared, like the Linear Discriminant Analysis classifier (LDA) [58], Support Vector Machines (SVM) [109], or k-Nearest Neighbors (kNN) [58], but the proposed scheme can work with almost any existing classifier.

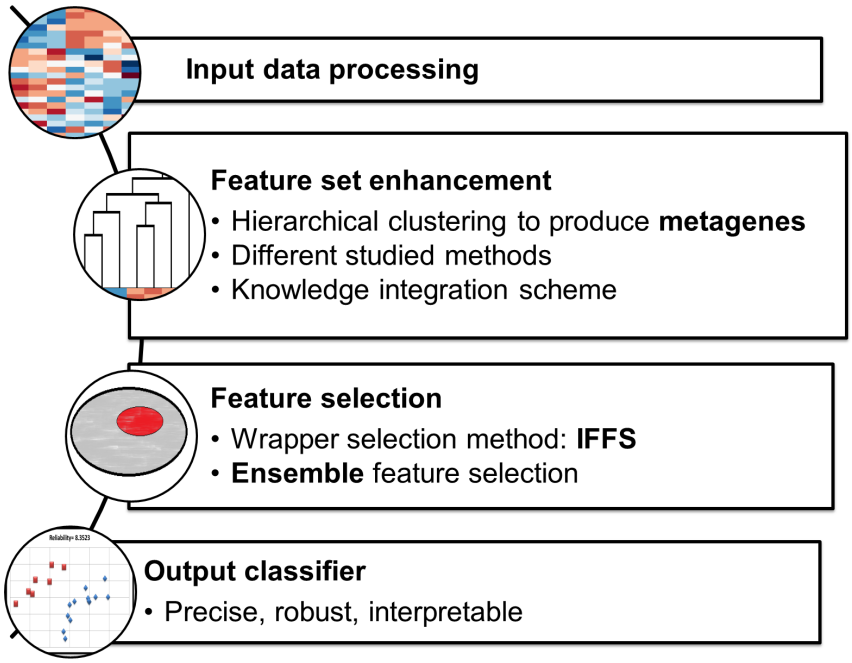

Figure 1-2: Current model framework for the binary classification case. The feature set enhancement phase and the feature selection phase have been studied in multiple configurations.

Due to the large amount of data provided by Microarray technology, the data analysis and feature extraction steps need the application of automatic and efficient processing techniques. The framework proposed in this thesis for binary classification is visualized in Figure 1-2. The core of the algorithm is the two-step process following the data preprocessing. A first phase infers a structure from the numerical data and produces new features called *metagenes*. Then, in a second step, different feature selection algorithms have been developed and compared. Finally, the algorithm output is the produced classifier. As far as the multiclass classification is concerned, the classification is obtained as a combination of multiple binary classifiers and it is detailed in Chapter 6 .

1.3 Contributions

This thesis aims at building an analysis framework for microarray data to output a precise and reliable classifier, improving the best alternatives in the state of the art. The goal is to produce a valid candidate for microarray data classification, which outputs an accurate and interpretable prediction model when analyzing new samples also in terms of biological

relevance. To achieve the proposed results, a variety of issues must be addressed to effectively extract information from the numerical data. The next sections describe the contributions from this thesis to each one of the faced microarray classification issues.

1.3.1 Feature set enhancement with metagenes

The first step in the proposed framework is to process the original data and extract a new set of features called *metagenes*. The objective is to infer a structure and to create a set of *metagenes* to be added to the original gene expression values. The newly created *metagenes* can improve the classification ability by expanding the feature space and reducing the noise when summarizing local clusters of correlated genes. Furthermore, since a data structure is created, it can be used to interpret the obtained results or to look for alternatives in case of practical implementation problems.

Metagenes are built from hierarchical clustering and are obtained as linear combination of the original features (i.e. gene expressions). This elaboration step aims at expanding the original feature set with useful alternatives. Algorithms like *Tree Harvesting*, [57], or *Pelora*, [35], highlight the usefulness of hierarchical clustering as a method to extract interesting new variables to expand the original feature set. The possibility to summarize groups of similar genes in a single feature as input for the classifier has many advantages. First, the interpretability of the selected feature as a combination of correlated genes that may be involved in the same biological process. Second, the robustness to chance because a group of correlated genes useful for classification is less likely to be due to chance than an individual gene. Third and last, classifying with a cluster-representing feature can highlight linear relations among groups of correlated genes.

The unsupervised analysis algorithm applied in this work aims at extracting new features representative of the original set. A structure is assigned to the data based on a similarity metric (details are included in Chapter 3) and this has a double utility:

- It defines the neighbors[1] of each gene, thus allowing a noise filtering effect when the *metagene* is defined. The common behavior of a gene cluster is encoded into the representing *metagene*. The result is a new set of features which emphasizes the

[1]The neighbor term, here, is used to define a gene close to another in the inferred structure.

common traits of gene groups, simultaneously reducing the residual noise on the measured values.

- It eases the results interpretation once the model is defined and also eases the model redefinition in case of practical inapplicability. Results can be more easily interpreted because, for each gene, groups of genes showing similar behavior are highlighted together with the extent of their similarity. The relations among genes are quantified by the similarity metric. About the practical issues once the model is defined, an example could be the high cost of a confirmation experiment for a specific gene. Thanks to the inferred structure by the clustering operation, alternatives to critical genes in the model can be found by looking at the produced tree.

The studied hierarchical clustering algorithm is one, but the actual inferred structure can change significantly depending on how it is implemented. We have performed studies to compare different clustering implementations, changing the similarity metric or the *metagene* generation rule, switching from the local Principal Component Analysis, PCA, proposed in [78], to a Haar-base feature fusion. From this study it emerged how the *metagene* generation allows producing better classifiers when compared to state of the art alternatives like those from [112].

Moreover, a knowledge integration scheme has been studied to include some specific prior knowledge in the clustering process. The objective is to produce a more interpretable and biologically meaningful clustering.

1.3.2 Feature selection

Feature selection is a compelling task when classifying microarray data to reduce the risk of overfitting. Its aim is to select an informative feature subset to be used for classification and it can be pursued in different fashions. Feature selection can be independent of the classifier as in the *t-test* [38]. This set of methods is usually referred to as *filters*. If the feature selection uses the classifier to evaluate the performance of each subset as in Sequential Floating Forward Selection (SFFS) [104], it is called *wrapper*. Otherwise, if the feature selection is coupled with the classifier design, as in recursive ridge regression [79], it is referred to as *embedded* methods. Different methodologies of feature selection exist, each

of which has benefits and drawbacks, and a more detailed discussion is reported in Section 2.2. In this thesis, *wrapper* methods have been implemented. Innovation elements have been introduced inside the selection phase, considering the microarray data characteristics in order to improve the final predictive ability. The main reason for modifying a feature selection algorithm is that, in microarray analysis, there is no fully reliable error estimator [19, 21, 20] due to the sample scarcity.

In addition to developing a novel *wrapper* algorithm, an indirect feature selection technique has also been studied. More precisely, it is an ensemble technique for classification based on the accuracy in diversity algorithm from [8], which is detailed in Section 4.3. It iteratively classifies samples with a majority voting scheme by selecting a subset of so called experts. Each expert has been chosen to be a classifier trained on a single feature, gene expression or microarray, and this is why it is an indirect form of feature selection. By selecting a subset of experts, a subset of features is selected since they are used to train the ensemble experts. This form of feature selection results in a significant improvement of the prediction properties of the classifiers when compared to state of the art alternatives on publicly available data.

1.3.3 Binary classification

This is the primary objective of this thesis: to correctly classify as many samples as possible. A key element for this task is the choice of the classification rule. It implies to choose a classifier which is precise, robust to overfit and with consistent results when applied to independent validation data. Numerous classification rules have been proposed during the last years and many comparative studies exist [58, 138, 74]. It is shown how good results can be obtained with both complex rule based classifiers (e.g. a neural network with hundreds of nodes [19], and with simple rule classifiers like linear discriminant analysis (LDA). [19] states that simple rules should be preferred in the absence of a reliable error estimator. As this is the case for microarray data classification, a set of simple rule classifiers like the linear discriminant analysis classifier (LDA) or linear Support Vector Machines SVM has been chosen. The choice is motivated by the rules simplicity, by their robustness to small training set changes and by the interpretability of the output results

as linear combinations.

From the performed experiments, it has been observed, coherently with previously gathered results [19, 58, 112], how the adopted simple classification rules allow obtaining results comparable, or even better, than state of the art alternatives. This without needing any fine parameter tuning or complex training schemes. The LDA classifier has been chosen as preferred method to compare all the different tested algorithm flavors, in terms of hierarchical clustering algorithm, feature selection algorithm or the multiclass classification scheme. Even so, once one specific alternatives has been selected, implementations with SVM or KNN has also been considered to evaluate possible benefits.

1.3.4 Multiclass classification

Multiclass cancer classification is still a challenging task in the field of machine learning. A novel multiclass approach has been developed as a combination of multiple binary classifiers. It is an example of Error Correcting Output Codes algorithms, applying data transmission coding techniques to improve the classification as a combination of binary classifiers. ECOC codes showed interesting properties but suffer of some issues which do not allow a remarkable prediction ability improvement. The proposed method combines the One Against All, OAA, approach with a set of classifiers separating each class-pair from the rest, called Pair Against All, PAA. The OAA+PAA approach has been tested on seven publicly available datasets and compared with the common OAA approach and with state of the art alternatives. The obtained results showed how the OAA+PAA algorithm consistently improves the OAA results, unlike other ECOC algorithms presented in the literature which did not lead to better results than OAA.

1.4 Thesis organization

This thesis work is organized as follows:

- In Chapter 2, a review of the state of the art concerning the studied elements in microarray classification is presented. A review of different classification approaches, as well as feature selection techniques or unsupervised learning techniques is presented

to offer a panorama of some of the most relevant developed algorithms.

- Chapter 3 presents the studied feature set enhancement algorithms to obtain a hierarchical structure from the data and to generate new features called metagenes, from the base Treelets algorithm [78], to all the studied variants.

- Chapter 4, is the core of this thesis and it is dedicated to the binary classification case. The studied feature selection algorithms are presented, as well as all the adaptations to the microarray scenario. In Chapter 4, the experiments to compare all the developed algorithms are presented, as well as the comparison with the state of the art.

- In Chapter 5, the knowledge integration scheme is introduced to explain how to integrate the numerical data with a priori known biological information. The proposed integration framework has been compared among its alternatives, with the state of the art and with the original Treelets implementation from Chapter 3.

- In Chapter 6, the developed algorithm for the multiclass classification case is presented and compared with baseline algorithms and with state of the art alternatives.

- Chapter 7 includes the conclusions for this thesis work and future research directions.

Chapter 2

State of the art

Through the last years, many methods tackled the high-throughput biological data classification problem with different angles, addressing the most relevant issues to produce an efficient classifier which conjugates high prediction performance with robustness to overfitting and with an interpretable biological meaning. In this thesis, the classification task is addressed by implementing a system composed of three main parts: the hierarchical data representation, the feature selection and the classification rule. The state of the art about the three main parts of this thesis work is summarized here to offer a panoramic view of the available techniques, with their strengths and limitations.

2.1 Hierarchical data representation

Microarrays do not have a known data structure that can be used to implement efficient filtering techniques for noise reduction. They provide unordered data which are considerably hard to read and interpret, due to the enormous amount of available variables. A large number of algorithms have been developed to make order from the unstructured gene expression data without using any previous information about the samples categories and are called unsupervised learning algorithms.

2.1.1 Unsupervised learning

Unsupervised learning refers to the problem of trying to find a hidden structure in unlabeled data. In the proposed framework, unsupervised learning techniques are implemented to find a hierarchical structure for the gene expression data and to generate a new set of features called metagenes. Unsupervised learning encompasses many techniques that seek to summarize and to explain key features from the data. Approaches to unsupervised learning include clustering algorithms (e.g. k-means, mixture models, hierarchical clustering) or blind signal separation using feature extraction techniques for dimensionality reduction (e.g. Principal component analysis, Independent component analysis, Non-negative matrix factorization, Singular value decomposition), for a detailed survey about these and more techniques refer to [38, 58, 93].

The goal of clustering is, roughly said, to assign a set of objects into groups called clusters so that objects in the same cluster are more similar to each other than to those in other clusters. Clustering algorithms differentiate themselves in the adopted similarity metric, which defines when two object are close to each other, and in the procedure to define the cluster number and their composition (i.e. the actual clustering algorithm). Popular clustering algorithms applied in microarray analysis are hierarchical clustering[42], k-means[85], partitioning around medioids (PAM)[120], self-organizing maps (SOM)[70], or bi-clustering methods [91]. A detailed explanation of these algorithms can be found in [38] and information about their utilization in microarray analysis is presented in [99, 63, 93].

Among the most popular clustering algorithms, the closest to this thesis objective is hierarchical clustering. It has been the first algorithm to be used in microarray research to group genes [42]. It is an iterative process in which, at first, each object is assigned to its own cluster, then, the two most similar clusters are joined, representing a new node of the clustering tree. This process is repeated until only a single cluster remains, including all the data. Variants to this algorithm exist, among which the simple process inversion is called top-down hierarchical clustering: the process starts from one cluster only, which is iteratively split into two clusters until one cluster for each feature is obtained. Hierarchical clustering outputs a tree of nested clusters. Each node in the tree represents a group of similar genes (i.e. the group composition depends on the chosen similarity

metric). Taking advantage from the tree resulting from hierarchical clustering, Lee's work in [78] presents a multi-resolution representation and eigen-analysis of the original data through an iterative pairwise hierarchical clustering algorithm called *Treelets*. This method produces a tree in which, at each level, the two most similar features are chosen and replaced by a coarse-grained approximation feature and a residual detail feature. This characteristic from *Treelets* will be used in the metagene creation process because it allows a local representation of common behavior of a gene cluster and more details are provided in Section 3.1.

2.1.2 Knowledge integration for clustering

A relevant theme addressed in this thesis within the hierarchical data representation and metagene generation, is the opportunity to include prior biological knowledge to drive the hierarchical clustering process. A relevant issue with high-throughput biological data is how to extract reliable knowledge from the vast amount of available data [3]. A whole set of analysis tools have been developed to help the interpretation task and to infer relationships between the gene signatures and biological knowledge databases [115, 82, 27, 67, 30].

Including and integrating prior biological knowledge has gained importance in the omics data analysis field throughout the years [3, 30]. Knowledge databases have been used in many directions, for example, to identify biologically relevant activated pathways by integrating Gene Ontology (GO) in the analysis process [105], or to integrate a gene ranking tool in the analysis [127]. Moreover, biological knowledge is also used in tools like Hanalyzer [77] to identify gene-to-gene relationships and facilitate the data interpretation.

Knowledge integration for microarray classification has been recently applied in modifications of classification methods like Nearest shrunken centroids [122] and Penalized partial least squares (PPLS) [133] called mPAM and mPLS, respectively [117]. Both methods implicitly contain a mechanism for selecting genes based on a penalty applied according to the discriminatory power of the gene. In [134, 98, 49] too, the biological information has been used to improve the gene-ranking and the filtering feature selection, increasing the classification results interpretability and robustness.

Prior knowledge has already been used to analyze microarray data. In [28] the prior

information has been used to analyze the patient survival prediction rather than for classification. The prior information in form of gene sets representing metabolic pathways has been used to summarize functionally related genes in a single variable called supergene by means of Supervised Principal Component Analysis (SPCA) [29]. In [25] the biological information is used to extract the common behavior of functionally related groups, generating *supergenes* like in [28] to be used for feature selection as substitutes of the original gene expressions and applied to the microarray classification rather than regression.

A common trait of all these works is that including some prior biological knowledge led to more interpretable results from a biological viewpoint, easing the scientist's task to formulate new hypotheses.

In this thesis, the biological information integration has been studied in a more extensive model than [25] or [77]. The information has been used to generate a whole hierarchical structure to generate a new set of features that do not substitute the original gene expressions. Moreover, in this work, the tested algorithms have been compared to a wide variety of state of the art classification algorithm on multiple publicly available datasets with a repeatable evaluation procedure recommended in [112].

Two key elements must be considered in including prior biological knowledge in a clustering process. The first on is the knowledge database and the second is how to determine the concept of biological similarity, so to include it in the actual clustering algorithm.

Concerning the knowledge database, in the last years many online and publicly accessible repositories have been implemented and maintained. Some relevant examples are the Gene Ontology database, GO [6], which annotates genes by three categories: Biological Process, Cellular Component, and Molecular Function, the KEGG database [65] which is a database resource for understanding high-level functions and utilities of the biological system, the Molecular signature Database [115] or the DAVID knowledge base [60]. The last two datasets are collections of external knowledge databases, processed and ordered in a computer friendly form, easier to use for data mining application. For a more complete and thorough list of knowledge databases and analysis tools, refer to [3, 13].

The biological similarity definition for the inclusion in the clustering process reduces

to finding an appropriate similarity measure for the biological data, which usually are in a binary or categorical form. The fundamental issue is then to find an appropriate categorical data similarity measure that considers the characteristics of a knowledge database like sparsity and incompleteness of the available data. Examples of categorical measures used to evaluate the similarity in microarrays can be found in [14, 77].

2.2 Feature selection

Feature selection is the process of choosing relevant features from the data set with respect to the task to be performed. In addition to the main goal of obtaining predictive and generalizable classifiers, two additional goals are pursued by feature selection: overcoming the curse of dimensionality and increasing the interpretability. The former is a concept introduced in [10] which is related to the relative amount of available training points and data dimensions. When there are too many dimensions compared to the available sample points, it is easy to find data discriminative patterns which are accidental and not generalizable, falling into data overfitting. The latter concept is related to making sense out of the data. A classification rule involving fewer features is easier to interpret and understand than a classifier using thousands of genes.

The selection of the best feature subset could be a solved problem if the problem would not be unfeasible computationally. Optimum subset selection algorithms already exist [48, 95, 58], which consist in testing every possible feature subset and finally choosing the best one in terms of some cost function.

Being this unfeasible, less computationally expensive methods must be considered. Some of the existing methods are introduced in the following Section using a commonly adopted taxonomy from - [54, 108], which divides the algorithms in three classes: *filters, wrappers* and *embedded*. In Section 4.3, methods adopting a different feature selection strategy are described. They are called *Ensemble methods* and are introduced since some of them are used within this thesis.

Filters are defined by a preprocessing step completely disconnected from the learning phase. A representative example are the ranking criteria such as [46, 129, 54], in which correlation, mutual information or other univariate criteria are used to assign a score to

each feature. Statistical tests like the Student t-test [38] or the Wilcoxon rank-sum test [131, 86] are commonly used as *filters* for feature selection. *Filters* methods typically have a short execution time because they are easy to calculate. The calculation speed is high because no classifier needs to be trained in the filtering phase. The filtering operation usually follows a univariate paradigm: the feature score is determined by the feature values without analyzing possible multivariate interactions. This independent feature evaluation leads to a feature ranking list, from which the top scoring features are chosen to train the classifier. Such univariate paradigm limits the interaction analysis in the classification phase, precluding a posterior interaction discovery by a multivariate classifier. The feature preselection limits the classifier to use features that usually are correlated, due to the univariate nature of the filtering phase selection. Numerous *filter* methods exist in the literature and for more details [75] can be referred as an exhaustive review.

Wrapper methods include the classifier results in the selection process. They search through the possible feature subsets and use the learning algorithm (i.e. the classifier) to evaluate the suitability of each candidate [69]. *Wrappers* have an advantage over *filters*, because they can identify multivariate interactions. However, when dealing with high dimensional data, this processing can be computationally expensive. Different families of *wrapper* algorithms exist, mainly divided into optimal and suboptimal. Optimal methods like extensive search or branch and bound algorithm are infeasible for microarray data [47]. The suboptimal family is then divided into deterministic and stochastic methods. The stochastic group includes evolutionary search algorithms like genetic algorithm [66], genetic programming [43] or NSGAA II [33]. These algorithms have shown good predictive ability [33] thanks to the mutation possibility of the selected feature subset during the search process. A typical framework for the search strategy implies evolutionary steps. At the beginning, many individual solutions are randomly generated to form an initial population and each solution is a feature subset. Each solution is evaluated and, afterwards, the best part of the population is more likely to be used to breed a new generation. In the generation process, the solutions can mutate and mix with some defined probabilities [126]. The process mimics the natural selection process, aiming at having a final popu-

lation well fitted for the classification task. This process, for its own nature, is random and strongly depends on the initial population, which can limit the solution space. That is why, usually, many parallel runs are needed to obtain a final solution. Furthermore, as noticed in [104], the performance of evolutionary tends to degrade when the feature number increases.

The deterministic algorithms group includes many commonly used algorithms like the Sequential Forward Selection (SFS) [130] or Sequential Backward Selection (SBS) [87]. The SFS algorithm starts from an empty set of selected features $Y_0 = \emptyset$, and sequentially adds the feature $\underline{f}_x$ that results in the highest objective function$J(\underline{f}_x, Y_k)$ when combined with the features $Y_k = \{\underline{f}_i | i \texttt{ selected before}\}$ that have already been selected. In this way Y_k is a set composed of k sequentially selected features. The SBS algorithm is the opposite of SFS and starts by selecting all the p available features, $Y_0 = \{\underline{f}_1 \dots \underline{f}_p\}$ and sequentially removing the worst feature from the subset Y_k. The worst feature is the one whose removal from Y_k allows to obtain the highest objective function $J(Y_k \setminus \underline{f}_x)$.

Deterministic search strategies like SFS or SBS always choose the same feature set if the starting conditions do not change, thus ensuring the result replication in successive tests. Within this group, algorithms introducing flexibility in the search have led to very competitive results [112, 39]. Common examples are the Sequential Floating Forward Selection algorithm (SFFS) [104], which is an evolution of SFS, allowing a backward correction stage in the search process, or the Improved Sequential Floating Forward Selection [94] which additionally includes a replacing step. Details about SFFS and IFFS are included in Chapter 4, since they are the reference wrapper algorithms adopted for feature selection.

Finally, *embedded* methods incorporate feature selection as part of the training phase. Examples are decision trees [24] or LASSO (Least Absolute Shrinkage and Selection Operator) [121, 135] or random forests [22, 23, 32]. These feature selection methods are strictly dependent on the chosen classifier and are not suited for the aim of this thesis, which is to propose a more general framework, applicable to more than one classifier. More details about embedded methods can be found in [38, 121, 58].

2.2.1 Ensemble learning for feature selection

In statistics and machine learning, ensemble methods use multiple experts to obtain better predictive performance than could be obtained from any of the constituent experts [103]. Ensemble techniques have been used in the literature to improve the stability and performance of feature selection and classification results [8, 136, 72]. In this thesis, a branch of ensemble techniques for classification has been studied to select a proper subset of classifiers to merge and produce a global classification outcome for microarray samples.

The idea is to use ensemble learning techniques by merging the prediction of a set of experts to produce a final outcome with improved generalization and precision [72]. The idea behind ensemble learning techniques is that the ensemble prediction ability can improve the one of the single classifiers. Many ensemble methods exist and they are applied in many research fields, for a review of ensemble methods and their applications in bioinformatic refer to [72, 97, 36]. To produce expert ensembles the adopted approaches in the literature can be categorized as follows, from [97]:

- Using different feature subsets for different experts
- Using different sample subsets for the different experts
- Using different types of classifiers to produce the different experts
- Using different parameters for the same classifier type
- Any combination of the above methods

The ensemble selection methods studied in this thesis pertain to the first category in the list. A set of expert is produced by applying the same classifier trained on different subsets. In Section 4.3, the details about the implemented algorithms are presented. As a general rule, the key elements in determining the expert selection are a fitness function, (e.g. training error), and the notion of diversity [73, 72]. Many diversity measures have been developed to capture how much an expert produces different decisions compared to another. Examples are the k-measure, yuleQ, PCDM [72]. Depending on the chosen diversity measure and the its integration with the fitness function, a plethora of ensemble selection algorithms have been evaluated and reviewed in [72]. Relevant examples are the

Pareto-optimal search [72], the Convex-Hull search in a properly defined search space [72] or the accuracy in diversity algorithm (AID) [8]. Among these, the AID algorithm will be detailed in Section 4.3, because it is the base of all the developed ensemble selection algorithms in this thesis thanks to both its good results in [72], and to its computational cost which eases the implementation [8].

For a deeper discussion on the other ensemble generation categories, [72, 97] can be referred, as well as for the description of popular ensemble methods to improve feature selection stability like bootstrapping [58], boosting [58] and many other variants that have been developed in the literature.

2.3 Classifiers

Sample classification assigns a class label to incoming samples following a precise rule. Such rule is obtained from a learning phase in which the classifier is trained on known data with previously assigned labels. The high dimensionality of the feature set of microarrays is an issue since the vast majority of classifiers are thought for cases in which the sample number is greater than the feature dimension. This problem is usually addressed through a feature selection operation and, sometimes, in developing new classifiers as adapted versions for the new scenario. Some standard algorithms have been more commonly adopted among all the possible techniques [138] and for more detailed surveys refer to [38, 58, 74]. These techniques include from simpler classification rules like K nearest neighbor (KNN) or discriminant analysis, to more complex systems like support vector machines (SVM) or artificial neural networks.

Simpler algorithms like KNN assign a class label depending on the classes of the K closest known samples to the current sample. Usually K is odd and, the classification boundaries are not robust to small training set variations [19]. KNN has been used in many works for microarray analysis [34, 112, 101] with some success. Nevertheless, KNN is a nonlinear classifier, whose boundaries can change importantly depending on the training set, making of KNN more sensitive to training set differences than other, more regularized, classification rules. The reduced robustness of KNN in a small sample scenario like microarrays classification, results in classifiers harder to replicate, thus making its

performances less stable [19, 112].

Another class of classifiers are discriminant analysis methods, which assume that different classes generate data based on different Gaussian distributions. The most popularly adopted algorithm among them is the Linear Discriminant Analysis (LDA). Linear discriminant analysis is also known as the Fisher discriminant, named for its inventor, Sir R. A. Fisher [58]. It is a statistical learning method which finds the best linear combination of features to separate two or more classes, under the Gaussian distribution assumption of the sample classes, moreover it considers that all classes have the same covariance matrix. [58]. This classifier usually obtains good predictive results with stable classification boundary and reliable performance estimation [112, 19, 15] and for these reasons has been chosen as a reference classifier throughout this thesis. Other relevant examples of discriminant analysis classifiers are the Quadratic Discriminant Analysis [38], QDA, which removes the identical covariance matrix assumption and includes quadratic components to the classifier training. It has also been used in microarray classification, [19]. It produces more flexible classifiers than LDA at a price of a higher computational cost. An important mention is also for a whole algorithm family born to overcome LDA limitation when the sample number is smaller than the classifier dimension. To do so, regularization, shrinkage or diagonalization techniques have been applied to evolve the original LDA, and QDA. Some relevant examples are the regularized LDA introduced in Friedman's work in [96], or the diagonalized LDA, DLDA, [39, 137], or the Shrinkage-based DLDA [123], and some application of these methods in the microarray analysis [53, 111]. Further LDA evolutions are known as generalized discriminant analysis [9] and kernel discriminant analysis,[81] is a kernelized version of linear discriminant analysis. Using the kernel trick, LDA is implicitly performed in a new feature space, which allows non-linear mappings to be learned to produce more complex classification boundaries. Such nonlinear classifiers can be very powerful but there is an increased risk of overfitting in a small sample scenario and it may be particularly tricky to obtain generalizable classifiers.

Support vector machines classifier was first proposed by Vapnik and Chervonenkis in [125]. The goal of the algorithm, in case of linearly separable data, is to find the hyperplane which maximizes the shortest distance from a sample point. When data are not linearly

separable, they might be transformed in a higher dimension space where data can be separable. SVM techniques encompass a universe of solution depending on the kernel function used for data transformation. Usually, the more complex the kernel, the more flexible and sensible is the classifier boundary. SVMs are commonly applied techniques and generally obtain good predictive results when linear, polynomial or Gaussian radial basis functions are used as kernels [58, 19, 112]. The SVM classifiers are chosen as a very popular alternative in high-throughput data analysis due to their properties of robustness to overfitting and good generalization properties [58]. Nevertheless, some of the best results are obtained when simple kernel are adopted [19], because the training of nonlinear SVM classifiers is indeed very susceptible to model parameter choices, which are harder to setup properly when only few samples are available [125, 109].

Other relevant classifiers are neural networks, which are a set of connected input/output units, like neurons in a biological neural network. There are many kinds of neural networks and neural networks algorithms, for a detailed introduction refer to [5]. Neural networks algorithms are usually tolerant to noisy data and obtain very good results on the training set when many samples are available. Drawbacks when using this kind of classifiers are the high number of parameters that need to be determined (typically empirically) [138], the long learning time and the possible overfitting due to the high complexity of the algorithm [19].

Conclusions from classifier surveys in the context of microarray analysis agree that better classification accuracy can be obtained with simple and robust methods like LDA or SVM with simple kernels [138, 19, 58, 112] along with a proper feature selection method. Training error estimations done with simpler rules are more likely to be maintained in a validation scenario, with respect to very complex methods estimations [19, 112].

2.4 Multiclass classification

Machine learning techniques have been extensively applied on microarray data for cancer classification, obtaining interesting prediction performances [112, 16, 138]. Most of the work in the field is focused on the binary classification, considering the multiclass case as a straightforward generalization. Different studies suggest however that in the multiclass

case, it is more complicated to obtain good prediction rates, especially when the class number is high and the class distribution is skewed [80, 114, 119, 128]. Many different approaches exist to tackle the problem and the majority proposes a combination of binary classifiers. In [106] a review is presented and it is explained how, among the plethora of developed algorithms, the most commonly adopted approaches are two simple algorithms: the One Against All (OAA) and the One Against One (OAO).

The OAA algorithm is composed of N binary classifiers, one for each sample class. Each classifier tries to separate one class from the rest and the final classification is then performed by predicting using each binary classifier, and choosing the prediction with the highest confidence score. Supposing $f_i(x)$ is the confidence for the i^{th} classifier in assigning the samples x to class i, the OAA decision is defined by Eq. 2.1

$$f(x) = \arg\max_i \left(f_i(x)\right) \tag{2.1}$$

The OAO classifiers builds $N(N-1)$ classifiers, one classifier to distinguish each pair of classes i and j. Let f_{ij} be the classifier where class i corresponds to examples and class j to negative so that $f_{ij} = -f_{ji}$. The final classification can be defined from Eq.2.2.

$$f(x) = \arg\max_i \left(\sum_j f_{ij}(x)\right) \tag{2.2}$$

More recent works about multiclass classification like [119, 128] introduce more sophisticated approaches applying data transmission algorithms for the sample classification. These algorithms are named Error Correcting Output Codes (ECOC) algorithms, which adopt a global approach which compares the sample classification using N binary classifiers as a transmission of N bit codeword over a noisy channel. Each binary classifier is the receiver for one of the N bits of the codeword. The sample class is then assigned depending on the received bits. With this parallelism, data transmission ideas can be adopted to improve the "bit error rate". In detail, redundancy and error correcting codes have been applied for the multiclass scenario.

An ECOC application example is discussed in [119], where recursive Low Density Parity Check (LDPC) codes have been implemented to code the M sample classes in N-

bits codewords. The application of LDPC codes for the multiclass classification is due to their outstanding performances in the data transmission field [92], where they can approximate the Shannon limit. In [118], a recursive way to produce LDPC codes is studied to apply for the multiclass case. The LDPC codes are used as ECOC approach for the multiclass case in different scenarios, showing interesting prediction abilities and highlighting the possibility to improve the classification performances with the adoption of ECOC approaches.

The common ECOC approach consists in building a code table relating each of the M sample classes to a N bit codeword to produce a suitable binary matrix (i.e. Hamming code restrictions or LDPC restrictions). This focus works well for the bit transmission but it does not take into account the aim of the classification task, which is to distinguish among elements pertaining to different classes. In the code matrix generation, all the class partitions are equally suitable, so a binary classifier separating one class from the rest can be chosen in the code table generation with the same probability of choosing a classifier separating three classes, with scarce biological relation, from the rest. This feature can lead to very interesting numerical code tables which however does not translate into the expected error correcting improvements at the time to classify microarray samples [106].

2.5 Discussion

In this state of the art review, the most relevant methods for the key points of the proposed classification framework have been presented. Spanning from the unsupervised learning focused on inferring a hierarchical structure and on producing new features, to the most relevant feature selection techniques, to the applied classification rules and to the adaptation for the multiclass classification.

Many alternative solutions exist to tackle the classification problem in microarrays and it has been chosen to take top performing elements and to combine them. The aim is to develop an organic framework which combines several key elements, tailored on the microarray data characteristics to obtain good predictive classifiers with generalization ability.

For example, it has been chosen to study and implement the feature selection either

with a wrapper algorithm or with an ensemble selection technique. Both these alternatives have greater potential than filter algorithms in finding multivariate interactions between features, and greater flexibility than embedded methods in changing the adopted classification rule and they will both be detailed in Chapter 4.

Regarding the first element of the classification framework, the feature set enhancement, it is explained in Chapter 3. The unsupervised learning techniques showed that they can make order out of the unstructured microarray data and that they can be used to produce a meaningful structure, reducing the noise of the individual genes. Finally, the integration of prior biological knowledge has been studied because it has obtained improved results in several examples in the literature of microarray data analysis.

Chapter 3

Hierarchical data representation

The hierarchical data structure inference is the first step in the presented framework illustrated by Figure 1-2. The aim of this processing step is to obtain an ordered structure from the unordered microarray data and from the structure generation process, a new feature set is extracted and combined with the original gene expression data. As remarked in Section 1.1, the original data are gene expression measurements which suffer from noise and are not endowed with a priori known structure. The noise effect can be minimized by inferring a structure from the numerical data. In such case, low-pass filtering techniques could be applied to correlated genes clusters.

The newly generated features are denominated *metagenes*, since they are aggregate patterns of gene expressions aiming at summarizing the common behavior of similar genes. The metagene notation has appeared with this definition in [61] in the context of the definition of breast cancer predictors. The metagene notion has since then been used to describe an aggregation of multiple gene expressions related by some closeness (numerical and/or biological), like for example in [31, 44].

To produce this new set of features, a hierarchical data representation is obtained through hierarchical clustering. Hierarchical clustering algorithms [93, 58] have been used for organizing and grouping the dataset variables because they offer an easily interpretable description of the data structure, clearly representable with a dendrogram as can be observed in Figure 3-1.

The only requirements to produce a tree structure are to define an aggregation rule

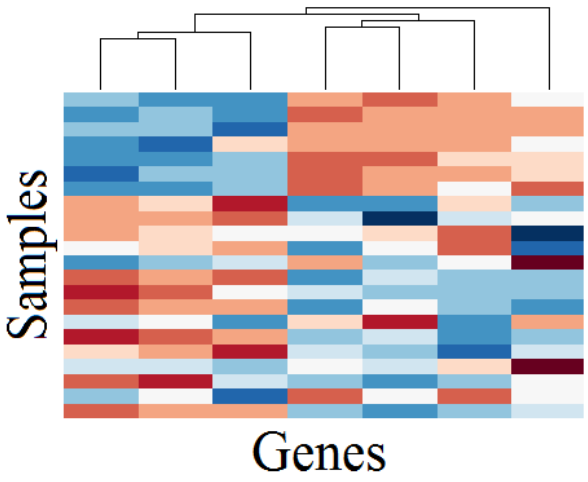

Figure 3-1: Example of a dendrogram representing a hierarchical structure for microarray data.

to form the clusters (i.e. the similarity metric) and to specify a generation rule for the metagene calculation as a combination of the individual genes. In this thesis, the chosen hierarchical clustering process is a bottom-up, pairwise approach based on Lee's work in [78], where an adaptive method for multi-scale representation and eigen-analysis of data called *Treelets* is presented. Treelets have been used as analysis tool to infer a hierarchical data structure both to analyze gene expression data [78, 17] and to create order to unstructured data in other research fields too [124]. Treelets has proven to be a powerful method to extract an underlying unknown structure from the data and this is why several variants of the original method from [78] have been tried in this thesis to analyze their potential in generating useful metagenes for classification.

The original implementation from [78] has been tested, as well as a set of alternatives to study possible improvements to the original algorithm. Among the infinite number of possibilities, a set of focused modifications have been chosen looking at previous results from [12, 71], and selecting those setups that may lead to better performances and that have a feasible computational implementation.

In Section 3.1, the original Treelets algorithm is introduced, while in Section 3.2, a variant adopting Euclidean distance instead of the Pearson correlation in the clustering process is described. In Section 3.3, a different modification is studied by applying Haar wavelet transform as the metagene generation process instead of the original Principal

Component Analysis decomposition. Such a modification simplifies the generation process by using constant combination weights to generate the metagene expression.

All the studied modifications to the original Treelets algorithm have been tested and compared, to analyze possible benefits for the predictive ability. The experiments setup and the results are included in Chapter 4.

3.1 Treelets clustering

The first studied technique to infer a hierarchical structure from gene expression data is the original Treelets algorithm, thus it has been chosen to call it *Treelets* clustering.

The clustering tree is produced in a bottom-up pairwise approach. At each level: the two most similar features are chosen and replaced by two features, a coarse-grained approximation feature and a residual detail feature. Taking advantage of this multi-scale data representation, with *Treelets* clustering, at each iteration, the two features are replaced by one feature only, the approximation one, while the residual detail feature is discarded because it represents what is different between the two merged features. This new approximation feature is called metagene and it is obtained as a linear combination of the two joined features. Afterwards, the newly created metagene is used as a feature to be compared in the next iterations. If the initial condition is a feature set of p individual genes, the final outcome from the feature set enhancement process is a metagene set of $p-1$ *metagenes*, one for each node in the hierarchical tree. This metagene set is then added to the initial feature set.

In Figure 3-2, a pseudo code for the hierarchical clustering and the metagene generation process is detailed. It is a general algorithm, which can be used to describe any of the implemented algorithm variants. What differentiates a clustering algorithm from another in this framework are either the similarity distance $d(\underline{f}_a, \underline{f}_b)$ or the metagene generation process $g(\underline{f}_a, \underline{f}_b)$.

The Pearson correlation is the similarity metric used to evaluate pairwise relations between features in the original *Treelets* clustering. It is a normalized correlation measure between two features and it is defined as in Eq. 3.1 for generic feature vectors $\underline{f}_a$ and $\underline{f}_b$. Each feature vector represents the samples of a specific feature, that is a gene or

Original feature set $\underline{\underline{G}}_0 = \{\underline{g}_1, \ldots, \underline{g}_p\}$
Active feature set $\underline{\underline{F}} = \underline{\underline{G}}_0$
Metagene set $\underline{\underline{M}} = \emptyset$

For $i = 1 : p\text{-}1$

1. *Calculate pairwise similarity metric* $d(\underline{f}_a, \underline{f}_b)$ *for all features in* $\underline{\underline{F}}$

2. *Find* $a,b : d(\underline{f}_a, \underline{f}_b) = \max(d(\cdot,\cdot))$

3. *New metagene* $\underline{m}_i = g(\underline{f}_a, \underline{f}_b)$ *generation:*

$$\underline{m}_i = \alpha_a \, \underline{f}_a + \alpha_b \, \underline{f}_b = \sum_{i=1}^{p} \beta_i \, \underline{g}_i; \quad \underline{\alpha} \in \Re^2 \quad \underline{\beta} \in \Re^p$$

   Each metagene can be seen either as a combination of its two child features $\{\underline{f}_a, \underline{f}_b\}$ *or as a linear combination of all the original features* $\underline{g}_i$

4. *Add the new metagene to the active feature set*

$$\underline{\underline{F}} := \underline{\underline{F}} \cup \{\underline{m}_i\}$$

5. *Remove the two features* $\underline{f}_a, \underline{f}_b$ *from the active feature set*

$$\underline{\underline{F}} := \underline{\underline{F}} \backslash \{\underline{f}_a, \underline{f}_b\}$$

6. *Join the metagene* $\underline{m}_i$ *to the metagene set*

$$\underline{\underline{M}} := \underline{\underline{M}} \cup \{\underline{m}_i\}$$

end

Define the new expanded feature set: $\underline{\underline{F}} = \underline{\underline{G}}_0 \cup \underline{\underline{M}}$ *as the union of metagenes and original gene expression profiles.*

Figure 3-2: General hierarchical clustering algorithm adopted in this thesis.

metagene.

$$d(\underline{f}_a, \underline{f}_b) = \frac{< \underline{f}_a, \underline{f}_b >}{\|\underline{f}_a\| \|\underline{f}_b\|} \tag{3.1}$$

The Pearson correlation $d(\underline{f}_a, \underline{f}_b) \in [-1, \ 1]$, measures the scalar product between two features (i.e. numerator in Eq. 3.1), divided by the product of l^2 norm of the two involved features. This criterion measures the profile-shape similarity of two features so that it is

invariant to a scaling factor: $d(\underline{f}_a, \underline{f}_b) = d\left(k\underline{f}_a, \underline{f}_b\right)$. The Pearson correlation assumes value equal to 1 when two features have the exact same pattern, while a correlation value of -1 implies a perfect profile anticorrelation, defining the farthest possible point in the similarity space spanned by the Pearson correlation.

About the *metagene* generation process with *Treelets*, the clustering process produces *metagenes* taking advantage of the multi-scale representation introduced in [78]. PCA can be described as a data representation and it is mathematically described as a change of basis in a vectorial space. It has been demonstrated that PCA can achieve a compact representation of the analyzed data. In its original formulation, PCA is a global feature transformation where the new representation is obtained as linear combination of its child features, but also as a linear combination of all the original components (in the microarray case it would be a combination of all the thousands of gene expressions). In *Treelets* clustering, PCA is instead used locally, inside each clustering step to produce a local data transformation, thus combining only two features at a time. In detail, for each node in the tree, a local Principal Component Analysis (PCA) [64] is applied on the child features. By this process, a hierarchical tree with multi-scale data representation is obtained. In each iteration, the local PCA calculates a Jacobi rotation on the two features $\underline{f}_a, \underline{f}_b$ [51] as in Eq. 3.2.

$$\begin{aligned} m &= \underline{f}_a \cos\theta_L + \underline{f}_b \sin\theta_L \\ d &= \underline{f}_a \cos\theta_L - \underline{f}_b \sin\theta_L \end{aligned} \tag{3.2}$$

In Eq. 3.2, θ_L is the rotation angle which decorrelates the two features $\underline{f}_a$ and $\underline{f}_b$ so that the two output features m and d will have 0 correlation. The m feature is the coarse-grained approximation feature in [78] (i.e the first principal component) and it is chosen as metagene in the *Treelets* clustering. On the other hand the d feature is the residual detail feature, which is not taken into account for further processing. The fact that the local PCA can be seen as a Jacobi rotation is visualized in Figure 3-3 in a case of two very similar features. On the left, the initial two-dimensional space formed by the original features $\underline{f}_a$ and $\underline{f}_b$ is visualized. On the right hand side, instead, data are visualized in the coordinate system of the two principal components. As can be seen, in this case, the

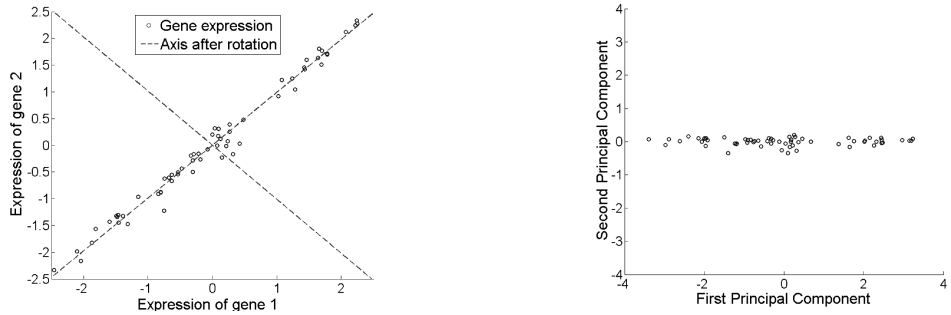

Figure 3-3: Example of how local PCA can be represented as a coordinate system rotation and how the first component well represents two similar features.

first principal component (the m feature chosen as metagene) represents well the common behavior of the two analyzed features.

A note about the linear coefficients calculated with PCA in the metagene creation algorithm in Figure 3-2: each metagene can be seen as a linear combination of all the individual genes, and PCA is an unitary transform so that $\|\underline{\beta}\|_2 = 1$. This l^2 norm equal to 1 states that PCA is an energy conservative transformation and this effect translates into producing metagenes of growing dynamic range as the number of represented genes grows.

The final output of the *Treelets* clustering is a hierarchical tree with a *metagene* for each node. The original feature set is enhanced by the addition of new features able to summarize the common behavior of gene clusters. This characteristic can reduce the noise thanks to the low-pass filtering effect from the linear combination of similar features.

3.2 Euclidean clustering

The second *metagene* creation technique is called *Euclidean* clustering. It adopts an iterative process like the one explained in Figure 3-2, but it introduces changes in the similarity metric $d(,)$ and in the metagene generation rule $g(\underline{f}_a, \underline{f}_b)$ with respect to the *Treelets* clustering technique.

The similarity metric adopted in the *Euclidean* clustering is the negative Euclidean distance between features, defined in Eq. 3.3. The negative Euclidean distance has a maximum in zero, when two features are equal. It has been chosen as alternative to the

Initial feature set of three equal genes

$$\underline{\underline{F}}_0 = \{\underline{f}_1, \underline{f}_2, \underline{f}_3\} \text{ with} \underline{f}_1 = \underline{f}_2 = \underline{f}_3$$

Two metagenes will be created

1. *metagene* $\underline{m}_1$ *joining* $\underline{f}_1$ *and* $\underline{f}_2$

$$\underline{m}_1 = \sqrt{1/2}\underline{f}_1 + \sqrt{1/2}\underline{f}_2$$
$$\underline{m}_{1scaled} = 1/2\underline{f}_1 + 1/2\underline{f}_2$$

2. *metagene* $\underline{m}_2$ *joining* $\underline{m}_1$ *and* $\underline{f}_3$

$$\underline{m}_2 = \sqrt{2/3}\underline{m}_1 + \sqrt{1/3}\underline{f}_3$$
$$\underline{m}_2 = \sqrt{1/3}\underline{f}_1 + \sqrt{1/3}\underline{f}_2 + \sqrt{1/3}\underline{f}_3$$
$$\underline{m}_{2scaled} = 1/3\underline{f}_1 + 1/3\underline{f}_2 + 1/3\underline{f}_3$$

Scaled versions $\underline{m}_{1scaled}$ *and* $\underline{m}_{2scaled}$ *the scaled versions are used to define the similarity with the Euclidean distance because they preserve the components dynamics. These versions are then used as metagenes, enhancing the original feature set.*

The non scaled versions, $\underline{m}_1$ *and* $\underline{m}_2$*, are used to compute the metagene from the two child features with PCA as they preserve the energy distribution among the elementary components.*

Figure 3-4: Example of metagene creation with *Euclidean* clustering.

Pearson correlation because the Euclidean distance can measure the point-wise closeness rather than the profile-shape similarity.

$$d\left(\underline{f}_a, \underline{f}_b\right) = - \left\|\underline{f}_a - \underline{f}_b\right\|_2 \tag{3.3}$$

The Euclidean distance has a different point of view with respect to the correlation measure adopted in *Treelets* clustering and might be able to extract similarity related to the actual gene expressions rather than to their pattern.

The change in the similarity measure implies a modification in the metagene generation rule $g(\underline{f}_a, \underline{f}_b)$. Due to the PCA transformation, which is energy conservative, a scaling factor is introduced on the produced metagenes. The obtained metagenes with *Treelets* clustering are scaled weighted averages of the genes, with a scale factor greater than 1.

To properly compare genes expression values (and not their shape as with the Pearson correlation) with *metagenes*, the latter must be a pure weighted average of the genes. An illustrative example of how the *metagene* creation process is performed with *Euclidean* clustering is presented in Figure 3-4. In this figure, a toy example with an initial feature set of three equal genes is shown. It can be seen how the metagenes obtained with the sole PCA transformation are scaled weighted average of the genes, moreover with a scale factor proportional to the number of genes. This scaling factor is not an issue when the Pearson correlation is concerned, but it affects the Euclidean distance measurement.

To obtain a proper comparison between genes and metagenes, when a metagene $\underline{m}_x$ is created, two versions of it are used. The first one is the same as in the *Treelets* case from the PCA transformation, while the second is a scaled version of the former $\underline{m}_{xscaled} = \underline{m}_x / \|\underline{\beta}\|_1$ The scaled version $\underline{m}_{xscaled}$ results to be a pure weighted average of the genes and it is used in the pairwise similarity measurement as metagene. The non scaled version, instead, is maintained and it is used when a new metagene is built from $\underline{m}_x$ to preserve the energy distribution among the individual component, as can be observed in Figure 3-4.

The differences in the similarity measure and in the generation rule lead to a different metagene set with respect to the *Treelets* clustering. To better visualize the differences between the *Treelets* clustering and *Euclidean* clustering, Figure 3-5 is introduced. There, it can be observed how the dendrograms are quite different even if only 4 initial features are considered. As expected, in *Treelets* clustering, the profile-shape prevails in defining the merging features, while in *Euclidean* clustering, the point-wise distance rules the process. It can be observed how, out of the three metagenes $\underline{m}_1$ $\underline{m}_2$ and $\underline{m}_3$, only $\underline{m}_3$ has the same profile in both the clustering techniques. This is an expected result because the final combination includes only three genes and the energy distribution among the individual components is determined in the same way by the two algorithms.

3.3 Haar wavelet for clustering

The possibility to change, simplifying, the metagene generation process inside the hierarchical clustering process has been evaluated by introducing Haar wavelet decomposition

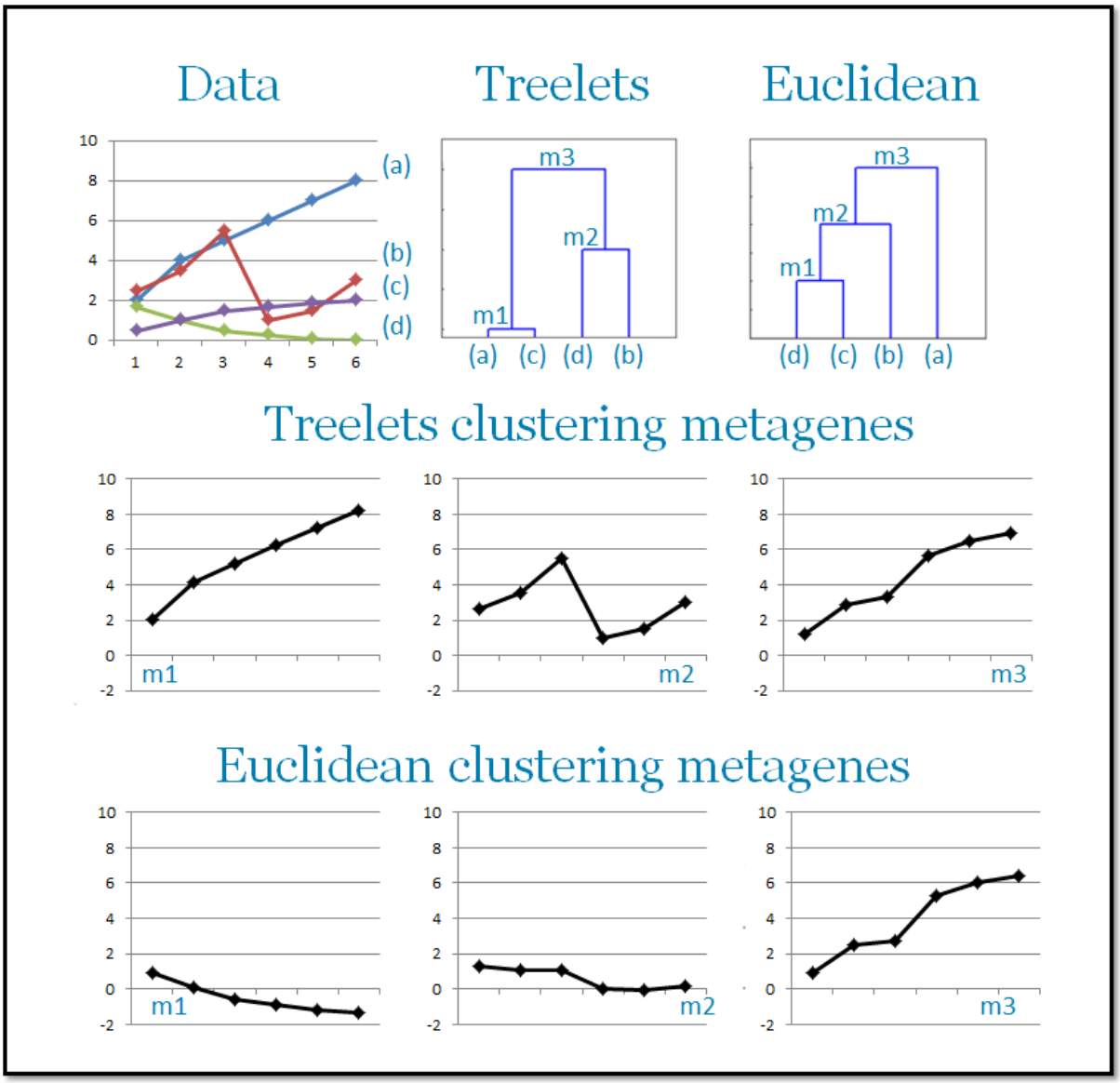

Figure 3-5: Example of metagene construction process differences between Treelets and Euclidean clustering. The vertical axis represent the gene expression value, while the bullets in the horizontal axis are the different samples. In the first row the original data and the two obtained clustering trees are shown. In the second and third rows, the created metagenes with Treelets or Euclidean clustering are represented.

[55] to define the metagene generation criterion $g(\cdot,\cdot)$. In the Treelets original version, each metagene is produced with a local PCA on the two merged features [78]. The studied alternative proposes to substitute the PCA with a Haar transformation on the two merged features.

The main difference between the two rules is in the linear combination weight assignment. Whether with PCA, the linear weights can be anything constrained to $\|\alpha\|_2 = 1$, being α the two dimensional coefficient vector, with the Haar wavelet transformation, the weights are fixed and equal to $\sqrt{2}/2$. Such weighting difference eases the structure information storage and retrieval, because the only needed information is the merging order, without caring about the coefficient values. A side effect of the Haar basis transform is the generation of a completely different metagene set.

3.4 Discussion

In this Chapter, techniques to infer a hierarchical structure from microarray data have been described. The produced output are binary trees associating genes in different orders and producing different sets of metagenes.

This processing step is done to obtain new features more able to summarize the behavior of related genes. To evaluate if this metagene generation process is useful and to decide which of the proposed alternative algorithms is the best, the inclusion of the metagenes in a classification framework must be done.

In the following Chapter, the proposed microarray classification framework is introduced with all the needed details to adapt the process to the microarray data characteristics.Moreover, all the metagene generation algorithms have been uniformly compared among them and with relevant state of the art alternatives. The results are measured in terms of predictive ability and robustness.

Chapter 4

Feature selection for binary classification

In Chapter 3, the metagene generation process has been used to enrich the gene expressions with a whole new set of features called metagenes. Metagenes can improve the classification ability since they expand the available feature space and because they can extract common traits of gene clusters, filtering out the residual noise. After the feature set enrichment, the main problem is to deal with the high dimensionality of the feature set, choosing an appropriate subset for classification. This task is even more compelling due to the increased sample scarcity condition since the total feature number has almost doubled. The feature selection task is needed to overcome the curse of dimensionality [10] by selecting a small amount of relevant features or, at least, by excluding a vast majority of irrelevant features, thereby improving generalization properties and the interpretability of the output prediction model.

The objective of this chapter is to present the studied classification frameworks for microarray classification and to compare them with the state of the art. As a general overview, to produce a final prediction model for new samples it has been chosen to use two fundamental building blocks: the metagene generation step and a subsequent feature selection stage, whose output is a prediction model for classification. The metagene generation process has been covered in Chapter 3, while in this chapter the two developed feature selection approaches are detailed in Sections 4.1 and 4.3. The first one aims at

developing and tuning a wrapper feature selection algorithm allowing mutation of previous choices, good stability and good scalability deriving from a deterministic approach. Several alternatives have been studied by introducing specific elements in the search process to deal with the small-sample scenario in microarray datasets. The second studied feature selection strategy is described in Section 4.3 and it consists in applying ensemble feature selection techniques for the specific case of microarray data.

In both cases, wrapper and ensemble feature selection, the opportunity to include metagenes in the selection process is evaluated, as well as a comparison among the different studied alternatives and with state of the art techniques is performed.

4.1 Wrapper feature selection

Wrapper feature selection has been chosen because of its flexibility in choosing features considering also multivariate relationships among them [46, 58]. Among the plethora of existing methods, we focus on evolutions of the sequential forward selection method, SFS, [130] because they add flexibility in the search process and in particular one algorithm has been implemented

- **Improved Sequential Floating Forward Selection (IFFS) [94]:** it is a sequential algorithm that allows backtracking after each sequential step to identify a better subset: after adding a feature to the subset, the algorithm looks for the possible benefits of eliminating one or more features. Furthermore, it introduces a replacing stage in case that backtracking does not improve the classification performance. The price to pay is a sensible increase of execution time in the replacing phase that does not grow linearly with the feature subset dimension. In Figure 4-1, the flowchart for the IFFS algorithm is presented.

4.1.1 The IFFS algorithm

The IFFS algorithm starts with an empty set and ends the search when a threshold value θ is reached. This threshold value is reached either because the selected number of features is equal to the desired maximum or because the algorithm is in a loop and has overcome

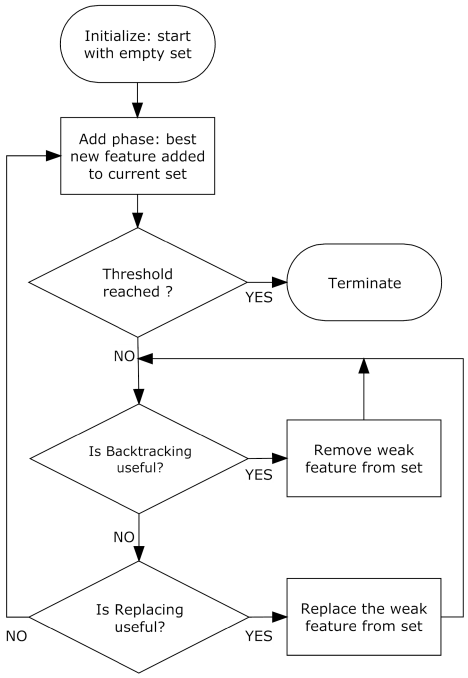

Figure 4-1: The IFFS framework with the three phases of addition, backtracking and replacing.

the maximum allowed iteration number. In the initialization block, the system simply sets to zero the cardinality of the feature set and begins a new search. After the initialization, IFFS selection process enters in a loop of tasks:

1. *Add Phase.* The algorithm looks for the best feature to add to the current feature set. It tests all the features that have not yet been selected one by one. The test implies the expansion of a current feature set with a new feature, then a classifier is trained and the corresponding classification score $J(\cdot)$ is calculated. After a comprehensive test of all the candidate features, the one obtaining the best $J(\cdot)$ score is included to the current feature set.

2. *Backtracking phase.* This is the block that differentiates IFFS from the greedy forward selection algorithm [104]. In this phase, the algorithm does a backtracking of its decisions and allows eliminating one of the already selected features. As a result, the feature subset has more evolutionary possibilities. In this step, one backtracking iteration is performed: it evaluates the potential benefits of removing one feature from the current subset. To this end, one feature is removed and the classification performances are then evaluated. This block identifies the weakest feature in the

subset (i.e. the feature whose elimination implies the minimum performance loss, or the maximum performance gain) and decides whether eliminating it or not. If the elimination implies no performance improvement, the algorithm keeps the feature in the current subset and goes to the *Add phase* again. Otherwise, if the classification gets better by eliminating the feature, the weak feature is removed from the subset and the algorithm starts a new *Backtracking phase* to evaluate if more than one feature can be eliminated.

3. *Replacing Phase.* Here, the algorithm looks for possible improvements by substituting one of the selected features. One feature at a time is removed from the current set and then, the best substitute among all the remaining features is found analyzing all features one at a time like in the *Add phase*. All the substitutions are ranked and, if the best substitution is useful (i.e. the score $J(\cdot)$ value with the substitution is better than without), the current set is updated and the algorithm then goes back to a *Backtracking phase*. If the replacing has not found any positive substitution, the subset remains unchanged and the algorithm goes to the *Add phase*.

In this thesis we chose to systematically adopt IFFS as wrapper feature selection method because it consistently obtained better results in feature selection when compared to simpler wrapper alternatives like the Sequential Floating Forward Selection, SFFS [104], algorithm, as found in[94, 17]. Nevertheless, algorithms like SFFS [104] could be easily applied to perform a much less computationally demanding feature selection than IFFS, but without guaranteeing to reach the predictive accuracy of IFFS [94, 17].

4.1.2 Fitness measure definition and feature ranking criteria

Once chosen the wrapper algorithm, its application to microarray data must consider the data characteristics and introduce elements to ensure that the feature selection is properly done. The main data characteristic to consider is the small-sample and high feature number present in microarrays which limits the number of features to be chosen and that introduces concerns about the measure which should be adopted as fitness $J(\cdot)$ in the search process. To adapt the feature selection process to microarrays, some novelty elements have been introduced in the definition of the fitness measure: $J(\cdot)$ score, which

measures the prediction ability of any proposed classifier. In wrappers, the classifier is iteratively applied throughout the selection process. The reference classifier in this thesis is the Linear Discriminant Analysis (LDA), due to its good properties [19, 112]. During the feature selection, then, LDA is applied multiple times and, in every case, a $J(\cdot)$ score is extracted from the classification results.

The most common way to measure the $J(\cdot)$ value in the literature is to use a classification error rate estimation. When sample scarcity is not a problem, usually the error rate is estimated on thousands of samples and a reliable (repeatable) evaluation is obtained. In the current microarray classification scenario, no such sample abundance is available, so different error estimation techniques have been developed [58, 19]. Among the possible alternatives, the stratified cross validation estimator has been used during the training phase. The cross validation estimation implies iterating several times the same process:

1. Divide the dataset in two parts, a training set usually composed of the majority of the available samples, and an internal test set.
2. Train the classifier on the training set.
3. Apply the obtained classifier to predict the internal test set.
4. Extract the results.

After iterating the process N times, the global results are obtained as the mean of the individual iteration outcomes. A common example is the 10-Fold cross validation, a 10 iterations process in which, for each iteration, the internal validation set is composed of the 10% of the available samples, while the training set is the remaining 90%. The cross-validation is an unbiased estimator of the error rate, but, in case of sample scarcity, it can show high estimation variance [19]. In order to obtain more robust error estimations, repeated runs of cross validation are performed, and the dataset partition is obtained in a random but stratified way. The stratified word means that the partition tries to maintain the same class distribution between the training set and the internal validation set.

Due to the microarray data characteristic involving few samples and many dimensions, a $J(\cdot)$ criterion based only on the error rate may not be enough in ranking features. Indeed, it is common to have a group of features with the same error rate, from which

only one feature must be selected. Furthermore, slight error differences can derive from an unfortunate data partition in the cross validation phase. For example, if a specific sample is included in the validation set in more iterations than another, it gains more weight in the error rate calculation. In this way, an apparently higher error rate might not reflect the actual prediction performance.

The reliability parameter

To overcome the error rate limitation as a fitness estimator in a small sample scenario, an additional value is introduced in this thesis to define the $J(\cdot)$ score: the reliability. It takes into account that a feature obtaining well separated classes is better than a feature in which the two classes are separated only by a very thin margin. It does so by trying to include the univariate t-tests concept (i.e. give more importance to features having large mean class separation and small intra class variance) to a multivariate scenario.

The reliability parameter r quantifies the estimation goodness as a weighted sum of sample distances from the classification boundary. It is calculated on the test set samples and the final value is the mean through the cross validation iterations. The reliability is calculated inside a cross-validation iteration for a two-class problem. It is defined in Eq. (4.1), where n_{test} is the test set dimension, c_l is the class of sample l (it can be 1 or 2), and $p(c_l)$ is the probability of class c_l in the test set. The value d_l is the Euclidean distance of sample l from the classifier boundary with positive sign in case of correct classification or negative sign otherwise.

$$r = \frac{1}{n_{test} \cdot \hat{\sigma}_d} \sum_{l=1}^{n_{test}} \frac{d_l}{p(c_l)} \tag{4.1}$$

Finally, $\hat{\sigma}_d = \sqrt{\frac{\hat{\sigma}_1}{n_1} + \frac{\hat{\sigma}_2}{n_2}}$, is an estimation of intra class variance of the sample distances from the classification boundary. In order to get a more complete estimation, the intra-class variance is estimated using all the samples from both the training and the test sets; n_1 and n_2 are the number of samples in class 1 and 2 respectively. The $\hat{\sigma}_d$ definition recalls the independent two-sample t-test denominator with classes of different size and variance, as it is the most general case for a two-class problem. In detail $\hat{\sigma}_1$ and $\hat{\sigma}_2$ are the estimated variances of sample distance from boundary for all samples of class 1 and

2 respectively. Dividing by $\hat{\sigma}_d$ guarantees that r is invariant to a scaling factor, thus obtaining the same value for metagenes that are perfect scale replicas of genes. Dividing by $p(c_l)$ assigns to each class the same relative weight and it is useful when the test set distribution is highly skewed. Reliability value, $r \in [-\infty\ \infty]$, is positively influenced by large mean class separation in the perpendicular direction to the classifier boundary, and by small intra class data variance. It is penalized by a factor proportional to error value so that greater errors produce greater penalties, allowing discrimination among features with equal error rates as visualized in Figure 4-2 where two classifiers with equal error rate are compared: in both cases the error rate is 0 but, in the left part, classes are well separated from the boundary in two close clusters while in the right side of Figure 4-2, the two classes are very close to each other with many samples almost over the boundary.

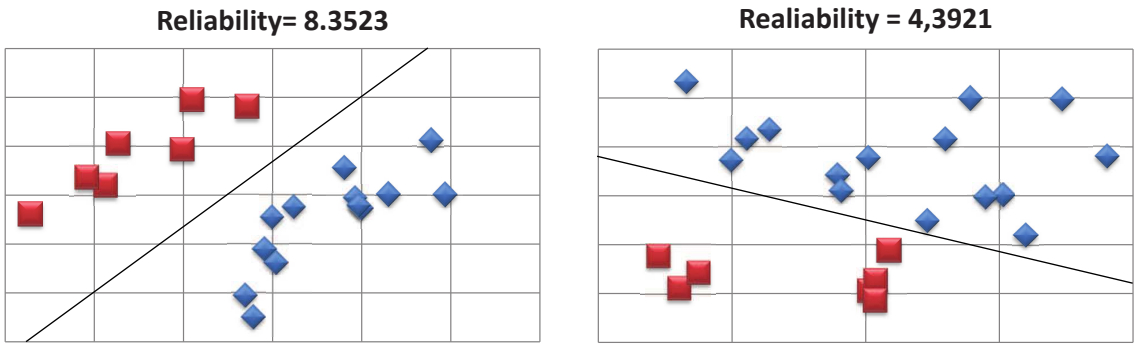

Figure 4-2: Example of how the reliability parameter can discriminate between two classifiers with equal error rate. In both cases the error rate is 0 but, in the left part, classes are well separated, while in the right part, the classes are very close to each other.

$J(\cdot)$ score calculation

The final $J(\cdot)$ value is determined by both the error rate and the reliability value along the cross validation iterations. A classifier is ranked to be better than another if its score is higher. The score definition is a key point for the feature selection operation to perform the best selection.

The first studied scoring scheme is introduced in [16] and it is a two-step ranking process. Features are firstly sorted by increasing error rate value, thereafter, reliability

is taken into account to discriminate among features sharing the same error rate. This criterion produces a lexicographic sorting of the features, in which the reliability parameter has a secondary role. Applying this ranking rule to the analysis of small, publicly available microarray datasets has produced interesting results [16], reducing the number of needed features to get a 0 estimated error rate with respect to state of the art alternatives. Nevertheless, the lexicographic sorting is by nature a rigid scheme. The derived benefits by the introduction of the reliability parameter can fade when the test set cardinality grows (either because the dataset has a fair number of samples or because more cross validation iterations are implied). In such a case, the probability to use the reliability information is reduced since it is harder to get features with the same error rate. This hypersensitivity of the lexicographic sorting to small error rate differences has showed to be a drawback when analyzing microarray datasets with higher number of samples, like those in the Microarray quality control study phase II (MAQC) [112].

To overcome that limitation and to make better use of the reliability information, two additional scoring rules have been designed. Both of them unify in a scalar value the two sources of information. The proposed score definition rules are influenced both by error rate and reliability, allowing a feature with higher reliability and slightly higher error rate to be considered better than another with poor reliability but with a smaller error rate. This flexibility is useful for small sample datasets like microarrays. It takes into account the seen data distribution from the classifier point of view, thus giving a higher score to features showing high mean class separation and small intra-class variation. The first of the two new scoring rules compares features in terms of the reliability value, properly penalized depending on the estimated error rate. The aim of this penalization is to introduce a fixed penalization factor to the reliability value for a constant error difference. Such a behavior can be obtained introducing an exponential penalization to the reliability value. For each feature, the $J(e, r)$ score is obtained as in Eq. 4.2, where r is the reliability value, e is the error rate value, and δ is a penalization parameter.

$$J \;=\; r \cdot \exp\left(-sign(r) \cdot \frac{100}{\delta} \cdot e\right) \tag{4.2}$$

$J(e, r)$ is a product of the reliability value with a penalization coefficient ≤ 1 and exponen-

tial behavior depending on the error rate value. The $-sign(r)$ factor in the exponent has been included to highly penalize features with negative reliability, while the δ parameter defines the steepness of the penalization. The δ value defines the e^{-1} penalization interval: between two features with equal reliability value, a $\delta\%$ difference in the error rate induces a e^{-1} penalization in the final score. So, when δ is small, the dominant parameter is the error rate (an extreme case is when $\delta \to 0$ the reliability has no influence at all), while when δ is large the dominant parameter becomes the reliability (when $\delta \to \infty$ the error rate is not taken into account).

The second scoring rule is a linear combination of error rate and normalized reliability. The linear combination score is defined by Eq. 4.3. It is a weighted sum of error rate e and normalized reliability value $r_n = (r - \min(r))/(\max(r) - \min(r))$. The δ parameter is bounded between 0 and 1 and it defines the relative weight of reliability with respect to the error rate.

$$J \; = \; \delta \cdot r_n \;\; + (1 - \delta) \cdot (1 - e) \;\; \delta \in [0, 1] \tag{4.3}$$

This simple scoring rule allows a more flexible comparison of reliability values among features with different error rates. It shows a linear trend both in the error rate and in the reliability direction. The main change with respect to the former exponential penalization scoring is that, here, a constant penalization is added (not multiplied) to a constant error rate increase. Figure 4-3 illustrates the three score functions. It shows the score value assigned to points in the Error-Reliability space for the exponential combination, the linear combination and the lexicographic sorting case of [16].

From Figure 4-3 it can be observed how in the exponential combination case, the score has an exponential decrease along the Error dimension, while it has a linear trend in the Reliability dimension. For the linear combination case, the scores lie onto a rotated plane in the space with the rotation axis passing through the $(0, 1)$ and $(1, 0)$ points. It shows linear trends in both dimensions (Error and Reliability) with slopes equal to $1 - \delta$ and δ respectively. The lexicographic scoring is here visualized in a very coarse scenario in which only 10 different error values are allowed (imagine a test set composed of 10 samples only) in order to visualize its behavior. It is a stairway-like surface showing how the main dimension is the Error value. Only if two features share the same error value the

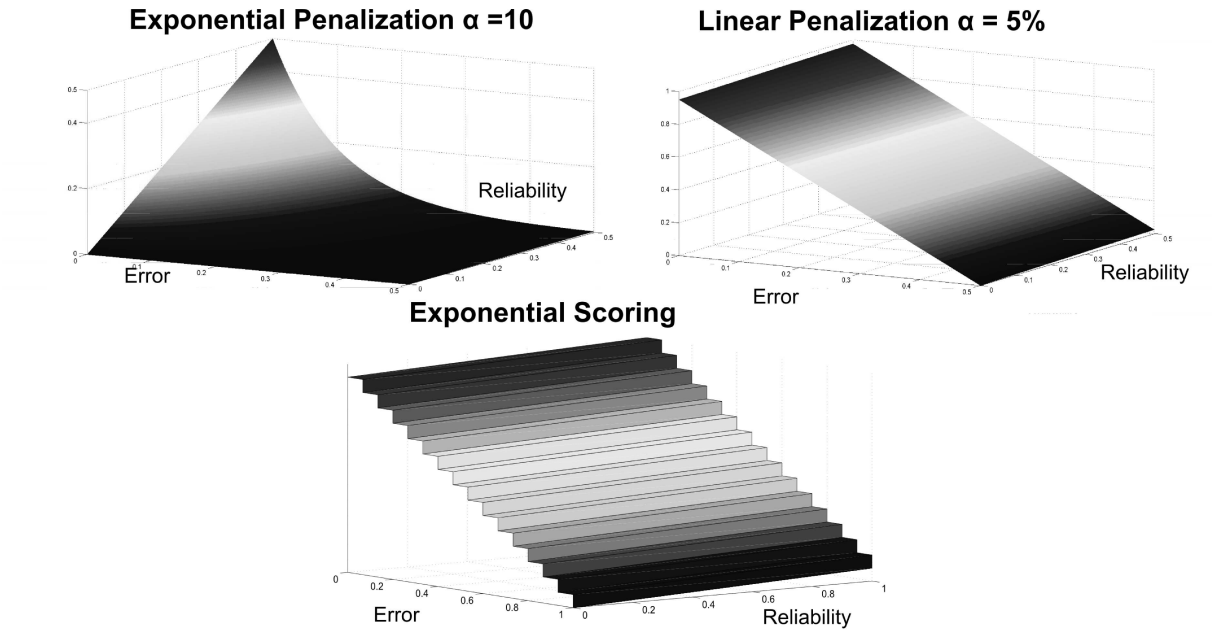

Figure 4-3: Score surfaces in the error-reliability space depending on the three scoring rules.

reliability is taken into account (linear trend in the reliability direction), otherwise the score of a feature with smaller error rate is higher, regardless of the reliability value. From Figure 4-3 it can be observed how both the scoring rules combining reliability and error rate radically change the score surface. From a stairway-like surface (with discontinuities among error rate values), the score surface is transformed to a continuous surface in which the reliability values have more decisional power. This change is more important in a case with many test samples, because in such a scenario, the lexicographic scoring would be like a stairway with many small steps, thus making the reliability parameter almost useless. Furthermore it would be extremely sensible to small error rate changes while the new scoring methods are able to mix error rate and reliability in a more flexible form.

As can be observed, the definitions of exponential combination and of linear combination in Eq. 4.2 and Eq. 4.3 depend both on a parameter (i.e. δ) that must be previously chosen. This parameter dependence implies an optimization study to choose the best δ value for classification. Thus, both the linear combination and the exponential penalization rules define a whole set of alternatives. It will be shown in Section 4.2 how the predictive ability also depends on the chosen parameter value too.

4.2 Experimental results for wrapper feature selection

In this section, the classification framework adopting the wrapper feature selection process is evaluated to determine the best setup considering all the introduced elements. The evaluation purpose is multiple: on one side, the usefulness of introducing the hierarchical structure and the metagenes is assessed and, on the other side, an evaluation protocol is defined to find the best setup in terms of clustering distance $f(\cdot,\cdot)$ (i.e. to compare between Treelets and Euclidean clustering from Chapter 3) and fitness measure (i.e. ranking score rules from 4.1.2).

The evaluation is performed by applying all the algorithms to analyze a data cohort of publicly available data from MAQC study [112] and are compared by means of predictive ability. Once the best alternative is chosen by defining the clustering type, Treelets or Euclidean, and the ranking score rule, among lexicographic sorting, linear combination or exponential penalization, additional studies are performed to assess the statistical robustness of the obtained results with Monte Carlo simulations and analyses on synthetic datasets.

In 4.2.3 and 4.2.4, different sources of variation are studied and the top performing algorithm is compared to alternatives changing the metagene generation rule $g(\cdot,\cdot)$ or the wrapper classifier. In 4.2.3, Haar wavelet is used instead of PCA to generate the metagenes, while in 4.2.4, linear SVM is chosen as classifier rather than LDA. In both cases, results on publicly available data are compared to their corespondent applying PCA with LDA classifier, chosen from the analysis in 4.2.2.

4.2.1 Dataset cohort

The analyzed data are a set of high quality datasets, provided by the Micro Array Quality Control study phase II as a common ground to test classification algorithms [112]. The analyzed data are a subset of the provided datasets by the MAQC II consortium: six datasets containing 13 preclinical and clinical endpoints coded A through M; for more information refer to [112]. Each endpoint corresponds to a different sample classification so that the same dataset can be classified following different criteria (e.g. treatment, outcome, sex, random, etc.). Four out of six datasets have been used, corresponding to

endpoints A,C to I endpoints of [112], available at `http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE16716`. A detailed explanation of the endpoint composition is included in Table 4.1. These data have been chosen because they are highly reliable, selected after a quality control process in order to provide a common test ground and because for each endpoint both a training set and an independent validation set are provided [112]. Furthermore, many different laboratories have tested their algorithm on the same datasets with the same evaluation protocol (i.e. train the classifiers on the training set with performance assessment on the validation dataset) and published their final outcome [112, 100, 83] thus an accurate benchmark can be performed to understand how well does a proposed algorithm perform with respect to a large number of state of the art alternatives. Results are compared in terms of Matthews Correlation Coefficient (MCC) [89] since, as stated in [112] it is informative when the distribution of the two classes is highly skewed, it is simple to calculate and available for all models with which the proposed method has been compared to. It is defined by:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{4.4}$$

where TP is the number of the true positives identified by the classifier, TN are the true negatives, FP are the false positives and FN are the false negatives. With true positive it is meant a sample categorized as positive, P, in Table 4.1 and correctly classified as positive by the classifier. The remaining values of TN, FP and FN are consequently defined. The MCC can assume values from 1 (perfect classification) to -1 (perfect inverse classification).

4.2.2 Clustering distance & scoring measure comparison

The aim of this section is to assess the usefulness of the metagenes, comparing the predictive results of classifiers built with features obtained with Treelets clustering, Euclidean clustering and without adding any metagenes. In all cases, the IFFS algorithm introduced in Section 4.1.1 has been consistently adopted. Meanwhile, the three scoring systems for IFFS feature selection presented in Section 4.1.2, (lexicographic sorting, exponential penalization and linear combination), are evaluated too in parallel analyses.

The experimental setup is a sequence of four main steps: data preprocessing, metagene

Table 4.1: Microarray datasets used for classification.

Dataset	Endpoint description		Microarray platform	Training set			Validation set		
				Samples	P	N	Samples	P	N
Hamner	Lung tumorigen vs. non tumorigen	A	Affymetrix Mouse 430.2.0	70	26	44	88	28	60
NIEHS	Liver toxicant vs. non toxicant	C	Affymetrix Rat 230.2.0	214	79	135	204	78	126
Breast cancer	Pre operative treatment response	D	Affymetrix Human U133A	130	33	97	100	15	85
	Estrogen receptor statur	E		130	80	50	100	61	39
Multiple Myeloma	Overall survival milestone outcome	F	Affymetrix Human U133Plus 2.0	340	51	289	214	27	187
	Event-free survival milestone outcome	G		340	84	256	214	34	180
	Sex of the patient	H		340	194	146	214	140	74
	Negative control, random assignation	I		340	200	140	214	122	92

creation, full-data analysis with some chosen δ values when the scoring depends on a parameter and a final performance assessment in terms of MCC and predictive accuracy, comparing the obtained results with state of the art alternatives.

The data preprocessing step for all the datasets, (except the Hamner), consists in setting the minimum value to $log_2 10$ in order to avoid considering small valued probe sets followed by a $log_2(\cdot)$ transformation and a mean removal operation along the samples direction (i.e. each feature is set to have zero mean) as it is a common practice in microarray analysis. The Hamner dataset, instead, needs to be normalized at first because an important batch effect has shown to worsen the performance of the validation analysis ([112] supplementary material). For this reason, data are firstly normalized using robust multi-array normalization (RMA) procedure on the whole data space, training and validation sets. Subsequently they are processed exactly like the other datasets.

The metagene creation phase is performed as explained in Chapter 3 applying *Treelets* clustering, the *Euclidean* clustering or without applying any clustering to assess the metagenes usefulness for classification.

In the following step, the predictive performance of the alternatives is measured. As shown in subsection 4.1.2, both the exponential penalization and the linear combination depend on a δ parameter, so the algorithm has been tested on multiple δ values, chosen after a small study on a reduced version of the available data. For the linear combination rule, a range of δ values between 0.05 and 1 with 0.05 interval has been tested. The best selected values are [0.05, 0.10, 0.15]. About the exponential combination a range of δ values from 5 to 100 with 5 interval has been tested, choosing $\delta = [5, 10, 15]$ for further evaluation.

Once the δ values have been chosen, the analysis is performed on the complete datasets (genes and metagenes) applying the feature selection algorithm to train classifiers up to five dimensions. In order to have a rigorous validation assessment, validation data are properly processed by setting the minimum to $log_2(10)$, subtracting the gene means calculated on the training set, and then producing the necessary metagenes using the coefficients from the hierarchical tree built on the training set. Results are collected for each δ and the classifier obtaining the best MCC value is considered as the measure of the prediction potential of the method.

Results analysis and assessment

The experimental results following the experimental protocol are presented and discussed here. In Table 4.2, the mean MCC and accuracy results across the analyzed endpoints from [112], A C D E F G H I, are showed. Each *datXX* expression identifies a different classifier developed by a different research group involved in the MAQC study. The *datXX* values are those whose results are reported in [112]. As it can be observed, the MCC results in Figure 4.2 span a range from 0.284 corresponding to *dat3*, to the 0.490 obtained by *dat24* group, while the accuracy values span from 65.43% of *Dat3* to 83.86% of *Dat20*. The best alternative is different depending on the chosen measure. This variation is linked to the class distribution skewness which can lead an algorithm to have a high accuracy but a very low or null MCC value. This is exactly what happens to *Dat20* analyzing endpoint F: it has 87.38% accuracy while MCC= 0 because it considers all the samples pertaining to a single class which corresponds to 87.38% of the validation set samples.

Table 4.2: MAQC mean MCC and mean Accuracy results

Group	*MCC*	*Accuracy*	*Group*	*MCC*	*Accuracy*
dat3	0.284	65.43%	*dat11*	0.453	75.59%
dat33	0.300	66.04%	*dat36*	0.457	79.18%
dat7	0.307	71.04%	*dat10*	0.458	78.39%
dat19	0.384	79.52%	*dat4*	0.468	81.49%
dat29	0.397	81.78%	*dat12*	0.476	82.54%
dat35	0.419	77.69%	*dat25*	0.477	80.81%
dat18	0.428	77.29%	*dat13*	0.488	80.67%
dat32	0.431	78.89%	*dat24*	0.490	81.13%
dat20	0.443	83.86%			

The MCC value better evaluates the performances of the scheme, particularly in cases of uninformative classification. The I endpoint is not considered in the mean calculations because it is a negative control dataset on which algorithms should produce bad results because class memberships have been randomly defined (see Table 4.1). Results in Table 4.2 are organized by increasing MCC value along each column.

In tables 4.3, 4.4 and 4.5 the results applying the proposed framework on the datasets from Table 4.1 are presented. Each table includes the results pertaining to a different scoring rule: the lexicographic sorting, the exponential penalization or the linear combination.

In Table 4.3, the mean MCC and accuracy values with the lexicographic scoring are showed. In each column the results corresponding to a different metagene generation method are reported: *Treelets* clustering, *Euclidean* clustering, or *None*. The *None* column corresponds to the results when no metagene has been considered. As for the method reported in [112], the I endpoint is not considered in the mean calculation due to its random nature. As can be seen, the introduction of metagenes allows obtaining higher mean MCC and accuracy values, thus producing better classifiers. With the lexicographic sorting the best MCC result is 0.423, with 77.46% accuracy, if *Treelets* clustering as metagene generation method is chosen.

Table 4.4 contains the collected values applying the exponential penalization scoring rule. Results are organized in four columns. The left column specifies the δ parameter, while the remaining three columns are organized as in Table 4.3. Changing the scoring rule leads to remarkably better results than those in Table 4.3. The simultaneous use

of both the error rate and the reliability allows us to reach better performances. Here also, results with metagenes are better than without and the best result is obtained when *Treelets* clustering is adopted and δ is equal to 10. Finally, the best mean MCC value is even higher than the best one of Table 4.2 from *Dat24*. There, the best MCC is 0.490, while here 0.495 is reached, supporting the proposed framework as an excellent alternative to state of the art methods. Concerning the accuracy values, with *Treelets* clustering and $\delta = [5, 10]$, better results than those in Table 4.2 are obtained. The highest accuracy value is 84.02%, obtained with $\delta = 5$.

In Table 4.5, the results relative to the linear combination score are showed. The organization is the same as in Table 4.4. In this case too, the metagenes have confirmed to be useful for classification because the results obtained with *Treelets* or *Euclidean* clustering are better than without. A comparison with the lexicographic sorting shows how, generally, the mean results are higher. In this case, the best mean MCC is 0.486 when *Euclidean* clustering is adopted and the δ parameter is between 0.1 and 0.15, while the highest accuracy value is 83.60% when δ is set to 10. Observing the results using both linear combination and exponential penalization rule, the MCC values are quite stable to small variation of the δ parameter. This is a good property because there is no need to precisely optimize the alpha value.

To visualize the proposed algorithm performance in comparison with the state of the art alternatives from [112], Figure 4-4 and 4-5 are introduced. In Figure 4-4, the results are sorted by increasing mean MCC value and are represented as columns. The MCC value for each alternative is printed above each column, and below the corresponding method is indicated. In Figure 4-5, the accuracy values are presented, sorted by increasing values. All the results from Table 4.2 are included and painted as uniform light gray bars. For space and clarity reasons, not all the results obtained with the proposed framework are included. A selection of them is proposed representing only the best three results for the exponential penalization and for the linear combination rule, and the overall best result with the lexicographic sorting. The result from the lexicographic sorting scheme is painted as a black bar and is identified by the *Lexicographic* label. Results applying the linear combination scheme are highlighted by a black and white horizontal

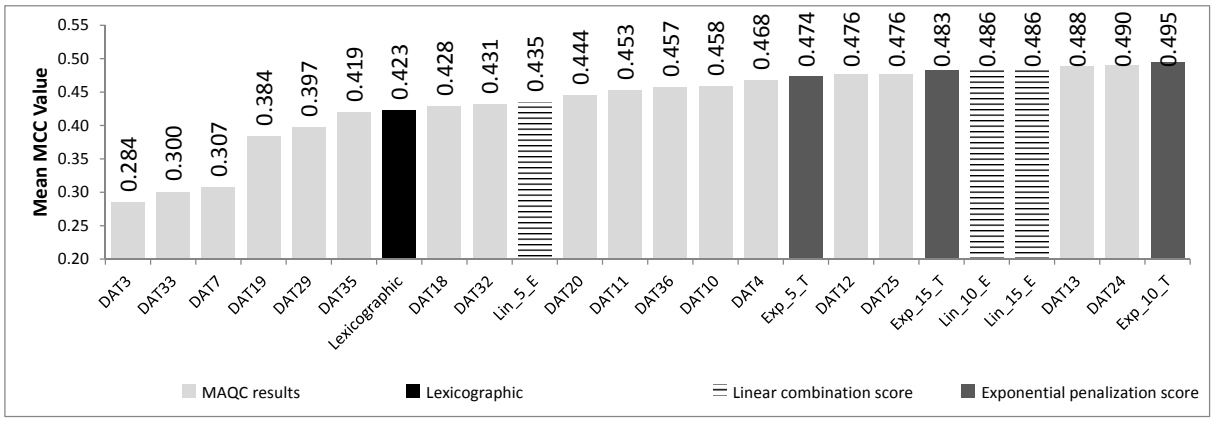

Figure 4-4: Mean MCC values comparison between MAQC results and the best alternatives for the different scoring techniques adopted.

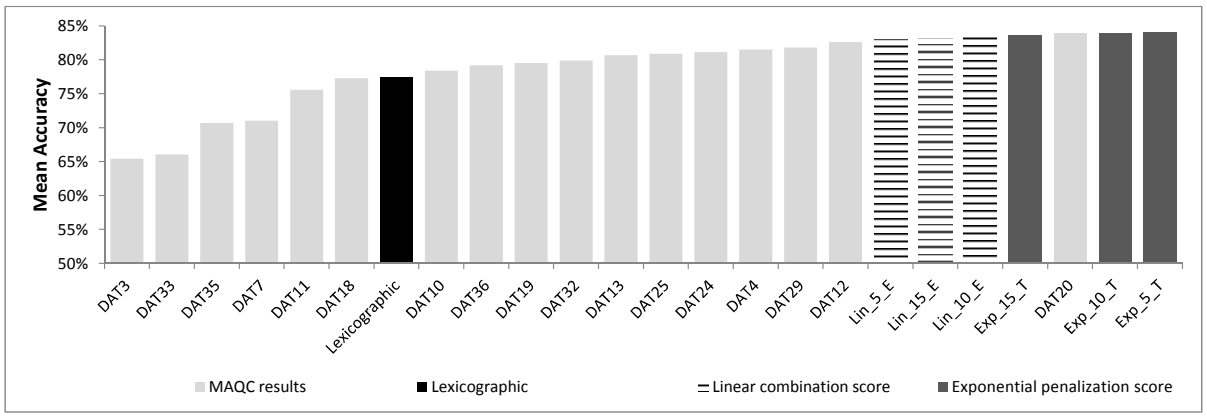

Figure 4-5: Mean accuracy values comparison between MAQC results and the best alternatives for the different scoring techniques adopted.

lines pattern. The labels start with *lin_xx_E*, where *_xx* is the δ value multiplied by 100 and *_E* indicates that the *Euclidean* clustering has been used. The values corresponding to the exponential penalization scoring rules are coded as dark gray columns. The labels are coded by *exp_xx_T*, where *_xx* is the δ value and *_T* indicates that the *Treelets* clustering has been adopted. As can be observed in Figure 4-4, the proposed framework obtains results comparable to the best state of the art alternatives when the linear combination scoring or the exponential penalization rule are used. Furthermore, the *exp_10_T* obtains the best overall mean MCC value. From Figure 4-5 it can be observed how both *exp_10_T* and *exp_5_T* obtain better values than the compared state of the art alternatives. Furthermore, it is shown how the accuracy value too is robust to small δ variations.

The mean number of chosen features by all the presented alternatives spans between 2.14 of *exp_10_T* to 3.43 of *lin_10_E*. As can be seen, the metagene creation process has almost doubled the number of features compared to the original number of genes, but the final classifier actually uses a very low number of features to perform the classification.

Statistical analysis of the top performing algorithm

The proposed framework provides competitive performances with respect to the state of the art alternatives. To validate this result, a further study has been performed to assess the robustness of the obtained performance. The study consists in a 50 runs Monte Carlo analysis of the classification endpoints. This 50 run setup has been proposed to have a broader range of experiments to assess the performance stability linked to the use of cross validation as performance estimation method, which is known to have a large variance [21]. In each run, the framework setup is the same as the best alternative: *Treelets* clustering as metagene generation method and exponential penalization with $\delta = 10$ as scoring rule for feature selection.

The results are shown in Figure 4-6 as a boxplot where the gray box represent the interval between the 25^{th} and the 75^{th} percentiles and the black crosses are values considered outliers. In Table 4.6 some statistical results are presented. Each column in Figure 4-6 corresponds to a different endpoint, labeled along the x axis. For each column, an asterisk identifies the MCC value obtained in the previous study (the values used to obtain the mean MCC value in Figure 4-4), whose values are included in the last column of Table 4.6 under the label of *run 0*.

The values are collected in the same way as the *run 0* iteration, for each endpoint, classifiers have been built up to five features and the best one is then considered in the mean calculation. Results for each endpoint are presented separately to better identify how the algorithm performance can change depending on the analyzed data.

What can be observed from both Figure 4-6 and Table 4.6 is that the results show a high robustness in the analysis of most endpoints. The obtained values are tight around their mean value for the endpoints A,C,D,E,H,G and I. The mean values are very close to the *run 0* results. The mean values in all these endpoints are slightly higher than the

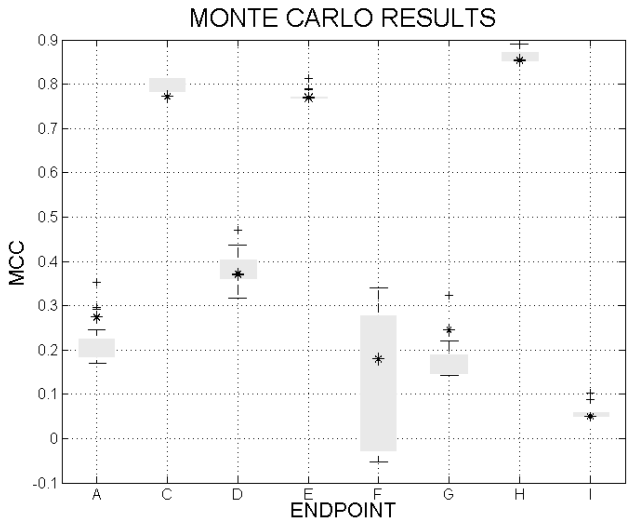

Figure 4-6: Boxplot of the obtained results along the 50 independent runs. Each column corresponds to a different endpoint.

run 0 results, except for the G and A endpoints, where the *run 0* results are well above the mean in the upper tail of the MCC distribution. About the F endpoint, it shows a considerably higher variability in the MCC distribution and this is mainly due to the class distribution skewness. In this endpoint, the positive class represents about the 15% of the training set. This eases the choice of uninformative features during the feature selection phase. The choice of an uninformative feature (a feature classifying all the samples to one class) biases the feature selection process and can lead to very different results. The Monte Carlo simulation confirmed that the predictive power is mainly determined by the analyzed dataset [112]. About the G endpoint, the Monte Carlo results are quite robust but inferior to the *run 0* result. This lead to think that the formerly obtained MCC value is due to some fortunate cross validation partition that allowed the selection of more useful features. Such lucky case is not unique as other runs may obtain even better values in the 50 run simulation (marked by the crosses outside the box). These results can be interpreted as “outliers" in the population distribution which underlines how the cross validation partition can change the final results. The MCC and accuracy values for the remaining endpoints are good and very consistent. This is an important feature of the proposed framework because it produces robust results. The results stability seems to be connected with the class distribution skewness which can lead to the choice of

uninformative features.

To analyze how the class distribution skewness influences the prediction performances, a final test with synthetic data has been performed. The experimental process follows the protocol introduced in [59], limiting the total feature number to 1000 and the Monte Carlo iterations to 30 due to calculation time reasons. Moreover, a skewness dimension has been introduced. In [59], the two classes have the same number of samples, while here three different setups have been tested. Class 1 may represent 50%, 70% or 90% of all the available samples. For each Monte Carlo iteration, a different dataset is built, the hierarchical tree is built with *Treelets* clustering and classifiers up to 10 features are trained with the exponential penalization rule and $\delta = 10$. For each iteration, the best classifier in terms of MCC is used for the following analysis.

Table 4.7 contains the summary of the study based on synthetic data. Results are organized in three subtables, one for each data generation model. In [59], three data generation models have been proposed: *Redundant, Synergetic* and *Marginal*, producing data with different distributions and detailed in [59] while a graphical representation of the feature distributions in the three models for feature pairs is shown in Figure 4-7. The general idea of the *Redundant* case is to divide features in blocks of similar characteristics with no correlation, the *Synergetic* model introduces positive covariance among each feature group introducing multivariate interactions, while *Marginal* is an extreme case of synergetic model in which each feature alone is useless. Each subtable in Table 4.7 presents the results organized by size of the training set because it is an important variable about the possible overfitting, and organized by skewness, which is the main variable in this study. For each, skewness-training set size, the mean MCC value, the standard deviation of the MCC (Std), and the mean feature number (# F) are presented. The mean is calculated not only along the Monte Carlo iterations, but also along the other varying parameters. This is done for sake of synthesis and because the focus is on the skewness. In a complete algorithm assessment, many more results should be presented taking into account dependencies while varying each one of the possible parameters.

Analyzing the results in Table 4.7, it can be observed how in both the *Redundant* and the *Synergetic* models, the high class skewness has a negative effect over the MCC

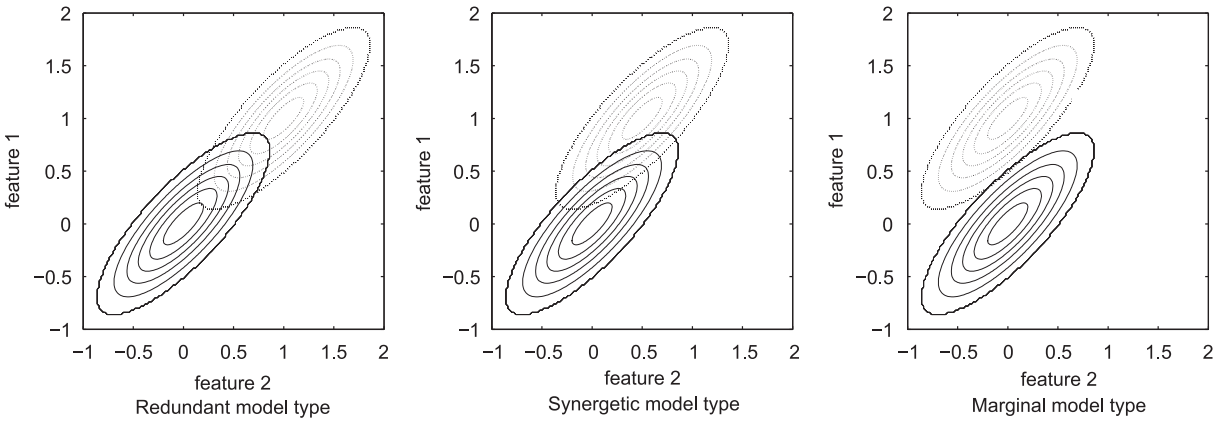

Figure 4-7: Graphical illustration of the three synthetic models used to generate features for feature pairs: Redundant, Synergetic and Marginal models are represented showing the densities for samples of two classes.

value. When the training set size is not too small, 120 and 180 samples, the MCC and skewness are inversely proportional along all the studied values. The *marginal* model instead presents a different behavior in which the best MCC values are constantly obtained with the intermediate skewness value and in one case, 180 training samples, the 50% case is the one obtaining the, slightly, worse performance. It can be stated how the skewness negatively influences the performance when the data have a redundant or synergetic model distribution, while with data represented by the marginal model, such direct relation does not hold.

What holds throughout all the results in Table 4.7 is the mean standard deviation of the MCC values. When the distribution is highly skewed (the 90% case) the standard deviation is always higher than the other cases, regardless of the mean MCC, whether it is better or worse. This is similar to what has been observed in the MAQC Monte Carlo study where the F endpoint results showed a much higher variability than any other endpoint.

About the mean selected feature number, the values span from 1.83 to 8.17. The best classifiers obtained by the proposed algorithm use also a reduced number of features in the synthetic case. This behavior helps in the training phase since the maximum feature number can be bounded by values of small magnitude.

Analysis of the selected features and additional benefit of the framework

The proposed analysis framework offers additional benefits other than the prediction accuracy thanks to the introduction of the hierarchical metagene structure. These benefits can make the use of this framework even more appealing from an analysis and hypothesis generation point of view. To illustrate them, a more detailed look to the *Run 0* results is provided. Tables 4.3, 4.4 and 4.5, show how using metagenes improves the classification results. Almost 15% of the chosen features in *Run 0* and in the Monte Carlo simulation are metagenes. These features extract the common behavior of gene clusters, reducing the noise thanks to the linear combination of individual genes. An example comes from the E endpoint classification, obtained applying LDA on two features: an individual probe set, 205225_at, and a metagene merging three probe sets named: 213462_at, 39548_at and 39549_at. These three probes show high pairwise correlation, higher than 90%, and, after a gene list analysis with DAVID [60] and GSEA [115], all refer to the neuronal PAS domain protein 2 (NPAS2). The chosen metagene is a summary of the NPAS2 behavior by merging three different probes expressing the same biological element.

Furthermore, the metagene structure can be useful for hypothesis formulation to infer biological relations between probe sets. An example of this potential in *Run 0* is the endpoint C analysis where the chosen metagene is formed by two elements, 13763271_at and 1379381_at. The first one, 13763271_at, corresponds to the tumor necrosis factor receptor superfamily member 14, (TNFRSF14), while no additional information can be found about the 1379381_at probe set neither in DAVID nor in GSEA. As a result, this metagene may suggest that further analysis and experiments on the 1379381_at probe set in relation with the tumor necrosis factor receptor superfamily could be initiated.

Finally, the proposed framework offers a model flexibility to deal with unpredictable problems during the numerical analysis and feature selection such as the probe set availability for further validating experiments. A practical case is when one of the chosen features is not available for a further validation with immunohistochemistry (IHC), due to the unavailability of the respective antibodies [116]. In that condition, the inferred hierarchical structure offers an efficient way to find alternatives to the best proposed model. Two cases are discussed here:

1. One of the metagene components is not available for validation;

2. An individual gene is not available for validation.

Both cases are illustrated analyzing the *Run 0* results about the E endpoint classification. The final system is a two dimensional classifier composed of a metagene and the 205225_at probe set. In the first scenario, assume that the metagene cannot be used because one of its three probe sets is unavailable for validation. In such case, the chosen metagene could be substituted by any of the available descendants in the hierarchical tree without loosing too much in terms of the prediction performance: at worst, an error rate of 11% and an MCC = 0.770 can be obtained instead that an error rate of 9% and MCC = 0.812 (see Figure 4-8). The second scenario is complementary to the first one. In this case,

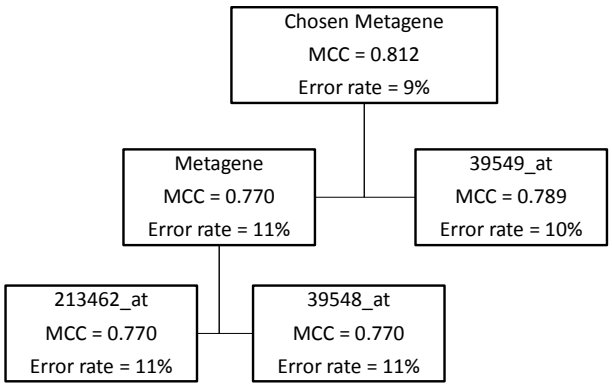

Figure 4-8: Hierarchical structure with the chosen metagene as root. In each node, the obtained MCC value and error rate are showed when the node is used instead of the chosen metagene. The best values are obtained with the original feature, root node, but the substitution with one of its descendant does not severely degrade the performances.

assume that the unavailable feature for validation is an individual probe set, 205225_at, used jointly with the previously chosen metagene. In this case, the hierarchical structure may be used to find the closest available nodes to the originally selected feature. The obtained results are shown in Figure 4-9. As can be seen, the best results (obtained with the 205225_at probe set) correspond to MCC = 0.812 and error rate = 9 %. The best alternative is obtained with the *sibling* node which is a metagene. It gives a MCC = 0.756 and an error rate = 13%. The *sibling* node is a metagene composed of five probe sets, 209602_at, 209603_at, 209604_at, 212956_at and 212960_at and obtains better performance than any of its descendants in the hierarchical structure.

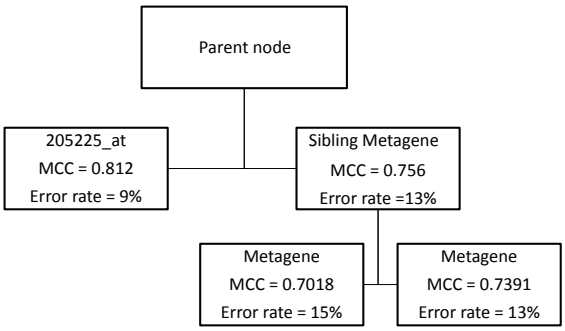

Figure 4-9: Substitution results for the 205225_at probe set. In each node the obtained MCC value and error rate are showed when the node is used instead of the chosen probe set. The best values are obtained with the original feature, 205225_at and the best substitution is with the sibling node, *Sibling Metagene*. The root node has no available values because it cannot be chosen as a substitute for the 205225_at node.

4.2.3 Metagene generation rule comparison

A side analysis has been performed evaluating the possibility to change, simplifying, the metagene generation process inside the hierarchical clustering process to substitute the default metagene generation rule $g(\cdot, \cdot)$. In the current version, each metagene is produced with a local PCA on the two merged features [17]. The studied alternative proposes to substitute the PCA with a Haar wavelet to be applied on the two merged features to produce a metagene.

The main difference between the two rules is in the linear combination weight assignment. With PCA, the linear weights can be anything constrained to $\|\alpha\|_2 = 1$, being α the two dimensional coefficient vector. With the Haar basis transformation, the weights are fixed and equal to $\sqrt{2}/2$. Such weighting difference eases the structure information storage and retrieval, because the only needed information is the merging order, without caring about the coefficient values. A side effect of the Haar basis transform is the generation of a completely different metagene set. To evaluate whether the Haar transform is a valid alternative, a set of experiments have been performed.

In detail, a Monte Carlo simulation on the MAQC datasets has been performed. The experimental conditions are exactly the same as the ones used for the results presented in Section 4.2: a 50 run monte carlo simulation has been performed on the MAQC datasets.

The mean simulation results are reported in Figure 4-10, where the mean MCC results

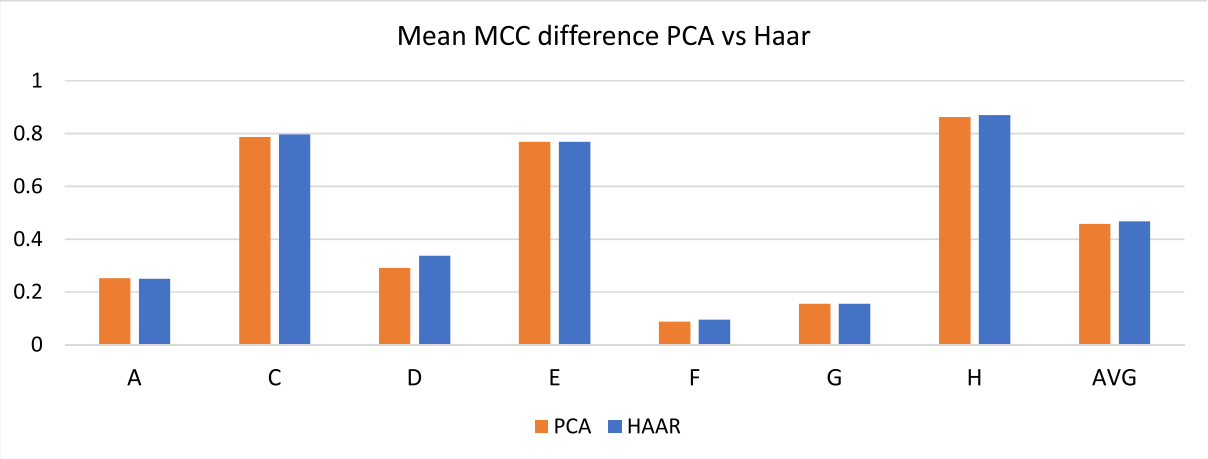

Figure 4-10: Mean MCC results comparison between PCA and Haar metagene generation rules.

applying the original PCA transform are represented by orange columns, while the current results applying the Haar basis transform are coded by blue bars. In Table 4.8, the same mean MCC value from the Monte Carlo simulation are reported. It can be observed how the Haar alternative obtains an overall MCC mean higher than the PCA original version.

Analyzing the results, it can be observed how there is little or no difference between the mean performances applying either Haar or PCA transform in generating a metagene. The difference is relevant in the F endpoint and, strictly speaking, using Haar basis to produce metagenes, leads to better results in 5 out of 7 datasets. As a general conclusion, Haar basis decomposition as metagene generation method can be a valid alternative to the PCA as metagene generation rule since the mean results are slightly better and the metagene generation process is easier than the original PCA implementation.

4.2.4 Classifier comparison: LDA and linear SVM

SVMs are very powerful tools for the learning and classification task. They are used in a very broad spectrum of applications, including the microarray classification with very simple kernels [125, 58]. Since SVMs are commonly used in machine learning and for microarray classification [112, 114], they have been considered as a possible alternative to the reference classifier, LDA.

The predictive results applying LDA have been compared with new results by changing LDA for a linear SVM. The reason for choosing linear SVM and not other nonlinear

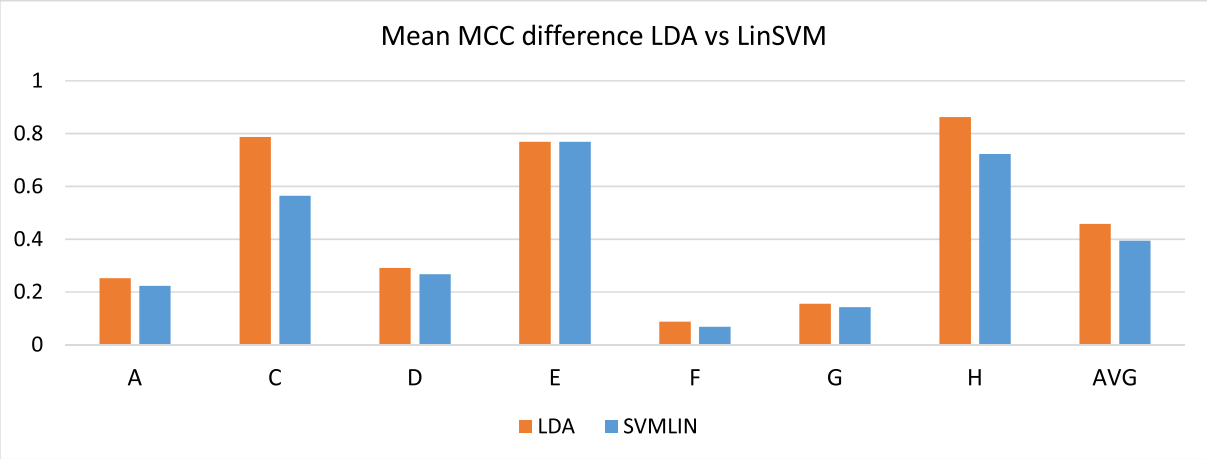

Figure 4-11: Mean MCC values on MAQC datasets comparing the LDA classifier and the linear SVM implementation.

kernels SVMs like radial basis functions or polynomial kernels [125] is due to the reliability measure formulation. The reliability has been considered for linear boundary classifiers which work in the space covered by the gene expressions. Its behavior when the decision space is augmented with a kernel is not known or well understood but there is a relevant probability that it may be biased by the nonlinear components which would reduce the discriminative effect of the current reliability formulation.

The linear SVM classifier has then been used with the default parameter for the libsvm implementation [26] because the sample size is too small to effectively perform a parameter estimation through internal cross validation, and because such a parameter estimation process would imply an enormous increase of the computation time.

The experimental process is the same as in Section 4.2, in which the MAQC datasets are analyzed with a 50 run Monte Carlo simulation, but for the SVM case, the iteration has been limited to 10 due to the much longer computation time than LDA implementation

The results in Figure 4-11 correspond to the mean Matthews Correlation Coefficient values (MCC) [89]. What can be observed on Figure 4-11 is that the results obtained with LDA are significantly better than those obtained using linear SVM. In 6 out of 7 datasets the mean MCC value is higher using LDA than using SVM.

Overall, it appears how the choice of LDA instead of SVM with linear kernel is the good one for the proposed feature selection algorithm. Probably, SVM classification can be improved with a proper parameter tuning but that would require more samples to be

effective and will surely imply an increase of the computation time (e.g. 10 fold for a 10 fold cross validation tuning).

4.3 Ensemble feature selection

Ensemble learning combines multiple learning algorithms, called experts, to improve the overall prediction accuracy and have been extensively adopted in the literature [136]. A plethora of ensemble methods has been developed to analyze biological data and there exist many alternatives reviewed for example in [136, 72]. They became popular because they allow to improve the classification by aggregating multiple experts to make decision over unseen data in a consensus way. In order to effectively improve the ensemble performances the experts should be accurate, (i.e. better than random), and diverse from each other [136].

An approach to ensemble learning called overproduce and select is described in [72] as a method to obtain good ensemble learners. It consists in producing a big set of experts and then select a subset which will be used for classification via majority voting. Several criteria of expert selection algorithms are studied in [72] and compared. Among the considered algorithms, the one called *Accuracy in diversity*, AID, [8] was able to reach the best prediction accuracy when compared to several alternatives [72].

In this thesis two versions of the AID algorithm from [8] have been implemented and studied as reference ensemble algorithm. One is the original AID implementation and the other is a simplified version from Kuncheva's book [72] and that will be named *Kun*. To produce a huge and diverse set of experts, we decided to use the overabundance of features microarray data. For each one of the available features, a Linear Discriminant Analysis classifier, LDA, is built and used as an expert. The available feature set is not only composed by the genes, but also by metagenes built as explained in Section 3.1 with the Treelets algorithm.

The microarray characteristics of small sample size and large feature number have been considered as possible issues for the ensemble search process, therefore novelty elements have been introduced to adapt the original thinning algorithm to the microarray scenario. In addition to including metagenes as experts, the notion of *nonexperts* that represent a

set of experts excluded from the thinning process due to they poor properties has been introduced as well as a rule to break ties in the thinning process.

4.3.1 The reference ensemble algorithms

The principles on which the AID algorithm is based are to include the most diverse and accurate classifiers by eliminating classifiers that are most often incorrect on examples that are misclassified by many experts. A pseudo code for the AID algorithm is shown in Figure 4-12. It is an iterative process in which, at each iteration, one expert is removed from the ensemble. At each iteration we consider to have a set of n samples and p experts [8]. To determine which expert E_i must be removed, some elements are calculated. The first one is an ensemble diversity measure called Percentage Correct Diversity Measure d [8], which is the percentage of samples which are correctly classified by a percentage of individual experts between 10 and 90 %. The d measure is then combined with other parameters, μ and β, defined in Figure 4-12 which are used to identify a set of relevant points S_p for the current iteration. The S_p set is composed of all samples which are correctly classified by a percentage of experts between the two calculated boundaries. Finally, the expert E_i to be removed from the ensemble is the one with lowest accuracy on the S_p set.

The rationale behind this is that the samples in S_p are those on which the ensemble is most uncertain, thus are those for which the elimination of an expert can be more relevant because it can change the ensemble majority voting. Therefore, excluding the expert that more poorly performs on these samples affects more positively the ensemble accuracy than simply excluding the expert with overall lowest accuracy. Since the ensemble changes throughout the iterations, the d value changes, as well as the boundaries, thus meaning that the set of relevant samples adapts to the ensemble changing characteristics.

In [8] is stated how the adaptive boundaries to define the S_p set are defined by considering the known relationship between the experts mean accuracy and the ensemble diversity [72]. On the other side, in [72] it is remarked how the AID algorithm could have equivalent performances with fixed boundary values, suggesting to use the ones in the calculation of the d measure: 10% and 90%. Since we could not find any works comparing

Table 4.3: Mean results adopting the lexicographic scoring scheme

Lexicographic sorting					
Treelets		*Euclidean*		*None*	
MCC	Accuracy	MCC	Accuracy	MCC	Accuracy
0.423	77.46%	0.418	76.18%	0.381	75.48%

Table 4.4: Mean results adopting the exponential penalization scoring scheme

Exponential penalization						
δ	*Treelets*		*Euclidean*		*None*	
	MCC	Accuracy	MCC	Accuracy	MCC	Accuracy
5	0.475	84.02%	0.457	81.57&	0.442	82.99%
10	0.495	83.95%	0.460	83.61%	0.421	82.66%
15	0.483	83.67%	0.451	83.187%	0.457	83.30%

Table 4.5: Mean results adopting the linear combination scoring scheme.

Linear combination						
δ	*Treelets*		*Euclidean*		*None*	
	MCC	Accuracy	MCC	Accuracy	MCC	Accuracy
0.05	0.483	83.45%	0.437	81.58%	0.444	81.46%
0.10	0.468	83.31%	0.486	83.60%	0.444	81.46%
0.15	0.469	83.25%	0.486	83.19%	0.444	81.46%

Samples $S = s_1 \dots s_n$
Experts $E = E_1 \dots E_p$

while $\#E > 1$
Calculate $S_d = \{s_i\} : 0.1 \leq f(s_i) \leq 0.9$
where $f(s_i)$ fraction of experts in the ensemble correctly classifying i^{th} sample.
Calculate $d = \frac{\#S_d}{n}$
Lower Bound $l_b = \mu \cdot d + \frac{1-d}{n}$
Upper Bound $U_b = \beta \cdot d + \mu(1-d)$
Define the set of relevant samples.
$S_p = \{s_i\} : l_b \leq f(s_i) \leq U_b$
$E_i =$ expert with lowest accuracy over the S_p set.
$E := E - E_i$
Remove E_i from E
end

$\mu =$ Mean experts accuracy
$\beta = 0.9$

Figure 4-12: Pseudocode for the AID algorithm.

Table 4.6: Statistical properties of the Monte Carlo simulation.

Endpoint	*MCC*	*Accuracy*	*Run 0 MCC*	*Run 0 Accuracy*
A	0.2176	67.37%	0.2750	65.91%
C	0.7949	90.25%	0.7700	89.22%
D	0.3869	80.49%	0.3690	80.00%
E	0.7732	89.17%	0.7680	89.00%
F	0.1147	86.3%	0.1800	87.85%
G	0.1723	79.57%	0.2430	82.71%
H	0.8609	93.21%	0.8550	92.99%
I	0.0564	55.14%	0.0510	52.68%

the two alternatives, we chose to apply both and keep the one with better performances.

4.3.2 Microarray adaptations for thinning

Considering the microarray data characteristics we propose some key points to obtain a good ensemble system:

Experts cohort We chose to build thousands of experts defining each expert as an LDA classifier trained on a different feature. Both genes and metagenes, obtained with the algorithm from Chapter 3 are considered as individual features since metagenes helped in finding better classifier than with genes only.

Nonexperts We introduce the notion of nonexpert to remove a whole set of "experts" with poor training characteristics. We decided to exclude from the thinning process all those experts that classify all the training sample with the same label. Considering that the expert is unable to distinguish two classes, it is not considered as a useful ensemble component. The nonexpert number can vary depending on the data type and it increases when the class distribution is highly skewed. Furthermore, the idea of nonexpert responds to the microarray data characteristic of feature overabundance: the major part of the available features is useless for prediction purposes since they are not related to the classified phenomenon. Thus, we included this simple criterion in the thinning process.

Tie breaking Considering the typical case of small sample number for microarrays and considering that the S_p sample is smaller or equal to the whole training sample number,

Table 4.7: Results of the study based on synthetic data. The three subtables correspond to the three different data distributions. Each subtable is organized showing the values depending on the skewness value and the different size of the training set. The *Train* column contains the size of the training set, the *MCC* columns shows the mean MCC value across the different experimental conditions and Monte Carlo iterations while *Std* and *#F* columns contain the MCC standard deviation and the mean number of selected features respectively.

Skewness - Class 1 percentage -									
	50%			*70%*			*90%*		
Redundant model									
Train	MCC	Std	# F	MCC	Std	# F	MCC	Std	# F
60	0.509	0.120	4.50	0.431	0.140	4.83	0.319	0.193	3.58
120	0.532	0.086	2.58	0.468	0.117	4.67	0.323	0.143	3.50
180	0.545	0.071	2.75	0.492	0.086	3.33	0.346	0.120	7.33
Synergetic model									
Train	MCC	Std	# F	MCC	Std	# F	MCC	Std	# F
60	0.343	0.184	4.58	0.315	0.187	2.92	0.325	0.239	1.83
120	0.431	0.133	5.42	0.351	0.143	5.83	0.266	0.221	4.75
180	0.475	0.108	5.50	0.407	0.109	5.92	0.257	0.189	6.50
Marginal model									
Train	MCC	Std	# F	MCC	Std	# F	MCC	Std	# F
60	0.509	0.159	6.58	0.555	0.150	3.25	0.490	0.193	2.17
120	0.549	0.148	7.50	0.610	0.135	4.92	0.542	0.211	2.92
180	0.570	0.139	7.75	0.631	0.137	8.17	0.572	0.193	4.00

	A	C	D	E	F	G	H	AVG
PCA	0.253	0.788	0.291	0.769	0.088	0.155	0.863	0.458
Haar	0.251	0.797	0.337	0.769	0.095	0.156	0.871	0.468

Table 4.8: Mean MCC results from Monte Carlo simulation on MAQC datasets. The two algorithms differ from the metagene generation rule, PCA versus Haar basis decomposition.

there is a relevant probability to have ties when comparing experts accuracies. To deal with this problem and introduce a rule, the metagene generation process is considered. When ties occur, the excluded expert is the one which has been generated at a higher level in the hierarchical tree, so that metagenes composed of many children with low similarity will be eliminated instead of another metagene with more correlated components. Indeed it is more likely that a metagene with more correlated children will replicate its behavior than another one merging many different individual genes. Finally, the ties between individual genes are randomly resolved since they all are on the same level of the hierarchical tree.

The usefulness of these three elements is assessed by experiments comparing the complete algorithm with three modified algorithms, each of which does not use one of the

Table 4.9: MCC results comparing the studied AID and *Kun* algorithms.

	A	C	D	E	F	G	H	MEAN
AID	0.293	0.793	0.459	0.789	0.221	0.231	0.813	0.514
Kun	0.407	0.812	0.459	0.789	0.221	0.236	0.828	0.533
Kun_{tie}	0.303	0.804	0.451	0.789	0.221	0.236	0.828	0.519
Kun_{genes}	0.346	0.781	0.366	0.773	—	0.313	0.817	0.485
Kun_{all}	—	0.792	—	0.789	—	—	0.031	0.230

proposed key elements.

4.3.3 Ensemble algorithms comparison

Two experiments are performed to evaluate the ensemble reference algorithms and to evaluate the usefulness of the introduced microarray adaptation elements. The first experiment evaluates whether the original AID algorithm [8] or the simplified version in [72] has better performances. They will be identified by *AID* and *Kun* respectively. Both algorithms are trained on the seven datasets. For each dataset they produce thousands of nested ensembles, one for each iteration. These ensembles are then applied on independent validation datasets and the best performing ensemble is taken as representative of the predictive potential of the algorithm as in [84, 15]. In order to avoid voting artifacts, only ensembles with an odd number of experts are considered.

The chosen performance metric is the Matthews Correlation Coefficient (MCC) [89], since, as stated in [112] it is informative when the distribution of the two classes is highly skewed, it is simple to calculate and available for all models with which the proposed method has been compared to. MCC values range from -1 (i.e. perfect inverse prediction) to 1 (perfect prediction).

The second experiment has the same setup as the first one, but it evaluates the usefulness of the introduced elements in Section 4.3.2: the nonexpert notation, the metagene inclusion and the tie breaking rule. Three algorithms are compared to the original one. Each one applies two of the three elements and is identified, for the *Kun* algorithm by:

- Kun_{all} : This algorithm does not exclude the nonexperts from the thinning process.
- Kun_{genes} : This algorithm excludes the nonexperts but it does not use any metagene.

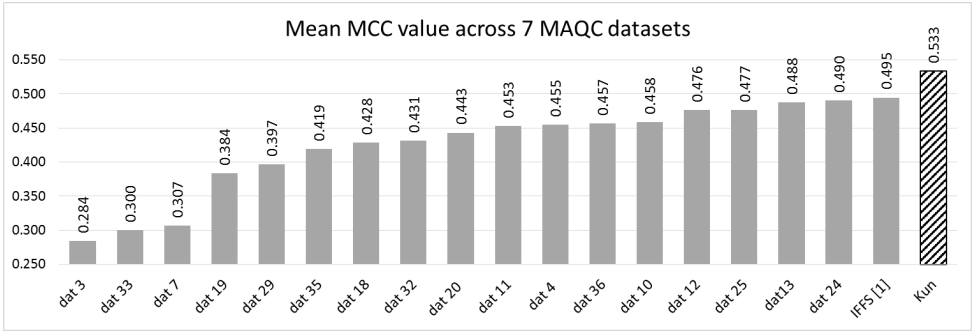

Figure 4-13: Mean MCC results comparison with state of the art results from [112] and from Section 4.2.2.

- Kun_{tie} : This algorithm resolves each tie without considering the tree structure, thus eliminating the first expert it encounters with lowest accuracy on S_p set.

Finally, the best performing algorithm is compared to state of the art alternatives from MAQC study [112] and from the best results in Section 4.2.2. In this way it is also possible to compare the differences introduced by the ensemble thinning algorithm with respect to the algorithm from Section 4.2.2, that uses the same features but with a different feature selection algorithm.

4.3.4 Comparison with state of the art

Table 4.9 shows the MCC results for all the studied algorithms in this work. Each dataset corresponds to a column and the last column is the mean MCC value across the datasets, the higher the value the better the algorithm is considered for prediction. The comparison between the AID and the simplified *Kun* algorithm can be done observing the first two lines in Table 4.9. The *Kun* algorithm obtains better overall MCC mean value and in every single dataset it obtains better or equal MCC values. It can be stated that the simpler *Kun* algorithm achieves better prediction results and it should be preferred to the AID algorithm.

In the last four rows of Table 4.9, the main proposed innovations are analyzed by comparing the full *Kun* algorithm, with three algorithms, each one excluding a different aspect. They are organized by decreasing mean MCC, so that it can be straightforwardly seen which algorithm obtains the best performances and how much each of the key elements affects the final results. Globally, the *Kun* algorithm obtains better results with

an overall MCC of 0.533 and the introduced elements have different impacts. The tie breaking rule is the least affecting factor since Kun_{tie} obtains a mean 0.519 MCC. The metagene inclusion as individual feature has a significant influence on the predictive ability, as an MCC of 0.485 is obtained. Here too, the metagenes are useful for classification as in Section 4.2.2 and not using them can lead to undesirable MCC values since the missing values represent an undetermined MCC due to the null denominator. This is obtained when all the validation samples are assigned to one class [15]. Finally, the most important of the introduced elements is the nonexpert definition. Not including this concept leads to very poor results and, more importantly, to undetermined MCC values in many of the analyzed datasets. This is due to the fact that all nonexperts agree on every sample, thus strongly biasing the ensemble vote.

From the results in Table 4.9, the best performing algorithm is the full *Kun* and all the introduced adaptations helped in obtaining such results. In Figure 4-13, the mean MCC value of *Kun* algorithm is compared with state of the art alternatives. The vast majority, all the dat_{xx} columns, correspond to the mean MCC value from the MAQC study [112]. In addition to them, the column labeled as *IFFS* is the mean MCC value from the best results from Section 4.2.2, which makes use of the same features, genes and metagenes, but adopts the IFFS feature selection algorithm. The state of the art algorithms are represented as solid gray columns, while the *Kun* mean MCC value is represented by a black and white straight lines pattern.

It can be observed how the *Kun* algorithm obtains a remarkable improvement when compared to state of the art alternatives and, comparing the shown results with the mean values in Table 4.9, it can be observed how various of the tested algorithms would have obtained better than state of the art results. This confirms the goodness of ensemble thinning as approach to combine multiple experts for classification [72].

4.3.5 Tuning the ensemble

From the results in Section 4.3.4, it appears how the *Kun* algorithm is a valid alternative for the feature selection task, and how the microarray adaptation elements have helped in obtaining the final result. To further explore the potential of the *Kun* feature selection

method, several alternatives have been implemented and analyzed. Changes have been made to the used classifier and to the nonexpert notation. The studied classifiers to train each expert are the following:

- LDA, which is the original *Kun* algorithm.
- SVM linear kernel implemented with libsvm [26] with a constant parameter $C = 1$
- SVM rbf kernel, implemented with libsvm [26] with default training parameters.
- K-Nearest Neighbors with 5 neighbors implemented with Matlab [88]. For this case the 3 and 11 neighbors have also been tested but obtained lower overall MCC results, so they are not shown.

The nonexpert notation has been switched among:

- Constant label, like the original *Kun* algorithm
- MCC ≤ 0 has been used as definition of a nonexpert because a random label assignment should give an MCC $= 0$.
- The binary Area Under the Curve [112] smaller than 0.5 :bAUC ≤ 0.5. The 0.5 threshold has been chosen because a random assignation should return a bAUC $= 0.5$.

In Figure 4-14 and in Table 4.10, the average MCC results by analyzing the MAQC datasets are presented. The obtained results are organized depending on the combination of nonexpert notation and the adopted classifier.

Analyzing the differences related to the nonexpert definition, it can be observed from Table 4.10 how the bAUC criterion constantly obtains lower results than the other two. On the other hand, the differences between the constant criterion, already introduced in the *Kun* algorithm in Section 4.3.1, and the criterion of MCC ≤ 0, are less evident. Depending on the chosen classifier, the best results is obtained with either the Constant definition, or the MCC ≤ 0 definition.

Observing the adopted classifier, it can be observed how the KNN classifier consistently obtains the lowest average MCC value, regardless of the nonexpert definition rule. The

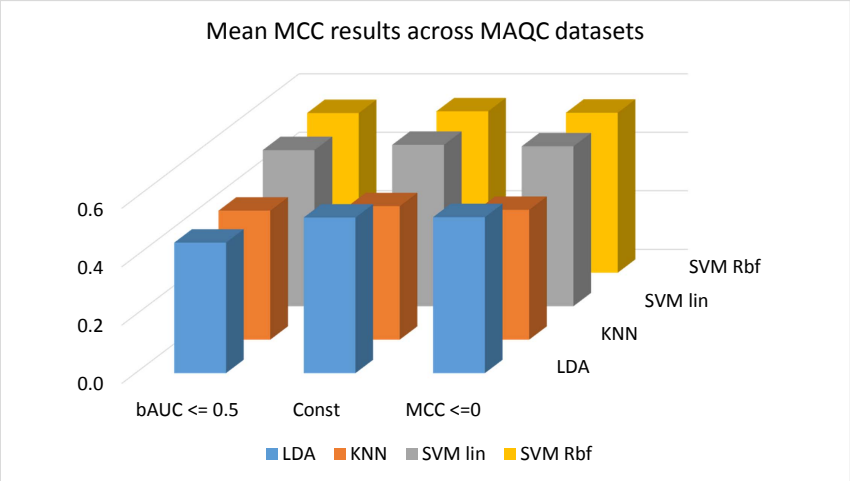

Figure 4-14: Mean MCC results comparison among all the tested alternatives for classifier and nonexpert condition. The values are the mean across the MAQC datasets.

Table 4.10: Mean MCC results comparing the alternatives in terms of nonexpert notation and adopted classifier.

Classifier	Nonexpert		
	bAUC ≤ 0	Const.	MCC ≤ 0
LDA	0.447	0.533	0.534
KNN	0.442	0.457	0.445
SVM linear	0.534	0.552	0.547
SVM rbf	0.546	0.552	0.547

best performances are consistently obtained with the SVM with RBF kernel, even if it obtains the exact same results as the linear SVM when the nonexpert notation is the Constant criterion or the MCC ≤ 0 criterion.

Overall, considering all the studied variables, it can be concluded that both KNN classifier and the bAUC nonexpert definition should not be used, because they consistently lead to worse MCC results. The best result is obtained using the Constant nonexpert definition like in the *Kun* algorithm, and either the linear SVM classifier or the SVM rbf classifier. Summarizing, the *Kun* algorithm can be improved by using a SVM classifier, either with a linear kernel or with a radial basis function kernel, thus pushing further the difference from state of the art results shown in Figure 4-14.

The obtained results with ensemble feature selection are better than those obtained with IFFS in Sections 4.2, showing higher predictive potential in terms of MCC. A difference from the results in Section 4.2 is that the ensemble algorithms have been tested on a single run experiment due to time reasons, while the IFFS results confirmed their robustness of the results on Monte Carlo simulation. The selected feature composition between best IFFS implementation from Section 4.2 and the *Kun* algorithm with SVM-RBF kernel, show some relevant differences. Across the seven datasets, the IFFS algorithm chooses always less than 5 features to produce a classifier, 15% of which are metagenes, each of which is composed of less than 10 genes on average. The *Kun* algorithm with SVM-RFE kernel classifier, chooses between 3 features for datasets E and H, up to more than 300 features for dataset D and G. The metagenes percentage increases up to a 50% and the average number of genes composing the metagenes grows up to more than 100. As a global comparison, the IFFS algorithm chooses less features and metagenes composed with less features than the ensemble *Kun* algorithm. The number of selected features is less relevant for overfitting in an ensemble algorithm than in a wrapper algorithm like IFFS because the experts are trained independently. Nevertheless, choosing metagnes with hundreds of composing genes should be avoided because the chance of having selected a feature that reduces noise from a group of correlated genes is lower than a metagene grouping only four or five genes.

4.4 Summary

In this chapter, the predictive ability of the proposed framework for binary classification has been evaluated. The overall classification framework of feature set enhancement and feature selection has been studied introducing the IFFS wrapper algorithm in multiple Monte Carlo simulations.

Among all the studied possibilities to generate a hierarchical structure and produce metagenes, the original Treelets formulation combined with LDA classifier has shown to have very good results and to improve state of the art alternatives when analyzing publicly available data when the IFFS algorithm is used for feature selection. The application of SVM classifier instead of LDA within the IFFS feature selection framework did not

suppose performance improvement but considerably increased the computation time.

Ensemble feature selection techniques have been studied to test the potential of such an approach with very interesting results. The performed single run experiment with different configurations highlighted how the ensemble feature selection approach allows to further improve the state of the art predictive ability when compared to the IFFS results. The studied algorithm has been enriched with key elements like the nonexpert notion that allows to boost the performance. Overall, the best results have been obtained with SVM classifier combined with the nonexpert notion introduced in Section 4.3.2.

Even if the ensemble techniques allowed to reach better prediction results in a single run experiments, the IFFS approach has been preferred to analyze the results in the following chapters to compare with the state of the art because its statistical robustness has been validated and it can be straightforwardly replicated in different scenarios like the knowledge integration in Chapter 5 or for the multiclass classification in Chapter 6. For ensemble techniques, a validation experimental model has not been designed for the multiclass case or to perform a sound Monte Carlo simulation.

Chapter 5

Knowledge integration for hierarchical clustering

Including and integrating prior biological knowledge has gained importance in the omics data analysis field throughout the years [3, 30]. Knowledge databases have been used in many directions, for example, to identify biologically relevant activated pathways by integrating Gene Ontology (GO) in the analysis process [105], or to integrate a gene ranking tool in the analysis [127]. Moreover, biological knowledge is also used in tools like Hanalyzer [77] to identify gene-to-gene relationships and facilitate the data interpretation. A common trait to all these works is that including some prior biological knowledge led to more interpretable results from a biological viewpoint, easing the scientist's task to formulate new hypotheses.

The aim of this chapter is to improve the microarray classification by combining prior biological knowledge with the numerical data when inferring a structure from the data and generating metagenes. The expectations are to build a classification framework able to compete with the best alternatives in the state of the art, to improve the prediction results robustness and the results biological interpretability.

To do so, we have studied modifications to the existing algorithm presented in Section 3.1, which demonstrated to have very good prediction performances when combined with IFFS feature selection in 4.2 and that works exclusively with numerical data. It relies on the same two-step approach, in the first step, a hierarchical clustering is applied over the

data to create an extended feature space with new features called *metagenes* based on the work on Treelets [78]. The second step takes care of the feature selection with a wrapper algorithm. Through the hierarchical clustering a binary tree is generated.

The proposed modification to the algorithm concerns the hierarchical clustering to include prior biological knowledge and to define the similarity between genes. A similar concept of knowledge integration has been implemented in Hanalyzer [77], where pairwise gene similarity is defined as a combination of numerical similarity and of knowledge similarity to infer gene regulatory networks. In [77], the pairwise similarities are used only once applying a threshold, while here they are used to infer a complete hierarchical structure and to produce new features. The aim is to generate *metagenes*, as in Chapter 4, able to help both in the noise reduction by inferring a structure and producing metagenes as combination of correlated genes, and in the data interpretation by summarizing genes with related biological functions. Different similarity metrics for the definition of biological similarity have been studied and chosen from the literature for their characteristics [14, 77]. Furthermore, the combination rule between biological similarity and the numerical correlation has been studied, comparing a simple average operation with a more elaborated value equalization which will be detailedly detailed in Section 5.3.

5.1 The knowledge database

There exist many available knowledge bases on the Internet, usually consulted to interpret the analysis results [67, 14, 115]. A subset of the Molecular Signature Database (MSigDB) [115] has been selected for this work. It is a collection of annotated gene sets provided with the Gene Set Enrichment Analysis (GSEA) software [115]. This database has been chosen because it is composed of high quality information, it is currently maintained and updated, and because it is publicly available and downloadable in an easy to use format.

The complete MSigDB database is composed of six gene set collections varying from manually *curated gene sets*, to *motif sets* or *Gene Ontology terms* related sets. Our choice has been to consider the C2, C3 and C5 collections. The C2 gene sets are manually curated sets from online pathway databases, publications in PubMed and knowledge of

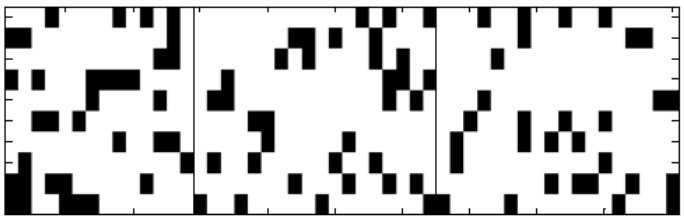

Figure 5-1: Toy example of a small knowledge database matrix where each row is a different gene while columns are attributes. Black dots represents that a gene has a specific attribute.

domain experts. C3 is composed of motif gene sets based on conserved cis-regulatory motifs from a comparative analysis of the human, mouse, rat, and dog genomes. Finally, C5 recollects gene sets sharing the same Gene Ontology term [115].

These data are publicly available and can be represented as a binary matrix **M** whose rows are the different genes, while the columns represent the MSigDB gene sets. A toy example of a possible knowledge matrix is shown in Figure 5-1, where each black dot represents the presence of a gene-gene-set correspondence. As can be observed, the matrix is sparse and this is a characteristic of the real knowledge matrix from the MSigDB data. The actual information from MSigDB C2, C3 and C5 gene sets is coded in a knowledge matrix **M** composed of 22680 unique gene identifiers and 5607 MSigDB gene sets. The **M** matrix is then used as knowledge database for the clustering process.

5.1.1 The hierarchical clustering process

The hierarchical clustering process is a pair-wise iterative process merging feature pairs to produce new hierarchical levels and new features called metagenes like described in Section 3.1 and illustrated in Figure 3-2.

In Section 3.1, features are merged measuring the Pearson correlation between two gene expressions and each metagene is built as the first principal component of the local Principal Component Analysis (PCA) over the two features to be merged [78]. Starting from the second merging step, metagenes and genes are considered as features, so the similarity must be calculated for all genes and metagenes.

In order to incorporate the information from the knowledge matrix $\underline{\underline{M}}$, changes to the similarity metric have been studied and are discussed in Sections 5.2 and 5.2. To this

end, for each feature pair $(\underline{f}_i, \underline{f}_j)$, two quantities are calculated: $d_n(\underline{f}_i, \underline{f}_j)$ which is the numerical similarity as in Section 3.1 and $d_k(\underline{f}_i, \underline{f}_j)$ the knowledge similarity. The global pairwise similarity is then defined as a combination of these two measures:

$$d(\underline{f}_i, \underline{f}_j) = f\left(d_n(\underline{f}_i, \underline{f}_j), d_k(\underline{f}_i, \underline{f}_j)\right) \tag{5.1}$$

In Section 5.2, the studied similarity measures to define $d_k(\underline{f}_i, \underline{f}_j)$ are presented and discussed, while in Section 5.2, the combination of d_n and d_k is analyzed, proposing two alternatives to define the final pairwise similarity.

5.2 Biological similarity measures

The introduction of the biological similarity in the clustering process brings to light some questions. The first one is, which measure should be adopted in quantifying how much two genes are alike and a second one is related to the clustering process and regards how the similarity measure can be integrated within the clustering process when generating metagenes as linear combinations of genes.

In the literature, there are plenty of similarity measures that have been proposed to work with binary data, categorical data or continuous data [14, 4, 132]. Since the knowledge matrix format is binary, we chose to search the literature for suitable measures in the binary and categorical field. From our research, we chose four different measures considering also the sparsity of the knowledge matrix and the computational feasibility of the measures. The chosen measures are the following:

- **Anderberg**: This measure has been proposed in [4] and assigns more importance to rare matches and rare mismatches. It ranges from [0; 1], the minimum value is attained when there are no matches, while the maximum value is reached when all attributes coincide between the compared features.

- **Godall**: This is an adaptation of the original measure proposed by Godall in [40], as presented in [14] under the name of *Godall3* to reduce the computational burden. This measure assigns higher similarity to a match if the value is infrequent than if

the value is frequent. Matches can be either of ones or zeros. Its range is between [0; 1] and it reaches one when all the attributes are the same.

- **NoisyOR**: This measure has been adopted in different works on microarray data analysis [77, 76]. It assumes the attributes independence and it computes the integrated likelihood for each feature pair through a noisy-OR function over each common attribute reliability. It is calculated with the consensus reliability estimate from [76]. This measure ranges from 0 to infinity, so it is normalized between [0 ; 1] by dividing by the maximum attribute reliability value as in [77].

- **Smirnov**: Smirnov [113] proposed a measure rooted in probability theory that not only considers a given value's frequency, but also takes into account the distribution of the other values taken by the same attribute. For a match, the similarity is high when the frequency of the matching value is low and the other values occur frequently. The range of this measure goes from [0; 2N] where N is the attribute number, so it is divided by $2N$ to be bounded between [0; 1].

All four measures adopt different criteria to define each attribute importance in the definition of a global similarity measure between two features. An additional concern arises when the metagene generation process is considered and, precisely, when the new metagene is generated via PCA. This step in the clustering process has not been touched and the new metagene is obtained as a linear combination of the two merged features considering only the expression value. This step has been preserved to maintain the metagenes benefit of noise reduction. As far as biological similarity is concerned, in order for the clustering process to progress, a knowledge profile must be assigned to the newly created metagene. The knowledge profile is represented by adding a new row to the knowledge matrix $\underline{\underline{M}}$ with the metagene corresponding attributes which are not necessarily binary values. The metagene generation formula for the numerical data is presented in Eq. (5.2) when it merges two features $(\underline{f}_i, \underline{f}_j)$ to build the $\underline{m}_k$ metagene.

$$\underline{m}_k = \alpha_1 \underline{f}_i + \alpha_2 \underline{f}_j \quad \text{with} \quad \alpha_1^2 + \alpha_2^2 = 1 \tag{5.2}$$

For the knowledge profile of the metagene $\underline{m}_k$ we chose to save as much as possible the

linear combination from PCA forbidding negative values which may occur in a PCA. The result is shown in Eq. (5.3), where $\underline{\underline{M}}i$ and $\underline{\underline{M}}j$ are the knowledge matrix rows for the i^{th} and k^{th} features.

$$\underline{m}_k = (|\alpha_1|\underline{\underline{M}}i + |\alpha_2|\underline{\underline{M}}j)/(|\alpha_1| + |\alpha_2|) \tag{5.3}$$

In this way the generated knowledge profile has non-negative values bounded between [0 , 1] which allow to use the chosen similarity metrics after a slight adaptation to accept continuous values instead of binary ones.

Table 5.1 shows the mathematical expression of the four studied similarity metrics. For each similarity metric, two formulas are shown, the first one is the original definition, while the second one is the continuous value adaptation. Some notations have to be introduced to properly read Table 5.1. First of all, the knowledge matrix $\underline{\underline{M}}$ is formed by N features, genes, and d attributes. Each column is a different attribute, while each row is a different feature. The notation $\underline{\underline{M}}i$ defines the i^{th} row of matrix $\underline{\underline{M}}$, which includes the attributes for an individual features. $\underline{\underline{M}}_{i,k}$ identifies the element from the i^{th} row and k^{th} column of the knowledge matrix. Table 5.1 presents the equations to measure the similarity between the i^{th} and j^{th} features, thus meaning the i^{th} and j^{th} rows of the matrix $\underline{\underline{M}}$. With $K_{\cap ij}$ the subset of shared attributes between the feature i and j is defined: $K_{\cap ij} = \{k\} \in \{1 \leq k \leq d : \underline{\underline{M}}_{i,k} = \underline{\underline{M}}_{j,k}\}$. While with $K^c_{\cap ij}$, the complementary subset of $K_{\cap ij}$ is defined : $K^c_{\cap ij} = \{k\} \in \{1 \leq k \leq d : \underline{\underline{M}}_{i,k} \neq \underline{\underline{M}}_{j,k}\}$. We also define the notions of $f_k(x)$, $\hat{p}_k(x)$, p_k^2 and r_k as in [14, 77]:

- $f_k(x)$ is the number of times that the k^{th} attribute assumes the value $x \in [0\ 1]$.
- $\hat{p}_k(x)$ is the sample probability for the value x for the k^{th} attribute.

$$\hat{p}_k(x) = \frac{f_x(x)}{N}$$

- p_k^2 is another probability estimate for the value x within the k^{th} attribute.

$$p_k^2 = \frac{f_k(x)(f_k(x) - 1)}{N(N-1)}$$

- r_k is the normalized consensus reliability estimate, $\hat{r}_k$ for the k^{th} attribute, calcu-

Table 5.1: Biological similarity measures formulas. For each measure the original formula and its adapted version for continuous variables are presented.

Method	$S_k(\underline{\underline{M}}_i, \underline{\underline{M}}_j)$
Anderberg	$\frac{\frac{1}{N}\sum_{k\in K_\cap}\left(\frac{1}{\hat{p}_k(\underline{\underline{M}}_{i,k})}\right)^2}{\sum_{k\in K_\cap}\left(\frac{1}{\hat{p}_k(\underline{\underline{M}}_{i,k})}\right)^2+\sum_{k\in K_\cap^c}\left(\frac{1}{2\hat{p}_k(\underline{\underline{M}}_{i,k})\hat{p}_k(\underline{\underline{M}}_{j,k})}\right)}$ $\frac{\frac{1}{N}\sum_{k=1}^{d}\left[\left(\frac{1}{\hat{p}_k(0)}\right)^2\left((1-\underline{\underline{M}}_{i,k})(1-\underline{\underline{M}}_{j,k})\right)+\left(\frac{1}{\hat{p}_k(1)}\right)^2\left(\underline{\underline{M}}_{i,k}\underline{\underline{M}}_{j,k}\right)\right]}{\sum_{k=1}^{d}\left[\left(\frac{1}{\hat{p}_k(0)}\right)^2(1-\underline{\underline{M}}_{i,k})(1-\underline{\underline{M}}_{j,k})+\left(\frac{1}{\hat{p}_k(1)}\right)^2\underline{\underline{M}}_{i,k}\underline{\underline{M}}_{j,k}+\frac{1}{2\hat{p}_k(0)\hat{p}_k(1)}\left(\underline{\underline{M}}_{i,k}+\underline{\underline{M}}_{j,k}-2\underline{\underline{M}}_{i,k}\underline{\underline{M}}_{j,k}\right)\right]}$
Godall	$\frac{1}{N}\sum_{k\in K_\cap}1-p_k^2(\underline{\underline{M}}_{i,k})$ $\frac{1}{N}\sum_{k=1}^{d}\left[(1-p_k^2(0))\left((1-\underline{\underline{M}}_{i,k})(1-\underline{\underline{M}}_{j,k}\right)+(1-p_k^2(1))\left(\underline{\underline{M}}_{i,k}\underline{\underline{M}}_{j,k}\right)\right]$
NoisyOR	$1-\prod_{k\in K_\cap}((1-r_k))$ $1-\prod_{k=1}^{d}\left((1-r_k)(\underline{\underline{M}}_{i,k}\underline{\underline{M}}_{j,k})\right)$
Smirnov	$\frac{1}{2N}\sum_{k\in K_\cap}\left[2+\frac{N-f_k(\underline{\underline{M}}_{i,k})}{f_k(\underline{\underline{M}}_{i,k})}+\frac{f_k(q)}{N-f_k(q)}\Big\vert_{q\neq\underline{\underline{M}}_{i,k}}\right]$ $\frac{1}{2N}\sum_{k=1}^{d}\left[\frac{N-f_k(0)}{f_k(0)}+\frac{f_k(1)}{N-f_k(1)}\right]\left((1-\underline{\underline{M}}_{i,k})(1-\underline{\underline{M}}_{j,k})\right)+\left[\frac{N-f_k(1)}{f_k(1)}+\frac{f_k(0)}{N-f_k(0)}\right]\left(\underline{\underline{M}}_{i,k}1-\underline{\underline{M}}_{j,k}\right)$

lated as in [77, 76] in order to bound its value between 0 and 1.

$$r_k = \frac{\hat{r}_k}{\max_k(\hat{r}_k)}$$

The calculation of the parameters $f_k(x)$, $\hat{p}_k(x)$, $p_k^2(x)$ and r_k are done over the initial knowledge matrix $\underline{\underline{M}}$ containing only the individual gene information. The information from the metagenes is not considered because these parameters must have a fixed value before starting to measure the feature similarity.

5.3 Combination of numerical and biological similarities

Once the different similarity metrics for the biological information are defined, the focus is on how to combine the two sources of information: numerical and biological. We have studied two different ways to combine the numerical correlation $d_n(\underline{f}_i, \underline{f}_j)$ and the biological similarity $d_k(\underline{f}_i, \underline{f}_j)$.

The first and easiest combination rule is a simple average of the two values, so that the overall similarity is defined as in Eq.(5.4).

$$d(\underline{f}_i, \underline{f}_j) = \frac{1}{2}\left(d_n(\underline{f}_i, \underline{f}_j) + d_k(\underline{f}_i, \underline{f}_j)\right) \tag{5.4}$$

In addition to the average combination, a more complex way to combine the two measures has been studied, based on the work from [77, 56], where the original similarity value is mapped to the range $[0\,,\,1]$ using a probability density function estimation assuming a logistic distribution with mean μ and variance $\nu = 6/\mu$. From [77, 56], it is highlighted how such a mapping can be beneficial to the discovery of important relationships between features. The underlying idea is to equalize the distribution of the calculated similarity values between 0 and 1 and making a more uniform combination of the values from the two sources of information. The equalization step would work if the assumption of the underlying distribution is correct, or if it suits the actual data. From the results we gathered the logistic assumption does not hold, in particular for the biological similarity data and especially with the imposed fixed ratio between μ and ν.

In order to operate an equalization step, we chose not to limit ourselves to a specific distribution type, but to estimate the density function on the real data. For this, 17 different parametric distributions are compared over a set of 10^5 pairwise similarity data as detailed in [90]. The best fitting distribution is then chosen in terms of Bayesian Information Criterion, BIC, and the equalization function is then obtained. After equalization, a new distribution $\tilde{d}_x(\underline{f}_i, \underline{f}_j)$ is obtained with a more uniform distribution of the values between 0 and 1. This work is done independently for the numerical correlation and the biological similarity so to equalize both distributions properly. The global similarity value

is then defined as in Eq. (5.5) as an average of the two equalized similarities.

$$d(\underline{f}_i, \underline{f}_j) = \frac{1}{2}\left(\tilde{d}_n(\underline{f}_i, \underline{f}_j) + \tilde{d}_k(\underline{f}_i, \underline{f}_j)\right) \tag{5.5}$$

5.4 Knowledge integration evaluation for classification

The evaluation of the usefulness of the knowledge integration framework for microarray classification is assessed in this section using the IFFS feature selection algorithm described in 4.1.1. The knowledge integration algorithms have been described in Sections 5.2 and 5.3, and combine the numerical correlation with four biological similarity measures (Anderberg, Godall, Noisy-OR and Smirnov) and with two combination schemes (average and distribution equalization).

The experimental protocol to evaluate both the predictive ability and the biological relevance of the different knowledge integrating schemes when classifying microarrays is detailed here. All algorithms have been analyzed in terms of predictive power and in terms of biological relevance of the found signatures. The objective of the study is to compare the different schemes to evaluate if introducing biological knowledge helps in obtaining better results, more robust and interpretable than in the original Treelets implementation using numerical data only.

5.4.1 Predictive power evaluation

To evaluate the predictive power of the different algorithms, a 50 run Monte Carlo simulation has been performed. For each run, the same protocol as in Section 4.2.2 has been followed, classifiers up to 5 dimensions have been built for each datasets. The best classifier in each case has been chosen by evaluating the Matthews Correlation Coefficient, MCC, [89] results when classifying the independent validation set.

Statistical properties of the algorithms predictive power have been extracted from the population of 50 run simulations in order to draw conclusions about the general behavior of each tested algorithm. The mean MCC value across the 50 iterations has been considered as well as the standard deviation of the results. The comparison among all the variants

considered in this Chapter and the algorithm presented Section 4.2.2 combines both the mean value and the standard deviation to consider also the stability and repeatability of the prediction results throughout the iterations.

A score, S is extracted as defined by Eq.(5.6) and all the algorithms are then sorted according to the S value.

$$S = \frac{\mu_{MCC}}{\sigma_{MCC} + \epsilon} : \epsilon = 0.02 = \frac{1}{50} \tag{5.6}$$

It is proportional to the mean MCC value so that higher means obtain higher scores, but it also is inversely proportional to the MCC standard deviation, so that more robust and stable results can obtain higher scores. The ϵ value at the denominator has been chosen to reduce the risk of giving too much relevance to the results robustness that could make the mean value irrelevant. The value has been chosen as the inverse of the Monte Carlo runs (i.e. $\epsilon = 0.02 = \frac{1}{50}$) and it is comparable to the obtained standard deviation values collected in Section 5.5.

5.4.2 Biological relevance evaluation

Besides the prediction ability, an additional evaluation is performed studying the differences among the found gene signatures about their biological usefulness. This kind of analysis aims at assessing the interpretability of the different solutions.

The aim is to see if the biological knowledge integration helps in selecting genes which are good for classification and also useful for biological interpretation. The biological usefulness assessment is an extremely complicated task. It is related to the specific problem under study and depends on the scientists' experience. Nevertheless, an established practice in the literature is to evaluate the different gene signatures with automatic analysis tools, for example to find enriched functions or to find genes related to an investigation topic from the literature. For each of the considered alternatives, the union of the used genes to build the classifiers throughout the Monte Carlo iterations is used as gene signature. When a metagene is chosen to be part of a classifier, all the genes composing it are included in the signature.

Four publicly available tools have been used to quantify the biological relevance of the

gene lists. They assess different characteristics of a gene list using different databases and references. The adopted tools are the following:

GSEA The first tool uses Gene Set Enrichment Analysis resources [115]`http://www.broadinstitute.org/gsea/msigdb/annotate.jsp`. For each gene list, it calculates an output p-value for each one of the selected MSigDB gene sets [115]. The p-values are calculated as hypergeometric distributions of overlapping genes between the analyzed gene signature and the MSigDB gene set. A low p-value indicates a high probability that the MSigDB gene set is represented in the gene signature and therefore that genes used for classification have something in common from a biological viewpoint (function, position, disease, etc.). For this analysis, the subsets C2, C4 and C5 from MSigDB have been used. Gene sets can be collected from various sources such as on-line pathway databases, publications in PubMed, knowledge of domain experts or from Gene Ontology databases.

Biograph The second tool is called Biograph [82] `http://www.biograph.be/`, it quantifies relationships between individual genes and a key term (e.g. the studied disease). Biograph analyzes each gene individually and quantifies their relationship with the key term based on a knowledge database. The output score is proportional to the gene key-term relationship. The method is based on the integration of heterogeneous biomedical knowledge bases and yields intelligible and literature-supported indirect functional relations. By assessing the plausibility and specificity of these hypothetical functional paths within a user-provided research context, the unsupervised methodology is capable of appraising and ranking of research targets, without requiring prior domain knowledge from the user. Since this method analyzes the relations between each gene and a relevant key-term, when analyzing the 7 MAQC datasets, different key-terms have been chosen, relating with the studied phenomenon: A dataset: lung neoplasms; C dataset: liver neoplasms; D and E datasets: malignant breast neoplasms; F dataset: Multiple Myeloma, G dataset: Survival Analysis and H dataset: sex differentiation.

Enrichr The third used tool is Enrichr [27] `http://amp.pharm.mssm.edu/Enrichr/index.html`, which is an integrative web-based and mobile software application that in-

cludes gene-set libraries, an alternative approach to rank enriched terms, and various interactive visualization approaches to display enrichment results. Enrichr contains 35 gene-set libraries where some libraries are borrowed from other tools while many other libraries are newly created and only available in Enrichr. It has been used to analyze the enrichment of the gene lists in terms of KEGG pathways. The chosen output for each different pathway is a combined score presented in [27].

Génie The fourth tool is Génie [45] `http://cbdm.mdc-berlin.de/~medlineranker/cms/genie`. With Génie, genes are ranked using a text-mining approach based on gene-related scientific abstracts. It prioritizes all of the genes from a species according to their relation to a biomedical topic using all available scientific abstracts and ontology information. Génie takes advantage of literature, gene and homology information from the MEDLINE, NCBI Gene and HomoloGene databases. This tool, like Biograph, analyzes each gene independently and its output is a p-value assessing the relevance of each gene with a search term. The used search term - dataset pairs are: A dataset: lung cancer; C dataset: liver cancer; D and E datasets: breast cancer; F and G datasets: multiple myeloma and H: sex.

The four analysis tools evaluate different characteristics of the gene lists. Tools like GSEA or Enrichr perform a gene-set analysis as a whole, while Biograph and Génie evaluate the individual gene relevance. To quantify in a simple way the collected results, an evaluation protocol is proposed. For each analysis tool, the first five outputs are averaged: for GSEA and Génie it is an average of the negative logarithm of the p-values while with Biograph and Enrichr it is an average of the first five output scores. In this way, the method with the highest average is the one obtaining the best result. The next step is to average all the values across the 7 datasets to obtain an average score ranking all the algorithms over multiple results.

Afterwards, to combine all the obtained results in terms of biological relevance and of predictive ability a voting scheme is adopted. We chose to combine the biological analysis results into a single score from the average of the Borda count of the four analysis tools. In Figure 5-2, a toy example is shown with only two analysis tools for biological information. For each tool, each method is assigned points depending on its ranking.

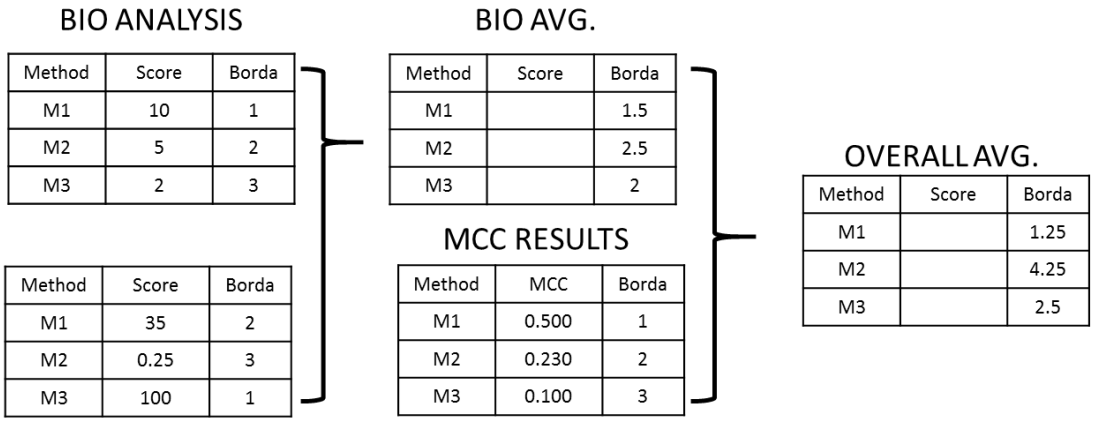

Figure 5-2: Toy example of the adopted ranking scheme using only two biological relevance analysis tools combined with Borda count.

Subsequently, the biological rankings are averaged to define a real valued unique score. Finally, the calculated scores are averaged with the rankings from the predictive power analysis defined in Section 5.4.1, in order to obtain a global score for all the considered methods. The one with the lowest final score is considered to be the best method in terms of combined predictive accuracy and biological relevance of the found gene lists.

5.4.3 Comparison with state of the art

The proposed algorithms including biological information in the metagene generation process have been compared with state of the art alternatives. Firstly, they have been compared with the algorithm from Section 4.2.2 which uses Treelets with only numerical correlation and that obtained very good predictive scores when compared with MAQC results in Section 4.2.2 and in [15].

In addition to that, a comparison with state of the art algorithms adopting different techniques has been done too. The two best algorithms from [84] have been chosen for comparison since they have been applied on the MAQC data and the relevant genes from their gene lists are publicly available. The algorithms of [84] analyze a subset of the MAQC datasets, in detail they are the two Breast Cancer datasets (called D and E) and the two Multiple Myeloma datasets (called F and G), but the gene lists information has been published only for the Breast Cancer datasets. Therefore, the comparison with the studied algorithms in this work has been done only over the D and E datasets from

MAQC.

The analysis protocol is the same, by applying the four biological analysis tools and the predictive power evaluation. The two chosen algorithms from [84] are the Support Vector Machine Recursive Feature Elimination with Support Vector Machine classifier, SVMRFE-SVM, because it is stated in [84] to be the one with the best gene lists, while the second algorithm is called Gradient based Leave-one-out Gene Selection with Nearest Mean Scale Classifier, GLGS-NMSC, because it is the one with the best predictive characteristics in [84].

5.5 Experimental results

After having introduced the experimental protocol in Section 5.4.3, the experimental results are here presented and discussed. There are eight studied algorithms for knowledge integration which are compared among themselves in terms of predictive ability and biological interpretability of the selected gene lists for classification in Section 5.5.1, and are also benchmarked to state of the art alternatives in Section 5.5.3 with an uniform experimental protocol. These eight algorithms are obtained from four different biological similarity metrics (i.e. *Anderberg, Godall, NoisyOr* and *Smirnov*), and two integration schemes for the numerical and biological similarities (i.e. simple average and a probability density function equalization scheme). The adopted notation to identify each algorithm has the form X-yyy, where X is the initial letter of the similarity metric (e.g. A for Anderberg or G for Godall), while yyy represents the combination scheme: avg for the average and pdf for the probability density function equalization. The studied algorithms in this work have also been compared to the algorithm from Section 4.2.2 which uses only the numerical correlation to define the similarity and that will be named COR in the results presentation.

5.5.1 Prediction results evaluation

In Table 5.2, the obtained results for the predictive ability are shown. For each dataset, the mean MCC value μ and its standard deviation σ are shown on two different rows,

Table 5.2: Comparison of the obtained MCC statistics on MAQC datasets.

		COR	A-avg	G-avg	N-avg	S-avg	A-pdf	G-pdf	N-pdf	S-pdf
A	Mean	0.278	0.255	0.291	0.271	0.249	0.249	0.264	0.251	0.253
	Std	0.055	0.042	0.080	0.063	0.028	0.025	0.056	0.027	0.033
C	Mean	0.797	0.825	0.793	0.817	0.789	0.802	0.828	0.828	0.804
	Std	0.025	0.018	0.013	0.015	0.023	0.016	0.010	0.010	0.007
D	Mean	0.315	0.285	0.276	0.279	0.294	0.266	0.334	0.297	0.288
	Std	0.085	0.083	0.081	0.063	0.088	0.013	0.016	0.020	0.066
E	Mean	0.773	0.749	0.773	0.746	0.754	0.741	0.738	0.742	0.779
	Std	0.019	0.019	0.012	0.033	0.017	0.016	0.013	0.017	0.018
F	Mean	0.249	0.060	0.065	0.060	0.036	0.008	0.180	0.067	0.048
	Std	0.045	0.053	0.041	0.053	0.083	0.044	0.056	0.059	0.077
G	Mean	0.162	0.217	0.215	0.208	0.217	0.217	0.218	0.212	0.204
	Std	0.042	0.033	0.032	0.037	0.031	0.028	0.038	0.031	0.041
H	Mean	0.866	0.786	0.782	0.866	0.863	0.869	0.862	0.863	0.784
	Std	0.014	0.010	0.019	0.017	0.015	0.018	0.015	0.017	0.008
Score	$\frac{\mu}{\sigma+\epsilon}$	10.879	11.225	11.286	10.265	10.865	12.012	13.439	12.731	12.997

while each column corresponds to a different algorithm. The final row in Table 5.2 contains the overall score calculated as in Eq. (5.6), with a combination of mean and standard deviation: the higher the obtained score is, the better the algorithm is considered in terms of predictive ability.

From the results in Table 5.2 we can observe how in terms of mean value all the algorithms obtain similar values for the majority of the datasets. This states how including biological information does not negatively affect the mean predictive power. In addition to that, we can observe how the σ values are in general smaller when the biological similarity is considered.

An exception of the observed behavior is represented by the datasets F and G, where two symmetrical behaviors are present. The COR algorithm obtains a noticeably higher mean value in the F dataset when compared to all the alternatives except G-pdf, in that case the mean difference is smaller. On the contrary, with the G dataset the situation is almost symmetrical, with the COR algorithm obtaining lower mean MCC values than the rest of alternatives.

This behavior can be explained looking more in detail the selected features in both the cases. About F dataset, the COR algorithm chooses a metagene as the first and most relevant feature. This metagene is built at a high level of the tree and joins three genes, MVP NCR1 and KLF11, that have a correlation smaller than 50%. What happens is

that this metagene allows a better generalization in the validation dataset, even if the probability of this result is low. The metagenes that are generated at higher levels of the tree are combining features with low similarity between them and the probability that the generated metagene is reducing the noise and extracting a robust common behavior is low. This fact is corroborated when observing what happens in all the alternatives except G-pdf. In all those cases, a metagene is chosen which has been built in a later stage of the hierarchical structure generation. On the contrary, G-pdf selects the MVP gene only as first feature and the LGLSE gene as second feature. The LGLSE gene is recognized as important by the Biograph analysis for survival and it helps in improving the classification in the G-pdf case.

In the G dataset the observed situation is that all the algorithms choose as first relevant gene the TGFA gene, which is found important by the Biograph analysis. As second feature, the COR algorithm chooses the MLF2 gene while all the alternatives choose a metagene in the lower levels of the tree. As example we consider the G-pdf metagene which merges two probe sets both corresponding to the ANXA2 gene which confirms that merging those features can help in reducing the noise. Furthermore, the ANXA2 gene is found as relevant by both the Biograph and the Génie analyses for the survival key term. Metagenes high in the tree can bring poorer results with higher probability as in the F dataset, while metagenes merging actually correlated genes more consistently improve the prediction results as in the G dataset.

An overall consideration about the MCC results is that the algorithms using the *pdf* combination rule achieve better scores, among which the G-pdf is the best one. On the other side, the COR algorithm is in the lower half of the algorithms mainly due to the high variance shown in the results. The methods including biological information allow us to obtain more robust results without compromising the mean MCC value. This is an interesting feature since the COR method showed to be better than the state of the art alternative methods from the MAQC study in Section 4.2.2, showing also predictive results robustness.

Table 5.3: Results from the biological evaluation of the gene signatures and the global ranking results.

		GSEA	BIOGRAPH	ENRICHR	GENIE	Bio Avg	MCC	Global Score
COR	mean	3.917	1.05E-03	4.909	2.91E+02	7	8	7.5
	rank	8	5	6	9			
A-avg	mean	5.132	1.07E-03	4.607	2.80E+02	6.5	5	5.75
	rank	7	4	7	8			
G-avg	mean	5.167	1.33E-03	4.450	5.09E+02	4.75	6	5.375
	rank	6	3	8	2			
N-avg	mean	5.298	2.00E-03	2.313	5.06E+02	5	9	7
	rank	5	1	9	5			
S-avg	mean	5.825	1.40E-03	7.797	5.06E+02	3.5	7	5.25
	rank	3	2	5	4			
A-pdf	mean	6.755	1.88E-04	8.734	5.07E+02	3.25	4	3.625
	rank	1	6	3	3			
G-pdf	mean	6.424	1.63E-04	9.357	5.10E+02	3	1	2
	rank	2	8	1	1			
N-pdf	mean	5.656	1.40E-04	8.830	5.03E+02	5.5	3	4.25
	rank	4	9	2	7			
S-pdf	mean	3.825	1.74E-04	7.883	5.06E+02	6.5	2	4.25
	rank	9	7	4	6			

5.5.2 Biological relevance evaluation

The biological relevance analysis of the found gene signatures has been done using four analysis algorithm: *GSEA, Biograph, Enrichr* and *Génie*. In Table 5.3, a summary of the obtained results is shown, as well as the ranking from the MCC results and the final global score.

The first four columns are dedicated to the adopted analysis tool, while the fifth column, *Bio score*, includes the overall score for the biological information evaluation. The sixth column, *MCC score*, contains the ranking for the MCC results from Table 5.2, while the last column is the Global score as the average of the *MCC score* and *Bio score*. The *Global score* values represent our evaluation of the algorithms ability to both predict new samples, and to identify biologically relevant genes.

From the results in Table 5.3 we can observe how the *pdf* methods consistently obtain better results than the other when gene sets as a whole are considered: *GSEA* and *Enrichr* analyses. About the individual gene analysis tools, it can be stated that both *pdf* and *avg* algorithm obtain good results, the *pdf* algorithms obtain better results when *Génie* is used, while the situation is reversed when considering *Biograph*. Both these analysis tools can have biased results since the scores are assigned individually and few relevant

genes are sufficient to highly increase the average value. When looking at the overall *Bio score* results, it can be observed how the COR algorithm is unsurprisingly last. This is expectable since it is the only algorithm that does not make use of prior biological information.

When looking at the final results, *Global score*, it is observed how the best algorithm is the G-pdf, corresponding to the Godall measure and *pdf* combination scheme. An additional observation is that the *pdf* combination achieves better scores than the *avg* methods, showing how the pdf equalization process led to better results. In synthesis, adding biological information is beneficial because it improves the biological interpretability of the results as well as the results stability without negatively affecting the mean predictive power.

5.5.3 Comparison with state of the art algorithms

The studied algorithms, including the COR algorithm, have been compared to state of the art alternatives from [84]. The two considered algorithms have been introduced in Section 5.4.3 and are from here on identified as SVMRFE-SVM and GLGS-NMSC.

The comparison has only been possible on the two D and E datasets because there are no published data about the selected gene lists for the other datasets. The obtained results are shown in Figure 5-3 where for each algorithm 5 columns and the global score value are presented. The algorithms have been analyzed with the same protocol as before. Columns in Figure 5-3 show the rankings for the different tools. The first 4 columns corresponds to the biological analysis tools and the fifth is the ranking associated to the MCC. The black line shows the global score for each algorithm, and it is used to sort the algorithms. An algorithm is considered to be better than another if it obtains a lower global score.

Analyzing the results, we can observe how in terms of predictive ability, the SVMRFE-SVM and GLGS-NMSC do not obtain good positioning and this is due to the high variance the presented results. The GLGS-NMSC algorithm obtains the overall best mean, less than a 2% improvement, but it has a standard deviation up to nine times higher than G-pdf algorithm. About the biological relevance analysis, the SVMRFE-SVM obtains an overall best score than the GLGS-NMSC as stated in [84], but even so it reaches the

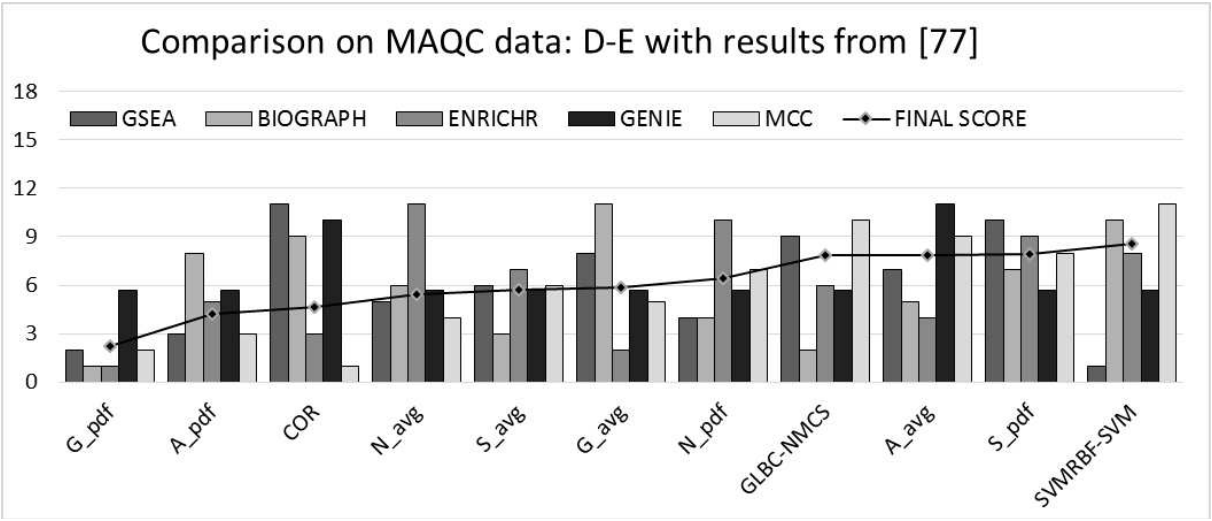

Figure 5-3: Score comparison with results from [84] on datasets D and E from MAQC datasets. All the algorithms are sorted by increasing final score, the black line. The best result is the one with the smallest overall score, which is G-pdf, consistently with the obtained results over a wider selection of datasets.

worst global score among the studied alternatives due to its prediction performances. An observation must be done about the Génie data because they are almost all the same. This is due to the fact that almost all algorithms are able to identify genes with zero p-value for both the datasets, (for example ESR1, IRS1, PHB or HRAS for dataset D and ESR1 for dataset E, which is a known gene related to breast cancer as it is an estrogen receptor), thus obtaining an ideally infinite value. This has been considered when evaluating all the datasets and a maximum threshold of 1000 has been set to avoid having infinite values in the algorithms comparison.

Looking at the global results, we observe how the G-pdf still is the best scoring algorithm even if the global score order has changed with respect to Table 5.3.

5.6 Summary

In this Chapter, the studied techniques to infer a hierarchical structure from microarray data combining both numerical information and prior biological information have been described and evaluated. They have been compared to state of the art alternatives and to the numerical information only solution from Section 4.2 which showed to have good and robust predictive properties.

The knowledge integration framework has been studied with different implementations, comparing four similarity metrics and two combination rules to merge the numerical correlation and the biological similarity.

The algorithms have been compared with Monte Carlo experiments on public datasets in terms of their predictive ability and biological interpretability of the chosen gene signatures. The knowledge integration has shown to be beneficial increasing the predictive power robustness without losing the mean performance value when compared to the numerical correlation only alternative, as well as producing more biologically interpretable gene signatures.

Among the studied alternatives, the G-pdf algorithm combining Godall similarity measure with the probability density function equalization is the best one. It consistently obtained the best performances when compared to the other knowledge integration alternatives as well as when it has been compared to state of the art algorithms.

As a general observation, a proper knowledge integration framework like G-pdf should be preferred to the bare numerical treelets, when possible, since it obtains more robust and interpretable results for classification.

Chapter 6

Multiclass classification

Machine learning techniques have been extensively applied on microarray data for cancer classification, obtaining interesting prediction performances [112, 16, 138]. Most of the work in the field is focused on the binary classification, considering the multiclass case as a straightforward generalization. Different studies suggest however that in the multiclass case, it is more complicated to obtain good prediction rates, especially when the class number is high and the class distribution is skewed [80, 114, 119, 128].

A novel multiclass approach has been studied in this thesis as a combination of multiple binary classifiers. It is an example of Error Correction Output Coding (ECOC) algorithms [37] applied to the microarray analysis. The ECOC algorithms obtained interesting results by applying built-in coding algorithms from the data transmission field [119]. Their direct application on biological data like microarrays has some drawbacks like the error independence assumption, the code matrix generation or the allowed binary partitions which will be detailed in the following sections. A new approach is introduced in this chapter to take advantage of the data transmission framework of ECOC algorithms without forgetting that what is decoded are biological data. To do so, the redundancy is used to reduce the error rate, but the binary classifiers are bounded to class partitions more likely to be significant than with other ECOC approaches.

The proposed ECOC scheme adds to the classical One Against All (OAA) approach a group of binary classifiers called Pair Against All (PAA), each of which focuses in separating a class-pair from the rest of the samples. The PAA choice is done because

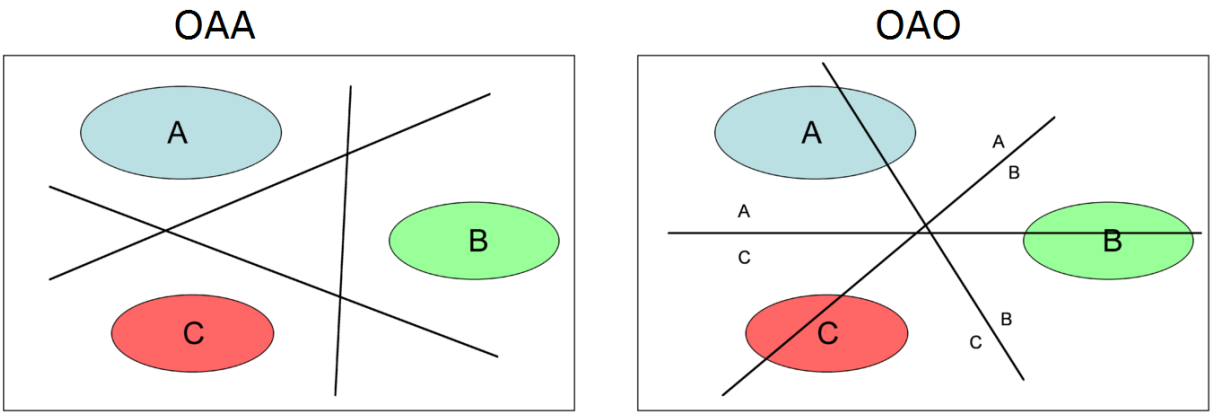

Figure 6-1: Example of OAO and OAA in a three classes problem with their associated classification boundaries.[1]

class pairs are more likely to have common biological features than larger class groups and it is common to find couples of variants of the same disease inside a microarray experiment.

The OAA+PAA algorithm has been tested on seven publicly available datasets through 50 run Monte Carlo simulations. Its performances have been compared with state of the art alternatives, showed in [119], where both the OAA approach and a state of the art ECOC algorithm applying Low Density Parity Check (LDPC) codes are studied with the application of linear SVM as classification algorithm.

6.1 ECOC algorithms and the OAA + PAA algorithm

In this section, the Error Correcting Output Coding application for microarray multiclass classification is discussed and the proposed OAA+PAA algorithm is detailed. The multiclass problem is addressed as a generalization of the two-class scenario in which multiple binary classifiers are used to obtain a final estimation. The two most common approaches are One Against All (OAA) and One Against One (OAO) [80, 38, 106]. In the OAA approach, M binary classifiers are trained, each one separating samples of one class from the rest of the samples. The final decision on the assignment of each sample is determined by a combination of the M outputs. In the OAO approach, $M(M-1)/2$ classifiers are trained, one for each possible class pair without considering samples from

[1] Images from:`http://courses.media.mit.edu/2006fall/mas622j/Projects/aisen-project/`

Table 6.1: Example of the ECOC representation of One Against All (OAA) classification in a 4 class case. Each bit is the output as a classifier separating one class from the rest.

Codeword:	OAA_1	OAA_2	OAA_3	OAA_4
Class 1	1	0	0	0
Class 2	0	1	0	0
Class 3	0	0	1	0
Class 4	0	0	0	1

the other classes. The class assignment is done on the basis of the partition of the decision space resulting from the combination of $M(M-1)/2$ produced boundaries. These two approaches are commonly used for multiclass classification with fairly good performances [128, 119] and a graphical representation of the difference between OAO and OAA is shown in Figure 6-1. It can be observed how the classification boundaries differ between the two cases and how OAO considers only a class-pair to define a boundary, rather than the whole samples population.

An interesting branch of multiclass classification approaches applies data transmission algorithms for the sample classification [119, 37]. These algorithms are called Error Correcting Output Codes (ECOC) algorithms.

The general approach compares the sample classification using N binary classifiers as a transmission of N bit codeword over a noisy channel. Each binary classifier is the receiver for a one of the N bits of the codeword. The sample class is then assigned depending on the received bits. With this parallelism, data transmission solutions can be adopted to improve the “bit error rate" such as error correcting codes.

In [119], recursive Low Density Parity Check (LDPC) codes have been implemented to code the M classes in N-bits codewords. The application of LDPC codes for the multiclass classification is due to their outstanding performances in the data transmission field [119], where they can approximate the Shannon limit. These codes showed very low bit error rate when used in the actual data transmission and are a great choice for that task, but their application to the sample classification needs to take into account some issues. First of all there is the error independence assumption, which assumes that errors on different bits are independent. This assumption is not true because here bits are connected to the sample classification [118]. Furthermore, LDPC codes are block codes which showed

good results for long codewords [118], thus a direct LDPC application for the microarray classification task would imply the training of thousands of classifiers, making their use unpractical. A LDPC related issue is the code-table generation because there is no unique and fast way to obtain them. These aspects are addressed in [118], where a recursive way to produce LDPC codes is studied and applied to the multiclass case.

Here, an alternative ECOC approach is presented, dealing with an additional issue of error correcting block codes: the equality of the binary classifier partitions. The common ECOC approach consists in building a code table relating each of the M sample classes to a N bit codewords to produce a suitable binary matrix (e.g. Hamming code restrictions or LDPC restrictions). This approach works well for data transmission but it does not take into account the aim of the classification task which is to distinguish among elements pertaining to different classes. In the code matrix generation, all the class partitions are equally suitable. A binary classifier separating one class from the rest can be chosen in the code table generation with the same probability as a classifier separating three classes with scarce biological relation from the rest. This feature can lead to very interesting numerical code tables but it does not translate into the expected error correcting improvements when classifying microarray samples [118, 106].

In the proposed approach, a simple error correcting scheme is proposed by adding redundancy to the OAA approach, which is the simplest ECOC approach and whose code table is represented in Table 6.1. The redundancy is obtained through multiple binary classifiers with class partitions more likely to be significant from a biological point of view than those obtained with LDPC codes or other more elaborate algorithms. The presented algorithm adds to the OAA approach a group of binary classifiers called Pair Against All (PAA), each of which focuses on separating a class-pair from the rest. Advantages of such a choice are in the simplicity of the code table generation, as a difference with respect to LDPC codes where the table generation is a complex process, and in the choice of possibly more significant class partitions. Limiting the possible binary partitions to single classes or pairs of classes reduces the risk of choosing meaningless partitions, which should result in the development of more reliable classifiers.

The PAA choice is done because class pairs are more likely to have common biological

Table 6.2: Code table for the OAA+PAA approach in a four classes scenario. There are four codewords of 10 bits, corresponding to the OAA case plus one bit for each class pair.

Bits	OAA_1	OAA_2	OAA_3	OAA_4	$PAA_{1,2}$	$PAA_{1,3}$	$PAA_{1,4}$	$PAA_{2,3}$	$PAA_{2,4}$	$PAA_{3,4}$
Cl. 1	1	0	0	0	1	1	1	0	0	0
Cl. 2	0	1	0	0	1	0	0	1	1	0
Cl. 3	0	0	1	0	0	1	0	1	0	1
Cl. 4	0	0	0	1	0	0	1	0	1	1

features than larger class groups, and it is common to find couples of variants of the same disease inside a microarray experiment. Binary partitions grouping unrelated classes can lead to the production of poor classifiers because the two partitions are not well separable, thus reducing the effectiveness of the code table redundancy. If some of the codeword bits are not trustworthy the class assignation is less likely to produce correct outcomes. The OAA+PAA for M different classes produces a code table with M lines, one for each class, formed of $M + M(M-1)/2$ bits. The codeword length is determined by the M bits deriving from the OAA approach, plus one bit for each possible class pair ($M(M-1)/2$). An example of how the code table is formed in a four classes case is shown in Table 6.2. As it can be observed, the code table from Table 6.2 includes the OAA code table, represented in Table 6.1.

In the proposed approach, each bit is received by a different classifier, built with the algorithm introduced in Section 4.1.1 with the IFFS feature selection algorithm. Each classifier can output a hard decision (i.e. a binary output of 1 or 0) or a soft estimation, each bit is a real value $\in [0,\ 1]$ representing estimated a posteriori probabilities from the LDA classifier. The code table represents the coding step of each sample before the transmission, while the decoding phase consists in receiving each one of the transmitted bits and in assigning an estimated codeword to each received block of bits. For each classified sample, a N dimensional word is received and the final class assignation depends on the distance of the received word from each one of the codewords in the code table.

In practice, assume that $\underline{x}_i$ is the produced word corresponding to the classification of the i^{th} sample, whose actual class is $Y(i) \in [1, \ldots, M]$. The decoding process can be seen as a function $f(\underline{x}_i) \rightarrow \hat{Y}(i)$ assigning an estimated class to the sample. The classification is correct if $Y(i) = \hat{Y}(i)$, otherwise an error is produced. The class estimation is obtained assigning the class whose codeword has the smallest distance from the received word:

$\hat{Y}(i) = min_{j \in [1,...,M]} \|\underline{x}_i - \underline{c}_j\|_1$, where $\underline{c}_j$ is the codeword corresponding to the j^{th} class. If the classifier output is a hard decision, the distance is a Hamming distance. Otherwise, if the output is a soft distance like a posteriori probability, the distances can be measured with L1 or L2 norm. More precisely, any N dimensional distance can be adopted to see whether it introduces some changes in the final output. In this work the hard decision has been paired with the Hamming distance and the soft decision case has been studied applying L1 and L2 distances.

6.2 Experimental Protocol

To assess the classification performance of the proposed algorithm in its two variants, Hard and Soft decision, the multiclass classifiers have been evaluated by means of a Monte Carlo simulation over 7 publicly available datasets described in Section 6.2.1. The results are then compared to those presented in [119]. Based on [39, 7] and similarly to what has been done in [119], 50 Monte Carlo 4:1 (4/5 for training and 1/5 for testing) partitions of the available data were considered. For each iteration, the single bit classifiers have been built up to 15 features as in Section 4.2. Afterwards the mean values for each feature number are measured and the best result is kept as performance level potential.

The performance is measured as the mean error rate predicting the independent test set along the 50 iterations of the Monte Carlo simulation. Inside each binary classifier training, a 10 fold cross validation process has been adopted.

The OAA+PAA algorithm has been studied in three variants, the Hard decision with Hamming distance ($OAA + PAA\_Hard$) the Soft decision version adopting the L1 distance in the class assignation task ($OAA+PAA\_L1$) and the Soft decision version adopting the L2 distance in the class assignation task ($OAA + PAA\_L2$). The simpler OAA and OAO approaches have been tested too since the focus of the experimental evaluation is to compare the performance of OAA+PAA with baseline methods.

Table 6.3: Brief microarray datasets description.

Name	Samples	Genes	Classes
SRBCT	63	2308	4
	http://research.nhgri.nih.gov/microarray/Supplement/index.html		
Brain	42	5597	5
	http://www.broadinstitute.org/mpr/CNS/		
NCI60	61	5244	5
	http://genome-www.stanford.edu/nci60		
Staunton	60	5726	9
	http://www.gems-system.org/		
Su	174	12533	11
	http://www.gems-system.org/		
GCM	190	16063	14
	http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi		
GCM RM	123	7129	11
	http://expression.washington.edu/publications/kayee/shrunken_centroid/		

6.2.1 The analyzed microarray datasets

Seven cancer microarray data sets were used in the evaluation of the analyzed multiclass algorithms. They are called Small Round Blue Cell Tumor dataset (SRBCT), the Brain dataset, the NCI60 dataset, the Staunton dataset, the Su dataset, the GCM dataset and the GCM RM dataset, derived from the GCM dataset with the purpose of improving multiclass classification with variability estimates of repeated gene expression measurements.

For a more detailed description of the datasets, the class distribution and the data preprocessing steps, refer to [119]. A basic description of the dataset composition including the sample number, the number of adopted genes, the class number and a public link to access to the dataset are given in Table 6.3.

6.3 Results

In this section, the experimental results are shown and discussed. Table 6.4 presents the mean Monte Carlo results over the seven datasets for the alternatives studied in this work, OAA, $OAA + PAA\_Hard$, $OAA + PAA\_L1$ and $OAA + PAA\_L2$, compared with the results obtained in [119]. The results from [119] are divided by those obtained with an OAA approach and those obtained adopting a recursive LDPC scheme for microarray classification. The differences between the OAA based method from [119] and the OAA baseline method tested in this work lie in the feature set that does not include metagenes in [119] OAA algorithm, in the classifier (LDA vs SVM) and in the feature selection algorithm to build the classifier, making of the two OAA based algorithm significantly different. The

Table 6.4: Experimental prediction error rates over the seven datasets.

Method	Brain	NCI60	SRBCT	Su
OAA	25.16%	41.37%	1.73%	9.22%
OAO	21.33%	38.50%	2.27%	12.29%
OAA+PAA_ Hard	19.83%	29.87%	2.00%	6.48%
OAA+PAA_ L1	18.67%	28.37%	0.55%	4.19%
OAA+PAA_ L2	18.67%	30.38%	0.67%	4.14%
[119] OAA	12.5%	23.08%	0.00%	8.57%
[119] LDPC	12.5%	30.77%	0.00%	8.57%
Method	Staunton	GCM RM	GCM	Mean
OAA	56.75%	5.91%	38.78%	25.56%
OAO	45.88%	7.16%	34.24%	23.10%
OAA+PAA_ Hard	41.25%	0.75%	24.26%	17.78%
OAA+PAA_ L1	37.75%	0.54%	20.17%	15.75%
OAA+PAA_ L2	37.75%	0.40%	20.17%	16.03%
[119] OAA	46.15%	0.00%	28.63%	16.82%
[119] LDPC	46.15%	0.00%	36.24%	19.07%

proposed algorithms in [119] are recent state of the art alternatives with good prediction results over a wide variety of datasets. Furthermore the validation procedure is clear and detailed, allowing a realistic error estimation thanks to the Monte Carlo simulation on independent test sets.

Table 6.4 indicates the mean prediction error rate for each dataset. In the last column, the mean error rate across all datasets is given to have a global indicator of the prediction ability of the different algorithms.

From Table 6.4 it can be observed how the proposed algorithm *OAA+PAA_L1* manages to obtain the smallest mean prediction error. The best mean error rate result is 15.75%, while the second best result comes from the *OAA+PAA_L2* , while the state of the art alternative from [119] implementing the *OAA* algorithm is third in terms of average error rate. As previously mentioned, the differences in error rates between the current *OAA* results and the [119] *OAA* results are due to the different feature selection algorithms, feature sets and used classification algorithms.

The proposed ECOC algorithm, *OAA+PAA*, is useful as a general method for multiclass classification since it consistently performs better than the *OAA* alternative, reducing the mean error rate of almost ten percent. This result is obtained with both the *OAA+PAA* implementations adopting Hard and Soft decision. Furthermore, it can be observed how using soft decision helps for the class assignment, since the *OAA+PAA_Hard*

consistently obtains worse values than the *OAA+PAA_L1* or *OAA+PAA_L2* . This result agrees with the data transmission parallelism, where the use of soft decision is generally better than relying on hard decision. An important difference of the *OAA+PAA* algorithm with respect to the recursive LDPC codes adopted in [119] is that it improves the *OAA* performance with equal experimental conditions (i.e. same feature selection and classification algorithms in the binary classifiers training).

6.4 Summary

In this chapter, a new algorithm for multiclass classification within the ECOC framework has been introduced. It addresses the issue of ECOC algorithms that consider equally probable all the possible class partitions. Here, it is proposed to restrict the partitions to single classes and class pairs.

The *OAA+PAA* algorithm has been tested on seven publicly available datasets and it has been compared to results obtained with the baseline OAA approach and with state of the art algorithms from [119] applying LDPC codes for multiclass classification.

The results showed how the *OAA+PAA* consistently outperforms the simple OAA in all the analyzed datasets. The performance improvement is due to the redundancy provided by the algorithm itself, and this is a difference with respect to other ECOC approaches that did not obtain substantial performance improvement when compared to OAA [119]. Applying *OAA+PAA* led to improve the best overall results when compared to [119], thus providing a valid alternative for the multiclass classification.

Chapter 7

Conclusions

In this chapter, an overview of the contributions of this thesis to the microarray data classification problem is detailed, with conclusions and a discussion about future research directions. The objective for this thesis was to develop algorithms for microarray data classification pursuing prediction accuracy, results robustness and biological interpretability. The thesis premise is that with algorithms and techniques from the signal processing world it is possible to develop prediction models for high throughput biological data like microarrays. For that matter, different strategies to tackle the classification problem have been developed and tested, showing how such a premise is right and that it is worth working on.

In Section 7.1, the initial research problem is reviewed together with the reasons and opportunities that motivated the whole research process. In Section 7.2, the thesis contributions are collected and detailed. Finally in Section 7.3, the overall thesis conclusions are drawn and the future research directions are suggested.

7.1 Microarray analysis: intersection between biology and signal processing

High-throughput data technologies are the current paradigm in genetic research and microarrays are the most prominent example. Microarrays contributed in a determinant way shift the gene-expression based research from *hypothesis-based* to *data-based* by pro-

viding the researchers with a huge amount of data from which to extract the relevant information.

With this new type of data, different problems arose since the classical statistical analysis tools are not suited to analyze this kind of matrices where noise is not negligible and where there are more variables than samples. The research community has addressed the microarray data analysis problem in a trial/error fashion, by applying algorithms that worked in other domains to probe their validity with the new kind of data and extract some sense about which techniques should be used or not. Through the years, a great deal of work has been dedicated to this task of developing microarray classifiers with several approaches, varying from statistical test, to deterministic algorithms, to neural network approaches and so on. Even if microarrays are considered as a consolidated research technology nowadays and the trends in high-throughput data analysis are shifting towards new technologies like Next Generation Sequencing (NGS) [102], an optimum method for sample classification has not been found yet.

This thesis went in the direction of improving the current state of the art in microarray classification and of contributing to understand how signal processing techniques can be developed and applied to analyze microarray data. The goal of building a classification framework needs an exploratory work in which algorithms are constantly tried and adapted to the analyzed data. With this in mind, a signal processing approach to this task has been tested, because many algorithms in the signal processing field exist that try to make sense out of vast amount of data and techniques exist to make order out of chaos.

To address the microarray classification task, three key data characteristics have been detected at first that contribute to the problem complexity:

- High feature set dimension with respect to the sample number also known as curse of dimensionality [11];
- Lack of a priori known data structural relations;
- Residual measurement noise even after applying normalization techniques.

The developed algorithms and classification frameworks in this thesis tackle the problem

with two essential elements. The first one is to deal with the lack of a priori structure by inferring a data-driven structure with unsupervised hierarchical clustering tools. The second key element is a proper feature selection tool to produce a classifier as an output and to reduce the overfitting risk. The pursued goal, towards which all the research work has been developed is a classifier with high prediction accuracy, with stable performances and able to output interpretable solutions.

7.2 Contributions

In this section, the obtained results throughout the thesis are analyzed in terms of contribution to state of the art knowledge about microarray classification.

7.2.1 Hierarchical structure and metagenes

The first key element that has been introduced is the structure inference from the data applying the hierarchical clustering algorithms derived from Treelets [78]. The obtained output is a binary tree iteratively merging the two most similar features and producing new features called metagenes.

Several alternative methods to the original one proposed in [78] have been tested, changing either the similarity metric to merge the feature, comparing Euclidean distance with Pearson correlation, or the way that two feature are merged, comparing local PCA with a Haar basis decomposition. The output features, the metagenes, are linear combination of similarly behaving genes. In this phase, almost no assumption is made on the data itself to infer a structure except the way to compare the similarity between features.

The outcome from the hierarchical clustering phase and metagene generation is to increase the available feature space by providing new features able to capture the common behavior of related genes, thus obtaining a noise reduction effect which should lead to better predictions.

The usefulness of metagenes has been evaluated comparing all the alternatives by including them in classification experiments. The chosen comparison metric has been the predictive ability measured in terms of Matthews Correlation Coefficient, MCC [89],

when analyzing publicly available data from MAQC study [112]. About the metagenes, they have proven to be useful for classification, allowing to reach better prediction results than using only the gene expression profiles. In all the experiments in Chapters 4 and 5, including metagenes in the classification framework led to better prediction rate than using individual genes only. Among all the compared alternatives, the original Treelets method was superior than the Euclidean distance alternative as shown in Section 4.2.2, while, on the other hand, generating metagenes with Haar basis decomposition proved to reach better overall MCC than the original PCA implementation.

Publications: The hierarchical structure generation has been described jointly with the different feature selection schemes in the following works: [17, 15, 1]

7.2.2 Binary classification

The main task for this thesis work was to develop effective binary classification tools for microarray data. In this direction, once the metagenes have been generated, the key element provided to complete the classification framework was a proper feature selection task. The global classification framework is then composed of a first block in which the metagenes are created, followed by a feature selection block in which features are selected.

For the feature selection, two alternative approaches have been studied: the first one is a modification of the IFFS algorithm [94] as a wrapper feature selection, while the second approach involved an ensemble learning focus.

To obtain good results from the IFFS algorithm, this has been adapted to the data characteristics by introducing two elements. The first one is the reliability parameter which allows to have more information about the sample distribution in the training phase. The reliability parameter helps to discern between classifiers obtaining the same error rate, which is a common case when dealing with the sample scarcity issue of microarray datasets. The second one is a score definition rule to choose the selected features for classification which is the way in which features are measured in the selection phase and it is a key factor to choose the right feature. The IFFS framework has been used to compare all the different metagene generation techniques as well as to compare alternative

classification algorithms like Linear Discriminant Analysis and linear-kernel SVM. From this last comparison, it showed how LDA should be preferred to SVM since it obtained better prediction performances analyzing MAQC datasets. As a global result, when IFFS is used jointly with Treelets clustering and LDA classifier, it obtained better results that the alternatives classifying MAQC data [112], improving the current state of the art.

The other studied feature selection approach is based on Ensemble learning techniques. In this direction, different alternatives have been tested as a proof of concept very interesting results. The performed single run experiment with different configurations highlighted how the ensemble feature selection approach allows to significantly improve the state of the art predictive ability when compared to the IFFS results in terms of predictive accuracy. The studied algorithm has been enriched with key elements like the nonexpert notion that allows boosting the performance. Overall, the best results for ensemble feature selection have been obtained with SVM classifier combined with the nonexpert notion introduced in Section 4.3.2.

Publications: Publications: The following publications in international conferences and journals are related to the aforementioned topics [17, 15, 1, 18].

7.2.3 Knowledge integration model for metagene generation

Techniques to infer a hierarchical structure from microarray data combining both numerical information and prior biological information have been described and evaluated with the aim to produce better metagenes and improve results stability and interpretability. They have been compared to state of the art alternatives and to the numerical information only solution from Section 4.2.2 which already showed to have good and robust predictive properties implementing IFFS as feature selection.

The knowledge integration framework has been studied with different implementations, comparing four similarity metrics and two combination rules to merge the numerical correlation and the biological similarity. The rationale behind it is to gather more high-quality external information about the genes so that the hierarchical clustering process can be more meaningful from a biological standpoint. In this way, the metagenes can

summarize the behavior of genes that are similar in both numerical expression and in biological functions.

Monte Carlo experiments on MAQC [112] datasets have been performed to evaluate the resulting algorithms in terms of their predictive ability and biological interpretability of the chosen gene signatures. When compared to the IFFS alternative using numerical data only, including prior knowledge in the metagene generation allows to obtain more stable prediction results and more biologically relevant signatures, all this without reducing the overall mean predictive performance.

Among the studied alternatives, the G-pdf algorithm combining Godall similarity measure with the probability density function equalization consistently obtained the best performances also when compared to state of the art alternative from [84]

As a general observation, a proper knowledge integration framework should be preferred to the bare numerical treelets when possible, since it obtains more robust and interpretable results for classification.

Publications: The result of this work has been published in [62].

7.2.4 Multiclass classification

Due to the good results for binary classification, the IFFS based framework has been generalized to work with multiclass problems. For that, a new algorithm for multiclass classification within the Error Correcting Output Coding framework [37] has been introduced,. It addresses the issue of ECOC algorithms of considering equally probable all the possible class partitions by limiting the partitions to single classes and class pairs and it has been named One Against All + Pair Against All: $OAA+PAA$.

The $OAA+PAA$ algorithm has been tested on seven publicly available datasets and it has been compared with results obtained with the baseline OAA and OAO approaches, as well as with state of the art algorithms from [119] applying LDPC codes for multiclass classification.

The proposed algorithm outperformed the baseline alternatives, showing how it can improve simpler algorithms. Such an improvement is due to the provided redundancy

from the algorithm by adding the Pair Against All part. This is a key difference with respect to other ECOC approaches that did not manage to substantially improve the performances when compared to the OAA approach [118, 106]. When compared to [119], the *OAA+PAA* algorithm obtained better results, showing how it is a valid alternative for the multiclass classification.

Publications: The result of this work has been published in [2].

7.3 Overview and Next steps

In this section, the overall conclusions from this thesis are detailed, together with intuitions and ideas about future research directions from this work.

The most important element for the whole framework prediction performance is the metagenes generation from gene expression data. In all the performed experiments, introducing metagenes consistently led to improved performances for classification. These newly introduced features have more reproducible behaviors than single genes between training and validation sets, supporting the statement that metagenes can reduce the residual noise on gene-expression. As a desirable development from the metagenes introduction, it is probably worth trying to further exploit the obtained hierarchical structure because in this thesis all the metagenes are considered equal, regardless of how many genes they merge. Making a better use of the inferred tree, for example to early eliminate some metagenes because of their unreliability (e.g. the tree-root or the highest level metagenes that combine thousands of features), or for example to drive the exploration of candidate regulatory genes for certain problems, could benefit the results. The tree structure is an asset that has not been used and that may help in making more sense out of the data.

About metagenes and how it is possible to improve them as well as the inferred structure, it has been shown how including prior biological information led to an overall improvement of the results. The predictive accuracy remained unaltered, but both the predictive stability and the interpretability are better than without it. These results can be interpreted how including data sources external from the gene-expression, helps in gaining more insight about the hidden data structure. A future work direction is therefore

to build systems integrating more and more information, in an automatic way to better define the tree construction. From automatic processing of different information sources, like the used gene ontology databases but also from natural language processing tools, it could be possible to extract meaning that otherwise would not be possible and it is an opportunity to integrate different signal processing areas.

Going from the structure to feature selection, feature selection algorithms are a key factor for the performances. Two alternative methods proved to reach good results from very different perspectives. On one side, the results using the wrapper algorithm led to predictive performance that are comparable with the best state of the art alternatives, with good performance stability and results interpretability. On the other side, applying ensemble feature selection led to a remarkable performance improvement, at a price of less interpretable results. Nothing can be said yet about stability because of the difficulty to design a proper experiment. Even if different, both methods share one common feature that should be remarked. In the development phase, both algorithms have been tailored to the data rather than simply being applied as is. For the feature selection task, future research works could be dedicated to deepen the knowledge of the ensemble learning potential, to the creation of experiments to assess the stability and how to incorporate the notion of stability in the selection process. Between ensemble learning and IFFS, the former has more potential to grow and explore.

In this thesis, the multiclass scenario has been considered too with the study of a new algorithm with interesting performances. Although we have proposed an improved algorithm compared to the the state of the art, the binary class classification should be preferred for further studies. The main reason is that there still is lot of room for improvement in that field and because the multiclass case could be reduced to multiple binary comparisons.

Finally, as a global conclusion, the application of signal processing techniques to the analysis of biological data like microarrays proved to be useful and interesting. It has been possible to develop tools comparable, and even better, than state of the art alternatives in both the binary and multiclass cases. The proposed frameworks led to good results in terms of predictive ability, predictive stability and results interpretability, meeting the

original thesis goal. Even though the theoretical optimum is still far from being reached, it has been possible to test some key elements like metagenes and the feature selection that are worth to be further studied.

Bibliography

[1] *Microarray classification with hierarchical data representation and novel feature selection criteria*, Larnaca, Cyprus, 11/2012 2012.

[2] *Multiclass cancer microarray classification algorithm with Pair-Against-All redundancy*, Washington, DC, USA, 12/2012 2012.

[3] G. Alterovitz and M. F. Ramoni. *Knowledge based bioinformatics : from analysis to interpretation*. John Wiley & Sons, Chichester, West Sussex, U.K., 2010.

[4] M. Anderberg. *Cluster analysis for applications*. Probability and mathematical statistics. Academic Press, 1973.

[5] J. A. Anderson. *An Introduction to Neural Networks*. The MIT Press, Mar. 1995.

[6] M. Ashburner. Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

[7] F. Azuaje. Genomic data sampling and its effect on classification performance assessment. *BMC Bioinformatics*, 4:5, 2003.

[8] R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer. A new ensemble diversity measure applied to thinning ensembles. In T. Windeatt and F. Roli, editors, *Multiple Classifier Systems*, volume 2709 of *Lecture Notes in Computer Science*, pages 306–316. Springer, 2003.

[9] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, 2000.

[10] R. Bellman and R. Kalaba. On adaptive control processes. *Automatic Control, IRE Transactions on*, 4(2):1–9, 1959.

[11] R. E. Bellman. *Dynamic Programming*. Dover Publications, Incorporated, 2003.

[12] P. Bellot Pujalte. Study of gene expression representation with treelets and hierarchical clustering algorithms. 2011.

[13] D. M. Bolser, P.-Y. Chibon, N. Palopoli, S. Gong, D. Jacob, V. D. D. Angel, D. Swan, S. Bassi, V. GonzÃ¡lez, P. Suravajhala, S. Hwang, P. Romano, R. Edwards, B. Bishop, J. Eargle, T. Shtatland, N. J. Provart, D. Clements, D. P. Renfro, D. Bhak, and J. Bhak. Metabase - the wiki-database of biological databases. *Nucleic Acids Research*, 40(Database-Issue):1250–1254, 2012.

[14] S. Boriah, V. Chandola, and V. Kumar. Similarity measures for categorical data: A comparative evaluation. In *SDM*, pages 243–254. SIAM, 2008.

[15] M. Bosio, P. Bellot, P. Salembier, and A. Oliveras-Vergés. Gene expression data classification combining hierarchical representation and efficient feature selection. *Journal of Biological Systems*, 20(04):349–375, 2012.

[16] M. Bosio, P. Bellot Pujalte, P. Salembier, and A. Oliveras. Feature set enhancement via hierarchical clustering for microarray classification. In *IEEE International Workshop on Genomic Signal Processing and Statistics (GENSIPS)*, San Antonio TX, USA, December 2011. IEEE.

[17] M. Bosio, P. Bellot Pujalte, P. Salembier, and A. Oliveras-Verges. Feature set enhancement via hierarchical clustering for microarray classification. In *Genomic Signal Processing and Statistics (GENSIPS), 2011 IEEE International Workshop on*, pages 226–229. IEEE, 2011.

[18] M. Bosio, P. Salembier, A. Oliveras, and P. Bellot Pujalte. Ensemble feature selection and hierarchical data representation for microarray classification. In *13th IEEE International Conference on BioInformatics and BioEngineering BIBE*, Chania, Crete, 11/2013 2013. 13th IEEE International Conference on BioInformatics

and BioEngineering„ 13th IEEE International Conference on BioInformatics and BioEngineering,.

[19] U. Braga-Neto. Fads and fallacies in the name of small-sample microarray classification - a highlight of misunderstanding and erroneous usage in the applications of genomic signal processing. *Signal Processing Magazine, IEEE*, 24(1):91 –99, jan. 2007.

[20] U. Braga-Neto and E. Dougherty. Bolstered error estimation. *Pattern Recognition*, 37](6):1267 – 1281, 2004.

[21] U. M. Braga-Neto and E. R. Dougherty. Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20(3):374–380, 2004.

[22] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[23] L. Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–215, 2001.

[24] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Statistics/Probability Series. Wadsworth Publishing Company, Belmont, California, U.S.A., 1984.

[25] D. Calvo-Dmgz, J. F. Gálvez, D. Glez-Peña, S. G. Meire, and F. Fdez-Riverola. Using variable precision rough set for selection and classification of biological knowledge integrated in dna gene expression. *J. Integrative Bioinformatics*, 9(3), 2012.

[26] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[27] E. Chen, C. Tan, Y. Kou, Q. Duan, Z. Wang, G. Meirelles, N. Clark, and A. Ma'ayan. Enrichr: interactive and collaborative html5 gene list enrichment analysis tool. *BMC Bioinformatics*, 14(1):128, 2013.

[28] X. Chen and L. Wang. Integrating biological knowledge with gene expression profiles for survival prediction of cancer. *Journal of Computational Biology*, 16(2):265–278, 2009.

[29] X. Chen, L. Wang, J. D. Smith, and B. Zhang. Supervised principal component analysis for gene set enrichment of microarray data with continuous or survival outcomes. *Bioinformatics*, 24(21):2474–2481, 2008.

[30] L. Chin, W. C. Hahn, G. Getz, and M. Meyerson. Making sense of cancer genomic data. *Genes & Development*, 25(6):534–555, 2011.

[31] D. Coppola, M. Nebozhyn, F. Khalil, H. Dai, T. Yeatman, A. Loboda, and J. J. MulÃ©. Unique ectopic lymph node-like structures present in human primary colorectal carcinoma are identified by immune gene array profiling. *The American Journal of Pathology*, 179(1):37 – 45, 2011.

[32] R. DÃaz-Uriarte and S. A. de AndrÃ©s. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7:3, 2006.

[33] K. Deb and A. Reddy. Reliable classification of two-class cancer data using evolutionary algorithms. *BioSystems*, 2003.

[34] S. Deegalla and H. Boström. Classification of microarrays with knn: comparison of dimensionality reduction methods. In *Proceedings of the 8th international conference on Intelligent data engineering and automated learning*, IDEAL'07, pages 800–809, Berlin, Heidelberg, 2007. Springer-Verlag.

[35] M. Dettling and P. Bühlmann. Finding predictive gene groups from microarray data. *J. Multivar. Anal.*, 90:106–131, July 2004.

[36] T. G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, pages 1–15, London, UK, UK, 2000. Springer-Verlag.

[37] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[38] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, 2001.

[39] S. Dudoit, J. Fridlyand, and T. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 457(97):77–87, 2002.

[40] G. D.W. A new similarity index based on probability. *Biometrics*, 1966.

[41] R. Edgar, M. Domrachev, and A. E. Lash. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, 2002.

[42] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.

[43] P. Espejo, S. Ventura, and F. Herrera. A survey on the application of genetic programming to classification. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(2):121 –144, march 2010.

[44] P. Farmer, H. Bonnefoi, P. Anderle, D. Cameron, P. Wirapati, P. Wirapati, V. Becette, S. André, M. Piccart, M. Campone, E. Brain, G. Macgrogan, T. Petit, J. Jassem, F. Bibeau, E. Blot, J. Bogaerts, M. Aguet, J. Bergh, R. Iggo, and M. Delorenzi. A stroma-related gene signature predicts resistance to neoadjuvant chemotherapy in breast cancer. *Nature medicine*, 15(1):68–74, Jan. 2009.

[45] J.-F. Fontaine, F. Priller, A. Barbosa-Silva, and M. A. Andrade-Navarro. GÃ©nie: literature-based gene prioritization at multi genomic scale. *Nucleic Acids Research*, 39(Web-Server-Issue):455–461, 2011.

[46] G. Forman. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305, Mar. 2003.

[47] K. Fukunaga. Introduction to statistical pattern recognition (2nd ed.). 1990.

[48] G. M. Furnival and R. W. Wilson. Regression by leaps and bounds. 1974.

[49] D. Glez-Pena, M. Peerez-Fernandez, M. Reboiro-Jato, F. Fdez-Riverola, M. Perez, and F. Diaz. Incorporating biological knowledge to microarray data classification through genomic data fusion. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–8, 2010.

[50] H. Gohlmann and W. Talloen. Gene expression studies using affymetrix microarrays. 2009.

[51] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.

[52] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 1999.

[53] Y. Guo, T. Hastie, and R. Tibshirani. Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, 8(1):86–100, Jan. 2007.

[54] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, Mar. 2003.

[55] A. Haar. *Zur Theorie der orthogonalen Funktionensysteme*. Georg-August-Universitat, Gottingen., 1909.

[56] D. Hanisch, A. Zien, R. Zimmer, T. Lengauer, and Thomas. Co-clustering of biological networks and gene expression data, 2002.

[57] T. Hastie, R. Tibshirani, D. Botstein, and P. Brown. Supervised harvesting of expression trees. *Genome Biology*, 2(1), 2001.

[58] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009.

[59] J. Hua, W. Tembe, and E. R. Dougherty. Performance of feature-selection methods in the classification of high-dimension data. *Pattern Recognition*, 42(3):409–424, 2009.

[60] D. W. Huang, B. T. Sherman, and R. A. Lempicki. Systematic and integrative analysis of large gene lists using david bioinformatics resources. *Nature Protocols*, 4(1):44–57, 2009.

[61] E. Huang, S. H. Cheng, H. Dressman, J. Pittman, M. H. Tsou, C. F. Horng, A. Bild, E. S. Iversen, M. Liao, C. M. Chen, M. West, J. R. Nevins, and A. T. Huang. Gene expression predictors of breast cancer outcomes. *The Lancet*, 361(9369):1590 – 1596, 2003.

[62] IEEE EMBS. *Hierarchical Clustering Combining Numerical and Biological Similarities for Gene Expression Data Classification*, Osaka, Japan, 07/2013 2013. IEEE EMBS.

[63] D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 16:1370–1386, 2004.

[64] I. T. Jolliffe. *Principal Component Analysis*. Springer, second edition, Oct. 2002.

[65] M. Kanehisa and S. Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res*, 28(1):27–30, January 2000.

[66] E. Keedwell and A. Narayanan. Genetic algorithms for gene expression analysis. In S. Cagnoni, C. Johnson, J. Cardalda, E. Marchiori, D. Corne, J.-A. Meyer, J. Gottlieb, M. Middendorf, A. Guillot, G. Raidl, and E. Hart, editors, *Applications of Evolutionary Computing*, volume 2611 of *Lecture Notes in Computer Science*, pages 191–192. Springer Berlin / Heidelberg, 2003. 10.1007/3-540-36605-9_8.

[67] J. H. Kim. Chapter 8: Biological knowledge assembly and interpretation. *PLoS Comput Biol*, 8(12):e1002858, 12 2012.

[68] L. Klebanov and A. Yakovlev. How high is the level of technical noise in microarray data. *Biol. Direct*, page 9, 2007.

[69] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97:273–324, December 1997.

[70] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.

[71] S. Kottah. Histogram based hierarchical data representation for microarray classification. 2012.

[72] L. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2004.

[73] L. Kuncheva and C. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51:181–207, 2003.

[74] C. Lai, M. Reinders, L. van't Veer, and L. Wessels. A comparison of univariate and multivariate gene selection techniques for classification of cancer datasets. *BMC Bioinformatics*, 7(1):235, 2006.

[75] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, and A. Nowe. A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 9(4):1106–1119, 2012.

[76] S. M. Leach, A. Gabow, L. Hunter, and D. Goldberg. Assessing and combining reliability of protein interaction sources. In R. B. Altman, A. K. Dunker, L. Hunter, T. Murray, and T. E. Klein, editors, *Pacific Symposium on Biocomputing*, pages 433–444. World Scientific, 2007.

[77] S. M. Leach, H. J. Tipney, W. Feng, W. A. B. Jr., P. Kasliwal, R. P. Schuyler, T. Williams, R. A. Spritz, and L. Hunter. Biomedical discovery acceleration, with applications to craniofacial development. *PLoS Computational Biology*, 5(3), 2009.

[78] A. B. Lee, B. Nadler, and L. Wasserman. Treelets - an adaptive multi-scale basis for sparse unordered data. *Annals of Applied Statistics*, 2(2):435–471, 2008.

[79] F. Li and Y. Yang. Analysis of recursive gene selection approaches from microarray data. *Bioinformatics*, 21(19):3741–3747.

[80] T. Li, C. Zhang, and M. Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20:2429–2437, 2004.

[81] Y. Li, S. Gong, and H. M. Liddell. Recognising trajectories of facial identities using kernel discriminant analysis. *Image Vision Comput.*, 21(13-14):1077–1086, 2003.

[82] A. Liekens, J. De Knijf, W. Daelemans, B. Goethals, P. De Rijk, and J. Del-Favero. Biograph: unsupervised biomedical knowledge discovery via automated hypothesis generation. *Genome Biology*, 12(6):R57, 2011.

[83] Q. Liu, A. H. Sung, Z. Chen, J. Liu, X. Huang, and Y. Deng. Feature selection and classification of maqc-ii breast cancer and multiple myeloma microarray gene expression data. *PLoS ONE*, 4(12):e8250, 12 2009.

[84] Q. Liu, A. H. Sung, Z. Chen, J. Liu, X. Huang, and Y. Deng. Feature selection and classification of maqcii breast cancer and multiple myeloma microarray gene expression data. *PLoS ONE*, 4(12):e8250, 12 2009.

[85] S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[86] H. Mann and D. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, pages 50–60, 1947.

[87] T. Marill and D. Green. On the effectiveness of receptors in recognition systems. *Information Theory, IEEE Transactions on,*, 1963.

[88] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[89] B. W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta*, 405(2):442–451, 1975.

[90] Mike Sheppard. Fit all valid parametric probability distributions to data - file exchange - MATLAB central, Apr. 2012.

[91] B. Mirkin. *Mathematical Classification and Clustering*. Mathematics and Its Applications. Kluwer Academic Pubishers, 1996.

[92] T. Moon. *Error Correction Coding: Mathematical Methods and Algorithms*. Wiley, 2005.

[93] S. Nagi, D. K. Bhattacharyya, and J. K. Kalita. Gene expression data clustering analysis: A survey. In *National Conference on Emerging Trends and Applications in Computer Science*, 2011.

[94] S. Nakariyakul and D. P. Casasent. An improvement on floating search algorithms for feature subset selection. *Pattern Recognition*, 42(9):1932 – 1940, 2009.

[95] P. M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. *Computers, IEEE Transactions on*, C-26(9):917–922, 1977.

[96] S. U. D. of Statistics, J. Friedman, and N. S. F. (U.S.). *Regularized Discriminant Analysis*. 1987.

[97] O. Okun, G. Valentini, and M. Re. *Ensembles in Machine Learning Applications*. Studies in Computational Intelligence. Springer, 2011.

[98] H. F. Ong, N. Mustapha, and M. N. Sulaiman. Integrative gene selection for classification of microarray data. *Computer and Information Science*, 4(2):55–63, 2011.

[99] V. P., R. T., and D. K. Survey on clustering algorithms for microarray gene expression data. *European Journal of Scientific Research*, 69:5–20, 2012.

[100] R. Parry, W. Jones, T. Stokes, J. Phan, R. Moffitt, H. Fang, L. Shi, A. Oberthuer, M. Fischer, W. Tong, and M. Wang. k-nearest neighbor models for microarray gene expression analysis and clinical outcome prediction. *Pharmacogenomics J*, 10(4):292–309, 2010.

[101] R. M. Parry, W. Jones, T. H. Stokes, J. H. Phan, R. A. Moffitt, H. Fang, L. Shi, A. Oberthuer, M. Fischer, W. Tong, and M. D. Wang. k-Nearest neighbor models for microarray gene expression analysis and clinical outcome prediction. *The Pharmacogenomics Journal*, 10(4):292–309, Aug. 2010.

[102] E. Pettersson, J. Lundeberg, and A. Ahmadian. Generations of sequencing technologies. *Genomics*, 93(2):105 – 111, 2009.

[103] R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45, 2006.

[104] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recogn. Lett.*, 15(11):1119–1125, Nov. 1994.

[105] A. Rao and A. Hero. Biological pathway inference using manifold embedding. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 5992–5995, 2011.

[106] R. Rifkin and A. Klautau. In defense of one vs all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

[107] D. M. Rocke, T. Ideker, O. G. Troyanskaya, J. Quackenbush, and J. Dopazo. Papers on normalization, variable selection, classification or clustering of microarray data. *Bioinformatics*, 25(6):701–702, 2009.

[108] Y. Saeys, I. n. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, Sept. 2007.

[109] B. Schölkopf and A. J. Smola. *Learning with kernels : support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning. MIT Press, 2002.

[110] E. Serpedin, J. Garcia-Frias, Y. Huang, and U. Braga-Neto. Applications of signal processing techniques to bioinformatics, genomics, and proteomics. *EURASIP Journal on Bioinformatics and Systems Biology*, 2009(1):250306, 2009.

[111] L. Sheng, R. Pique-Regi, S. Asgharzadeh, and A. Ortega. Microarray classification using block diagonal linear discriminant analysis with embedded feature selection. In *ICASSP*, pages 1757–1760. IEEE, 2009.

[112] L. Shi, G. Campbell, W. D. Jones, F. Campagne, and Z. Wen. The microarray quality control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models. *Nature biotechnology*, 28:827–38, 2010 Aug 2010.

[113] E. Smirnov. *On exact methods in systematics*, volume 17. Systematic Zoology, 1968.

[114] A. R. Statnikov, L. Wang, and C. F. Aliferis. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinformatics*, 9, 2008.

[115] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–15550, 2005.

[116] S. Suster and C. A. Moran. Applications and limitations of immunohistochemistry in the diagnosis of malignant mesothelioma. *Adv Anat Pathol*, 13(6):316–29, 2006.

[117] F. Tai and W. Pan. Incorporating prior knowledge of predictors into penalized classifiers with multiple penalty terms. *Bioinformatics*, 23(14):1775–1782, 2007.

[118] E. Tapia, P. Bulacio, and L. Angelone. Recursive ECOC classification. *Pattern Recogn. Lett.*, 31(3):210–215, Feb. 2010.

[119] E. Tapia, L. Ornella, P. Bulacio, and L. Angelone. Multiclass classification of microarray data samples with a reduced number of genes. *BMC Bioinformatics*, 12:59, 2011.

[120] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Elsevier Science, 2008.

[121] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

[122] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Class Prediction by Nearest Shrunken Centroids, with Applications to DNA Microarrays. *Statistical Science*, 18(1):104–117, 2003.

[123] T. Tong and Y. Wang. Optimal Shrinkage Estimation of Variances With Applications to Microarray Data Analysis. *Journal of the American Statistical Association*, 102(477):113–122, Mar. 2007.

[124] S. Vantini, V. Vitelli, É. de France, and P. Zanini. Treelet analysis and independent component analysis of milan mobile-network data: Investigating population mobility and behavior. *Analysis and Modeling of Complex Data in Behavioural and Social Sciences*, page 87, 2012.

[125] V. Vapnik and A. Chervonenkis. A note on one class of perceptrons. *Automation and Remote Control*, 25, 1964.

[126] M. D. Vose. *The Simple Genetic Algorithm: Foundations and Theory*. MIT Press, Cambridge, MA, USA, 1998.

[127] C. Wang, J. Xuan, H. Li, Y. J. Wang, M. Zhan, E. P. Hoffman, and R. Clarke. Knowledge-guided gene ranking by coordinative component analysis. *BMC Bioinformatics*, 11:162, 2010.

[128] S. Wang and X. Yao. Multiclass imbalance problems: Analysis and potential solutions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(4):1119 –1130, aug. 2012.

[129] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping. Use of the zero norm with linear models and kernel methods. *J. Mach. Learn. Res.*, 3:1439–1461, Mar. 2003.

[130] A. Whitney. A direct method of nonparametric measurement selection. *Computers, IEEE Transactions on*, 1971.

[131] F. Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[132] D. R. Wilson and T. R. Martinez. Improved heterogeneous distance functions. *CoRR*, cs.AI/9701101, 1997.

[133] S. Wold, M. Sjostrom, and L. Eriksson. Pls-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems*, 58:109–130, 2001.

[134] X. Xu and A. Zhang. Selecting informative genes from microarray dataset by incorporating gene ontology. In *Bioinformatics and Bioengineering, 2005. BIBE 2005. Fifth IEEE Symposium on*, pages 241–245, 2005.

[135] A. Y. Yang, S. S. Sastry, A. Ganesh, and Y. Ma. Fast l1-minimization algorithms and an application in robust face recognition: A review. In *ICIP*, pages 1849–1852. IEEE, 2010.

[136] P. Yang, Y. H. Yang, B. B. Zhou, and A. Y. Zomaya. A review of ensemble methods in bioinformatics. *Current Bioinformatics*, (5):296–308, 2010.

[137] J. Ye, T. Li, T. Xiong, and R. Janardan. Using uncorrelated discriminant analysis for tissue classification with gene expression data. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 1(4):181–190, 2004.

[138] W.-K. Yip, S. B. Amin, and C. Li. A survey of classification techniques for microarray data analysis. In H. H.-S. Lu, B. Schölkopf, and H. Zhao, editors, *Handbook of Statistical Bioinformatics*, Springer Handbooks of Computational Statistics, pages 193–223. Springer Berlin Heidelberg, 2011. 10.1007/978-3-642-16345-6_10.