



UNIVERSITÉ CATHOLIQUE DE LOUVAIN

INFORMATION AND COMMUNICATION TECHNOLOGIES,

ELECTRONICS AND APPLIED MATHEMATICS (ICTEAM)

MACHINE LEARNING GROUP

Inductive Biases for Stable Feature Selection in High Dimensional Spaces

—

Applications to Gene Profiling and
Diagnosis from DNA Microarrays

Thibault Helleputte

Thesis presented for the Ph.D. degree
in Engineering Sciences

Thesis Jury:

Prof. **Pierre Dupont** (University of Louvain), Advisor

Dr. **Alexandros Kalousis** (University of Geneva, Switzerland)

Dr. **Yvan Saeys** (University of Gent)

Prof. **Etienne Sokal** (University of Louvain), Advisor

Prof. **Michel Verleysen** (University of Louvain), Thesis Committee, President

Prof. **Louis Wehenkel** (University of Liège), Thesis Committee

September 2010

“What shall we use to fill the empty spaces?”

Roger Waters

Acknowledgments

I deeply thank my advisors Pierre Dupont and Etienne Sokal for their commitment to guide my work during the last four years, which count amongst the most intellectually stimulating of my life. I much enjoyed the cross-domain work with Etienne Sokal's team at the medicine faculty. It made my research more concerned by practical applications. The central theme of my thesis made me have far more interactions with Pierre Dupont. By his side, I have learned scientific rigor and care of the details while not losing the broader picture of my research, and its applicative context. I also thank the members of my thesis committee, Michel Verleysen and Louis Wehenkel for their time and advice. They, along with the other members of the jury, Alexandros Kalousis and Yvan Saeys, made lots of constructive comments on a preliminary version of this text, for which I am grateful.

My thanks also go to my direct colleagues Nizar Touleimat, Roman Zakharov, Juan-Pablo Narino-Mendoza and Daniel Hernandez-Lobato, and to all the people at INGI – I cannot list them all here. And what would be INGI without the system administration team, Pierre Reinbold and Nicolas Detienne? Thank you guys for your world-class support. I also thank the Machine Learning Group researchers for the good atmosphere and for the numerous research discussions and seminars we had. Thanks to Guillaume Bernard and Pierre Geurts for the implementation of the random forest and its related feature selection method. Most computations have been run on the FYNU's INGRID cluster thanks to the Institute for Intensive Computation and Mass Storage (CISM) of the UCL (particular thank to Bernard Van Renterghem, Thomas Keutgen and Fabrice Charlier). Thanks to Shane O'Neill for english corrections.

I am grateful to all the people with whom I collaborated. One chapter

of this thesis is dedicated to ensemble methods for classification. This is a joint work with the bioinformatics group of the Department of Plant Systems Biology at Ghent University. I thank Yvan Saeys and Thomas Abeel for this nice collaboration. Many aspects of my work are related to practical applications of my research. I deeply thank Bertrand Bearzatto for all his work on microarrays and the good time we spent together. I also thank all the Cristall team and particularly Françoise Smets, Floriane Andre and Catherine Lombard. I thank Prof. Olivier Feron and Romain Boidot for the collaboration on the Hypoxia project. I also did some work in collaboration with GSK Biologicals. For different aspects of this collaboration, I thank Jean-Louis Ruelle, Swann Gaulis and Jamila Louahed.

I am also deeply grateful to my family. First to my parents Didier and Marie-Françoise for their support during the last 27 years. All those math and english courses were useful, finally, as well as the conviction that there is more in this world than material things. Thanks to my brothers and sisters (in-law or not) Benoit, Diane & Diong, Xavier & Magali, Vincent & Aline for asking me from time to time *"But finally what do you do as a job?"*¹. I also here seize the opportunity to specially thank Xavier for somehow having triggered all that! Many thanks also go to Jean & Rita, for their kindness and their support. You all provide a warm, happy and stimulating environment.

I dedicate this thesis to my wife Anne-Sophie for supporting (and moderating) me in my numerous projects and for making my life as awesome as it is, and to our daughter Clara.

My work has been funded by a FRIA grant (1.E091.07) delivered by the FNRS.

¹I even tend to believe they were sometimes really interested in the answer.

Abstract

In many technological or industrial fields, the amount of high dimensional data is steadily growing. The number of dimensions is however often growing much faster than the number of points available. The field of genomics is a typical illustration of this trend.

This thesis is concerned with classification and feature selection of high dimensional data with a small number of points. This setting makes many machine learning applications subject to the curse of dimensionality. This phenomenon is the problem caused by the exponential increase in volume associated with adding extra dimensions to a (mathematical) space. It makes difficult the estimation of models that will generalize well.

This thesis focuses on the design and application of techniques achieving highly sparse feature selection while allowing the estimation of models with good classification performance in a context where only few points are available, and those points lay in a high dimensional space. It deploys several means of regularization to mitigate the lack of extra samples in that high-dimensional setting.

The text first introduces all the necessary machine learning background and details relevant performance metrics among which balanced classification rate and selection stability, and discusses the relevance of several validation protocols in that context.

The thesis then proposes three feature selection approaches embedded within linear classification models. The first is an ensemble technique for feature selection. It builds a consensus selection from many different selections performed on subsets of the data. This

technique induces an increase in stability and classification performances. The second approach makes possible the incorporation of expert knowledge to bias a multivariate feature selection towards some features while letting the selection depart from that prior knowledge if not improving performances. The method is based on a linear model regularized by an approximation of a zero-pseudo-norm with iterative rescaling of the inputs. The impact on the selection is a clear gain in stability even with a limited amount of prior knowledge with classification performances either improved, or not degraded. A variant of this technique proposes to rely on related datasets to extract that prior knowledge and transfer it to the dataset of interest. In that case, both the stability and the classification performances are shown to be improved on the considered datasets. The third approach goes a step further than transfer learning, and performs feature selection in a multi-task setting. In this case, the selection is performed for several datasets at once, corresponding to different but related tasks. Several variants are tested, each with its specificities. Depending on the applicative context, one can improve within or between-task stability, classification performances, or sparsity. This family of techniques is based on logistic regression and group-regularizers. Those regularizers take the form of $L_{i,j}$ matrix-(pseudo-)norm where $i \in \{0, 1\}$ and $j \in \{1, 2, \infty\}$. An original solver is introduced for the corresponding optimization problem which is linear in terms of the total number of features. Finally, it is shown that this last technique can be applied not only to multi-task but also to multi-class problems (i.e. with more than two classes). In the case of the $L_{1,\infty}$ matrix-norm regularizer, this technique is able to identify a single set of features for all the pairwise class discriminations.

All the proposed methods are tested over several gene expression microarray datasets. This technology measures at once the rate of transcription (the expression) of thousands of genes into mRNA, the intermediate messengers between genes and proteins production. Those datasets are typically made of few samples (a few tens) and thousands of dimensions (genes).

The end of the thesis is dedicated to a case study, where an analysis of gene expression microarray data is performed in the context of the Cristall project. This project is an attempt to identify allergy risk factors in newborns.

Contents

I	Lost in Empty Space	1
1	Introduction	3
1.1	Context	3
1.2	Issues Raised By Microarray Data	6
1.3	Contributions Summary	11
1.4	Thesis Roadmap	15
II	Machine Learning Background	19
2	Classification	21
2.1	Preliminaries	21
2.2	Linear Models	25
2.3	Statistical Learning	26
2.4	Support Vector Machine	33
2.5	Logistic Regression	38
2.6	Tree-Based Ensemble Methods	41
3	Feature Selection	47
3.1	Feature Selection	47
3.2	Filters	51
3.3	Wrappers	55
3.4	Embedded Methods	57
4	Evaluation Criteria	69
4.1	Metrics for Selection Quality	69
4.2	Classification Performances	72
4.3	Combining Several Criteria	75

5	Validation Protocols	77
5.1	Validation in High Dimensional Settings	77
5.2	Cross-Validations	79
5.3	Bootstraps	80
5.4	Resamplings	92
5.5	Random Experiments	94
5.6	The Choice of a Protocol	95
6	Preprocessing	97
6.1	Normalization	97
6.2	Non-Specific Filtering	101
III	Inductive Biases for High-Dimensional Learning	107
7	Ensemble Feature Selection	109
7.1	Selection within SVM	111
7.2	Ensemble Feature Selection	113
7.3	Experiments and Results	114
7.4	Discussion	123
8	Partially Supervised Feature Selection	125
8.1	Partially Supervised AROM	126
8.2	Experiments and Results	130
8.3	Discussion	141
9	Transfer Learning for Feature Selection	143
9.1	Transfer Learning with PS- L_2 -AROM	146
9.2	Experiments and Results	148
9.3	Discussion	156
10	Feature Selection for Multi-Task Classification	161
10.1	Multi-Task Regularizers for Generalized Linear Models	164
10.2	Multi-Task Penalized Logistic Regression	167
10.3	Optimization of a Sparse Multi-Task Logistic Regression	168
10.4	Experiments and Results	174
10.5	Discussion	182
10.6	Feature Selection for Multi-Class Classification	183

11 Conclusion	191
11.1 Contributions	191
11.2 Perspectives	193
 IV Microarray Case Study	 199
 12 Cristall	 201
12.1 Project Description	202
12.2 Data Description	204
12.3 Prognosis Model and Biomarkers Identification	220
12.4 Confounding Factor: An Illustration	228
12.5 Discussion	239
 V Appendix	 241
 A Molecular Biology for Dummies	 243
A.1 The Central Dogma	244
A.2 Genomic Technology	252
A.3 Perspectives	264
 B Frequently Asked Questions (for Biologists)	 267
 C Affymetrix Genechip Microarrays	 271
C.1 Measuring Expression	271
C.2 Summarization	274
C.3 Discussion	280
 D Datasets	 283
 E Non-Specific Filtering	 287
 F Ensemble Selection	 293
F.1 Stability and Classification Performance	293
F.2 Number of Bootstrap Rounds	295
F.3 Number of Features to Eliminate	297
F.4 60 Bootstraps and a Single Run of an SVM	299

G	The Birthday Paradox	301
G.1	Is Randomness a Competitor?	301
G.2	Probability of Non-Selection	306
G.3	Effect in Practice	306
H	Cristall: Supplementary Material	311
H.1	List of a Priori Relevant Features	311
H.2	Expression Profiles for the Preliminary Cohort	324
H.3	L_2 -AROM and PS- L_2 -AROM Comparison	327
	References	331
	Index	345

Part I

Lost in Empty Space

Introduction

1

This introductory chapter opens with a description of the context of this work (Section 1.1). The main issues raised by high-dimensional data analysis, and in particular by gene expression microarray data analysis, are detailed in Section 1.2. The contributions of this thesis are summarized in Section 1.3. The introduction ends with an overview of the document's structure in Section 1.4. But first of all, let us clarify an important point.

Biology or Machine Learning?

Reading the title, one might be wondering if this thesis is more about biology or more about machine learning. In a single sentence, it is clearly a machine learning thesis, since all the contributions are in that field. However, the target applications belong to the fields of biology and medicine. Some choices are thus motivated by biology, or the type of data molecular biology tools produce. The biological aspects within the contributive Chapters 7 to 11 (Part III) are as limited as possible. An introduction to molecular biology and genomic technology is available in Appendices A and C.

1.1 Context

Since the discovery of the double helix structure of DNA by [Watson and Crick \(1953\)](#), the number of measures done by molecular biology tools in one experiment has kept growing amazingly. It started with binary present/absent detection of a single protein with the oldest electrophoresis techniques in the eighties, to end up in 2010 with next-generation sequencing (NGS). This technology is able to measure at once the three billions of nucleotides of a human

cell as well as quantitative expression of the whole cell transcriptome (Metzker, 2010). The number of experiments that are done is however not growing at the same pace. This is due mainly to time and cost reasons. The number of experiments is typically of the order of a few hundreds in the best cases.

DNA Microarray refers in this work to gene expression microarray. In fact, *microarray* has become a generic word encompassing several types of biological measurements such as gene expression, single nucleotide polymorphism (SNP) or copy number variation (CNV). Most of them share a common characteristic: a single experiment outputs a very high number of measurements. This is a common trait of modern molecular biology tools, even outside the field of microarrays. It is also the case for example of spectrometry data, cell imaging and above all it is the case of next-generation sequencing technology. They fall therefore in the category of high-throughput technology. This work is particularly based on gene expression microarray data. This technology measures expression levels from several hundreds to several (tens of) thousands genes in a single experiment, which is their main strength. The expression of a gene is a measure of its transcription rate. The motivation behind gene expression measurement is to infer the current physiological state of a biological entity, i.e. the current activity of its proteins. However, directly measuring proteins activity is still technically difficult nowadays. Gene expression microarrays rely on strong hypotheses. First, that the expression gives an estimate of the produced amount of the corresponding proteins. Second, that the proteins activity is proportional to their production rate. Two main variants of gene expression microarrays are currently being used: high-density microarrays (up to about fifty thousands monitored genes) and low-density microarrays (few hundreds monitored genes). High-density arrays are the most expensive ones, but performs a complete screening of (candidate-)genes expression at a given time. This is particularly convenient when no subset of genes has previously been identified as the only relevant set for a given biological condition. The experiments reported in this thesis mainly focus on those high-density arrays, and in particular one of the most widespread technology: the Affymetrix Genechip. The main drawback of such technology is its high cost and as a direct consequence, the generally low number of samples analyzed within a given study.

Experiments (genomic or not) just generate data. What people are gen-

erally interested in is knowledge, not data itself. Machine learning, also known as statistical inference or computational intelligence, is a scientific field which purpose is to build mathematical models able to learn a concept (i.e. to generalize) from a limited amount of data. In other words, machine learning attempts to extract knowledge from observations. Microarray data are particularly interesting if external annotations, or labels, are available. Here, annotations should not be understood as *gene annotations*, but rather as annotations on the samples. Those can be clinical information about patients corresponding to biological material hybridized on microarray chips. It can also be wet lab observations about a cell line, etc. With those labels available, mathematical models can be built on both microarray data and the extra annotations in order to perform diagnosis or prognosis. Such models are trained to predict the annotations based on the expression data. A diagnosis is a prediction about the current state of the patients. A prognosis is a prediction of the future state of the patients. In that case, gene expression data are collected at a given time, and annotations are collected later. An example of diagnosis is to predict the current healthy/diseased status of a patient based on gene expression data. An example of prognosis, is the Cristall project ([Smets, 2006](#)) which is aimed at predicting the risk that children develop allergy at 18 months, based on their expression data at 6 months. Another type of prognosis task is the response prediction. It comes to predict if a patient will react or not to a treatment or a drug, before giving it to the patient. This leads to personalized medicine and is a hot topic for pharmaceutical companies, but also for medical and national insurances. That kind of prognosis has been done in the context of the design of a candidate treatment against Melanoma ([Louahed et al., 2008](#); [Dupont et al., 2010](#)). From a machine learning point of view, those three tasks of diagnosis, prognosis and response prediction are addressed by classification techniques. Classification also serves as a basis for another biological application: survival analysis.

The number of measurements provided by a microarray experiment can be viewed as the number of dimensions of the classification problem. Each microarray experiment is a data point or a sample defined by a vector of measurements. The number of dimensions (genes) for that kind of data is generally hundred or thousand times higher than the number of samples (patients, cell lines,...). In the best cases, the number of hybridized chips hardly reaches one hundred. Given their high number of dimensions, those data fall in the category of high dimensional data, and feature selection is

often performed on them. Feature Selection in the context of classification comes to identify a small number of dimensions on which a predictive model with satisfying performances can be built. In the case of gene expression data, building a model on fewer dimensions, i.e. fewer genes, can improve the model interpretability and also its running time. To some extent, good predictions obtained with a small number of genes lead to infer a high probability that those selected genes are implied in the biological process corresponding to the annotations. Those few selected genes are called biomarkers. A set of biomarkers is called a signature. Several objectives can be targeted. Some applications, such as producing clinical diagnosis or prognosis kits require to identify the smallest/best-performing signature, namely for cost concerns. Some other applications more driven by explanatory concerns, attempt to identify all genes implied in the biological topic under study. In the industrial area, gene selection for prognosis or diagnosis is known as gene profiling.

1.2 Issues Raised By Microarray Data

Gene expression microarray data analysis is a complex topic and is a typical representative of high dimensional data with few points. The high number of dimensions alone makes difficult or impossible the application of techniques for which the computational complexity is strongly depending on the number of dimensions, even regardless of the number of samples available. This does not only affect the kind of classification model considered, but also the optimization techniques used to estimate their parameters. This is the case of most techniques involving matrix inversions. As a typical example, the naive Newton method for gradient descent is an iterative procedure requiring the inversion the Hessian¹ of the function to optimize. The computational complexity of each iteration is thus $\mathcal{O}(p^3)$, where p is the number of dimensions of the problem². The Iteratively Reweighted Least Square technique (IRLS) (Bishop, 2006) is an efficient implementation of the Newton method. It still has a complexity of $\mathcal{O}(np^2)$ (Minka, 2007), where n is the number of samples of the problem. Those costs are unaffordable when p is several tens of thousands. The situation gets even worse if that kind of algorithm is included in a validation loop.

¹Matrix of the second derivatives

²This complexity assumes that the Gauss-Jordan elimination method is used to achieve the inversion step

This time complexity issue is also faced when dealing with feature selection. Given the number of dimensions, an exhaustive space search is not tractable. Searching for all the possible sets of, say, 50 features among 50,000 comes to compare 2.85×10^{170} distinct sets of features, which is about hundred orders of magnitude greater than the number of particles in the universe³. In most applications, there is even an extra difficulty: the number of features to be selected to get an optimal solution is not known a priori. Several strategies like forward selection or backward elimination (or combinations of both) have been proposed but lead then to potentially sub-optimal solutions.

Apart from the computational complexity point of view, the data high dimensionality alone is not intrinsically problematic for model estimation. It has been shown that most classification models are not affected by potential impacts of high-dimensionality on distances (Francois, 2007). The problem originally stated as the *curse of dimensionality* is related to the density of points in a space more than just to the number of its dimensions (Bellman, 1961). Consider a p -dimensional r -side cube (a hypercube), inside which a number of points proportional to p is available. With k points available by dimension, the density d of the hypercube can be expressed as:

$$d(k, r, p) = \frac{p \times k}{r^p} \quad (1.1)$$

This expression shows that when the number of points available in a hypercube grows linearly with the number of dimensions p , the density drops exponentially. This is illustrated in Figure 1.1. It means that for a large number of dimension, an exponentially large number of points are required to keep a constant density. It is thus the combination of a high number of dimensions with a low number of samples that is problematic and that generates *empty spaces*. This characteristic is often referred to as the *small-n-big-p* setting. In practice, empty spaces are the main source of the following issues for gene expression microarray data: poor generalization performances and lack of signature robustness. Those issues are developed in the next two paragraphs.

Gene expression microarray data classification typically leads to poor accuracy with test error rates around 15 – 25% (Alon *et al.*, 1999; Singh *et al.*, 2002; Weigelt *et al.*, 2005). Classification errors can even be much worse in some cases, for example in (van 't Veer *et al.*, 2002) when authors use a rigorous protocol for performances evaluation. This is on average much

³Current estimations between 10^{72} and 10^{87}

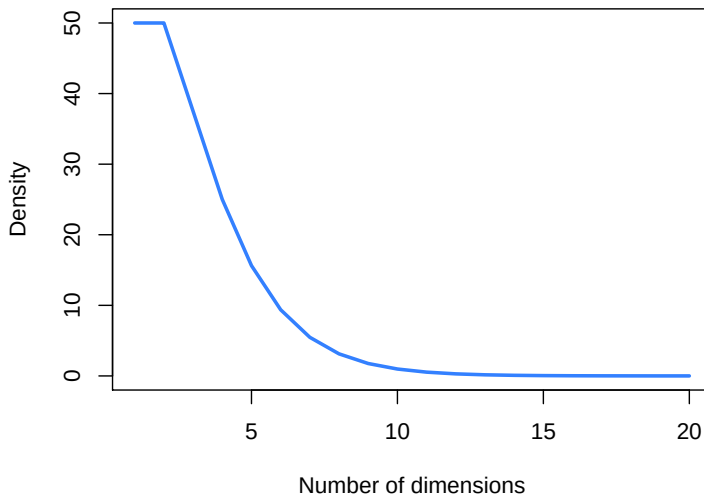


Figure 1.1 Density of a cube of side $r = 2$ with $k = 100$ points for each dimension.

higher than for other applicative fields. In optical character recognition, it is frequent to obtain test error values around 1% (Schölkopf and Smola, 2001; Vapnik, 1995) on public benchmarks, though this is a multi-class problem. Several millions of belgian tax declarations are read by an optical character recognition system each year, and nearly no error occurs. Automated spam-filtering for email is another example of widespread -though less critical- application of classification with satisfactory performances. Automated diagnosis based on gene expression microarray is however rarely (if ever) applied in a routine clinical setting, because classification performances delivered by baseline techniques are most of the time far from satisfactory. Reading literature with some sense of critics leads to the conclusion that models for prognosis and diagnosis based on expression data generally offer poor generalization performances which are mainly due to overfitting, in turn due to that typical empty spaces phenomenon. Even with a simple linear model, a set of parameters leading to a perfect separation on training samples is guaranteed to exist as soon as $p + 1 \geq n$ (Vapnik, 1995). In the case of microarray data, $p \gg n$. A learning algorithm typically trains a model until it meets a quality criterion (generally the accuracy) on the training samples. The problem here is that a 100% accuracy is guaranteed to be reached too easily, for many parameter value combinations. Generalization performances on an independent test set will generally drop dramatically if no particular

care is taken. In other words, the challenge for high-dimensional data with few samples is not to find just *a* model seemingly good on a training set, but well to find one that will lead to good generalization performances on new test samples.

It has been highlighted that for a given pathology, even on a single given expression dataset, several non-overlapping gene lists could offer the same predictive power (Ein-Dor *et al.*, 2005). Different models or different subsamplings of the patients lead to different signatures. This lack of signature robustness makes their biological interpretation tricky. It can also make biologists suspicious about machine learning techniques. Indeed, it is a reasonable hypothesis that the biological processes explaining the clinical outcome are mostly common among different groups of patients. Similar signatures are thus expected regardless of the subsampling of the patients on which the model is built. Such robustness is called stability. A good feature selection technique is ideally highly stable. It has been shown that with baseline feature selection techniques, thousands of samples are needed to obtain a reasonable overlap ($\approx 50\%$) between signatures built on several breast cancer expression microarray datasets (Ein-Dor *et al.*, 2006). A high stability might be difficult to obtain in the cases where the size of signatures wanted are below the number of genes involved in a biological process. Another issue is that stability alone is not a sufficient criterion for a good signature. Indeed, a selection method always returning the same arbitrary signature regardless of the sampling will exhibit a maximal stability, but is not likely to make possible the estimation of a highly performing classification model.

Other challenges raised by microarray data are related to methodological aspects. One of the pitfalls most frequently encountered in literature is called the selection bias (Ambroise and McLachlan, 2002). It consists of identifying a gene signature on all the samples of a dataset, and to evaluate the performances of a model based on that signature on the same samples, be it with or without a cross-validation setting (Guyon *et al.*, 2002; Shipp *et al.*, 2002; van 't Veer *et al.*, 2002). Such a bias leads to optimistic results. Similarly, a normalization bias is also frequently observed, with normalization being done on an entire dataset (both training and test samples). Another methodological bias is the ignorance of confounding factors. van de Vijver *et al.* (2002) did not realize that their model is highly correlated to the fact

that patients underwent or not a chemotherapy (Ahmed and Brenton, 2005). The same issue is present in (Alon *et al.*, 1999) where samples corresponding to non-tumor patients are mainly composed of muscle tissue whereas tumor samples are not (Guyon *et al.*, 2002). The signature proposed by Alon *et al.* (1999) precisely contains genes related to tissue composition instead of the patient status. Another important lack in most microarray study papers is the absence of standard deviation evaluation. Since there are only a few samples, using classical k-fold cross-validation leads to large standard deviations around the estimated classification performances. Ignoring variability could lead to infer that two models are offering significantly (and largely) different classification performances, whereas it is not the case. It can also lead to deduce that a classification model performs better than a random decision when in fact it does not. Finally, some performances are assessed by inadequate statistics. It is the case of the accuracy to assess classification performances when dealing with unbalanced datasets. In practice, most gene expression microarray datasets are unbalanced (Louahed *et al.*, 2008; Dupont *et al.*, 2010; Alon *et al.*, 1999; Golub *et al.*, 1999; Shipp *et al.*, 2002; Chandran *et al.*, 2007; Welsh *et al.*, 2001). Typical ratios for those two-classes datasets are between 2:1 and 3:1, meaning that one class contains two thirds to three quarters of the samples and the other class contains the last third or quarter. A naive model classifying every given sample as one belonging to the majority class will already have an accuracy of 66.67% to 75%. Accuracy thus sometimes leads to deceiving conclusions.

Microarray technology is also subject to technological issues. Summarization, the operation of computing one unique value for a probeset (a gene) based on its corresponding sub-part expressions (probe expressions) is a matter of debate. Several methods have been proposed but no consensus on the adequate summarization technique has been reached. Three frequently used methods are MAS (Hubbell *et al.*, 2002), RMA (Irizarry *et al.*, 2003) and GC-RMA (Wu *et al.*, 2004). The difficulty of answering this question resides in the lack of datasets for which the expected summarization outcome is known. Finally, microarray technology also raises other biological issues. Sources of variability may come from methods of RNA extraction, or different types of probe preparation, labeling and hybridization (Ahmed and Brenton, 2005). Another source of noise in the data is that the considered experiments are led on samples from patients whose metabolism is not only ruled by the phenomenon under study. If cancer is under study, it may still happen that a

given patient in the cohort has flu, another one might have allergy, etc. Those phenomena are also reflected in the gene expressions, scrambling the signal of interest. Those biological issues are beyond the scope of this work.

1.3 Contributions Summary

This introductory chapter follows with a summary of the original contributions of this thesis. Part III (Chapters 7 to 11) details much more the main contributions. Some side-contributions are presented in Part II, Chapter 5 and Chapter 6.

1.3.1 Main Contributions

1. **Ensemble feature selection (Abeel *et al.*, 2010, 2009).** This method builds consensus of several signatures selected on different subsamplings of the data through selectors embedded in linear support vector machines (SVM). It improves the stability of the selected gene lists with respect to sampling variations as well as the classification performances. Ensemble feature selection is covered in Chapter 7.
2. **Partially supervised feature selection (Helleputte and Dupont, 2009c).** This method incorporates prior knowledge about features known or guessed to be more relevant than others for the considered classification task. It makes possible a direct interaction with field experts (MD or biologists). This feature selection technique is embedded in linear SVM with iterative rescaling of the inputs. The prior knowledge is incorporated through this rescaling step. It has been shown to largely improve stability with respect to sampling variations, particularly for signatures of interesting sizes (few tens of genes), and even in some cases to also improve classification performances. Another strength of that method is its ability to depart from the prior knowledge if it leads to poor performances. Partially supervised feature selection is detailed in Chapter 8.
3. **Feature selection by transfer learning (Helleputte and Dupont, 2009b,a).** Based on the partially supervised feature selection, a transfer learning technique has been designed. It allows the use of previously published datasets, for example available on Gene Expression Omnibus (Edgar and Barrett, 2006), as a source of information to improve the quality of the gene selection on a dataset of interest, even in absence of prior knowledge from field experts. Transfer learning is a

way to address the lack of samples in microarray studies at no cost (financially speaking). Here again, an improvement of the selection stability with respect to the sampling variations has been shown, along with a gain in classification performances. Feature selection by transfer learning is covered in Chapter 9.

4. **Feature selection for multi-task learning.** This approach allows the selection of features for several different but related tasks at once. The selection is embedded within a multi-task logistic regression classification model penalized by a group-regularizer. Six such regularizers are proposed in the form of mixed norms $L_{i,j}$, where $i \in \{0, 1\}$ and $j \in \{1, 2, \infty\}$. Those approaches are evaluated according to four criteria: classification performances, within- and between-tasks stability, and sparsity. The $L_{1,2}$ regularizer tends to provide the best overall classification performances, but less sparse solutions. When compared at the same level of sparsity, $L_{1,\infty}$ tends to be the best possible solution in terms of between-task stability. The $L_{0,j}$ regularizers tend to produce sparser solutions. A dedicated solver is developed which makes iterations of linear complexity with respect to the number of features. Feature selection for multi-task settings are also compared to single task approaches, with results in favor of the multi-task approach. Multi-task learning is detailed in Chapter 10.
5. **Feature selection for multi-class learning.** The previous approach of multi-task learning can be shown to be also adequate in applications on single-task but multi-class problems (multi-class understood as *more than two classes*). Usually, feature selection for multi-class problems leads to several signatures (one for each binary discrimination sub-task). The multi-task method applied to multi-class problems with the $L_{1,\infty}$ regularizer makes possible the production of a single signature for every discriminative task, with comparable sparsity and classification performances. Feature selection for multi-class learning is also covered in Chapter 10.
6. **Cristall Project.** A part of the Cristall project is to predict the allergic status of children at 18 months based on their gene expression data at 6 months, and to identify biomarkers making this prognosis possible. An extensive study of the Cristall expression data can be found in Chapter 12. Our analysis fails to identify a set of biomarkers making possible the estimation of a prognosis model with high classification perfor-

mances. A sufficient amount of elements and experimental results are provided to justify this outcome. Nevertheless, the benefits of several other contributions of this thesis are confirmed on Cristall data. An increase of stability with partially supervised feature selection or ensemble methods for feature selection is observed.

1.3.2 Side-Contributions

1. **Generalized bootstrap .632+.** This contribution is not as central to the thesis as the previous ones. It is rather a methodological contribution. The bootstrap .632+ has been developed to reliably estimate a statistic of interest, even in conditions of overfitting. As such, the only estimate it is designed for is the accuracy. In this thesis, an extended version of this protocol, called the generalized bootstrap .632+, is proposed. In principle, it allows one to compute any function of a confusion matrix for binary classification problems through a bootstrap scheme, in presence of overfitting. In practice, five such functions are detailed: accuracy (the standard bootstrap .632+), balanced classification rate, positive accuracy, negative accuracy, and the positive predicted value. The generalized bootstrap .632+ is introduced in Section 5.3.2 and has not been published previously.
2. **Non-Specific Filtering Evaluation.** Most feature selection techniques discussed in this document are informed, meaning that they somehow use the labels of the training samples. Another kind of feature selection not making use of those labels is called non-specific filtering. It is particularly useful as a first step of selection when dealing with datasets with several tens of thousands of dimensions. An empirical study shows that nearly 75% of the features can be filtered out without harming the classification performances and the subsequent specific selection. In some cases, improvements on classification performances are even observed. Non-specific filtering is covered in Chapter 6.
3. **Other methodological aspects.** Throughout this text, a particular attention is paid to avoid methodological biases and to provide rigorous results evaluation, including relevant statistical hypothesis testing when possible. Protocols avoid the selection and normalization bias and consist of robust validation loops with several hundreds of resamplings, reducing standard deviations of the estimated performances. Balanced classification rate (BCR) is generally preferred to accuracy to avoid ef-

fects from class unbalance. A general framework to evaluate both stability and classification performances is used in most experiments. Chapters 5 to 6 are dedicated to methodological aspects. This concern is also noticeable throughout all publications related to this thesis.

The main contributions are related to the following list of publications:

- Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., and Saeys, Y. (2009). Robust biomarker identification for cancer diagnosis using ensemble feature selection methods. In *Machine Learning in Systems Biology*
- Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., and Saeys, Y. (2010). Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics (Oxford, England)*, **26**(3), 392–398
- Helleputte, T. and Dupont, P. (2009a). Biomarker selection by transfer learning with linear regularized models. In *Machine Learning in Systems Biology*
- Helleputte, T. and Dupont, P. (2009b). Feature selection by transfer learning with linear regularized models. In *European Conference on Machine Learning (ECML)*, number 5781 in Lecture Notes in Artificial Intelligence, pages 533–547
- Helleputte, T. and Dupont, P. (2009c). Partially supervised feature selection with regularized linear models. In *International Conference on Machine Learning (ICML)*
- Louahed, J., Gaulis, S., Helleputte, T., Dupont, P., Gruselle, O., Spatz, A., Kruit, W., Dreno, B., Lehmann, F., and Brichard, V. (2008). Clinical response to the mage-3 immunotherapeutic in metastatic melanoma patients is associated with a specific gene profile present prior to treatment. In *33th European Society for Medical Oncology (ESMO) Congress*

The research work behind this thesis also led to US and UK patents, which results have been produced in the spirit of ensemble methods for feature selection described in Chapter 7:

- Dupont, P., Gaulis, S., Helleputte, T. (Inventors), GlaxoSmithKline Biologicals S.A., Dupont, P., Gaulis, S., and Helleputte, T. (Applicants) (2010). Method for classifying a cancer patient as responder or non-responder to immunotherapy. Patent, WO/2010/029174

1.4 Thesis Roadmap

Apart from this introduction (Part I), this document is made of four parts. In a nutshell, Part II gives some machine learning background, Part III proceeds to a detailed description of the contributions of this thesis, Part IV is dedicated to a microarray data analysis case study and Part V contains the appendices. The publications related to the original contributions of each chapter are recalled at the beginning of each chapter.

Part II. Background.

The chapters of this part can be further divided in two parts. Chapters 2 to 4 are pure reviews of related topics useful for the full understanding of the rest of the text. Classification, regularization, feature selection and evaluation metrics are introduced respectively in those chapters. Chapters 5 to 6 are centered on tools such as validation protocols, normalization and non-specific filtering. Some methodological side-contributions are contained in those chapters, such as an extension of the bootstrap .632+ protocol to other metrics than accuracy (Section 5.3.2) or the study of non-specific filtering impact on classification performances (Chapter 6).

Part III. Inductive Biases for High-Dimensional Learning.

This part holds the main contributions of the thesis. It begins with Chapter 7 where an ensemble method for feature selection is proposed. Chapter 8 introduces a novel approach called partially supervised feature selection. It makes possible the incorporation of expert prior knowledge within the selection of the features relevant for classification. Chapter 9 further adapts the method of the previous chapter to face the (common) case where no prior knowledge is available. In that case, knowledge can be extracted from other datasets on related tasks, and transferred on the dataset of interest via partially supervised feature selection. Yet improving on the availability of related datasets, Chapter 10 proposes a multi-task approach for feature selection. It intends to select features for several related tasks at once in order to perform classification on each task. Finally, it is also shown in that chapter that this approach can be applied not only to multi-task, but also to multi-class settings. All those original approaches are validated on numerous microarray datasets.

Part IV. A Microarray Case Study.

This part is dedicated to the Cristall project (Chapter 12). The aim of this study is to identify allergy risk-factors in newborn. To this end, the gene expression of 77 patients at the age of 6 months have been measured on microarrays. The allergic/non-allergic clinical status is however determined once the children got 18 months. The task is to predict at 6 months the children status at 18 months, and to do so on a limited number of genes. It is thus a prognosis task with gene profiling. This chapter also illustrates many aspects discussed in the previous parts of this thesis. In addition, it discusses other practical aspects not covered so far such as data visualization, outlier detection, confounding factor detection, etc.

Part V. Appendix.

Appendices cover themes that are less central to this thesis, but that may help the reader better understand some aspects of the work. They also provide some extra material not displayed in the core of the text. Appendix A introduces the basics of molecular biology, and presents the main tools of genomic technology and perspectives in that field. Appendix B is an attempt to answer some frequently asked questions on the interactions between machine learning and biology. Most datasets used in this thesis have been produced on Affymetrix Genechip microarrays. Affymetrix microarray technology is detailed in Appendix C. The usual preprocessing steps applied to Affymetrix microarray data (the summarization) are also covered in this appendix. These steps tend to normalize each hybridized chip, and to produce one expression value by probeset ($\approx$ gene). Many summarization protocols are available and show advantages and drawbacks. The principle of a new summarization protocol is also proposed (Appendix C.2.4), based on a machine learning perspective. Appendix D gives more information on all the datasets used through the text. Appendix E and F provide supplementary material for Chapters 6 and 7 respectively. A counter-intuitive probabilistic phenomenon similar to the *birthday paradox* is introduced in Appendix G. This phenomenon comes into play when performing permutation experiments, and induces misleading results. Finally, Appendix H provides a large amount of supplementary material for the Cristall project.

Datasets

In many chapters, datasets are used to validate the proposed methods. Those datasets are public datasets. Through the text, each of them is referred to with a code of three small capital letters (and possibly a digit). For example, one of the prostate cancer datasets is coded as PRO2. See Appendix D for more information on every dataset. The code is displayed before each dataset description. This coding does not apply to the Cristall dataset (see Chapter 12). We choose to mainly use real data to validate our approaches, and rarely deal with artificial datasets. This choice is motivated by the will to address real-life problems and also by the high difficulty of adequately modeling gene expression data, precisely because the underlying biological processes are largely to be discovered yet.

Implementations

Most of the experiments and methods have been coded in R language (R Development Core Team, 2010), with a frequent use of the Bioconductor packages suite (Gentleman *et al.*, 2004). Some parts of the code have also been developed in C/C++. Note that R recently received guidelines from the American Food and Drug Administration (FDA), making it particularly suitable for biotech applications⁴.

⁴See <http://www.r-project.org/doc/R-FDA.pdf>

Part II

Machine Learning Background

Classification

2

This chapter opens Part II with a review of several classification techniques that are used throughout the text. First, Section 2.1 gives the reader a few elements in order to easily follow the rest of the text. It includes mathematical notations, a one-paragraph introduction on microarray data and the definition of basic concepts related to optimization. Section 2.2 is about linear models for binary classification. Section 2.3 defines some notions of statistical learning, among which the notions of risk, VC dimension and regularization. Both Sections 2.2 and 2.3 also offer arguments in favor of linear models given the applicative context of small- n -big- p problems addressed by this thesis. This point of view is completed in the next chapter on feature selection.

The chapter follows with three sections dedicated to three families of models. Section 2.4 introduces the support vector machines. The discussion is mainly restricted to linear support vector machines. Section 2.5 reviews the logistic regression and points out the similarities and differences of this approach with respect to the support vector machines. Section 2.6 ends this chapter by introducing another type of classification models, based on ensemble methods and decision trees: bagging, the random forests and extra trees.

2.1 Preliminaries

2.1.1 Notations

Through the text, a vector in p dimension is noted by a bold lowercase letter, such as $\mathbf{x}$. Matrices are noted by bold uppercase letters. A set of n points in

$\mathbb{R}^p$ is thus an n rows and p columns matrix noted $\mathbf{X}$. Bold lowercase letters with a subset index, like $\mathbf{x}_i$, generally refer to the i^{th} row of the matrix $\mathbf{X}$ (the i^{th} point). In some cases, it might refer to the i^{th} column of the matrix $\mathbf{X}$ (the i^{th} dimension)¹. The difference is either obvious or explicitly specified. Scalars are noted with lowercase non-bold letters. x_{ij} is the entry of $\mathbf{X}$ corresponding to the i^{th} row and j^{th} column.

2.1.2 Microarrays in One Paragraph

Readers not familiar with gene expression microarray data and who do not feel the need to get more information on it in Appendices A and C just have to know that microarray data can be represented as a wide and flat $n \times p$ matrix $\mathbf{X}$ where $n \ll p$ such as depicted in Table 2.1. Each row is a microarray experiment, each column corresponds to a probeset, which represents a gene, or a portion of genetic material suspected of playing some role in metabolism regulation. Each entry x_{ij} of the matrix is a measure of the expression of probeset j in the microarray experiment i . The expression of a gene might be understood as the activity of that gene. Expression is variable through time and environment. The last column of the table is a vector of labels, generally clinical annotations, related the the microarray experiment. In our context, it is the binary target value to predict based on the expression values.

Table 2.1 A matrix of gene expression values for n microarray chips with p probesets. The last column contains (clinical) annotations.

	probeset 1	probeset 2	...	probeset p	label
sample 1	$x_{1,1}$	$x_{1,2}$	...	$x_{1,p}$	y_1
sample 2	$x_{2,1}$	$x_{2,2}$	...	$x_{2,p}$	y_2
...	...	...	...	...	...
sample n	$x_{n,1}$	$x_{n,2}$	...	$x_{n,p}$	y_n

2.1.3 Convex Optimization

This thesis covers many optimization problems. An important characteristic of these problems is their *convexity*. An objective function that has to be optimized (minimized or maximized) can be either convex, strictly convex or non-convex. Entire books are dedicated to convex optimization and the interested reader is invited to refer to (Boyd and Vandenberghe, 2004) for more

¹In that case, the j letter is generally preferred

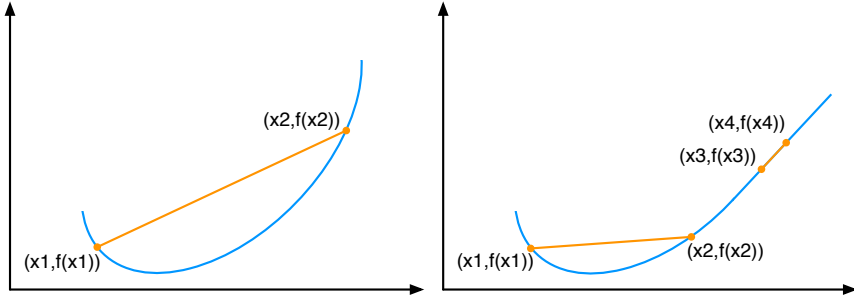


Figure 2.1 Left: Graph of a strictly convex function. The chord (i.e. orange line segment) between any pair of points of the graph lies strictly above the graph (blue curve). Right: graph of a convex but not strictly convex function. The chord between $(x^3, f(x^3))$ and $(x^4, f(x^4))$ is not below the graph, but not strictly above it either.

information on this topic. Let us just define the convexity and the strict convexity of a function, as it comes into play at various places across the thesis.

Convexity

A function $f(\mathbf{x}): \mathbb{R}^p \rightarrow \mathbb{R}$ is *convex* if and only if its domain $\text{dom}(f)$ is a convex set, and for any two points $\mathbf{x}^1, \mathbf{x}^2 \in \text{dom}(f)$:

$$f(\theta \mathbf{x}^1 + (1 - \theta) \mathbf{x}^2) \leq \theta f(\mathbf{x}^1) + (1 - \theta) f(\mathbf{x}^2) \quad (2.1)$$

with $0 \leq \theta \leq 1$. A function which does not satisfy this condition for any pair of points is not convex. A function f is said to be *concave* if $-f$ is convex, and vice-versa. Note that a set $\text{dom}(f)$ is convex if the line segment between any two points in $\text{dom}(f)$ lies in $\text{dom}(f)$. The definition of convexity is illustrated in Figure 2.1. Geometrically, this definition means that the line segment between $(\mathbf{x}^1, f(\mathbf{x}^1))$ and $(\mathbf{x}^2, f(\mathbf{x}^2))$ lies above the graph of f .

Strict Convexity

A function is *strictly convex* whenever Equation (2.1) is transformed into a strict inequality, $\mathbf{x}^1 \neq \mathbf{x}^2$ and $0 < \theta < 1$. If f is strictly convex, $-f$ is strictly concave, and vice-versa.

Second-Order Conditions for Convexity

In practice, the convexity (strict or not) of a function is hard to check based on the definition of Equation (2.1). First order conditions are available², but again require a check over every possible point pairs in $\text{dom}(f)$ and are thus difficult to assess. Second-order conditions³ are available to assess the convexity of a function. Their validity is easier to check. A function is convex if and only if its domain is convex and its Hessian is positive semidefinite:

$$\nabla^2 f(\mathbf{x}) \succeq 0, \forall \mathbf{x} \in \text{dom}(f). \quad (2.2)$$

For a function $f(x)$ on $\mathbb{R}$, this necessary and sufficient condition is equivalent to require that the second derivative of f be positive. This can be interpreted as the need for a function to show a positive (upward) curvature at any point in order to be convex. A function is strictly convex if:

$$\nabla^2 f(\mathbf{x}) \succ 0, \forall \mathbf{x} \in \text{dom}(f). \quad (2.3)$$

For a function $f(x)$ on $\mathbb{R}$, this is equivalent to require that the second derivative of f be strictly positive. Note that this condition for strict convexity is sufficient but not necessary. A function failing to satisfy this condition might still be strictly convex⁴.

Why Convexity?

In an optimization problem, the fact that the objective function $f(\mathbf{x})$ is convex guarantees that it has a unique optimal value $f^*(\mathbf{x})$. It is thus possible to globally solve the problem on its whole domain. This is an advantage with respect to non-convex optimization that might get stuck into local optima. The strict-convexity of an objective function furthermore guarantees that this optimal value can be obtained only by a single combination of the function inputs:

$$\nexists \mathbf{x}^1, \mathbf{x}^2 \text{ s.t. } f(\mathbf{x}^1) = f^*(\mathbf{x}) \wedge f(\mathbf{x}^2) = f^*(\mathbf{x}) \wedge \mathbf{x}^1 \neq \mathbf{x}^2.$$

²i.e. conditions relying on the first derivatives of the function

³i.e. conditions relying on the second derivatives of the function

⁴A function f can also be *m-strongly convex*. This is the case if the necessary and sufficient condition $\nabla^2 f(\mathbf{x}) \succeq m\mathbf{I}$ is met, with $m > 0$ and $\mathbf{I}$ the identity matrix. So every strongly convex function is also strictly convex. The concept of strong convexity is however not required in the text.

2.2 Linear Models

This section defines linear models for classifying samples (points, examples) in a binary context, where a point belongs to one out of two classes. In this thesis, most classification tasks are binary. The multi-class setting is only addressed in the last section of Chapter 10. By multi-class, we mean more than two classes. By default, from a k -classes problem, it is possible to define $k \times (k - 1) / 2$ one-against-one binary classification tasks or k one-against-the-rest binary classification tasks. All binary techniques described in this chapter are thus applicable to multi-class problems.

Hyperplane

Given n examples $\mathbf{x}_i$ in an input space $\mathcal{X} \in \mathbb{R}^p$ with $i \in \{1, \dots, n\}$, a linear model $h(\mathbf{x})$ of parameters $\mathbf{w} \in \mathbb{R}^{p+1}$ can be defined as:

$$h(\mathbf{x}) = w_0 + \sum_{j=1}^p w_j x_j, \quad (2.4)$$

where w_0 is the bias term (sometimes also called the *offset*). To simplify the notations throughout this text, we assume that each $\mathbf{x}$ vector is complemented with a 0^{th} element equal to one, corresponding to the w_0 parameter. The true number of dimensions of the data remains p , not $p + 1$. Equation (2.4) can then be rewritten:

$$h(\mathbf{x}) = \langle \mathbf{x}, \mathbf{w} \rangle. \quad (2.5)$$

The set of points for which $h(\mathbf{x}) = 0$ is the generalization in p dimensions of what is a line in 2 dimensions, and a plane in 3 dimensions⁵. Such a generalization is called a *hyperplane*. Such a hyperplane is also called a *hypothesis*.

A classifier f based on such a linear model typically makes decisions based on the sign of h . The sign of h corresponds to the side of the hyperplane where the considered sample lays. Examples of such classifiers are illustrated in Figure 2.4. The classifier f predicts the class of any point $\mathbf{x} \in \mathbb{R}^p$ as follows:

$$\hat{y} = f(h(\mathbf{x})) = \text{sign}(\langle \mathbf{x}, \mathbf{w} \rangle). \quad (2.6)$$

The value of the function $h(\mathbf{x})$ for a particular point $\mathbf{x}$ is also called the *decision value* of the classifier f .

⁵In one dimension, it defines just a threshold.

Typically, extra data are also available in the form of labels, or target values, y_i in a space $\mathcal{Y} \in \{-1, +1\}$. The techniques from machine learning dealing both with samples and their labels constitute what is called the supervised learning. A set of couples $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ available for model estimation is called a *training set*.

The main difference between classification and regression is the nature of the target values. In the case of regression, labels are defined as variables on a continuous domain. Although a small change, it has a deep impact on the type of model considered, and on the way their parameters are estimated. In this thesis, only classification is considered.

Linear models have the advantage of interpretability. A direct mapping can be done between each coefficient w_j and an original dimension, feature, of the data. A large (resp. small) $|w_j|$ means that the corresponding feature j has a high (resp. low) weight in the decision process, which is useful to understand the generative process. The sign of each w_j is also useful to understand the type of contribution of each feature. These properties will be used many times through this text. Linear models have other interesting properties related to statistical learning theory that will be illustrated in the next section.

2.3 Statistical Learning

Statistical Learning Theory is mainly due to Vapnik, and most of this section is inspired from his work (Vapnik, 1995). Sticking to the previously introduced notations, we assume that pairs of points and labels are independently and identically distributed (*i.i.d.*), drawn from a distribution $P(\mathcal{X}, \mathcal{Y})$. The distribution P is assumed to be fixed, and is most of the time unknown. The main aim of statistical learning is to choose a decision function $f: \mathcal{X} \rightarrow \mathcal{Y}$ within a space of functions $\mathcal{F}$. For classification based on linear models, choosing f comes mainly to choose the right hypothesis h within the space of hypotheses $\mathcal{H}$, i.e. the right hyperplane. Choosing the hyperplane is the learning process. But what would make a good choice?

2.3.1 Loss Functions

A *loss function* $\mathcal{L}$ is generally used to measure the quality of a predictive model. In most cases, loss functions tell how *bad* a model is rather than how *good* it is. The higher the value of the loss, the poorer the model. To provide such an evaluation, a loss function needs both the output of the learned model $\hat{y} = f(\mathbf{x})$ or its decision value $h(\mathbf{x})$, as well as the true value to predict, y . It is typically computed for every element of a set of points, then averaged. We discuss here four well known loss functions. Three of them are mainly used for classification: the 0/1-loss, the hinge loss and the logistic loss. The fourth one, the square loss, is more related to regression problems. It is nevertheless discussed here since it is used in several models covered by this thesis. Those four loss functions are pictured in Figure 2.2.

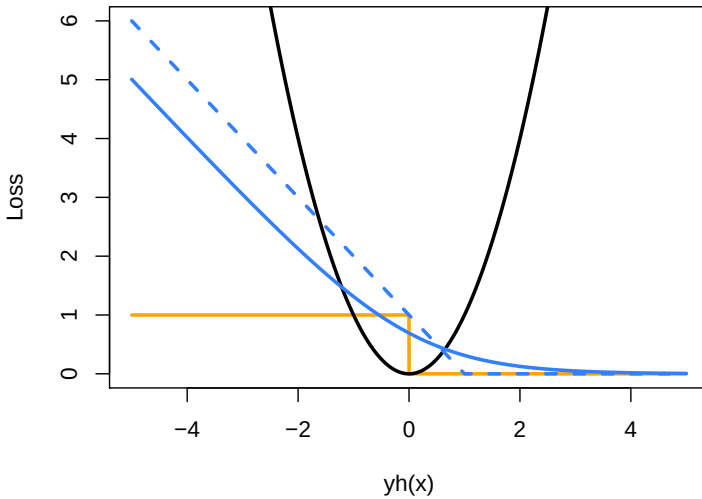


Figure 2.2 Four loss functions: The 0/1-loss (orange), the hinge loss (dashed blue), the logistic loss (blue) and the square loss (black).

The 0/1-loss $\mathcal{L}_{01}$ simply counts one if the prediction is wrong, and zero otherwise. An expression directly translating this behavior is:

$$\mathcal{L}_{01}(y, f(\mathbf{x})) = \begin{cases} 0 & \text{if } y = f(\mathbf{x}), \\ 1 & \text{if } y \neq f(\mathbf{x}). \end{cases} \quad (2.7)$$

The 0/1-loss is a step function. It has the drawbacks of being not continuous

and not convex, making difficult its optimization. For comparison purposes, it can be rewritten as a function of $yh(\mathbf{x})$:

$$\mathcal{L}_{01}(yh(\mathbf{x})) = \frac{1 - \text{sign}(yh(\mathbf{x}))}{2}. \quad (2.8)$$

The hinge loss $\mathcal{L}_{\text{hinge}}$ is defined by:

$$\mathcal{L}_{\text{hinge}}(yh(\mathbf{x})) = \max(0, 1 - yh(\mathbf{x})). \quad (2.9)$$

Like the 0/1-loss, its value is 0 if the model is right. Unlike the 0/1-loss, the value of the hinge loss is not necessarily 1 in case of error. It outputs a value proportional to the *strength* of the error. This loss is convex, but has two limitations. First, it is not strictly convex which is a hint that it has no single minimum value. Indeed, it is in this case obvious that it has a constant and minimal value of 0 for all $h(\mathbf{x})$ such that $yh(\mathbf{x}) > 1$. Second, even if convex, the hinge loss differentials are not defined for $yh(\mathbf{x}) = 1$. The hinge loss is namely used by the support vector machine introduced in Section 2.4. The third loss for classification that we consider is the logistic loss. It is used for logistic regression introduced in Section 2.5. Its expression is:

$$\mathcal{L}_{\text{logistic}}(yh(\mathbf{x})) = \log(1 + \exp^{-yh(\mathbf{x})}). \quad (2.10)$$

It is convex, continuous, and its differentials are defined over its whole domain. The logistic loss is hence one of the easiest loss functions for classification to deal with in optimization problems. Theoretically, the logistic loss is strictly convex, since its second derivative is strictly positive. Numerically, however, it has the behavior of a non-strictly convex function, as its second derivative tends towards zero as $yh(\mathbf{x})$ increases. The square loss $\mathcal{L}_s$, more appropriate for regression, is defined by:

$$\mathcal{L}_s(y, h(\mathbf{x})) = (y - h(\mathbf{x}))^2. \quad (2.11)$$

It, too, can be rephrased as a function of $yh(\mathbf{x})$ for comparison purposes:

$$\mathcal{L}_s(yh(\mathbf{x})) = \left(\frac{y^2 - yh(\mathbf{x})}{y} \right)^2. \quad (2.12)$$

The square loss is strictly convex: it has a unique optimum. It is used as a building block for models such as the lasso, introduced in Chapter 3. Just like the logistic loss, the square loss is easy to deal with in optimization problems.

The 0/1-loss is a direct measure of the classification error rate of a model,

which is 1 minus the accuracy of this model. If, on a set of 100 points, a model makes wrong predictions about 25 points, the average of the 0/1-loss will be 0.25, and the model has thus an accuracy of 0.75, or 75%, on this set of points. Except for this loss, it is difficult to directly interpret the exact value of loss functions. Consequently, the other loss functions are frequently used as objectives to minimize in order to learn a model, as explained in the next section, but rarely used as performance metrics to evaluate a model quality afterwards. It is the opposite situation for the 0/1-loss which is easy to interpret, but difficult to use in optimization problems. Chapter 5 discusses extensively how to rigorously assess model performances.

2.3.2 Notions of Risk

Functional Risk

The functional risk is the measure of the loss of a given hypothesis $h \in \mathcal{H}$ on the whole $P(\mathcal{X}, \mathcal{Y})$ distribution (i.e. on every possible input). Let us remain very general here by not specifying the type of loss considered. The functional risk can thus be written:

$$R(h) = \int \mathcal{L}(y, h(\mathbf{x})) dP(\mathcal{X}, \mathcal{Y}). \quad (2.13)$$

The best model h_0 is the one minimizing the functional risk R :

$$h_0(\mathbf{x}) = \underset{h \in \mathcal{H}}{\operatorname{argmin}} R(h). \quad (2.14)$$

However, R is most of the time not available because P itself is rarely known and the training set has generally a finite size.

Empirical Risk

Since the minimization of the functional risk (2.14) is most of the time not tractable, the *empirical risk* may be minimized instead. The empirical risk is the value of the loss computed only on the available training set of size n :

$$R_{\text{emp}}(h) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}(y_i, h(\mathbf{x}_i)). \quad (2.15)$$

If $\mathcal{L}$ and $\mathcal{H}$ are chosen adequately⁶, it becomes possible to find the model h_{emp} that minimizes R_{emp} :

$$h_{\text{emp}}(\mathbf{x}) = \underset{h \in \mathcal{H}}{\operatorname{argmin}} R_{\text{emp}}(h). \quad (2.16)$$

⁶In our case, $\mathcal{H}$ is restricted to the family of hyperplanes.

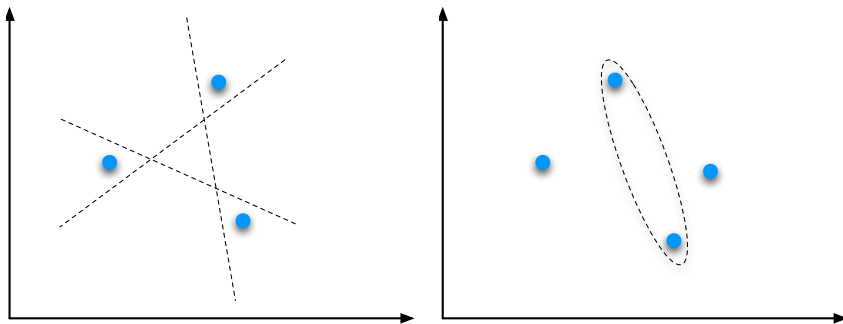


Figure 2.3 Left: The VC dimension of hyperplanes in 2 dimensions is 3. Right: Four points cannot be separated into two classes in all $2^d = 4$ possible ways using hyperplanes. For example, the set of two points in the dashed shape cannot be separated from the two other points by a hyperplane.

This is called the empirical risk minimization principle (ERM principle).

Overfitting

Without entering the details of statistical learning theory, minimizing the empirical risk as such is not a satisfactory condition to obtain models with good generalization performances. Without restricting $\mathcal{H}$ (without an *inductive bias*), a model cannot learn anything except the training data itself. This phenomenon is known as *overfitting*. A model behaving perfectly on the training data, and returning a random value for other input values is a minimizer of the empirical risk (2.15), but is clearly not what one would expect as the result of a learning process. Such a model *overfits* the training data. On the other extreme, restricting too much $\mathcal{H}$ could prevent the learning process to choose h close enough to h_0 , the minimizer of the functional risk (2.13). The next section introduces some more concepts and explains how to adequately choose an inductive bias.

2.3.3 Regularized Risk

An indicator function separates a given set of points into two subsets: those for which this indicator takes the value 1, and those for which it takes the value 0. Most binary classification models are indicator functions, or real functions approximating indicator functions.

VC Dimension

In the case of $\mathcal{H}$ being a class of indicator functions, the Vapnik-Chervonenkis dimension, the *VC dimension*, d_{VC} of $\mathcal{H}$ is the maximum number d of points $\mathbf{x}_1, \dots, \mathbf{x}_d$ that can be separated into two classes in all 2^d possible ways using functions in $\mathcal{H}$.

The VC dimension of $\mathcal{H}$ being the class of hyperplanes in $\mathbb{R}^p$ is $d_{VC} = p + 1$. It is illustrated in Figure 2.3 for $p = 2$, thus with a VC dimension of 3. This class of function has a VC dimension equal to the number of free parameters it has (one for w_0 and p for the w_j 's). This is a particular case which is not often verified. This holds both for hyperplanes in Equation (2.5) and for binary classification linear models of Equation (2.6). Generally speaking, the VC dimension of a class of functions is different from the number of its parameters, either larger or smaller and can even be infinite, as it is the case for some sinusoidal functions (Vapnik, 1995).

Statistical learning theory upper-bounds the true, functional risk (2.13) by a sum of two terms:

$$R(h) \leq R_{emp}(h) + \Phi\left(\frac{d_{VC}}{n}\right). \quad (2.17)$$

On the right side, the first term is the empirical risk (2.15). The second term Φ is a confidence interval. Its complete expression is more complex than the one reported here, and contains a confidence measure. The inequality in (2.17) thus holds with a given confidence. The larger this confidence, the larger the second term. A major advantage of this bound is that it does not depend anymore on $P(\mathcal{X}, \mathcal{Y})$. Without considering the extended expression of Φ , what really matters is that it depends on n , the size of the training set, and d_{VC} , the VC dimension of the considered class of model $\mathcal{H}$. If n is small, the confidence interval is large. If the VC dimension is large, the confidence interval is large. This is another point of view on the curse of dimensionality with the same conclusion: it is hard to learn when the ratio between the number of dimensions and the number of points is high. But here, the VC dimension is considered rather than the input space dimensionality, and this makes a difference. It is thus the VC dimension rather than the number of parameters of a model that matters for generalization ability of learning machines.

In that respect, linear models are convenient since their VC dimension

is easy to determine, and is equal to the number of input features plus one. Many other common models have a VC dimension larger than linear models respectively to their input feature space. For example, the one-dimensional classifier defined by:

$$f(x) = \text{sign}(\sin(wx)), \quad w, x \in \mathbb{R}, \quad (2.18)$$

can shatter an arbitrarily large number of points, just by increasing w . It has thus an infinite VC dimension, although it has only one parameter (Vapnik, 1995). Polynomial models of degree k in $\mathbb{R}^p$ are another example (Ben-David and Lindenbaum, 1998). Their VC dimension is defined by:

$$\binom{p+k}{k} = \frac{(p+k)(p+k-1)(p+k-2)\dots(p+1)}{k(k-1)(k-2)\dots 1} \quad (2.19)$$

Polynomial models of degree $k = 2$ when $p = 2$ dimensions have a VC dimension of $\binom{4}{2} = 6$, for example. Polynomial models thus have a higher VC dimension than linear models, except when $k = 1$, in which case they degenerate into linear models, with an equivalent VC dimension of $p + 1$.

Regularized Risk

The bound (2.17) is known to be loose. This implies that it is useless to try to minimize its exact expression to minimize the functional risk. The optimization community has not been stopped by this argument and designed many optimization problems for machine learning in the spirit of bound (2.17). This technique of optimization is known as *regularization*. It follows a pattern in which the objective function to minimize is the sum of a loss function $\mathcal{L}$ and a *regularizer* $\mathcal{R}$:

$$\text{argmin } \mathcal{L} + \lambda \mathcal{R}. \quad (2.20)$$

The minimization of the loss corresponds to the minimization of the empirical risk. The minimization of the regularizer tends to minimize the second term of the bound (2.17), without a direct mapping to this term. A regularizer is a term penalizing the *complexity* of the models (assuming that complexity is related to the VC dimension). The λ constant is known as the regularization constant, which tunes the respective importance of $\mathcal{L}$ and $\mathcal{R}$ in the optimization process. Through this thesis, many models fit within this *loss & regularizer pattern* for optimization. The pattern is recalled as such models appear, beginning in the next section with support vector machines. Four (pseudo-)norms are frequently used for regularization, with different effects on the learned models.

Table 2.2 Four common regularizers for linear models. The first one is the zero-pseudo-norm (card is the set cardinality). The three others are respectively the one-, two- and infinite-norm.

$L_0 =$	$ \mathbf{w} _0$	$=$	$\text{card}\{w_j w_j \neq 0\}$
$L_1 =$	$ \mathbf{w} _1$	$=$	$\sum_{j=1}^p w_j $
$L_2 =$	$ \mathbf{w} _2$	$=$	$\sqrt{\sum_{j=1}^p w_j^2}$
$L_\infty =$	$ \mathbf{w} _\infty$	$=$	$\max \mathbf{w} $

Those regularizers are summarized in Table 2.2. Their effects are discussed throughout the text. Note that even if we note $\mathbf{w} = w_0, \dots, w_p$ to simplify notations of hyperplanes for example, the w_0 term is generally **not** included in the regularization. This is visible in Table 2.2 where sums include w_j with $j = 1$ to p , not 0 to p .

2.4 Support Vector Machine

The support vector machines (SVM) are a direct by-product of statistical learning theory (Boser *et al.*, 1992) and have been shown to deliver state-of-the-art performances (Mukherjee, 2003; Ben-Hur *et al.*, 2008). The simplest form of SVM is the linear SVM, and is the one considered in this thesis. It makes decisions based on a hyperplane, just like in Equation (2.6). The central theme of this text is not a discussion on kernel methods. SVM in their kernelized aspects are thus not touched here. For an extensive discussion on SVM, the reader is invited to refer to the work of Schölkopf and Smola (2001). The linear SVM offers the advantage of model interpretability in term of the input features, like any linear model, but has the extra advantage of minimizing the risk in the spirit of Equation (2.17). This section first covers the case where classes are linearly separable, and later on the case of linearly non-separable data.

2.4.1 Hard-Margin SVM

Maximum Margin

A linear SVM is not just any linear model. An SVM is a binary classification model based on the maximum margin principle. A set of points is separated

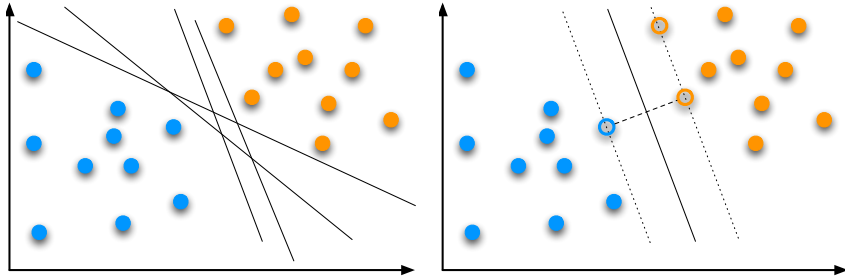


Figure 2.4 Left: Examples of separating hyperplanes in 2 dimensions. Separable classes can be discriminated by an infinite number of hyperplanes. Right: A maximum margin hyperplane. The points on the margin are the support vectors. For separable classes, the maximum margin hyperplane is unique.

by a maximum margin hyperplane if it is separated without error and the distance M between the hyperplane and the closest points on each side of it is maximal. This is illustrated in Figure 2.4 (right). The distance between any point $\mathbf{x}$ and the hyperplane h of parameters $\mathbf{w}$ is:

$$\frac{|h(\mathbf{x})|}{\|\mathbf{w}\|_2}. \quad (2.21)$$

The points for which the distance to the hyperplane is exactly equal to the margin M are called the *support vectors* (SV) of the decision boundary. Those are the closest points to the hyperplane. They correspond to the circles in Figure 2.4 (right). The margin can thus itself be defined in terms of the support vectors:

$$M = \min_i \frac{y_i h(\mathbf{x}_i)}{\|\mathbf{w}\|_2} \quad \forall i \in \{1, \dots, n\}, \quad (2.22)$$

and this margin has to be maximized with respect to the hyperplane parameters. This leads to the following minimax problem:

$$\max_{\mathbf{w}} \min_i \frac{y_i h(\mathbf{x}_i)}{\|\mathbf{w}\|_2} \quad \forall i \in \{1, \dots, n\}. \quad (2.23)$$

This problem is undetermined: an infinity of equivalent solutions in terms of $\mathbf{w}$ could be found (each solution being a multiple of another). We can arbitrarily fix the distance of the closest points to the hyperplane equal to one unit of the norm of $\mathbf{w}$:

$$\frac{y_i h(\mathbf{x}_i)}{\|\mathbf{w}\|_2} = \frac{1}{\|\mathbf{w}\|_2} \quad \forall i \in SV. \quad (2.24)$$

This is equivalent to fix the margin to one unit of the norm of $\mathbf{w}$. Consequently, for all points (either support vectors or not):

$$y_i h(\mathbf{x}_i) \geq 1 \quad \forall i \in \{1, \dots, n\}. \quad (2.25)$$

SVM Primal

The maximization of the margin in Equation (2.22) comes then to minimize its denominator ($\|\mathbf{w}\|_2$) under the constraints defined above in Equation (2.25). For convenience in further developments, $\|\mathbf{w}\|_2$ is replaced by $\frac{1}{2}\|\mathbf{w}\|_2^2$, leading to an equivalent solution. This is called the SVM primal problem:

$$\underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{w}\|_2^2, \quad (2.26)$$

subject to

$$y_i h(\mathbf{x}_i) \geq 1 \quad \forall i \in \{1, \dots, n\}. \quad (2.27)$$

This problem can be rewritten to fit within the loss & regularizer pattern (2.20):

$$\underset{\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^n (y_i h(\mathbf{x}_i) - 1)_\infty + \lambda \|\mathbf{w}\|_2^2. \quad (2.28)$$

The $(z)_\infty$ function is 0 if $z \geq 0$ and ∞ otherwise. Stated this way, the support vector machine is not a convenient problem to solve.

SVM Dual

The SVM primal constrained minimization Problem (2.26) can be replaced along with its constraint of Equation (2.27) by an unconstrained equivalent form with Lagrange multipliers:

$$L(\mathbf{w}, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|_2^2 - \sum_{i=1}^n \alpha_i (y_i h(\mathbf{x}_i) - 1). \quad (2.29)$$

The $\alpha_i \geq 0$ are the Lagrange multipliers introduced by the constraints of Problem (2.26). The strategy now consists of expressing the derivatives of L w.r.t. $\mathbf{w}$ and equalling them to zero. It leads to the following equalities corresponding to the optimality conditions of L :

$$\begin{cases} w_j = \sum_{i=1}^n \alpha_i y_i x_{ij} & \forall j \in \{1, \dots, p\}, \\ 0 = \sum_{i=1}^n \alpha_i y_i. \end{cases} \quad (2.30)$$

Using those conditions in Equation (2.29) allows the elimination of $\mathbf{w}$ and leads to the SVM dual problem, expressed only in terms of α :

$$L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{l=1}^n \alpha_i \alpha_l y_i y_l \langle \mathbf{x}_i, \mathbf{x}_l \rangle, \quad (2.31)$$

subject to the following constraints:

$$\begin{cases} \alpha_i \geq 0 \quad \forall i \in \{1, \dots, n\}, \\ 0 = \sum_{i=1}^n \alpha_i y_i. \end{cases} \quad (2.32)$$

This problem has to be minimized with respect to the α .

What advantage is there in the dual form (2.31)? First, the number of variables to optimize is not $p + 1$ (the length of $\mathbf{w}$), but is now n (the length of α). In the context of this thesis where $n \ll p$, this change is a major benefit. Second, once optimized, the number of non-null α_i is generally small. This is due to the fact that the conjunction of $\alpha_i \geq 0$ and $\sum_{i=1}^n \alpha_i$ in Equation (2.31) corresponds to a one norm penalty $\|\alpha\|_1$ enforcing sparsity over the samples when it is minimized⁷. The non-null α_i correspond to the support vectors. The decision function can be expressed only in terms of the support vectors:

$$h(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i \langle \mathbf{x}, \mathbf{x}_i \rangle = \sum_{i \in SV} \alpha_i y_i \langle \mathbf{x}, \mathbf{x}_i \rangle. \quad (2.33)$$

The w_0 parameter still has to be computed. This can be done by injecting Equation (2.33) into Equation (2.24) for any single chosen support vector. In practice this is done for every support vector, and then averaged for numerical robustness concerns. Another advantage of SVM with respect to, for example, multi-layer-perceptron (MLP) (Bishop, 2006), is that it is a convex optimization problem with a unique optimum. Finally, yet another appealing characteristic of SVM is the low level of meta-design required. MLP require to carefully design a specific architecture for every new application. SVM do not need to take the problem structure into account to already deliver competitive results.

2.4.2 Soft-Margin SVM

So far, we have assumed that classes are linearly separable (non-overlapping). It might not be the case in practice. A modified version of the support vector

⁷The L_1 norm and its sparsity-inducing character are more extensively discussed in Section 3.4.2

machine has been introduced by Cortes and Vapnik (1995) to face this setting: the soft-margin SVM. In this case, having points *inside* the margin, or on the wrong side of the hyperplane is tolerated, but penalized. The penalization for point $\mathbf{x}_i$ is encoded in a *slack variable* ξ_i . The constraints of Equation (2.25) are rewritten:

$$y_i h(\mathbf{x}_i) \geq 1 - \xi_i \quad \forall i \in \{1, \dots, n\}, \quad (2.34)$$

where $\xi_i \geq 0$. Points for which $\xi_i = 0$ are correctly classified and at a distance from the hyperplane greater or equal to the margin. Points for which $0 < \xi_i < 1$ are correctly classified, but inside the margin. Points for which $\xi_i = 1$ are right on the decision boundary, i.e. the hyperplane. Points for which $\xi_i > 1$ are wrongly classified. The objective function of the soft-margin SVM still tends to maximize the margin, but also to minimize the penalties of the slack variables:

$$\underset{\mathbf{w}}{\operatorname{argmin}} C \sum_{i=1}^n \xi_i + \frac{1}{2} \|\mathbf{w}\|_2^2, \quad (2.35)$$

subject to:

$$\begin{cases} y_i h(\mathbf{x}_i) \geq 1 - \xi_i & \forall i \in \{1, \dots, n\}, \\ \xi_i \geq 0 & \forall i \in \{1, \dots, n\}. \end{cases} \quad (2.36)$$

Again, this problem fits in the loss & regularizer pattern. The C constant rules the tradeoff between the slack variables minimization and the margin maximization. It is the inverse of a regularization constant: the larger C , the less the model is regularized. When $C \rightarrow \infty$, Problem (2.35) is equivalent to the hard margin Problem (2.28). The loss that is minimized in Problem (2.35) is the hinge loss, illustrated in Figure 2.2. Its expression, given by Equation (2.9), is obtained by manipulating the constraints of Problem (2.35). The hinge loss is convex, but not strictly convex since it contains a plateau at the optimal value of 0. Its global optimum can be obtained for an infinite number of values of its argument. The primal SVM problem is the natural SVM form, but is rarely the one that is optimized in practice. A dual formulation of the soft-margin problem can be formulated, similar to the hard-margin case. It leads back to the separable case problem in Equation (2.31). Only the constraints are modified:

$$\begin{cases} 0 \leq \alpha_i \leq C & \forall i \in \{1, \dots, n\}, \\ 0 = \sum_{i=1}^n \alpha_i y_i. \end{cases} \quad (2.37)$$

In the simplest case of linear hard-margin SVM, no meta-parameter has to be tuned. In the soft-margin case, the number of meta-parameter to tune is one: the C constant in Equation (2.35). The values of the α 's can be interpreted as in

the hard-margin case, but with slightly modified meanings. Like in the hard-margin case, points $\mathbf{x}_i$ for which $\alpha_i = 0$ are correctly classified and outside the margin. The other points contribute to the decision boundary and are the support vectors. What changes is that if $\alpha_i = C$, point $\mathbf{x}_i$ lies inside the margin, either correctly ($\xi_i < 1$) or wrongly classified ($\xi_i > 1$). If $\alpha_i < C$, point $\mathbf{x}_i$ lies on the margin.

Joachims Heuristic

A heuristic due to [Joachims \(2002\)](#) is available to automatically fix the C constant. This heuristic is often used in the experiments presented in this thesis. It generally leads to results comparable to those obtained with an optimized parameter value. Its expression for the linear SVM is:

$$C_J = \frac{1}{\frac{1}{n} \sum_{i=1}^n \sqrt{\langle \mathbf{x}_i, \mathbf{x}_i \rangle}}. \quad (2.38)$$

The intuition behind this automatic setting is that $\frac{1}{C}$ can be interpreted as a regularization constant. So the regularization is proportional in this case to $\frac{1}{n} \sum_{i=1}^n \sqrt{\langle \mathbf{x}_i, \mathbf{x}_i \rangle}$, which is the mean of the euclidean norm of all point vectors. The larger this mean, the stronger the regularization. With a much stronger regularization, the data itself would be masked by the regularization. Conversely, a much weaker regularization would be without effect. This heuristic is a way of automatic regularization calibration.

2.5 Logistic Regression

It is convenient that the decision value of a model has some interpretable properties. In the case of SVM, it is difficult to say much about this decision value, except for its sign.

Probabilistic Outputs

Suppose we want a function $g(h(\mathbf{x}_i))$ to predict the *probability* of sample i to belong to class 1, i.e. the probability that $y_i = 1$. If $P(y_i = 1|\mathbf{x}_i)$ is the probability of a binary event, then $P(y_i = 1|\mathbf{x}_i) / (1 - P(y_i = 1|\mathbf{x}_i))$ is known as the *odds* of this event. The *logit function* ([Berkson, 1944](#)) is defined as:

$$\text{logit}(P(y_i = 1|\mathbf{x}_i)) = \log \frac{P(y_i = 1|\mathbf{x}_i)}{1 - P(y_i = 1|\mathbf{x}_i)}. \quad (2.39)$$

The logit function is represented in Figure 2.5 (left). We can impose that the linear model h is equivalent to the logit function before estimating the model parameters $\mathbf{w}$:

$$\text{logit}(P(y_i = 1|\mathbf{x}_i)) = \log \frac{P(y_i = 1|\mathbf{x}_i)}{1 - P(y_i = 1|\mathbf{x}_i)} = \langle \mathbf{x}_i, \mathbf{w} \rangle = h(\mathbf{x}_i). \quad (2.40)$$

Then, the probability that sample i belongs to the class 1 can be expressed as:

$$g(h(\mathbf{x}_i)) = P(y_i = 1|\mathbf{x}_i) = \frac{1}{e^{-\langle \mathbf{x}_i, \mathbf{w} \rangle} + 1}, \quad (2.41)$$

which is known as the *logistic function*, displayed in Figure 2.5 (right), that can be rewritten:

$$g(h(\mathbf{x}_i)) = \frac{1}{e^{-h(\mathbf{x}_i)} + 1}. \quad (2.42)$$

The logistic function g has several convenient properties. It is continuous, defined for $h \in [-\infty, +\infty]$, and is differentiable on its whole domain. The image of the function is naturally contained in $[0, 1]$, which is expected for a probability. In binary classification problems, the probability that sample i belongs to the other class is $1 - P(y_i = 1|\mathbf{x}_i)$. A distinction must be made here between the decision process and the probabilistic interpretation. The decision of classifying sample i in class 1 can be made either based on the sign of h , i.e. f as in Equation (2.6), or as follows:

$$\hat{y}_i = \begin{cases} +1 & \text{if } g(h(\mathbf{x}_i)) > 0.5, \\ -1 & \text{if } g(h(\mathbf{x}_i)) < 0.5. \end{cases} \quad (2.43)$$

The case of $g(h(\mathbf{x}_i)) = 0.5$ corresponds to an equivalent probability of belonging to any class (see the dashed line in Figure 2.5, right). Such cases can be untied according to the classes priors for example. The decision can thus be made by two different but equivalent means, g or f . The probabilistic interpretation, however, is only valid for g , not for f .

The parameters of the logistic regression, the $\mathbf{w}$ vector, have to be estimated from the data. This can be done in many ways and a good review on the topic is the work of Minka (2007). Most techniques minimize, via different procedures, the *logistic loss* over the set of available points:

$$\sum_{i=1}^n \mathcal{L}_{\text{logistic}}(y_i h(\mathbf{x}_i)) = \sum_{i=1}^n \log \left(1 + \exp^{-y_i \langle \mathbf{x}_i, \mathbf{w} \rangle} \right). \quad (2.44)$$

This loss is the negative log-likelihood of Equation (2.42) over all available points. Minimizing it comes thus to identify the model that best fits the data.

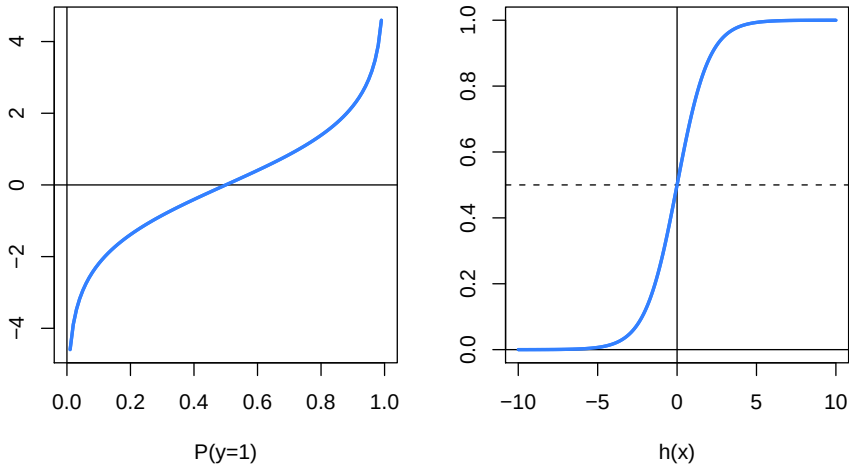


Figure 2.5 Left: The logit function. Right: The logistic function. The dashed line represents the decision threshold (a probability of 0.5 to belong to any class).

2.5.1 SVM and L_2 -Regularized Logistic Regression

SVM have been successfully used in many applicative contexts and are well principled. Logistic regression on the other hand offers a nice probabilistic interpretation. SVM is naturally regularized while logistic regression, as such, is not. Logistic regression has however been extended, mainly with an L_1 regularizer, leading to the generalized lasso (Roth, 2002), another example of the loss & regularizer pattern of Equation (2.20). The L_1 regularizer induces model sparsity with respect to the input features. The next chapter details such approaches providing feature selection embedded within classification.

Logistic Ridge Regression

Another classification model is the L_2 -regularized logistic regression, also known as logistic ridge regression. The L_2 regularizer generally offers model robustness by penalizing large values of any w_j but it does not induce sparsity with respect to the input features as the L_1 regularizer does. We show here that linear SVM is an approximation of the L_2 -regularized logistic regression. The objective function of the L_2 -regularized logistic regression is:

$$\operatorname{argmin}_{\mathbf{w}} \sum_{i=1}^n \log \left(1 + \exp^{-y_i \langle \mathbf{x}_i, \mathbf{w} \rangle} \right) + \lambda \|\mathbf{w}\|_2^2. \quad (2.45)$$

The regularization term in both models are similar: a squared two-norm over the parameters vector. The other term of the objective function is the loss. In the case of SVM, it is the hinge loss. In the case of logistic regression, it is the logistic loss. Those two losses are highly similar, as illustrated in Figure 2.2. The linear support vector machine can thus be seen as an affine approximation to the L_2 -regularized logistic regression, without the extra advantage of offering a probabilistic output. Vapnik (1995) explains how to get as close as desired to L_2 -regularized logistic regression with linear SVM by increasing the number of affine components of the hinge loss.

2.6 Tree-Based Ensemble Methods

This section covers classification techniques that are quite different from the ones introduced in previous sections. The methods described here are based on two key ingredients: decision trees and ensemble methods. Decision trees are a classification or regression model having typical properties like a tree structure, a convenient graphical representation, an easy interpretation, and many other interesting properties. They are described in Section 2.6.1. Ensemble methods have originally been developed to enhance classification performance (Diettrich, 2000). The general idea of that family of techniques consists in combining lots of different models in a global, more robust model. Section 2.6.2 introduces three types of ensemble methods based on decision trees: bagging, the random forests and extremely randomized trees.

2.6.1 Decision Trees

A decision tree is a hierarchical structure making decision according to a sequence of simple decision steps. Many types of decision trees have been proposed with varying properties and designs, such as ID3 (Quinlan, 1990), C4.5 (Quinlan, 1993) and CART (Breiman *et al.*, 1984). We focus here on CART (Classification And Regression Trees) as typical representatives of decision trees.

CART produces binary decision trees, meaning that each node in the tree has two children. Such a binary tree corresponds to a recursive binary partitioning of the input space. Figure 2.6 shows such a binary tree and the corresponding partition of the two-dimensional input space. Advantages of decision trees can be noticed on this figure, on top of their convenient graphical representation. Decision trees are naturally adequate for dealing

with more than two classes. They do not make strong assumptions on classes distributions (they allow many other distributions than gaussians for example). Decision trees like CART lend themselves both to classification and to regression. They accept any type of features: binary, continuous, categorical. Finally, they can be translated into human-readable decision rules. For example on Figure 2.6: *If x_1 is smaller than or equal to i and if x_2 is smaller than j , then the sample belongs to class A.*

The learning process in the case of decision trees comes to identify the structure of the tree as well as the split criteria at each node and also the decision value in each leaf node. Given a training set of n samples $\mathbf{x}_i \in \mathbb{R}^p$ and the corresponding labels $y_i \in \{-1, +1\} \forall i \in \{1, \dots, n\}$, the enumeration of all possible binary trees is generally infeasible for problems of relevant sizes. It is thus in most cases impossible to identify, for example, the minimal-size and optimal tree according to a loss $\mathcal{L}$. The search for a convenient tree through the whole search space generally follows a greedy strategy. It starts with a single root node corresponding to the whole input space, and adds nodes one at a time. At each internal node k , the space is split in two parts. Each split corresponds to a particular feature $j \in \{1, \dots, p\}$ and to a particular threshold value θ so that a decision rule is created:

$$\text{go to } \begin{cases} \text{left child} & \text{if } x_j \leq \theta_k, \\ \text{right child} & \text{otherwise.} \end{cases} \quad (2.46)$$

At each terminal node, a prediction value is output, corresponding to the majority class of samples from the training set falling into this node⁸. Note that CART produces internal nodes in which the split is done according to a univariate decision, based on a single feature at a time. Globally, decision trees are however multivariate. Now how to choose j and θ of Equation (2.46) at a new node k ?

Gini Index

The choice of the feature used to make the split at node k is based on the Gini impurity function (Bishop, 2006). A node is considered pure if all examples assigned to it have the same class label. The Gini index measures the impurity

⁸This holds for classification. For regression, the predicted value is the average of the y_i corresponding to the training points falling into this terminal node

at a node k according to:

$$\text{Gini}(\mathbf{X}_k) = 1 - \sum_{i=1}^{|C|} (p_{\mathbf{X}_k}^{c_i})^2, \quad (2.47)$$

where $\mathbf{X}_k$ is the set of points at node k , $|C|$ is the number of possible classes and $p_{\mathbf{X}_k}^{c_i}$ is the proportion of points in $\mathbf{X}_k$ belonging to class c_i . In a binary classification setting, this comes to:

$$\text{Gini}(\mathbf{X}_k) = 1 - ((p_{\mathbf{X}_k}^+)^2 + (p_{\mathbf{X}_k}^-)^2) = 2p_{\mathbf{X}_k}^+ p_{\mathbf{X}_k}^- = 2p_{\mathbf{X}_k}^+ (1 - p_{\mathbf{X}_k}^+). \quad (2.48)$$

The Gini index is maximal when $p_{\mathbf{X}_k}^+ = p_{\mathbf{X}_k}^- = 0.5$. Conversely, it vanishes when samples from only one class are present at node k (when node k is pure). Splitting samples of $\mathbf{X}_k$ according to feature j produces two children nodes, and j is chosen so that the drop of impurity is maximal. This drop is measured as the difference between the Gini index value at node k and the sum of Gini index values at the left and right children node:

$$\text{Drop}(\mathbf{X}_k, j, \theta) = \text{Gini}(\mathbf{X}_k) - \left(\frac{|\text{left}|}{|k|} \text{Gini}(\mathbf{X}_{\text{left},j,\theta}) + \frac{|\text{right}|}{|k|} \text{Gini}(\mathbf{X}_{\text{right},j,\theta}) \right), \quad (2.49)$$

where $\mathbf{X}_{\text{left},j,\theta}$ is the set of points from $\mathbf{X}_k$ falling into the left child of node k according to a comparison of feature j with value θ , as in Equation (2.46). The choice of j and θ is thus made in order to maximize this drop:

$$\underset{j,\theta}{\text{argmax}} \text{Drop}(\mathbf{X}_k, j, \theta). \quad (2.50)$$

Another issue for the learning of CART trees is when to stop growing the tree. Two conditions may apply in this case: either the Gini index falls below a predefined threshold ($\text{Gini}(\mathbf{X}_k) < G_0$) or the number of samples at node k is smaller than a predefined threshold ($|\mathbf{X}_k| < n_0$). G_0 and n_0 are meta-parameters of the learning algorithm. Usually in CART, a post-pruning strategy is also considered, discarding some nodes of the tree. This post-pruning is intended to avoid some overfitting, but has to be tuned on an independent validation set or by cross-validation (see Chapter 5).

2.6.2 Ensemble Methods Based on Trees

Ensemble methods based on decision trees have a common trait. They generate lots of trees and combine them in a voting system to output a likely more

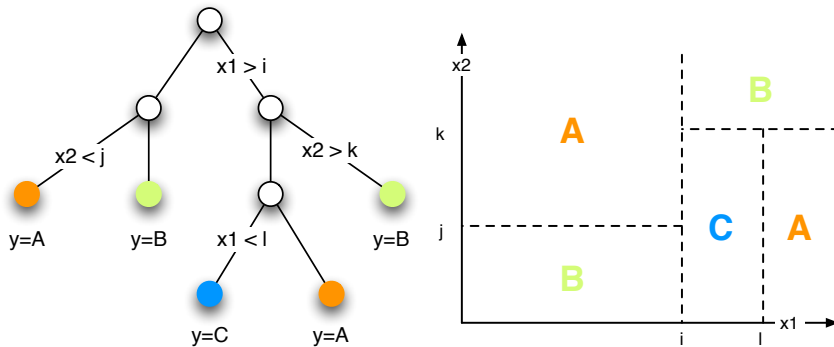


Figure 2.6 Left: A binary tree. Right: The corresponding partition of the input space.

robust prediction. To make such system work, trees have to be different and complementary. If a deterministic decision tree learning algorithm is run on the same data to generate many trees, all trees of the ensemble method would be identical. Tree-based ensemble methods thus make heavy use of randomization to generate many different trees. This randomization can take place at three different levels:

1. **Samples:** Variability can be introduced by building each tree on a randomly drawn subsampling of the data points.
2. **Features:** Another option is to build the trees on different randomly drawn subsets of the features.
3. **Decision threshold:** When a tree is grown, each node k corresponds to a split. The decision to send a sample towards the left or the right sub-tree depends on the decision threshold θ_k . This value can also be selected more or less randomly.

Three different methods are introduced in the following subsections. They mainly differ in the type of randomization they consider. The bagging technique uses the first one. The random forests use the two first ones, on the samples and the features. The extremely randomized trees use the last two, on features and split threshold value. These three methods are described next.

Bagging

Decision trees have been shown to offer a low bias in a large number of datasets, but a large variance ([Breiman, 1996a](#)). The idea of ensemble method

for classification based on trees has been introduced by [Breiman \(1996b\)](#) and is known as bagging. Bagging tends to produce classifiers more robust than single trees. The diversity from tree to tree is induced by a resampling of data points.

Let us consider the bootstrap sample $\mathbf{X}^*$, which is a random draw with replacement of n points among $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$. A number $nTree$ of such bootstrap samples are drawn. An unpruned CART tree is grown on each of these bootstrap samples.

To obtain a classification decision for a new point $\mathbf{x}$ from the bag of $nTree$ decision trees, each tree classifies $\mathbf{x}$ and the majority vote is the one that is finally kept.

Random Forests

The random forests ([Breiman, 2001](#)) are an ensemble method for classification based on an ensemble of decision trees (hence the name of *forest*), each of which is learned on a random subsampling of the data (hence the term *random*). It can be seen as a refinement of bagging where an extra level of randomization is added.

Random forests consider the same bootstrapping strategy on the data points as in bagging. A number $nTree$ of bootstrap samples of n points are drawn with replacement among $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$. The other subsampling strategy is jointly applied, this time on the features of the original data. A number $mTry$ of features are selected with equal probability, without replacement. A decision tree is learned on each bootstrap sample $\mathbf{X}^*$ restricted to the set of selected features according to the CART setting with no post-pruning of the trees. This set of trees naturally forms the forest. The randomization at the feature level is inspired from the *random subspace method* restricting the learning of a model on a subset of features. It was first introduced by [Ho \(1998\)](#).

To obtain a classification decision for a new point $\mathbf{x}$ from a random forest of size $nTree$, each tree classifies $\mathbf{x}$ and the majority vote is the one that is finally kept.

Random forests have many interesting properties, such as a bound on

the generalization error. A consequence of this bound is to prove that the growing of the forest is not likely to produce overfitting (Breiman, 2001). Extensions of the random forests for feature selection have also been proposed and are discussed later on in Chapter 3.

Extremely Randomized Trees

Extremely randomized trees, also known as extra trees, are also an ensemble method for classification based on decision trees (Geurts *et al.*, 2006). They have two main differences with respect to the random forests. First, each decision tree is grown on all the samples available. No bootstrap samples are considered. The usage of the full original learning sample rather than bootstrap samples is intended to minimize the bias of the model. Second, the split threshold θ_k is chosen randomly at each node k .

For each of the $mTry$ considered features at node k , a threshold value is chosen randomly according to a uniform distribution over the interval between the minimum and maximum value appearing for that feature. The pair feature/threshold at node k is selected as the best one among the $mTry$ randomly chosen ones. This procedure might seem strange. How a random choice of candidate threshold could lead to an improvement on the robustness of the model? In fact, it has been observed that the variance of the threshold with respect to samplings variation in bagging or random forest is so high that it can be considered as implicit pseudo-random (Wehenkel, 1997). Extra trees only make this random character explicit, while taking advantage of a larger number of samples for the estimation of each tree. They thus undergo less variance with respect to the training data by using them less than other approaches for the computation of this threshold.

To obtain a classification decision for a new point $\mathbf{x}$ from the set of $nTree$ decision trees of an extra tree model, each tree classifies $\mathbf{x}$ and the majority vote is the one that is finally kept.

Feature Selection

3

This chapter discusses the second major aspect of this thesis: feature selection. Section 3.1 introduces the basics of feature selection and states some differences between feature selection and extraction or univariate and multivariate selection. The remaining of the chapter presents three widely used selection patterns: filters, wrappers and embedded methods. Those three approaches are the topics of three sections. Section 3.2 presents several filter approaches, like the Golub's ratio or hypothesis testing. Section 3.3 explains how wrappers can be implemented in practice, and Section 3.4 details several selection methods embedded in classification techniques. It covers the recursive feature elimination, the generalized lasso, the AROM methods and the random forests as a mean of feature selection. In this thesis, mainly filters and embedded methods are considered for reasons detailed in this chapter. Two reviews on feature selection can be found in the works of [Guyon and Elisseeff \(2003\)](#) and [Saeys *et al.* \(2007\)](#), the latter focused on biological applications. Metrics for assessing feature selection quality in a context of classification are discussed in Chapter 4, not in this chapter.

3.1 Feature Selection

Feature selection refers to the set of dimensionality reduction techniques aiming at identifying *important* features of the input space. This definition is a bit vague, since the notion of importance may vary according to the context. The importance of a feature depends on the subsequent use of this feature. Is it data visualization? Generative process understanding? Reducing the measurement and storage requirements? Reducing training and test times? Or improving prediction performance in an empty space setting?

Classification of microarray data is a challenging problem as it typically relies on a few tens of samples but several thousand dimensions (genes). Feature selection techniques are commonly used in this context. *Biomarker selection* specifically refers to the identification of a small set of genes, also called a *signature*, related to a pathology or to an observed clinical outcome after a treatment. Two main objectives are central to this thesis. The selection should provide sets of features...

1. ... sparse enough to be interpretable and to reduce the costs involved by gene expression measurements and ...
2. ... on which high-performance classification models can be built.

Interpretability

The selection increases the interpretability of the predictive models. To some extent, acceptable predictions obtained with a small number of genes lead to infer a high probability that those selected genes are implied in the biological process corresponding to the labels to predict. The use of “implied” and not “responsible” or some other causal term is totally on purpose, since dependence does not imply causality. Causality is not considered in this thesis, but the selected dimensions are likely to be somehow involved in the biological process. Feature selection thus serves explanatory purposes.

Financial Issues

Another advantage of sparse selection is to make the identified signatures usable in a clinical environment, for example in a diagnosis kit. Microarrays are expensive. If a few tens of biomarkers can be identified, clinical tests can be implemented based on those markers via other, cheaper, technological means. Feature selection can thus reduce the financial cost of further applications. The reduction of the computational cost of subsequent tests is also an advantage of feature selection.

Classification Quality

The second objective states that the features that we look for are important for classification purposes. Feature selection has been shown to sometimes improve the classification accuracy ([Krishnapuram et al., 2004](#)) as it may suppress noisy or irrelevant features. This result is ideal, but rare. The meaning

of this second objective should be understood slightly differently. Given the huge amount of features with respect to the number of samples available, we aim at producing highly sparse feature selection while not loosing too much in classification performances. For example, a move from 55,000 features to 50 features producing a decrease from 95% to 90% in accuracy is a good result, given the advantages provided by the dimensionality reduction explained in the previous paragraphs. Another argument related to classification quality is developed next.

Feature Selection & Statistical Learning Theory

Feature selection for classification can be directly related to regularized learning, at least in the context of linear models. The bound on functional risk in Equation (2.17) is made of a term measuring the empirical risk, which is the classification performances on the training set, and a second term depending on the number of samples available and the VC dimension of the considered models. The number of points is fixed by the training set size. The VC dimension of a hyperplane is equivalent to the number of its free parameters, which is $p + 1$, the number of input features plus the bias term. Feature selection reduces p to $p' \ll p$ by selecting only a small subset of the input features. Feature selection for linear models can thus be seen as a VC dimension reduction, and thus a form of regularization. By reducing the VC dimension while keeping acceptable performances, one increases the probability of good generalization of the learning process. This is only true when the learning of the model and the dimensionality reduction are performed jointly, as it is the case for methods of Section 3.4.

Selection vs. Extraction

Dimensionality reduction techniques other than feature selection are available. For example, principal component analysis (PCA), multidimensional scaling (MDS), as well as non-linear developments in the same spirit, referred to as manifold learning, allow one to strongly reduce the number of dimensions of the data (Lee and Verleysen, 2007). Independent component analysis (Hyvärinen *et al.*, 2001) or auto-encoding MLP (Hinton and Salakhutdinov, 2006) are other options. The result of this dimensionality reduction can be fed into a learning algorithm. Those techniques are generally known as feature extraction and are not considered in this thesis for two main reasons. First, most of such techniques do not make use of the target variable and do not optimize the separability of the classes. They make part of unsupervised

learning. It may be the case in some applications that those techniques improve the classification performances, but it is not their objective and nothing guarantees it. Second, those techniques reduce the dimensionality at the price of loss of interpretability and in many case of non-sparsity in the input feature space. Indeed, most of these techniques are (non-linear) combinations of the original features. The features of the central layer of an auto-encoding MLP are also combinations of (combinations of) the input features.

Univariate vs. Multivariate

Feature selection techniques are either *univariate* or *multivariate*. Univariate techniques typically compute a score corresponding to each feature quality separately while multivariate methods take into account groups of features or interaction between features. An advantage of univariate methods is that their complexity is generally $\mathcal{O}(p)$, while multivariate approaches might consider all possible subsets of the p features, which is not feasible for large p . Drawbacks of univariate methods are numerous. [Guyon and Elisseeff \(2003\)](#) provide several examples where a univariate criterion is likely to miss important interactions between features. Two features apparently useless on their own can be useful with others (think of the XOR-classification problem for example). Multivariate techniques also seem to make more sense in genomic applications, because gene expressions are rarely modulated on their own. Genes work in groups (pathways) and influence each others.

Selection Bias

Deciding on which samples the feature selection should occur is an issue as important as deciding which particular technique to use. If the selection is done on all the samples of a dataset (both training and test samples), the information contained in the test samples potentially induces an optimistic bias in the final performances of the model ([Ambroise and McLachlan, 2002](#)). This is the selection bias. The solution is to perform the selection only on the training samples and then to restrict the test samples to those selected features before making a prediction on them. For example, the success of the recursive elimination technique described next in Section 3.4.1 is partly due to the impressive performances apparently obtained with it, as reported in the work of [Guyon et al. \(2002\)](#). Authors obtain 100% accuracy for several gene expression classification problems. The evaluation protocol in this case precisely comprises a selection bias, overestimating by much the performances, as acknowledged

in (Guyon, 2002). Other papers have the same methodology flaw, as mentioned by Ambroise and McLachlan (2002) and Guyon (2002).

Selection Patterns

Feature selection techniques are generally implemented according to three different patterns, depending on how they interact with the estimation of the classification model: filters, wrappers, and embedded methods. Those three patterns are detailed in the next three sections.

Although we develop the here-above arguments in the context of microarray data analysis, most of the discussion could be transposed to other applicative fields where small- n -big- p data are generated.

3.2 Filters

Filters are the most simple of the three approaches considered in this chapter and perform feature selection by looking at intrinsic properties of the data. Those techniques are disconnected from any further treatment, like classification, and may be applied for their own sake. Their principle is illustrated in Figure 3.1. Most of filters are ranking techniques: they order the features according to a given criterion. In this thesis, mainly two filters are considered: the Golub's ratio, and the t -test, which are described next. A third type of filters, based on information theory, is considered as well. However, those information theoretic filters are not applied in the context of this thesis.

Other filters exist, like the χ^2 (Liu and Setiono, 1995) or correlation-based feature selection (Hall, 2000). The χ^2 statistics is designed for discrete variables, and gene expressions are continuous variables. Discretization is required, which constitutes a loss of information. Correlation-based feature selection can be univariate or multivariate. In both cases, it generally requires to compute a correlation between a feature and a binary discrete class label, although correlation is a measure of linear dependence between variables. Multivariate correlation-based selection methods favor groups of features highly correlated to the target variable, and additionally showing a low correlation between them. Those approaches are not used in this work.

The techniques presented in this section are *specific filters*, meaning that they use an extra information on top of the data points $\mathbf{x}$ themselves. In this

case, they use the class label y . Non-specific filtering techniques (not making use of the labels) are also described and evaluated in Chapter 6. They may be used as a quick preprocessing step.

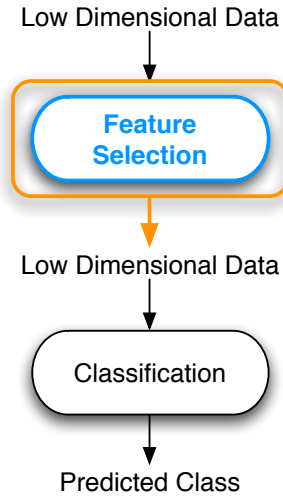


Figure 3.1 Feature selection: the filter pattern.

3.2.1 Golub's Ratio

The Golub's ratio is a univariate feature selection technique that has been used in one of the first successful microarray study (Golub *et al.*, 1999)¹. Its principle is simple: the features to be favored are those which behave the same way within each class, but differently in different classes. It is the ratio between a difference of average values in two classes and the sum of variances in both classes. That is why it is sometimes described as a *signal to noise* ratio. Its exact expression for feature j is:

$$S2N(j) = \frac{|\mu_{j_1} - \mu_{j_2}|}{\sigma_{j_1} + \sigma_{j_2}}, \quad (3.1)$$

where μ_{j_i} is the mean value of feature j across samples of class i and σ_{j_i} is the standard deviation of feature j across samples of class i . This value can be computed for each feature $j \in \{1, \dots, p\}$. The larger the value of the

¹This technique is a standard approach that has not been originally proposed by Golub, but is known under his name in the field of bioinformatics, due to this publication.

ratio, the more the feature is considered as interesting. Once those values are computed, a ranking on the features is obtained.

With such a ranking, one has two options to perform a selection:

1. The signature size s is known in advance, and the s features having the largest ratio values are selected.
2. The signature size is not known, but a threshold value t is defined for the Golub's ratio. All features having a value larger than the threshold are selected.

Once the features are selected, a classifier may be built. This technique has the advantage of being conceptually simple, and often works well in practice. The drawback is that s is rarely known in advance, and that the ratio value has no intrinsic interpretation. A given value t cannot be translated in a measure of confidence. This issue is addressed by the next filter.

3.2.2 Hypothesis Testing

Some filters offer a stronger statistical interpretation than the Golub's ratio. It is the case of filters based on hypothesis testing. An example is the t -test based on the Welch statistic. This hypothesis test assesses the equality of the means of several normal populations that cannot be assumed to have the same variance. It is a univariate approach. The Welch statistics is defined by:

$$J(j) = \frac{|\mu_{j_1} - \mu_{j_2}|}{\sqrt{\sigma_{j_1}^2/n_1 + \sigma_{j_2}^2/n_2}}, \quad (3.2)$$

where μ_{j_i} is the mean value of feature j across samples of class i , σ_{j_i} is the standard deviation of feature j across samples of class i and n_i is the number of samples in class i . The spirit of such a metric is close to the Golub's ratio, but the output value $J(j)$ is used to compute a p-value for feature j . This p-value is the probability of wrongly reject the hypothesis according to which the true means of j for each class are identical. Consequently, unlike with the Golub's ratio, the smaller the p-value, the more the feature is considered as interesting. Here again, a ranking can be built based on the p-value of each feature.

Once such ranking is built, the selection can be performed according to one of the two following principles:

- The signature size s is known in advance, and the s features having the smallest p-values are selected.
- The signature size is not known, but a confidence threshold value t is defined for the p-values. All features having a value smaller than the threshold are selected.

The second option is clearly an advantage of statistical tests over other techniques like the Golub's ratio. Such a test is performed once for every considered feature. Multiple-test correction strategies should thus be considered to account for false positives. Many corrections have been proposed. The Bonferroni correction multiplies every p-value by the number of tests performed (the total number of features). This correction has been criticized mainly because it increases the number of false negatives (features wrongly made non-significant) by being too stringent (Perneger, 1998). Another, less stringent, correction is the Benjamini-Hochberg correction (Benjamini and Hochberg, 1995). Note that those corrections do not change the order of the features within the ranking. If the first strategy is applied (select a given number of s features), the correction is useless.

3.2.3 Information Theoretic Criteria

A whole branch of feature selection is based on information theory (Guyon and Elisseeff, 2003). It would be difficult to write a chapter on feature selection without mentioning information theoretic approaches. A standard filter approach is to rank the features according to the mutual information I between each feature $\mathbf{x}_j = \{x_{1j}, x_{2j}, \dots, x_{nj}\}$ and the target variable:

$$I(\mathbf{x}_j, \mathbf{y}) = \int_{\mathbf{x}_j} \int_{\mathbf{y}} P(\mathbf{x}_j, \mathbf{y}) \log \frac{P(\mathbf{x}_j, \mathbf{y})}{P(\mathbf{x}_j)P(\mathbf{y})} d\mathbf{x}d\mathbf{y}, \quad (3.3)$$

where $P(\mathbf{x}_j)$ and $P(\mathbf{y})$ are the probability densities of $\mathbf{x}_j$ and $\mathbf{y}$ respectively, and $P(\mathbf{x}_j, \mathbf{y})$ is the joint density. Equation (3.3) is a measure of dependence between the density of each feature and the target variable to predict. The larger $I(\mathbf{x}_j, \mathbf{y})$, the more feature j is likely to be predictive of $\mathbf{y}$. The lower $I(\mathbf{x}_j, \mathbf{y})$, the more j and $\mathbf{y}$ are independent: knowing feature j brings no information on $\mathbf{y}$. The densities $P(\mathbf{x}_j)$, $P(\mathbf{y})$ and $P(\mathbf{x}_j, \mathbf{y})$ are all unknown and are difficult to estimate from data. Density estimation is an ill-posed problem because there are infinitely many probability distributions that could have produced the observed finite dataset. The estimation can be done in practice via k-nearest neighbors or histogram-based techniques. These methods converge to the

true probability density as $n \rightarrow \infty$ (Bishop, 2006). This is clearly not suitable to our context where only few points are available. Torkkola (2003) presents a deeper discussion on the issues of mutual information computation on continuous and high-dimensional features.

Mutual Information is also used in a multivariate setting, in a pattern similar to multivariate correlation-based feature selection. This is the case of the MRMR approach (Minimum Redundancy, Maximum Relevance). This criterion favors signatures $\mathbf{S}$ of features having a high value of $\frac{1}{|\mathbf{S}|} \sum_{j \in \mathbf{S}} I(\mathbf{x}_j, \mathbf{y})$ (Maximum Relevance) and a low value of $\frac{1}{|\mathbf{S}|^2} \sum_{i,j \in \mathbf{S}} I(\mathbf{x}_i, \mathbf{x}_j)$ (Minimum Redundancy). How to combine those two criteria is a matter of debate and influence the quality of the selection (Gulgezen *et al.*, 2009).

The difficulty of density estimation can be overcome in the case of categorical features and targets. In this case the integrals become sums and the estimation of the densities can be achieved by frequency counts (Bekkerman *et al.*, 2003). The case of microarray data is harder since all features are continuous. A possibility to circumvent the density estimation is to discretize the features, implying a loss of information.

Finally, multivariate feature selection techniques based on the mutual information also exist, but require to estimate the joint distributions of several variables at a time, which is even harder than the univariate case.

For those reasons, information theoretic criteria are not considered in this thesis.

3.3 Wrappers

Wrappers are a family of iterative feature selection methods making use of a classification information, as illustrated in Figure 3.2, but otherwise using the classifier as a black box (Guyon and Elisseeff, 2003). At each step they use for example the classification performance to assess the quality of a feature subset. In practice, one has to specify how to move across the feature subsets space, the quality criterion to measure and to choose a given classifier. A review of wrapper techniques is proposed by Kohavi and John (1997).

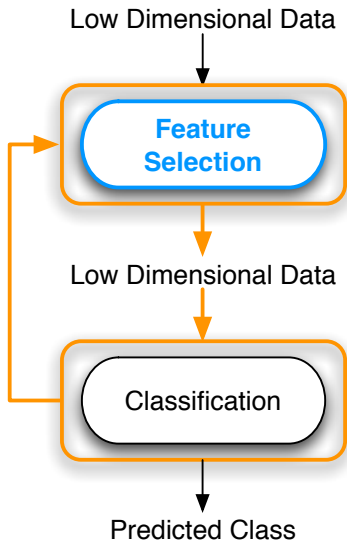


Figure 3.2 Feature selection: the wrapper pattern.

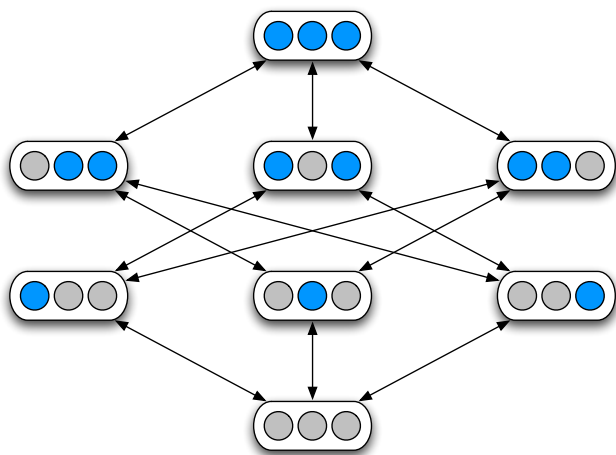


Figure 3.3 Forward and backward approaches.

Wrappers can be subdivided in two main categories: forward and backward wrappers. The forward wrappers start from an empty set of features, and run p times a given learning algorithm on each of the p features individually.

Performances are estimated for each option, and the single feature leading to the optimal performance is selected. At a second step, $p - 1$ classifiers are built on the previously selected feature and each of the $p - 1$ remaining features. Performances are once again estimated and the feature leading to the best performance is added to the first one, and so on. The backward approach works similarly, but starts from the full set of the p features and discards the single feature which elimination leads to the best performances. Those two approaches are illustrated in Figure 3.3 and both produce nested subsets of features. In that figure, forward selection starts at the bottom with an empty set, and backward selection starts from the top with the full feature set.

The main motivation for forward selection is computational: starting from only one feature, the first steps are fast to compute. The backward elimination approach is said to be less prone to miss a relevant feature since all features are considered at the initial step. Both approaches are in fact sub-optimal since only a small part of the feature subsets space is considered. Other heuristics than forward and backward approaches can be designed: forward-backward selection, simulated annealing, genetic algorithms etc., which do not necessarily produce nested subsets anymore. Readers interested in such local search approaches should refer to (Hoos and Stützle, 2004). Wrapper methods are finer than the filter approaches because first, they make use of an information from the classifier (its performance), and second, they consider groups of features jointly. By construction, those approaches are multivariate. On the other hand, they are generally much more computationally expensive.

3.4 Embedded Methods

Embedded methods make use of much more information from a classifier: they make use of its very structure, and the selection process occurs while training the classifier. Embedded methods are sometimes known as joint methods because they achieve two objectives at the same time. They are illustrated in Figure 3.4. They are yet finer than wrappers, and generally not more computationally intensive. That is why they are frequently preferred to them. The four next subsections describe four embedded methods for feature selection: two based on the SVM (the recursive feature elimination and the AROM methods), a method based on the logistic regression (the generalized

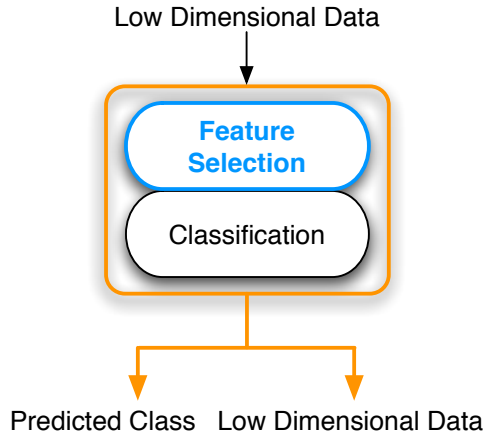


Figure 3.4 Feature selection: the embedded pattern.

lasso) and a last one based on random forests. The three first methods can be formulated in the form of the loss & regularizer pattern. This section ends with a discussion on a particular, maybe counterintuitive procedure to apply in the context of embedded methods.

3.4.1 Recursive Feature Elimination

SVM are known to scale well to high-dimensional spaces, and have shown state-of-the-art performance, as discussed in Chapter 2. Furthermore, a linear SVM offers the additional advantage that it contains an embedded capability for feature selection. As a linear SVM essentially consists of a separating hyperplane in the input space, the absolute values of the weights of each dimension in the hyperplane can be regarded as the contribution (importance) of each dimension (feature) to the multivariate decision of the hyperplane. As a result, these weights can be used to rank the features from most important to least important, which is the rationale for the recursive feature elimination algorithm (RFE, (Guyon *et al.*, 2002)).

RFE adopts a backward elimination strategy to iteratively remove features. Starting from the full feature set, a linear SVM is estimated from the training samples and features are sorted according to the absolute value of their weight in the hyperplane. Subsequently the least important features are eliminated and a linear SVM is re-estimated on the same samples but restricted to the remaining set of features. This process is iterated until a

desired number of features is reached.

RFE & Margin

At each iteration of RFE, a linear SVM is estimated. This means that the norm of the $\mathbf{w}$ vector of model parameters is proportional to the inverse of the maximum margin width, as expressed by Equation (2.22). Discarding the feature corresponding to the smallest component of $\mathbf{w}$ comes thus to discard the feature modifying the less the margin width.

An internal parameter of the RFE method is the fraction E of features to eliminate at each step, which greatly influences the computational complexity of the method. Decreasing E increases the computational cost since less features are dropped at each iteration but possibly offers a more refined selection.

RFE has been shown many times to be a state-of-the-art selection technique, as reported in (Guyon, 2002). Finally, note that RFE can be seen either as an iterative method for selection similar to a wrapper, or as a feature selection technique making use of the classifier structure, hence, an embedded method. A non linear version of RFE has been proposed (Guyon *et al.*, 2006). It allows one to use any kind of kernel instead of just the linear SVM. It is however computationally much more demanding. The non-linear RFE is much more a wrapper than an embedded method, because it requires to test the elimination of each feature at a time, and then to measure which drop leads to the smallest difference in terms of the norm of the margin. The next section on L_1 regularization presents pure embedded methods such as the lasso and the generalized lasso.

3.4.2 L_1 Regularization and Lasso Methods

L_1 Regularization

The L_1 norm is a sparsity-enforcing regularizer, which is not the case of the L_2 norm. To obtain sparse predictive models, the L_1 norm is frequently combined with various loss functions. The study of the L_1 regularization has now become mature and two main theoretical results should be outlined:

1. **Support recovery condition:** The recovery concept is more extensively detailed in the next chapter. Briefly, it corresponds to the ability of a selection method to select the features truly used by the generative process and only those. It has been proved by Zhao and Yu (2006) that a

necessary and sufficient condition of recovery with a model including an L_1 regularization is that the features should have a small correlation between each other. It has been shown experimentally by Bach (2009) that in the case of strong correlation, the L_2 regularization achieves better classification performances than L_1 regularization.

2. **Exponential number of features:** The second important result about L_1 regularization is that the number p of dimensions it is able to deal with can grow exponentially with the number n of samples available (Ng, 2004). This property is written as: $\log(p) = \mathcal{O}(n)$. To be more precise, it specifies in fact that the number of *irrelevant* features only can grow exponentially with n .

A link can be made between the L_1 regularization and the Bayesian framework (Seeger, 2008). This norm corresponds to the log density of the Laplace distribution. It can thus be seen as a Laplacian prior on the features:

$$P(\mathbf{w}) \propto \prod_{j=1}^p e^{-\lambda|w_j|}. \quad (3.4)$$

The L_1 regularization is most often performed on the model weights parameters, in the original domain of the objective function. Other forms of sparse regularization have also been implemented via the L_1 norm, namely to impose sparsity on the gradient of a decision function. In this case, the regularizer that is used is referred to as the total variational norm, or TV-norm (Rudin *et al.*, 1992)².

(Generalized) Lasso

The lasso model combines an L_1 regularizer with the square loss $\mathcal{L}_s$ of Equation (2.11) for regression (Tibshirani, 1996):

$$\operatorname{argmin}_{\mathbf{w}} \sum_{i=1}^n (y - \langle \mathbf{x}_i, \mathbf{w} \rangle)^2 + \lambda \|\mathbf{w}\|_1. \quad (3.5)$$

The generalized lasso is a specific case of penalized logistic regression using a logistic loss for classification and the L_1 norm as regularizer (Roth, 2002). The

²The TV in the name of the TV-norm does not only refer to the *total variation* but also to the motivations for such a method, mainly in image processing where it has been developed. It has been noticed that cartoons generally display wide zones of constant intensity or color, and only smooth shapes. Cartoon images have thus a very low TV-norm, hence the link with television.

objective function of the generalized lasso is:

$$\operatorname{argmin}_{\mathbf{w}} \sum_{i=1}^n \log \left(1 + \exp^{-y_i \langle \mathbf{x}_i, \mathbf{w} \rangle} \right) + \lambda \|\mathbf{w}\|_1. \quad (3.6)$$

Many solvers have been designed to optimize this objective function. In the case of $n \ll p$, an efficient solver has been proposed by [Shevade and Keerthi \(2003\)](#). This solver is discussed in Chapter 10 where it serves as a basis for another solver for multi-task learning.

Group Lasso

An extension of the lasso model further structures the feature space by explicitly making use of groups of features. In the context of genomic applications, it is known that groups of genes work together to form pathways. When groups can be user-defined group lasso ([Yuan and Lin, 2006](#)) makes use of this information and optimizes the following objective:

$$\operatorname{argmin}_{\mathbf{w}} \sum_{i=1}^n (y - \langle \mathbf{x}_i, \mathbf{w} \rangle)^2 + \lambda \|\mathbf{w}\|_{1,2}, \quad (3.7)$$

where $\|\mathbf{w}\|_{1,2}$ is the regularization of the k groups of features $\mathcal{G}_1, \dots, \mathcal{G}_k$, each of them containing a given number of features. Its detailed expression is:

$$\|\mathbf{w}\|_{1,2} = \sum_{i \in \{\mathcal{G}_1, \dots, \mathcal{G}_k\}} \sqrt{\sum_{j \in \mathcal{G}_i} w_j^2}. \quad (3.8)$$

With this method, all features within a group are either selected or dropped together. The number of selected groups is small since there is a L_1 regularization on it. However, the weights corresponding to each feature within a group can take different values, which are L_2 -regularized. DRAGS is another technique adopting the same grouping approach although not based on L_1 regularization ([Yu et al., 2008](#)). In this feature selection method, groups are discovered from the data via kernel density estimation (also known as Parzen window). Once the groups are defined, the relevance of each group is defined based on the average relevance of every feature it contains. Such an individual relevance is assessed through a filter similar to a t -test. Only the most relevant groups are selected. One feature in each group (the most relevant one) is finally used to perform classification with any classification algorithm.

Convexity

The L_1 norm itself is convex, but not strictly convex. Its combination with a strictly convex loss function leads to a globally strictly convex objective function. This is the case of the square loss, and thus the lasso and group-lasso techniques are strictly convex. This means that they have a unique global optimum. The generalized lasso is the combination of the L_1 norm and a logistic loss function. As mentioned in Chapter 2, the logistic loss is in theory strictly convex but numerically tends to behave as a simply convex function. The generalized lasso has thus a unique optimal value for its objective function, but there is no guarantee in practice that this optimum is unique: it could be achieved by several combinations of the model parameters.

Compressed Sensing

Sparse learning is closely related to compressed sensing in the signal processing community. Similar theoretical results about the relation between the logarithmic number of dimensions and the number of samples in the case where a signal can be assumed sparse have also been independently produced (Candès and Tao, 2005). An interesting discussion on the fact that this result is a generalization of the Shannon Sampling theorem (Shannon, 1949) to other bases than the frequency domain (Fourier domain) is considered by Candès (2006). This discussion is out of the scope of this thesis. Compressed sensing often makes use of the L_1 norm in convex optimization objectives to ensure sparsity. However, the concerns of this community are complementary to those of the machine learning community. In compressed sensing, one of the main aim is to implement sparsity and the related theoretical results directly within the sensors of a given technology. Such a way of thinking gives rise to devices making sparse measurements, i.e. avoiding to ever deal with high-dimensional data. As a consequence, a lower number of samples are required in order to correctly reconstruct a signal.

3.4.3 AROM methods

L_0 Regularization

Feature selection for linear classification models is closely related to a specific form of regularization of the decision function to enforce sparsity of the weight vector $\mathbf{w}$. Weston *et al.* (2003) study in particular the zero-norm mini-

mization subject to linear margin constraints³:

$$\min_{\mathbf{w}} \|\mathbf{w}\|_0, \quad (3.9)$$

subject to

$$y_i \langle \mathbf{x}_i, \mathbf{w} \rangle \geq 1 \forall i \in \{1, \dots, n\}. \quad (3.10)$$

L_1 -AROM

Since Problem (3.9) is NP-Hard, a log 1-norm minimization is proposed instead.

$$\min_{\mathbf{w}} \sum_{j=1}^n \log(|w_j| + \epsilon), \quad (3.11)$$

subject to

$$y_i \langle \mathbf{x}_i, \mathbf{w} \rangle \geq 1 \forall i \in \{1, \dots, n\}, \quad (3.12)$$

where $0 < \epsilon \ll 1$ is added to smooth the objective when some $|w_j|$ vanishes. The logarithm in the objective is a natural logarithm and facilitates parameter estimation with a simple gradient descent procedure (an extended version of this procedure can be found in (Weston *et al.*, 2003)). The resulting algorithm L_1 -AROM simply optimizes the 1-norm of $\mathbf{w}$ with iterative rescaling of the inputs, as illustrated in Algorithm 1. A smooth feature selection occurs during this iterative process since the weight coefficients along some dimensions progressively drop below the machine precision while other dimensions become more significant. A final ranking on the absolute values of each dimension can be used to obtain a fixed number of features. AROM stands for **A**pproximation of **zeRO**-norm **M**inimization. Figure 3.5 illustrates why Problem (3.11) is a good approximation to the zero-norm minimization. This objective is nearly flat on the whole space of parameters $\mathbf{w}$ except when a specific w_j tends towards zero. The objective is there strongly minimized.

L_1 -AROM vs. Adaptive Lasso

The L_1 -AROM technique is highly similar to two other techniques: the reweighted- L_1 (Candès *et al.*, 2007) and the adaptive lasso (Huang *et al.*, 2008). Both methods use an iterative reweighting scheme and a L_1 regularizer. The adaptive lasso and the reweighted- L_1 are highly similar since they both use a square loss. They were independently proposed by two different research

³The constraints of Problem (3.9) could be rewritten $y_i \langle \mathbf{x}_i, \mathbf{w} \rangle > 0$ since the 0-norm is insensitive to the scale of $\mathbf{w}$. The use of margin constraints is motivated by the subsequent approximations to this problem.

Algorithm 1 The L_1 -AROM algorithm. The component-wise product is denoted by $*$.

Init:

$$\mathbf{w}_0 = (1, \dots, 1), \mathbf{w}_0 \in \mathbb{R}^p$$

$$k = 0.$$

Output: $\bar{\mathbf{w}}$

while Convergence not reached **do**

Solve:

$$\bar{\mathbf{w}} = \operatorname{argmin}_{\mathbf{w}} \|\mathbf{w}\|_1$$

$$\text{subject to: } y_i \langle \mathbf{x}_i * \mathbf{w}_k, \mathbf{w} \rangle \geq 1$$

Update:

$$\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k * \bar{\mathbf{w}}$$

$$k \leftarrow k + 1$$

end while

communities. The former is from the machine learning community while the latter is from the signal processing community (compressed sensing). The main difference between those two techniques and the L_1 -AROM approach is that L_1 -AROM uses a hinge loss.

L_2 -AROM

The L_2 -AROM method further approximates the L_1 -AROM Algorithm 1 by replacing the 1-norm by the 2-norm. Even though such an approximation may result in a less sparse solution, it is very efficient in practice when $n \ll p$. Indeed, a dual formulation may be used and the final algorithm boils down to a linear SVM estimation with iterative rescaling of the inputs. A standard SVM solver can be iteratively called on properly rescaled inputs. The procedure of L_2 -AROM is illustrated in Algorithm 2. The convergence of the algorithm is generally fixed to a given number of null components of $\mathbf{w}$. Classification performances of the original L_1 -AROM and L_2 -AROM methods do not significantly differ while the computational time is in favor of the latter (Weston *et al.*, 2003). This similarity of classification performances has been shown on microarray data. It might be explained by the generally high correlation observed between expressions of different genes. As explained in the previous section, in such cases the L_1 regularization is not likely to outperform the L_2 approach.

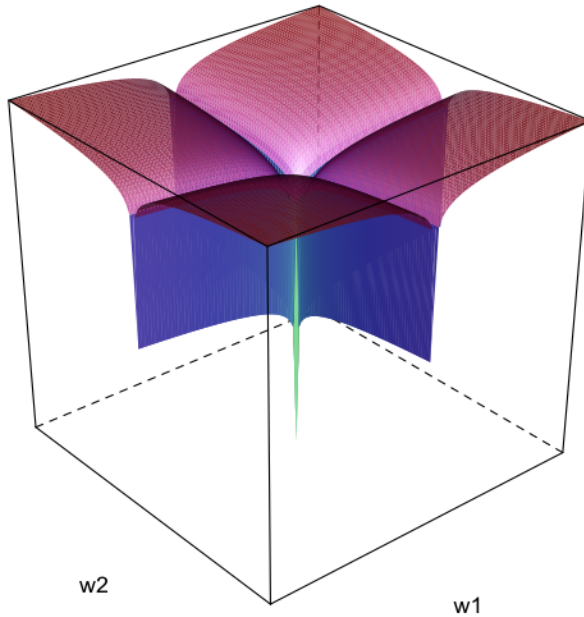


Figure 3.5 2D-representation of the Zero-Norm Approximation by L_1 -AROM.

L_2 -AROM vs. RFE

The L_2 -AROM selection technique is similar to the RFE technique. Replacing the update of $\mathbf{w}_{k+1}$ of Algorithm 2 by a product with 1 for the largest components of $\tilde{\mathbf{w}}$, and a product with 0 for the smallest ones leads to the RFE procedure. L_2 -AROM can be seen as a smoother, finer, version of RFE.

L_2 -AROM vs. RM-SVM

The L_2 -AROM algorithm is based on a standard linear SVM, which optimizes the margin of a classifying hyperplane. Some generalization bounds for the SVM do not only imply the concept of margin, but also the concept of the radius of the sphere containing the training points. This is the case of the radius-margin bound (Chapelle *et al.*, 2002; Cristianini and Shawe-Taylor, 2000). When the feature space is fixed, the radius cannot be optimized and an optimization limited to the margin makes full sense. When feature selection is considered, the feature space is modified, and the radius is likely to be affected. Including the radius into the optimization process is then sound. Do

Algorithm 2 The L_2 -AROM algorithm. The component-wise product is denoted by $*$.

Init:

$$\mathbf{w}_0 = (1, \dots, 1), \mathbf{w}_0 \in \mathbb{R}^p$$

$$k = 0.$$

Output: $\bar{\mathbf{w}}$

while Convergence not reached **do**

Solve:

$$\bar{\mathbf{w}} = \operatorname{argmin}_{\mathbf{w}} \|\mathbf{w}\|_2^2$$

$$\text{subject to: } y_i \langle \mathbf{x}_i * \mathbf{w}_k, \mathbf{w} \rangle \geq 1$$

Update:

$$\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k * \bar{\mathbf{w}}$$

$$k \leftarrow k + 1$$

end while

et al. (2009) propose such an approach and additionally impose a sparsity constraint on the components of the radius, embedding feature selection within the estimation of a maximal margin hyperplane. This problem named MR-SVM (Margin-Radius-SVM) is solved by an iterative optimization procedure similar to L_2 -AROM, but where the rescaling factors are related to the optimization of the radius. Unlike L_2 -AROM, MR-SVM does not offer any control on the sparsity of the solution, which is *discovered* from the data by the optimization problem. In some cases, it ends up in totally non-sparse solution.

AROM Convexity

L_1 -AROM uses a hinge loss, which is convex but not strictly convex. Its combination with the L_1 norm thus makes it a globally convex but not strictly convex problem. L_2 -AROM uses a L_2 regularization and is thus strictly convex.

3.4.4 The Random Forests as a Selection Technique

Random forests described in Chapter 2 have been extended to provide a ranking over the features (Diaz-Uriarte and de Andres, 2006). This ranking is based on an embedded measure of the importance of each feature for the classification accuracy. This measure of importance is computed as follows.

Each tree t of the forest is built on a bootstrap sample $\mathbf{X}^*$ of the training

set, on a restricted set of features preselected randomly. Data not in the bootstrap sample are referred to as the out-of-bag sample (OOB). The tree built on the bootstrap sample is used to predict the labels of the data from the OOB sample, and the accuracy A_t of this prediction is stored. The values of a given feature j are permuted among the points of the OOB sample, a new classification is made with the same tree on this permuted sample, and the accuracy is once again computed and noted A_t^j . The difference in accuracies before and after permutation is a measure of the importance of the feature. This measure is averaged over each tree containing feature j . This number is noted $nTree^j$.

$$\text{Importance}(j) = \frac{1}{nTree^j} \sum_{t=1}^{nTree^j} A_t - A_t^j. \quad (3.13)$$

Finally, $\text{Importance}(j)$ is computed for each feature of the original dataset. A global ranking on the features is thus obtained, according to this importance measure. [Diaz-Uriarte and de Andres \(2006\)](#) recommend not to recompute this measure in an iterative fashion (as is done for RFE) because it could lead to instability. It is instead recommended to compute the ranking once and then to adjust the number of features retained for classification purposes.

3.4.5 Two-Stages Procedure

Embedded approaches are said to solve at the same time feature selection and classification. Empirically, it has been shown that embedded methods performances with respect to classification can be improved if an additional learning step is applied. In some cases, this extra step consists of model weights rescaling ([Zou and Hastie, 2005](#)). In other cases, it consists of a *two-stages procedure* ([Guyon et al., 2002](#); [De Mol et al., 2009](#)). The first stage reduces the dimensionality via feature selection technique embedded within a classification model. In the second stage, a standard non-sparse classification (or regression) technique is applied to the resulting set of features (for example, a standard SVM, a ridge regression, etc.). This latter option is adopted in most experiments of this thesis. In fact, the procedure is similar to a situation where filters or wrappers are used. The selection step is followed by a predictive model learning step. The extra computational load due to the second step is minor, since it is done on a small number of selected features.

3.4.6 Relevance Vector Machine: Another Form of Sparsity

The Relevance Vector Machine (Tipping, 2001), or RVM, is often presented as a Bayesian competitor of the Support Vector Machine. It is built over the Automatic Relevance Determination (ARD) probabilistic framework (MacKay, 1994). It has the same functional form as the SVM (hyperplane with potential use of kernels). Instead of using support vectors, RVM defines *relevance vectors*. Like support vectors for SVM, relevance vectors serve as basis for the definition of the RVM decision function.

RVM are often promoted arguing that they are more *sparse* than the SVM. However, we stress here that the higher sparsity achieved by the RVM concerns the number of points involved in the definition of the decision function, and not the number of features finally involved in the decision function. That means that on the same data, an RVM will generally have fewer relevance vectors than a SVM will have support vectors. However, RVM do not imply any feature selection at all.

With regard to classification, RVM has a training complexity of $\mathcal{O}(n^3)$ while SVM has a complexity of $\mathcal{O}(n^2)$. RVM generally do not outperform SVM with respect to classification quality. To be complete, an advantage of RVM over SVM is to offer a probabilistic output, but then we already introduced the L_2 regularized logistic regression offering the same opportunity in Chapter 2. For these reasons, we do not consider RVM in this thesis.

Evaluation Criteria

4

This chapter defines some metrics that are relevant to assess the performances of the classification and feature selection techniques covered by this thesis in Chapters 2 and 3 as well as those of contributing chapters of Part III. Section 4.1 describes several feature selection quality criteria. Among those is stability. Several ways of measuring it are discussed. Section 4.2 introduces several metrics for assessing classification performances. This chapter ends with a discussion on the need for several criteria considered together to evaluate the quality of feature selection for classification tasks.

4.1 Metrics for Selection Quality

4.1.1 Several Quality Criteria

Several quality criteria can be found in literature for evaluating feature selection techniques. Among others are recovery, stability, and interpretation.

Recovery

The recovery, also known as consistency ([Bach, 2008](#)), refers to the ability of a feature selection technique to select the right features among a large feature set when data have indeed been generated by a sparse pattern of those features. It can be measured for example in terms of precision and recall. Although sound, it presents two drawbacks. First, assessing the recovery ability of a selection technique requires that the truly relevant features are known. This is rarely the case when facing real data. Second, it assumes that the generative process of the data is indeed sparse. It can be the case that this generative process is not sparse, and still a user may want to perform classification on

few dimensions. In that case, a feature selection technique should drop some of the relevant features for the sake of sparsity. In the case of genomic data, features are generally highly correlated. Genes are known to work together in networks compounded of several pathways and are rarely expressed on their own without affecting other genes expression. It is thus probable that the generative feature patterns are not so sparse and are anyway rarely known. That is why recovery is not much considered in this thesis.

Stability

The stability of a feature selection method is a measure of similarity between sets of selected features. It allows one to study the robustness of a selection method with respect to variations such as sampling variations, or meta-parameters variation. It also makes possible to study how similar several selection methods are. If two selection methods tend to produce similar sets in many conditions, their behaviors can be thought as being close. A high stability means that the considered sets are highly similar, and vice-versa. Stability is often required in medical applications based on genomic data for several reasons, among which the reproducibility and the easiness of the biological validation of the extracted signature. The underlying hypothesis is that small changes in the data sampling should not yield dramatic changes in the set of selected markers. In other words, the genes responsible for a given biological condition are expected to be close if not identical regardless of the data sampling since the metabolism of all patients (belonging to a given class) is supposed to be regulated the same way (at least qualitatively). In the case where the size of the considered signatures is smaller than the number of genes involved in the biological process studied, it can happen that several (non-overlapping) signature have an equivalent predictive power and are biologically sound. Obtaining a high stability could be harder in such a situation. Still, a high stability might be wanted for example in order to design a unique an small diagnosis kit.

Interpretation

In many applicative contexts, signatures extracted by feature selection are interpretable. In the case of text classification, it is often easy to manually check if a selected bag of words makes sense to discriminate between two text categories. In the case of genomic data, the genes that are selected can be also interpreted. This implies two things: the selected genes have to be already extensively studied and annotated, and the interpreter should have

a deep background in molecular biology or in medicine. This makes the interpretation of a gene signature less easy for a computer scientist, but is maybe the ultimate validation: *does a signature make sense?* Selection methods leading to meaningful signatures should be preferred to those that do not.

Recovery and interpretation assume prior knowledge (stronger in the case of recovery), and can be applied to a single signature. Stability can only be applied to a set of signatures, and is thus more relevant in the case of method validation than in the case of single signature validation. Recovery can also easily be applied to the evaluation of a method rather than just a single signature. Interpretation is more difficult to apply to a method, because it generally takes time to interpret a single signature. Interpreting many is thus difficult. A high recovery ability would lead to a high stability, except if the enforced sparsity is stronger than the underlying one. The reverse is however not guaranteed to be true: a high stability does not necessarily implies a high recovery rate. In this thesis, we choose to use stability as our main evaluation criteria for feature selection since we are more interested in methods evaluation. When it comes to evaluate a single signature for practical applications, interpretation should be the ultimate validation, as illustrated in the Cristall case study of Chapter 12.

4.1.2 Stability Indices

In this thesis, two main stability indices are used. The Kuncheva index (Kuncheva, 2007) and the Jaccard index (Kalousis *et al.*, 2007).

Kuncheva Index

The Kuncheva index $\mathcal{K}$ measures to which extent two sets $\mathbf{S}_i$ and $\mathbf{S}_j$ of s selected features share common features:

$$\mathcal{K}(\mathbf{S}_i, \mathbf{S}_j) = \frac{|\mathbf{S}_i \cap \mathbf{S}_j| - \frac{s^2}{p}}{s - \frac{s^2}{p}}, \quad (4.1)$$

where p is the total number of features, and $\mathbf{S}_i, \mathbf{S}_j$ are two signatures. The $\frac{s^2}{p}$ ratio in this formula corrects the bias of selecting common features in both sets by chance. This correction motivates our use of this particular stability index when possible. This index satisfies $-1 < \mathcal{K} \leq 1$ and the greater its value the larger the number of commonly selected features in the two sets.

It is undefined for both extreme cases $s = 0$ and $s = p$. This index can be generalized to k signatures:

$$\mathcal{K}(\{\mathbf{S}_1, \dots, \mathbf{S}_k\}) = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^k \mathcal{K}(\mathbf{S}_i, \mathbf{S}_j). \quad (4.2)$$

Jaccard Index

For two sets of selected features $\mathbf{S}_i$ and $\mathbf{S}_j$, the Jaccard stability index $\mathcal{J}$ is defined as:

$$\mathcal{J}(\mathbf{S}_i, \mathbf{S}_j) = \frac{|\mathbf{S}_i \cap \mathbf{S}_j|}{|\mathbf{S}_i \cup \mathbf{S}_j|}. \quad (4.3)$$

The Jaccard index does not take into account the probabilistic effects of selecting a large proportion of the available features, unlike the Kuncheva index. This index satisfies $0 \leq \mathcal{J} \leq 1$ and the greater the index value the larger the proportion of commonly selected features in the two sets. The Jaccard index is well defined even when $\mathbf{S}_i$ and $\mathbf{S}_j$ do not have the same cardinality. This property motivates the choice for this particular index when there is no exact control on the number of features selected by the considered methods. It can be generalized to k signatures:

$$\mathcal{J}(\{\mathbf{S}_1, \dots, \mathbf{S}_k\}) = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^k \mathcal{J}(\mathbf{S}_i, \mathbf{S}_j). \quad (4.4)$$

4.2 Classification Performances

The results of a binary classification experiment can be summarized in a confusion matrix, such as the one in Table 4.1. This representation can easily be extended to more than two classes by a row and a column for each additional class.

4.2.1 Accuracy

Probably the most used performance criterion is the correct classification rate, known as accuracy. It can be written in terms of entries of the confusion matrix 4.1:

$$\text{Acc} = \frac{n_{(-,-)} + n_{(+,+)} }{n_{(-,-)} + n_{(-,+)} + n_{(+,-)} + n_{(+,+)}} = \frac{n_{(-,-)} + n_{(+,+)}}{n}. \quad (4.5)$$

Table 4.1 A confusion matrix for a binary classification problem. $n_{(-,+)}$ is the number of samples classified as having $y_i = -1$ although they truly correspond to $y_i = +1$. Other $n_{(.,.)}$ are defined similarly. The sum of all entries is n .

		True	
		-1	+1
Predicted	-1	$n_{(-,-)}$	$n_{(-,+)}$
	+1	$n_{(+,-)}$	$n_{(+,+)}$

Its interpretation is the proportion of correctly classified samples. The use of this metric is ubiquitous. It is the criterion that is optimized by standard SVM (Boser *et al.*, 1992) and by the generalized lasso (Roth, 2002)¹. It also serves as basis for an internal measure of feature importance in Random Forests (Diaz-Uriarte and de Andres, 2006). When class priors are strongly dissimilar, accuracy may lead to treacherous results. Suppose that one class contains 90% of the samples, and that the other class contains the remaining 10%. A naive model always classifying every given sample in the majority class will produce an accuracy of 90% although it performs really poorly on one of the two classes.

4.2.2 Sensitivity and Specificity

In terms of the entries of confusion matrix 4.1, sensitivity is defined as:

$$\text{Sensitivity} = \frac{n_{(+,+)}}{n_{(-,+)} + n_{(+,+)}} , \quad (4.6)$$

and specificity is defined as:

$$\text{Specificity} = \frac{n_{(-,-)}}{n_{(+,-)} + n_{(-,-)}} . \quad (4.7)$$

Sensitivity can be interpreted as the accuracy of the positive class (or *True Positive Rate*, TPR), while specificity can be interpreted as the accuracy of the negative class (or *True Negative Rate*, TNR). Sensitivity is also known as the *recall*.

¹In practice, an approximation of the accuracy is generally used in objective functions because accuracy is a step function, thus not convenient for optimization.

4.2.3 Positive and Negative Predicted Value

Positive Predicted Value (PPV) and Negative Predicted Value (NPV) are orthogonal to sensitivity and specificity. In terms of the entries of confusion matrix 4.1, PPV is defined as:

$$\text{PPV} = \frac{n_{(+,+)} }{n_{(+,-)} + n_{(+,+)} } , \quad (4.8)$$

and NPV is defined as:

$$\text{NPV} = \frac{n_{(-,-)} }{n_{(-,-)} + n_{(-,+)} } . \quad (4.9)$$

The PPV is equivalent to the *precision*, and in medical context, to the *clinical efficacy*.

4.2.4 Balanced Classification Rate

The balanced classification rate (BCR) is another measure of classification performance intended to offer an alternative to the misleading behavior of accuracy when classes are unbalanced. The expression of the BCR is:

$$\text{BCR} = \frac{1}{2} \left(\frac{n_{(+,+)} }{n_{(-,+)} + n_{(+,+)} } + \frac{n_{(-,-)} }{n_{(-,-)} + n_{(+,-)} } \right) . \quad (4.10)$$

BCR can be interpreted as the average *by-class* accuracy. The balanced error rate (BER) is defined as $1 - \text{BCR}$. Balanced classification rate is preferred to accuracy when classes have unequal priors. This is the case of most microarray datasets. Other metrics are also able to deal with unbalanced classes, like the *F*-measure (van Rijsbergen, 1979), which is the harmonic mean between precision and recall. Another point in favor of BCR is that it is the average between specificity and sensitivity, and those are two very common measures in the medical domain. BCR can also be generalized to multi-class problems more easily than *F*-measure or ROC analysis. For k classes, BCR can be redefined as:

$$\text{BCR} = \frac{1}{k} \left(\sum_{i=1}^k \frac{n_{(i,i)} }{\sum_{j=1}^k n_{(j,i)} } \right) . \quad (4.11)$$

4.2.5 Area Under the Receiver Operator Characteristic Curve

The receiver operator characteristic curve, or ROC curve (Provost and Fawcett, 1997), is a function of sensitivity and specificity. It is often possible to make some parameter of a model vary. For example, think of the bias

term in a linear classification model. For each value of the bias, a measure of the sensitivity and the specificity is performed, and a point corresponding to (1-specificity) vs. sensitivity can be plotted on a unit square. The set of points generated forms a curve. The area under the curve, also known as AUC, is a measure of quality of the model, independently of the considered parameter value. AUC has a maximal value of 1. An AUC value of 0.5 is the performance achieved by a random classifier. ROC are plotted for a good, a poorer, and a random classifier in Figure 4.1. AUC analysis cannot be easily extended for classification problems of more than two classes, unless pairwise class comparisons are performed, but conclusions are harder to draw in that case.

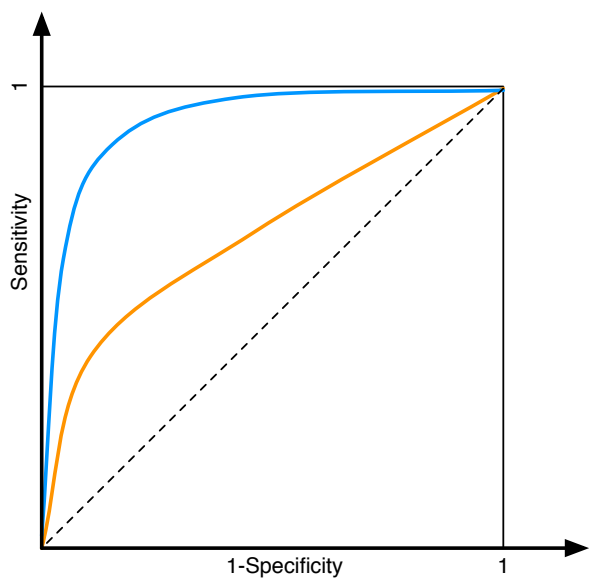


Figure 4.1 ROC Curves. The classifier corresponding to the blue curve should be preferred to the one corresponding to the orange curve since it has a greater area under its curve. The dashed line corresponds to a random classifier (area under the curve: 0.5).

4.3 Combining Several Criteria

For feature selection method evaluation, stability seems the most appropriate metric. Another metric not discussed so far, is of course the sparsity of the obtained models, which can be critical in some applications. Neither stability,

nor sparsity alone can characterize sufficiently the quality of a feature selection method in a context of classification. Indeed, if an arbitrary set of features were purely forced in every signature, the stability would be high, but the model built on those features would likely have a poor classification performance. The same applies to sparsity. A method returning always one feature is highly sparse, but again, likely to produce poor classification performance. That is why the quality of feature selection in a context of classification should take into account both stability and classification performance at the same time (and more or less critically sparsity, depending on the application). BCR should be preferred to accuracy in case the classes are not balanced for the classification performance part.

It is possible to combine selection and classification metrics in a single metric, as done by [Saeys et al. \(2008\)](#). However, the advantage of keeping them apart is a gain of interpretability of the results. In this thesis, the different metrics are kept separated but always considered together to perform a fine analysis of the methods behavior.

Validation Protocols

5

The previous chapters introduced classification and feature selection technique, as well as metrics to evaluate their quality. The experiments in the next chapters comprise many validation aspects. This chapter details *how* to measure these quality criteria. Validation of the performance of a model trained on n samples i.i.d. from $P(\mathcal{X}, \mathcal{Y})$ on other, unseen, samples assumed to be drawn from the same distribution P is an old issue that has not totally been solved yet, be it for feature selection quality or classification performance.

This chapter starts with a discussion on validation in small- n -big- p settings in Section 5.1. This discussion deals with the selection bias and shows why it is difficult to rigorously evaluate *one* model performances in practical applications. The remaining of the chapter discusses three related techniques of validation: cross-validation (Section 5.2), the bootstrap (Section 5.3) and resampling strategies (Section 5.4). Section 5.3 proposes the first contribution of this thesis: a generalization of the bootstrap .632+ to other metrics than the accuracy. Permutation experiments as validation are also briefly introduced in Section 5.5. The chapter ends with Section 5.6 discussing the choice of an adequate protocol for this thesis experiments among the proposed ones.

5.1 Validation in High-Dimensional Settings: the Dog Chasing its Tail

The discussion in this section is illustrated by the context of microarray technology, but is applicable to most high dimensional data where few samples are available. Suppose one wants to design a complete diagnosis system in-

cluding a classification model and a signature on gene expression microarray data. At the end, he has to end up with a single gene signature, a unique set of normalization parameters, and a single classification model (the model parameters numerical values, not just a learning algorithm). How to evaluate the performance of that system? The approach to avoid is to estimate a model and estimate its performances on the same data. This setting is prone to all kinds of biases (normalization and selection biases, overfitting, ...). Two possibilities can be considered instead:

- **The train-test setting, or how to lower performances and poorly validate.** The first sound possibility is to split the data into training and test sets. Without any bias, summarization, normalization, feature selection and classifier training are done on the training set. The final model is then applied to the test samples to assess its quality. This setting would have been ideal if plenty of samples were available and no empty spaces phenomenon had to be faced. With the few samples available, the model built is not only likely to be significantly weaker than if it had been built on the whole dataset, but its performances are poorly evaluated on fewer samples, leading to a non-robust estimation. The result is a unique model, likely weak and with unknown performances.
- **The validation loop setting, or how to not have a model.** A second option is to incorporate the whole process in a validation loop (be it a bootstrap, a resampling or a cross-validation loop). Differences between the different kinds of validation loops are discussed in the next sections, but do not change anything for the current discussion. At each validation loop, the dataset is split in two, a bias-free model is built on one large part of the available samples and evaluated on the remaining, much smaller part of the dataset. Performances will be much better evaluated, averaged on potentially several hundreds of runs, on different partitions of the dataset. The problem is then that there is not *one* model anymore, but as many models as validation loops, as mentioned in (Kalousis *et al.*, 2004). This does not only hold for the classifier, but also for the selected biomarkers. There are potentially as many signatures as validation loops. The result is a plenty of models, with an average and reliable performance computed.

This issue is not strictly solvable. The original methods proposed in this thesis aim at improving classification performances and stability of the selected signatures in small- n -big- p settings so that the simple train-test setting can be

applied. The dilemma between both perspectives should be approached from several viewpoints, depending on the context.

From a machine learning research perspective, the second option including validation loops is a perfect setting to globally compare several methods. It is however not so helpful for the biologist or the medical doctor awaiting for a signature and a predictive model. From a computational biology practitioner perspective, having to deliver a unique diagnosis or prognosis model, the best solution to recommend is probably to normalize, select genes and build a classification model with the globally best performing methods (known from previous experience) on the whole dataset available. An evaluation of this model on the same samples will be prone to all kinds of biases among which normalization and selection biases. Normalization bias, and overoptimistic classification performances due to the classifier itself being evaluated on the same samples can be corrected by applying a validation loop, using however the signature identified on the whole dataset. This will suppress the normalization bias, and reduce the optimism in classification performances evaluation. However, the selection bias will still be present.

Identified biomarkers should anyway be validated by other means, like domain expertise. Another way of getting more confidence in obtained models, even in presence of selection bias, is by random permutation experiments, as discussed at the end of this chapter.

5.2 Cross-Validations

K-Fold Cross-Validation

In the k-fold cross-validation (or KCV) setting ([Stone, 1974](#)), a sample is split in K non-overlapping subsets of points. K models are built. Each model $k \in \{1, \dots, K\}$ is built on all subsets except the k^{th} one. Performances of model k are evaluated on the k^{th} subset, unused for that model estimation. The K independently measured performances are then averaged. This resulting average is supposed to be equal to the performance of a model built on the available training data and tested on an infinite test set, potentially the whole distribution. The variability of the estimated performance can be computed via the standard deviation of the K performances, since the K test sets are independent (given that they do not overlap and points are assumed to be i.i.d.).

Leave-One-Out

A KCV procedure with $K = n$, the number of available points, is called a leave-one-out cross-validation (LOO). The KCV procedure generally provides a low bias but shows a large variance, particularly if used in the LOO setting (Efron and Tibshirani, 1997). Such a large variance is an even more likely outcome if the number of points available is really low, as it is generally the case with genomic data for example. Here, *bias* refers to the difference between the average of KCV estimations on several sets drawn from the same distribution and the true value one wants to estimate, not to the difference between a single KCV estimation and that true value. Similarly, the *variance* refers here to the variance of those KCV estimations on several sets drawn from the same distribution, not to the variance associated with a single KCV estimation.

5.3 Bootstraps

Another resampling technique has been proposed to overcome this high variance issue: the bootstrap .632 Efron (1983). It is also a resampling procedure, drawing a series of what is called *bootstrap samples*. Each bootstrap sample corresponds to the drawing of n observations with repetitions. Any point in the sample has in this case roughly 36.8% chance not to be selected (i.e. 63.2% chance to be selected), hence the “.632” in the name of the technique. To each bootstrap sample corresponds a set of points not drawn. This set forms what is called an out-of-bag sample (OOB). A model is trained on the bootstrap sample and evaluated on the OOB sample. This operation is repeated B times, and the average out-of-bag error $\hat{Err}^{(1)}$ is obtained. Another model is trained on the full set of available points, and its performance evaluated on the very same points. This produces an error rate $e\bar{r}r$ which is generally optimistic. The final performance of the model is evaluated via a combination of the optimistic $e\bar{r}r$ and its average out-of-bag error ($\hat{Err}^{(1)}$), which has been shown to be pessimistic. Efron showed that this bootstrap .632 estimator has a bias similar to cross-validation, but a much lower variance.

However, the bootstrap .632 does not take overfitting properly into account. In cases where strong overfitting is indeed observed, bootstrap .632 is optimistically biased, sometimes heavily, as shown in the work of Kohavi (1995). Efron and Tibshirani replied to this analysis by proposing two years later an extension to the bootstrap .632 (Efron and Tibshirani, 1997). This ex-

tension, named bootstrap .632+, includes a measure of the model overfitting. It offers relatively small bias and variance, and is robust to overfitting.

Both bootstrap .632 and .632+ are limited to the estimation of the classification error rate, which is the complement of the model accuracy. According to the confusion matrix pictured in Table 4.1, this error corresponds to $1 - \frac{n_{(-,-)} + n_{(+,+)}}{n}$. However many practical applications require the estimation of other statistics than accuracy, like the sensitivity, the specificity, etc. A critical example is in healthcare, when classification models based on clinical information and/or genomic data are built to predict if cancerous patients should undergo a chemotherapy or not. Although the number of unnecessarily prescribed chemotherapy should be reduced as much as possible, it is worse not giving it to a patient actually requiring it, than the reverse. Another typical example is spam filtering, where it is worse trashing an important email than letting from times to times an undesired email reach the inbox. The bootstrap estimation of statistics corresponding to such situations is not described in previous work. Efron and Tibshirani (1997) study in details the bootstrap for classification problems with 0-1 loss. They stress that other prediction problems may exhibit qualitatively different behavior, and that the statistical approach could also differ (see (Efron, 1986; Efron and Tibshirani, 1997; Wahba, 1980) for more details on this discussion). It is shown experimentally later in this section that bootstrap .632+ in its traditional form leads to incoherent results when used for estimating other statistics than the error related to accuracy. The extension of the bootstrap .632+ to other metrics is precisely the topic of this section.

Section 5.3.1 reviews in details the bootstrap .632 and .632+ approaches, sticking to the notations defined in previous chapters and those used by Efron and Tibshirani when possible. Section 5.3.2 develops an upgrade of the bootstrap .632+, called the generalized bootstrap .632+, allowing robust estimations of other statistics than the classification error rate of a model. This enhanced version allows the estimation of any statistics defined as a function of the entries of a confusion matrix defined as in Table 4.1. In particular, this chapter proposes four such extensions of the bootstrap .632+, namely for sensitivity, specificity, balanced classification rate and positive predicted value. Computational complexity of the bootstrap variants is considered throughout Sections 5.3.1 and 5.3.2. It is shown that the new generalized bootstrap .632+ has the same complexity as the bootstrap .632+, although it

offers to compute several statistics at once. Conclusions and perspectives are addressed in Section 5.3.3.

5.3.1 Bootstrap and Bootstrap+

This section reviews in more details the bootstrap both in its and + versions. A complexity analysis is led in each case.

Bootstrap

Like in the work of [Efron and Tibshirani \(1997\)](#), consider the 0/1-loss $\mathcal{L}_{01}$ in Equation (2.7). The true error rate of a function f estimated on the n samples of a training set $\mathbf{X}$ can be defined as:

$$Err = Err(\mathbf{X}, P) = \int \mathcal{L}_{01}(y, f_{\mathbf{X}}(\mathbf{x})) dP(\mathcal{X}, \mathcal{Y}). \quad (5.1)$$

The subscript in $f_{\mathbf{X}}$ denotes that model f has been estimated on the set of points $\mathbf{X}$ only. Note Equation (5.1) corresponds to the functional risk of Equation (2.13) with a 0/1-loss. It measures the error that a model estimated on a limited number of points would do on the whole points distribution. Since the whole distribution P is typically not available, the computation of the true error rate is not tractable. An estimated, or empirical error rate is thus defined as:

$$e\bar{r}r = Err(\mathbf{X}, \hat{P}) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}_{01}(y_i, f_{\mathbf{X}}(\mathbf{x}_i)), \quad (5.2)$$

with $\hat{P}$ meaning the empirical distribution, i.e. the training sample, putting a probability of $1/n$ over each $\mathbf{x}_i \in \mathbf{X}$. Note that Equation (5.2) corresponds to the empirical risk R_{emp} in Equation (2.15). The inference of an error only based on $e\bar{r}r$ is optimistic, because points of $\mathbf{X}$ have been used both to build the model and to test it. The cross-validation setting avoids this issue by holding observations used for the test out of the set of points used for the estimation of the model.

Let us consider the bootstrap sample $\mathbf{X}^*$, which is a random draw with replacement of n points among $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$. Many (e.g. B) bootstrap samples are considered for a single Bootstrap estimation: $\mathbf{X}^{*1}, \mathbf{X}^{*2}, \dots, \mathbf{X}^{*B}$. Let us define I_i^b as an indicator function being equal to 0 if $\mathbf{x}_i$ is selected *at least* one time in bootstrap sample b , and being equal to 1 if $\mathbf{x}_i$ is *not* selected in bootstrap sample b . Then, an estimate of the out-of-bag

error $\hat{Err}^{(1)}$ is defined as follows:

$$\hat{Err}^{(1)} = \frac{1}{n} \sum_{i=1}^n \hat{E}_i, \quad (5.3)$$

with

$$\hat{E}_i = \frac{\sum_{b=1}^B I_i^b \mathcal{L}_{01}(y_i, f_{\mathbf{X}^{*b}}(\mathbf{x}_i))}{\sum_{b=1}^B I_i^b}. \quad (5.4)$$

This estimator of the error has been shown to be pessimistic, unlike $e\bar{r}r$, mainly because $f_{\mathbf{X}^*}$ is built on fewer different samples than $f_{\mathbf{X}}$. Based on all the introduced notations, the standard .632 estimator first introduced in (Efron, 1983) can be expressed as a combination of the optimistic $e\bar{r}r$ and the pessimistic $\hat{Err}^{(1)}$:

$$\hat{Err}^{(.632)} = 0.368 \times e\bar{r}r + 0.632 \times \hat{Err}^{(1)}. \quad (5.5)$$

B is not upper-bounded to the number of samples n , unlike K in KCV even in its extreme LOO configuration, and is generally set to a much higher value than n in contexts where n is small. The average bootstrap error is originally presented as having a bias comparable to the KCV, but a much lower variance. More precisely, Efron showed that $\hat{Err}^{(1)}$ has a variance equivalent to the one of LOO computed on 1.60 times more samples.

The bootstrap .632 has a computational complexity of $\mathcal{O}(B\mathcal{C}_{f_{\mathbf{X}^*}})$ where $\mathcal{C}_{f_{\mathbf{X}^*}}$ is the complexity of the estimation of f on the Bootstrap sample of the same size as $\mathbf{X}^*$, which is n . Such complexity can be compared to the one of the KCV, which is $\mathcal{O}(K\mathcal{C}_{f_{\mathbf{X}^{(cv)}}})$. Generally, the bootstrap will require more time to run than KCV, mainly because:

1. In practice, B is typically larger than K when n is small.
2. The size of a bootstrap sample, n , is greater than the size of $\mathbf{X}^{(cv)}$, which follows $\approx \frac{n(K-1)}{K}$, and most classifiers have a complexity at least linear in n .

This slight increase of time complexity is the price to pay for similar bias and lower variance.

Bootstrap .632+

The estimator in Equation (5.5) may still show optimistic performance evaluation in case of strong overfitting (Efron and Tibshirani, 1997). The bootstrap

.632+ overcomes this issue by taking into account a measure of the overfitting of the model. To cite Efron: “The .632+ rule puts greater weight on $\hat{Err}^{(1)}$ in situations where the amount of overfitting as measured by $\hat{Err}^{(1)} - e\bar{r}r$ is large”. Its expression is:

$$\hat{Err}^{(.632+)} = (1 - \hat{w}) \times e\bar{r}r + \hat{w} \times \hat{Err}^{(1)}, \quad (5.6)$$

where $\hat{w}$ takes the overfitting into account. It is defined as follows:

$$\hat{w} = \frac{0.632}{1 - 0.368 \times \hat{R}}. \quad (5.7)$$

The *Relative overfitting*, $\hat{R}$, is defined by:

$$\hat{R} = \frac{\hat{Err}^{(1)} - e\bar{r}r}{\hat{\gamma} - e\bar{r}r}. \quad (5.8)$$

The concept of γ , the *no-information error rate*, deserves some explanation. It is the equivalent of Err in Equation (5.1) but for a distribution P_{ind} where the y_i are independent from the x_i . An estimate of γ can be obtained via a simple permutation experiment. In this experiment, every point is given every label, constituting a new dataset of n^2 points. A decision rule f_X is trained over the n original points-label pairs, and is used to make decisions on the new, permuted dataset. This permutation experiment produces a confusion matrix illustrated in Table 5.1, where $\hat{p}$ is the observed proportion of responses y_i actually equal to +1 in the original dataset and $\hat{q}$ is the observed proportion of predictions f_X equal to +1 on the original dataset X . The error rate of the model on the permuted dataset can be computed as follows, as in the original bootstrap .632+ paper:

$$\hat{\gamma} = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \mathcal{L}_{01}(y_i, f_X(x_j)). \quad (5.9)$$

Table 5.1 A confusion matrix for a binary classification problem with permutation, where each observation has been assigned every other observation label value once. $\hat{p}$ is the observed proportion of responses y_i actually equal to +1 and $\hat{q}$ is the observed proportion of predictions f_X equal to +1.

		True	
		-1	+1
Predicted	-1	$(1 - \hat{p})(1 - \hat{q})n^2$	$\hat{p}(1 - \hat{q})n^2$
	+1	$(1 - \hat{p})\hat{q}n^2$	$\hat{p}\hat{q}n^2$

Equation (5.9) corresponds to the traditional classification error rate, being one minus the accuracy. It can be rewritten in terms of the entries of Table 5.1:

$$\hat{\gamma} = \frac{1}{n^2} \left(\hat{p}(1 - \hat{q})n^2 + (1 - \hat{p})\hat{q}n^2 \right) = \hat{p}(1 - \hat{q}) + (1 - \hat{p})\hat{q}. \quad (5.10)$$

The computation of $\hat{\gamma}$ occurs only once. Let us comment a bit on this upgraded bootstrap. The heart of this technique relies in $\hat{R}$. It is where the overfitting is estimated. If no overfitting occurs, i.e. when $\hat{Err}^{(1)} = e\bar{r}r$, $\hat{R}$ is equal to 0. In that case, $\hat{w} = 0.632$, as in the original bootstrap .632 setting. On the other hand, when an extreme overfitting is observed, $\hat{R}$ is equal to 1 ($\hat{Err}^{(1)} = \hat{\gamma}$). In that case, $\hat{w} = 1$ and $\hat{Err}^{(.632+)}$ only takes the OOB error $\hat{Err}^{(1)}$ into account.

In practice, two adjustments are introduced because it may happen that $\hat{\gamma} \leq e\bar{r}r$ or $e\bar{r}r < \hat{\gamma} \leq \hat{Err}^{(1)}$. In that case, $\hat{R}$ falls outside the range $[0, 1]$. The first modification proposed by [Efron and Tibshirani \(1997\)](#) is to replace $\hat{Err}^{(1)}$ by:

$$\hat{Err}^{(1)'} = \min(\hat{Err}^{(1)}, \hat{\gamma}). \quad (5.11)$$

The second one is to replace $\hat{R}$ by:

$$\hat{R}' = \begin{cases} \hat{R} & \text{if } \hat{Err}^{(1)'}, \hat{\gamma} > e\bar{r}r, \\ 0 & \text{otherwise.} \end{cases} \quad (5.12)$$

The time complexity of the bootstrap .632+ is $\mathcal{O}(n^2 + BC_{f_{X^*}})$. The n^2 term is due to the computation of $\hat{\gamma}$ that requires a permutation experiment and is the price of achieving robust estimation even in occurrence of overfitting. In practice, the permutation experiment is not strictly needed, because $\hat{\gamma}$ can be obtained based on $\hat{p}$ and $\hat{q}$ only and the method thus has a complexity of $\mathcal{O}(n + BC_{f_{X^*}})$. This complexity might even be reduced to $\mathcal{O}(BC_{f_{X^*}})$ in cases where the considered model is guaranteed to perform perfect classification on the training set. This is the case of the 1-Nearest Neighbor for example. In this case, the computation of $\hat{\gamma}$ can be achieved in $\mathcal{O}(1)$ since $\hat{p} = \hat{q}$ and $\hat{\gamma} = 2\hat{p}(1 - \hat{p})$.

5.3.2 Generalized Bootstrap .632+

The bootstrap makes an estimate of the classification error rate of a binary classifier. Other statistics might be of interest for diverse applications, such

as the balanced classification rate (BCR) or other functions of the confusion matrix. The current section proposes an upgrade of the bootstrap .632+ to make possible the estimation of any function of a confusion matrix. A complete development of the cases corresponding to the BCR, the sensitivity, the specificity, the positive predicted value is provided.

We first motivate our work by showing that the standard form of the bootstrap .632+ is not able to provide correct estimates of other functions of the confusion matrix. To do so, let us consider a dataset X_1 of 200 points in $\mathbb{R}^5$ corresponding to an unbalanced binary classification problem. The prior of the $y_i = -1$ class is 90%, while the prior of the other class, $y_i = +1$, is 10%. The negative class follows $\mathcal{N}(0, 1, 0, 1, 0)$ with unit standard deviation, and the positive class follows $\mathcal{N}(-1, 0, -1, 0, -1)$, also with unit standard deviation. We apply 500 iterations of the classical bootstrap .632+ with $\hat{\gamma}$ defined as in Equation (5.10), with two different classification models: a 1-Nearest-Neighbor, and a linear SVM (hard margin, $C=1000$). The 1-NN is known to be prone to overfitting while the linear SVM involves a regularization mechanism tending to avoid this issue. The experimental results are shown in columns 1 and 4 of Table 5.2. Columns 2 and 5 of Table 5.2 display the results of the same experiment with the appropriate generalized bootstrap .632+ that is introduced next. Those two columns are discussed at the end of this section. Column 3 and 6 show approximations of the true values obtained on a dataset of 100.000 points drawn from the same distributions as the 200 points dataset, with the same 90%/10% class priors with the 1-NN and the SVM, respectively.

Table 5.2 Bootstrap experiments. Columns 1-3: 1-NN. Columns 4-6: linear SVM. Columns 1 and 4: bootstrap .632+ (original $\hat{\gamma}$). Columns 2 and 5: generalized bootstrap .632+ (modified $\hat{\gamma}$). Columns 3 and 6: expected values approximations.

	1-NN			linear SVM		
	Orig. $\hat{\gamma}$	Mod. $\hat{\gamma}$	Exp.	Orig. $\hat{\gamma}$	Mod. $\hat{\gamma}$	Exp.
Acc.	92.17	92.17	89.77	93.46	93.46	93.66
$\hat{\gamma}$	0.18	0.18	-	0.18	0.18	-
$\hat{R}$	0.549	0.549	-	0.175	0.175	-
$\hat{w}$	0.792	0.792	-	0.675	0.675	-
$e\bar{r}r$	0	0	-	0.05	0.05	-
$\hat{E}rr^{(1)}$	0.099	0.099	-	0.073	0.073	-
BCR	82.00	75.97	65.64	82.00	79.22	78.88
$\hat{\gamma}_{BCR}$	0.18	0.5	-	0.18	0.5	-
$\hat{R}$	1.00	0.594	-	1.00	0.272	-
$\hat{w}$	1.00	0.809	-	1.00	0.702	-
$e\bar{r}r$	0	0	-	0.139	0.139	-
$\hat{E}rr^{(1)}$	0.18	0.297	-	0.18	0.237	-
Sens.	82.00	55.74	35.48	79.42	61.44	60.40
$\hat{\gamma}_+$	0.18	0.9	-	0.18	0.9	-
$\hat{R}$	1.00	0.605	-	0.00	0.294	-
$\hat{w}$	1.00	0.813	-	0.632	0.709	-
$e\bar{r}r$	0	0	-	0.25	0.25	-
$\hat{E}rr^{(1)}$	0.18	0.544	-	0.18	0.441	-
Spec.	96.50	96.15	95.80	96.91	96.91	97.36
$\hat{\gamma}_-$	0.18	0.1	-	0.18	0.1	-
$\hat{R}$	0.276	0.498	-	0.032	0.067	-
$\hat{w}$	0.704	0.774	-	0.639	0.648	-
$e\bar{r}r$	0	0	-	0.028	0.028	-
$\hat{E}rr^{(1)}$	0.05	0.05	-	0.033	0.033	-
PPV	82.00	61.65	48.40	79.42	69.94	71.75
$\hat{\gamma}_{PPV}$	0.18	0.9	-	0.18	0.9	-
$\hat{R}$	1.00	0.540	-	0.00	0.118	-
$\hat{w}$	1.00	0.789	-	0.632	0.661	-
$e\bar{r}r$	0	0	-	0.25	0.25	-
$\hat{E}rr^{(1)}$	0.18	0.486	-	0.18	0.327	-

The interest of the experiment is double. First it shows why it is interesting to compute other statistics than the accuracy, specially in the case where class

priors are unbalanced. Indeed, the high accuracy (around 90% for 1-NN, around 94% for SVM) is due to the fact that the dataset is unbalanced. The point is not that the accuracy is not correct, but rather that it is misleading. A look on other statistics, like the BCR, shows that models tend to classify more points than they should in the majority class. It might thus be useful to have robust estimators for other statistics than just accuracy. Second, it shows that computing those other statistics without modifying the way the bootstrap .632+ works leads to poor and inconsistent results. For the 1-NN for example, the BCR estimation is more than 20% higher than the expected value, and the sensitivity nearly 50% higher than the expected value. On top of being poor, those estimation are inconsistent. Indeed, the arithmetic mean of the sensitivity and specificity does not lead to a value close to the BCR, as it should normally be the case by definition of the BCR. For the 1-NN, $\frac{1}{2}(82.00 + 96.50) = 89.25$ is far from 82.00. For the SVM, $\frac{1}{2}(79.42 + 96.91) = 88.165$ is far from 82.00.

Modifications of the standard bootstrap .632+ have thus to be performed in order to estimate other statistics than the accuracy. Although the meaning of all components of Equation (5.6) have to be modified to reflect the statistics of interest, the key modification is the computation of the adequate $\hat{\gamma}$'s. A distinct $\hat{\gamma}$ has to be defined for every function of the entries of a confusion matrix.

The rest of this section is subdivided in four parts, each of which is dedicated to the derivation of the bootstrap .632+ estimation of a different function of the confusion matrix.

Sensitivity

The first function considered is the sensitivity of a model. Like for accuracy, we want a robust estimator of its corresponding error, defined as 1-sensitivity. Both $e\bar{r}r$ and $\hat{E}rr^{(1)}$ have to be replaced by a measure of that error instead of the standard error rate. This is achieved by replacing n by n_+ in Equations (5.2) and (5.3). The corresponding $\hat{\gamma}$ is named $\hat{\gamma}_+$ and can be computed similarly to Equation (5.10), based on the confusion matrix in Table 5.1 resulting from

the permutation experiment:

$$\hat{\gamma}_+ = 1 - \text{sensitivity} \quad (5.13)$$

$$= 1 - \frac{\hat{p}\hat{q}n^2}{\hat{p}\hat{q}n^2 + \hat{p}(1 - \hat{q})n^2} \quad (5.14)$$

$$= 1 - \frac{\hat{p}\hat{q}}{\hat{p}\hat{q} + \hat{p}(1 - \hat{q})} \quad (5.15)$$

$$= \frac{\hat{p}\hat{q} + \hat{p}(1 - \hat{q}) - \hat{p}\hat{q}}{\hat{p}\hat{q} + \hat{p}(1 - \hat{q})} \quad (5.16)$$

$$= \frac{\hat{p}(1 - \hat{q})}{\hat{p}\hat{q} + \hat{p}(1 - \hat{q})} \quad (5.17)$$

$$= \frac{(1 - \hat{q})}{\hat{q} + (1 - \hat{q})} \quad (5.18)$$

$$= 1 - \hat{q}. \quad (5.19)$$

Specificity

The second function considered is the specificity. We thus want a robust estimator of the corresponding error, 1-specificity. Both $e\bar{r}r$ and $\hat{Err}^{(1)}$ have to be replaced by a measure of that error. This is achieved by replacing n by n_- in Equations (5.2) and (5.3). The same line of reasoning can be applied to discover the $\hat{\gamma}$ adequate for the error corresponding to the specificity, named $\hat{\gamma}_-$:

$$\hat{\gamma}_- = 1 - \text{specificity} \quad (5.20)$$

$$= 1 - \frac{(1 - \hat{p})(1 - \hat{q})n^2}{(1 - \hat{p})(1 - \hat{q})n^2 + (1 - \hat{p})\hat{q}n^2} \quad (5.21)$$

$$= 1 - \frac{(1 - \hat{p})(1 - \hat{q})}{(1 - \hat{p})(1 - \hat{q}) + (1 - \hat{p})\hat{q}} \quad (5.22)$$

$$= \frac{(1 - \hat{p})(1 - \hat{q}) + (1 - \hat{p})\hat{q} - (1 - \hat{p})(1 - \hat{q})}{(1 - \hat{p})(1 - \hat{q}) + (1 - \hat{p})\hat{q}} \quad (5.23)$$

$$= \frac{(1 - \hat{p})\hat{q}}{(1 - \hat{p})(1 - \hat{q}) + (1 - \hat{p})\hat{q}} \quad (5.24)$$

$$= \frac{\hat{q}}{(1 - \hat{q}) + \hat{q}} \quad (5.25)$$

$$= \hat{q}. \quad (5.26)$$

Balanced Classification Rate

The third considered statistics is the balanced classification rate (BCR), to which corresponds the balanced error rate (BER=1-BCR). Both $e\bar{r}r$ and $\hat{Err}^{(1)}$ have to be replaced by the BER measure instead of the accuracy. This requires to redefine $e\bar{r}r$ and $\hat{Err}^{(1)}$ as follows:

$$e\bar{r}r_{BCR} = Err_{BCR}(\mathbf{x}, \hat{P}) \quad (5.27)$$

$$= \frac{1}{2} \left[\frac{1}{n_+} \sum_{i=1}^{n_+} \mathcal{L}_{01}(y_i, f_{\mathbf{x}}(\mathbf{x}_i)) + \frac{1}{n_-} \sum_{i=1}^{n_-} \mathcal{L}_{01}(y_i, f_{\mathbf{x}}(\mathbf{x}_i)) \right], \quad (5.28)$$

$$\hat{Err}_{BCR}^{(1)} = \frac{1}{2} \left[\frac{1}{n_+} \sum_{i=1}^{n_+} \hat{E}_i + \frac{1}{n_-} \sum_{i=1}^{n_-} \hat{E}_i \right], \quad (5.29)$$

with $\hat{E}_i$ defined in Equation (5.4). The $\hat{\gamma}$ is named $\hat{\gamma}_{BCR}$ and is the arithmetic mean between $\hat{\gamma}_+$ and $\hat{\gamma}_-$. Its development is easier since it relies on the two previous ones:

$$\hat{\gamma}_{BCR} = \frac{1}{2} (\hat{\gamma}_+ + \hat{\gamma}_-) \quad (5.30)$$

$$= \frac{1}{2} ((1 - \hat{q}) + \hat{q}) \quad (5.31)$$

$$= \frac{1}{2}. \quad (5.32)$$

Positive Predicted Value

The fourth function of interest in this section is the positive predicted value (PPV). Both $e\bar{r}r$ and $\hat{Err}^{(1)}$ have to be replaced by a measure of 1-PPV in Equations (5.2) and (5.3). The complete development of the adequate $\hat{\gamma}$ is as fol-

lows:

$$\hat{\gamma}_{PPV} = 1 - PPV \quad (5.33)$$

$$= 1 - \frac{\hat{p}\hat{q}n^2}{\hat{p}\hat{q}n^2 + (1 - \hat{p})\hat{q}n^2} \quad (5.34)$$

$$= 1 - \frac{\hat{p}\hat{q}}{\hat{p}\hat{q} + (1 - \hat{p})\hat{q}} \quad (5.35)$$

$$= \frac{\hat{p}\hat{q} + (1 - \hat{p})\hat{q} - \hat{p}\hat{q}}{\hat{p}\hat{q} + (1 - \hat{p})\hat{q}} \quad (5.36)$$

$$= \frac{(1 - \hat{p})\hat{q}}{\hat{p}\hat{q} + (1 - \hat{p})\hat{q}} \quad (5.37)$$

$$= \frac{(1 - \hat{p})}{\hat{p} + (1 - \hat{p})} \quad (5.38)$$

$$= 1 - \hat{p}. \quad (5.39)$$

Computational Complexity

The computational complexity of the generalized bootstrap is the same as the one of the bootstrap .632+. Even in the case where a user is interested in several bootstrap statistics at once, the complexity remains unchanged. Indeed, the computation of all $\hat{\gamma}$ can be based on a single permutation experiment.

Experimental Validation

The same experiment as the one described at the beginning of this section is performed again, but with the generalized bootstrap .632+ with modified $\hat{\gamma}$'s. Results are shown in columns 2 and 5 of Table 5.2. By definition, the $\hat{\gamma}$ corresponding to the classification error rate does not change and identical results are obtained (comparing column 1 to 2, and column 4 to 5 respectively). The $\hat{\gamma}$'s of the other statistics are however strongly affected, ranging from 0.1 to 0.9 instead of being constantly set to 0.18. The relative overfitting measured by $\hat{R}$ are modified accordingly. The quality of the estimates is improved by the use of the generalized bootstrap .632+. Those are much closer to the expected values both for 1-NN and SVM than in with traditional bootstrap .632+. For example, the difference between the expected sensitivity and the bootstrap .632+ estimate of the sensitivity with the SVM is 19.02%. This difference is reduced to 1.04% with the generalized bootstrap .632+ estimate. The increase or decrease in $\hat{R}$ values in the case of generalized bootstrap .632+ combined to this quality improvement proves that it detects much better the overfitting. In any case, it is lower for SVM than for 1-NN, as expected. A last point in

favor of the generalized bootstrap .632+ is that the mean of sensitivity and specificity estimates is much closer to the BCR estimate than previously. For the 1-NN, $\frac{1}{2}(96.15 + 55.74) = 75.945$, which is close to 75.97. For the SVM, $\frac{1}{2}(96.91 + 61.44) = 79.175$, which is close to 79.22. All those arguments plead to use the generalized version of the bootstrap .632+ rather than the former bootstrap .632+. The switch to this new version should be also encouraged by the fact that the computational complexities are equivalent.

5.3.3 Conclusion and Perspectives

We improve over the bootstrap .632+ validation procedure to make possible the estimation of other statistics than the accuracy of a binary classifier. We introduce in details the computation of a bootstrap estimate of the sensitivity, the specificity, the balanced classification rate and the positive predicted value robust to overfitting, as is the bootstrap .632+ for accuracy. The process can be extended to any other function of the confusion matrix of a binary classifier in a similar way.

The computational complexity of the generalized bootstrap .632+ is equivalent to the one of the bootstrap .632+ although it can compute several statistics at once. This complexity is slightly higher than the one of the bootstrap .632, in turn a bit higher than the one of k-fold cross-validation. The increase in computational complexity is the price for low bias and low variance, with robustness to overfitting.

The 1997 paper of Efron and Tibshirani details how to compute the standard deviation of $\hat{Err}^{(1)}$ in the case of the bootstrap .632+ estimator. The development is rather long. The computation of the standard deviation of $\hat{Err}^{(1)}$ for the generalized bootstrap .632+ is totally unchanged with respect to the previous one. That is why it is not included in this chapter. However, Efron and Tibshirani also mention that the estimation of the standard deviation of the bootstrap .632+ itself is complex and do not derive it. A much welcome future contribution could be an unbiased estimation of the (generalized) bootstrap .632+ standard deviation.

The extension of the bootstrap in this chapter only applies to binary classification problems, as the original bootstrap .632 is designed to work only in that setting (Efron and Tibshirani, 1997). Being the latest of a series of improvements for robust performance estimation, the generalized bootstrap

.632+ leaves room for another improvement: the multi-class generalized bootstrap .632+, as well as the extension to other metrics, such as the AUC.

5.4 Resamplings

5.4.1 Resampling Strategy

In cases where the number of available samples n is really low, it is difficult to build relevant models on only 63.2% of the samples, as it is the case with bootstrap. Another validation setting is to draw a higher proportion of the samples in the dataset at random, for example 90%, without repetition, and to train a model on it. Its evaluation is then done on the remaining 10% of unused samples.

Formally, given a data set $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ with n instances and p features, J subsamplings of size $\lceil n_1 \times n \rceil$ ($0 < n_1 < 1$) are drawn randomly from $\mathbf{X}$. The model is trained on each of those subsamplings, and tested on the independent test set made of the $\lfloor n_2 \times n \rfloor$ samples not selected, where $n_2 = 1 - n_1$. A typical value for J in this thesis is a few hundreds, and $n_1 = 0.9$.

This setting is thus similar to the 10-fold cross-validation (10CV) in the way it allows the models to be built on more different points than in a bootstrap procedure. Unlike 10CV however, the fact that this procedure is repeated much more than 10 times (hundreds or thousands) provides an average performance computed over more estimations. This technique, called N-Resamplings is prone to another pitfall not present in KCV: the different test sets are not independent. This issue is addressed next.

5.4.2 A Corrected Significance Test

Assessing a significant difference between two models performances evaluated via a N-Resamplings strategy requires a special care to take into account the test sets dependence. The same applies to confidence interval estimation. [Nadeau and Bengio \(2003\)](#) propose a corrected statistical test for this particular situation, as well as a corrected confidence interval. Both are described in the following paragraphs.

Corrected resampled t -test

Tests of the hypothesis $H_0: \mu = \mu_0$ (at significance level δ) have the following form:

$$\text{Reject } H_0 \text{ if } |\hat{\mu} - \mu_0| > c\sqrt{\hat{\sigma}^2}. \quad (5.40)$$

Corrected Confidence Interval

The confidence interval for μ at a confidence level of $1 - \delta$ is:

$$\mu \in [\hat{\mu} - c\sqrt{\hat{\sigma}^2}; \hat{\mu} + c\sqrt{\hat{\sigma}^2}]. \quad (5.41)$$

For an N-Resamplings procedure of $n_1\%$ of the samples as training set, $n_2\% = 100 - n_1\%$ of the samples as test set, with J iterations, the values of $\hat{\mu}$, $\hat{\sigma}$ and c in Equations (5.40) and (5.41) are estimated as follows.

- $\hat{\mu} = \frac{1}{J} \sum_{j=1}^J \hat{\mu}_j$,
- $\hat{\sigma} = \left(\frac{1}{J} + \frac{n_2}{n_1}\right) \times S_{\hat{\mu}_j}^2$,
- $c = t_{J-1, 1-\delta/2}$,

where

- $\hat{\mu}_j$ is the estimation of μ on the $n_2 \times n$ samples of the j^{th} test set output by the model estimated on the $n_1 \times n$ samples of the j^{th} training set,
- $S_{\hat{\mu}_j}^2$ is the variance computed on the J values of $\hat{\mu}_j$,
- $t_{J-1, 1-\delta/2}$ is the $1 - \delta/2$ quantile of a student distribution with $J - 1$ degrees of freedom. This tends to equal 1.9599 when $\delta = 0.05$ and $J \rightarrow \infty$.

With such corrected statistics available, and given the typically low number of samples available in microarray data, the most used validation protocol in the experiments reported in this thesis is a N-Resamplings protocol with $n_1 = 90\%$.

5.5 Random Experiments

Model performances cross-checking can be achieved by other means than those presented in previous sections. Such additional checks may be sound, specially if the model is subject to some bias, such as selection bias. Permutation experiments are another way to get confidence in a classification model. Many random permutation schemes can be deployed, for example:

- Within a validation loop, permute the test set labels. If the model still behaves well a significant number of times, then the concept learned is not specific to the initial target variables distribution. The model could have learned something else.
- Within a validation loop, permute the training set labels. If the model still behaves well a significant number of times on the test set, then it means that other models are predictive of the target variables distribution.

Such tests have been applied in the context of the Melanoma (Louahed *et al.*, 2008; Dupont *et al.*, 2010). In the PS- L_2 -AROM paper (Helleputte and Dupont, 2009c), another test is applied. Noisy prior knowledge is simulated by randomly favoring some genes through the selection process. The technique shows that it can depart from that random (and thus probably poor) knowledge (see Chapter 8). Reasoning with permutation experiments can also lead to counter-intuitive results as detailed in Appendix G.

5.6 The Choice of a Protocol

In this thesis, numerous experiments are led in order to estimate the performance of some models. There come validation protocols into play. In most cases, we choose the N-Resamplings protocol detailed above. We argue that it is adequate in the current applicative context, though the other protocols introduced might find their utility in other situations.

First the number of evaluations that can be carried out with N-Resamplings is not upper-bounded by n , as it is the case for the KCV in its extreme LOO configuration. This is important since n is assumed to be small in our context. Second, there is no replacement scheme in N-Resamplings, unlike with the bootstrap. Such drawings with replacement could lead to optimistic biases in the case of nested validation loops. Nested loops are useful when a meta-parameter has to be tuned, for example.

The N-Resamplings protocol is the most used throughout this thesis. The other protocols are nevertheless used on few occasions.

Preprocessing

6

Data may require some preprocessing, be it for classification or feature selection. This chapter covers two aspects of preprocessing: normalization (Section 6.1), making possible the comparison of every piece of data on a fair basis, and feature pre-filtering (Section 6.2), allowing the quick drop of an important proportion of the features without degrading (and sometimes improving) the quality of the final predictive models.

6.1 Normalization

6.1.1 Feature and Sample Normalization

Normalization is an important issue in data analysis and many models can suffer from the lack of it. Two orthogonal types of normalization can generally be considered: the sample normalization and the feature normalization.

Summarization

In the context of microarray data analysis, the sample normalization is carried out through a preprocessing step known as summarization. This preprocessing is specific to the microarray (Affymetrix) technology and is more extensively covered in Appendix C.2. Nevertheless, the motivation behind samples normalization is more general. In the case of gene expression high-density microarray data analysis, samples (patients) normalization stems from a biological assumption. It is assumed that a given biological process (for example, a disease) is regulated by a limited number of genes, that this number is small relative to the total number of probesets on a chip and that this set of genes is the same for significant parts of the population. Apart from those genes, we

make the hypothesis that every patient is *globally* equivalent. The expression profiles based on all the measured genes should thus be almost equivalent. To enforce this, normalization standardizes the global expression level of each microarray chip of a dataset corresponding to a cohort. This hypothesis might not be valid for low-density microarrays. Patients normalization also intends to erase the global intensity level differences due to experimental variance. After the summarization steps, histograms of every chip expression intensities are ideally highly similar.

Feature Normalization

Another preprocessing step, feature normalization, is also generally required to make most machine learning techniques work properly. For example, this step is recommended for SVMs (Hsu *et al.*, 2009) by the authors of one of the most used SVM library (`libsvm`), for Artificial Neural Networks (Sarle, 1997), and most of those considerations also hold for a vast majority of models. Intuitively, the rationale behind feature normalization can be understood as the will not to favor a particular feature just because of numerical effects. This kind of normalization is not, by default, included in summarization. The two following sections introduce two variants of feature normalization respectively known as the Z-Score and the IQR normalization. The first one is a standard normalization technique sometimes referred to as a standardization procedure. The second one is a robust version of the Z-Score particularly convenient when facing a low number of samples, likely to be stained by noise.

6.1.2 Z-Score

In a given dataset containing n samples in $\mathbb{R}^p$, the Z-Score transform of a given feature $\mathbf{x}_j = \{x_{1j}, x_{2j}, \dots, x_{nj}\}$ with $j \in \{1, \dots, p\}$ corresponds to:

$$\mathbf{x}'_j \leftarrow \frac{\mathbf{x}_j - \mu(\mathbf{x}_j)}{\sigma(\mathbf{x}_j)}. \quad (6.1)$$

It means that $\mu(\mathbf{x}_j)$ is subtracted from each component of $\mathbf{x}_j$ and that the result is divided by $\sigma(\mathbf{x}_j)$. The Z-Score technique consists of estimating $\mu(\mathbf{x}_j)$ as the mean of entries of $\mathbf{x}_j$:

$$\mu(\mathbf{x}_j) = \bar{\mathbf{x}}_j = \frac{1}{n} \sum_{i=1}^n (x_{ij}), \quad (6.2)$$

and $\sigma(\mathbf{x}_j)$ as the standard deviation of $\mathbf{x}_j$:

$$\sigma(\mathbf{x}_j) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_{ij} - \bar{\mathbf{x}}_j)^2}. \quad (6.3)$$

After such a normalization, every dimension mean equals 0 and every dimension standard deviation equals 1. In other words, x'_{ij} is a dimensionless quantity indicating how many standard deviations an observation x_{ij} is above or below the mean of $\mathbf{x}_j$.

Normalization Bias

Beyond the mathematical way of normalizing features, deciding on which samples μ and σ should be computed is an important issue. If those quantities are computed on every sample of a dataset (both training and test samples), the information contained in the test samples potentially induces an optimistic bias in the final performances of the model. This is the normalization bias, similar to the selection bias when selecting features on both training and test samples. The solution is to compute μ and σ for each feature only on the training samples and then to apply them on test samples before making a prediction on them. In fact, this is even closer to the applicative setting where all aspects of a model have been computed and fixed (normalization, signature, model parameters) and routinely applied on real new patients: the normalization parameters of the model will be applied.

6.1.3 IQR Normalization

Some extra precautions can be taken to make normalization of Equation (6.1) more robust in our context where we have only few samples. Modifications are proposed about the way μ and σ are computed.

Breakdown Point

We consider the concept of *breakdown point*. The breakdown point is the proportion of arbitrarily large values that a statistic can tolerate without returning an arbitrarily large value (Huber, 1981). First, we propose to define μ as the median of a feature's values across the samples instead of the mean. The mean of a set of values can be dramatically influenced by a single extreme value, while the median cannot. The mean has a breakdown point of 0%. For example, suppose that a single value $x_{ij} \rightarrow \infty$, then $\bar{\mathbf{x}}_j \rightarrow \infty$. Noise in any single measurement has thus an impact on the mean. The median has a

breakdown point of $\lfloor \frac{n-1}{2n} \rfloor \approx 50\%$, which is the maximal value of breakdown point for any statistic (Huber, 1981). Intuitively, a breakdown point cannot exceed 50% because if more than half of the observations are contaminated, it is not possible to distinguish between the true distribution and the contaminating distribution. Median is thus much more robust to noise than the mean.

This difference of robustness between mean and median is reinforced by the small sample size we deal with. The fewer points we have, the larger the impact on the mean of a single outlier. Given that most microarray datasets are highly subject to both experimental and biological noises (see introduction and Appendix C), and generally count only few samples, the median is preferred to the mean. Without outliers, if the data are distributed normally, and as the number of samples grows, both statistics become equivalent.

Second, we suggest to define $\sigma(\mathbf{x}_j)$ as the inter quartile range (IQR) of $\mathbf{x}_j$ entries. A quartile is any of the three values which divide a sorted vector of values into four equal parts, so that each part represents one fourth of the values. The IQR is the difference between the third and the first quartiles:

$$IQR(\mathbf{x}_j) = Q_3(\mathbf{x}_j) - Q_1(\mathbf{x}_j). \quad (6.4)$$

Just like the median is more robust to outliers than the mean, IQR is more robust than standard deviation. It has a breakdown point of 25%, while standard deviation in Equation (6.3) has a breakdown point of 0%: any single outlier will impact the standard deviation, potentially dramatically.

We are interested in determining the link between the IQR and the standard deviation both computed on a normal distribution. The IQR of a continuous distribution can be calculated by integrating the probability density function between adequate bounds. The lower quartile Q_1 is a number such that the integral of the probability density function from $-\infty$ to Q_1 equals 0.25. Similarly, the upper quartile Q_3 is a number such that the integral of the probability density function from $-\infty$ to Q_3 equals 0.75. It can be shown that in the case of a normal distribution, the IQR of feature j is 1.349 times the standard deviation of that feature (Tukey, 1977). In practice, we thus choose to use $\frac{IQR(\mathbf{x}_j)}{1.349}$ as an estimator of $\sigma(\mathbf{x}_j)$ since it gives equivalent results to the standard deviation in cases where the data are distributed normally.

The robust version of the Z-Score is called the IQR normalization procedure. This robust normalization is used in all the experiments reported in this thesis.

6.2 Non-Specific Filtering

Non-specific filtering is a form of feature selection making no use of the class labels. It is intended to be very fast and used as a preprocessing step in order to reduce the number of features, before applying a finer but probably more time-consuming informed selection. Two methods are discussed in this chapter. The concept of pre-filtering has been applied for example by [Alon *et al.* \(1999\)](#). In that case, a proportion of the features with the lowest minimal value across all patients were dropped.

Another non-specific filtering technique has also been applied in the context of Melanoma ([Louahed *et al.*, 2008](#); [Dupont *et al.*, 2010](#)), where about 55.000 features were available. A gene never varying much across every patients, regardless of the class labels, is not likely to vary much across sub-groups of the patients, even if those groups coincide with the labels. The proposed non-specific filtering evaluates the IQR of every gene across patients, and drops a defined proportion of the features corresponding to the less varying ones. The next section illustrates the impact of such pre-filtering techniques on classification performances.

6.2.1 Impact on the Performances

Several issues of pre-filtering are addressed in this section. First, what is the ideal proportion of features to drop? Is the impact of pre-filtering the same in combination with further univariate and multivariate specific selection methods? Among the two considered non-specific filtering approaches, which one should be preferred?

A bootstrap protocol is used to test the impact of the pre-filtering step on the classification results. Both methods are tested on two datasets, LEU ([Golub *et al.*, 1999](#)) and PRO ([Singh *et al.*, 2002](#)), in combination with two univariate selection methods (t -test and Golub index) and two multivariate selection methods (RFE and L_2 -AROM). More information on those datasets can be found in Appendix D. The classifier used is a linear SVM with the C constant tuned according to the Joachims heuristic of Equation 2.38. A robust IQR normalization step is performed on the training part.

On each bootstrap sample, normalization parameters are computed, and a pre-filtering step occurs, dropping either 5, 25, 50, 75 or 95% of the features. On the remaining of the features, one of the four considered selection method is applied to select various signature sizes (smaller than the number of features kept after pre-filtering). The linear SVM is learned on the bootstrap sample restricted to the selected features. Points in the out-of-bag sample are normalized and classified, based only on the finally selected features. The balanced classification rate is then computed. Results for PRO are shown in Figures 6.1 to 6.4. Each curve corresponds to a given size of signature resulting of a specific selection subsequent to the non-specific filtering step. Similar results on LEU can be seen in Appendix E.

The conclusions are the following:

- Both methods are very fast.
- In combination with multivariate selection methods, BCR performances of the second technique (based on IQRs) are improved by non-specifically filtering out from 5% to 75% of the features and then applying the specific multivariate technique.
- In combination with univariate selection methods, BCR performances of the second technique (based on IQRs) are not degraded by non-specifically filtering out from 5% to 75% of the features and then applying the specific multivariate technique.
- In combination with univariate or multivariate selection methods, BCR performances of the first technique (based on minimal values) are not degraded by non-specifically filtering out from 5% to 75% of the features and then applying the specific univariate or multivariate technique.
- In any cases (first or second non-specific filtering, uni- or multivariate selection), a decrease in performances is generally observed if aggressive non-specific filtering is applied ($> 75\%$).
- All those effects are stronger in case of very small signatures.

As a conclusion, not-too-aggressive non-specific pre-filtering is a good heuristic to speed up the selection process if a really large number of features is encountered (filtering out $\leq 75\%$ of the features). Pre-filtering based on variability should be preferred since it sometimes improves the classification performances.

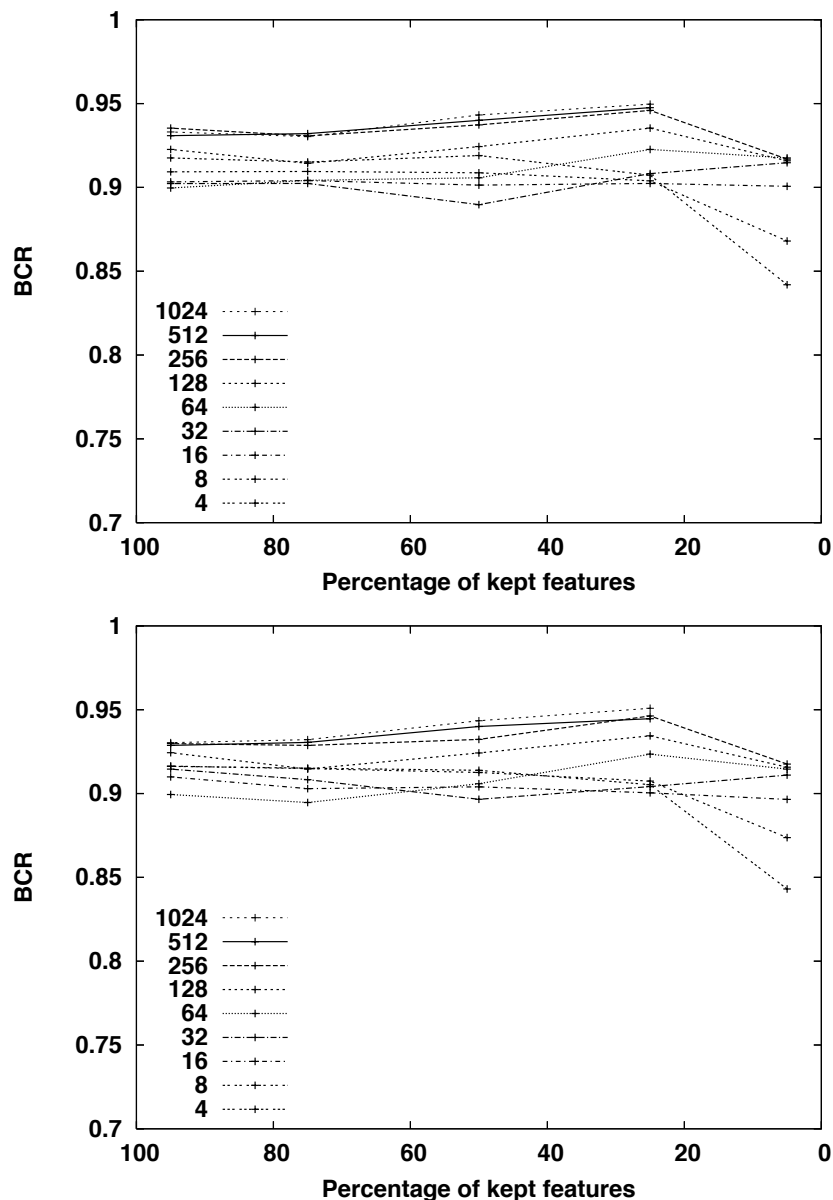


Figure 6.1 Non-specific filtering on PRO based on features minimal value. Top: t-Test used as a second selection method. Bottom: Golub index used as a second selection method

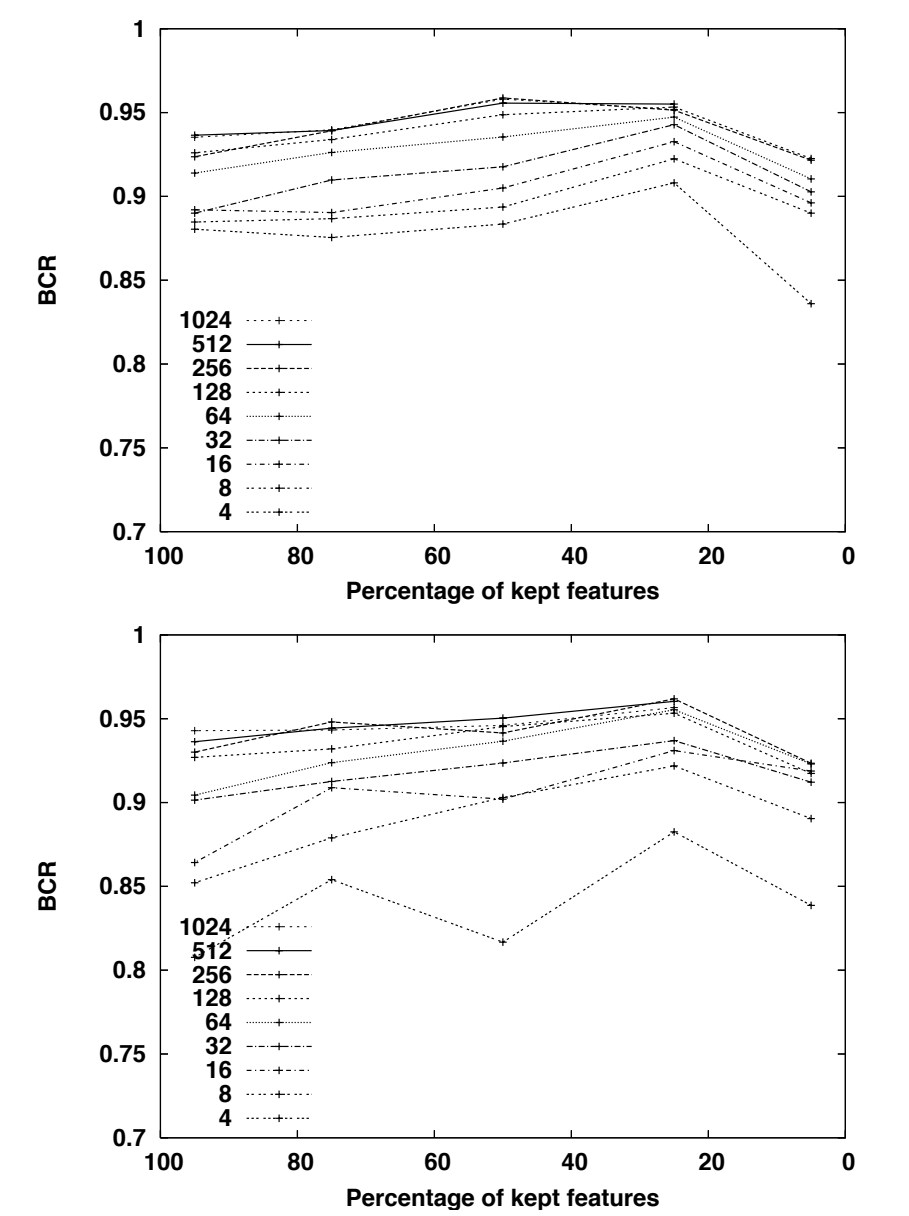


Figure 6.2 Non-specific filtering on PRO based on features minimal value. Top: RFE used as a second selection method. Bottom: L_2 -AROM used as a second selection method

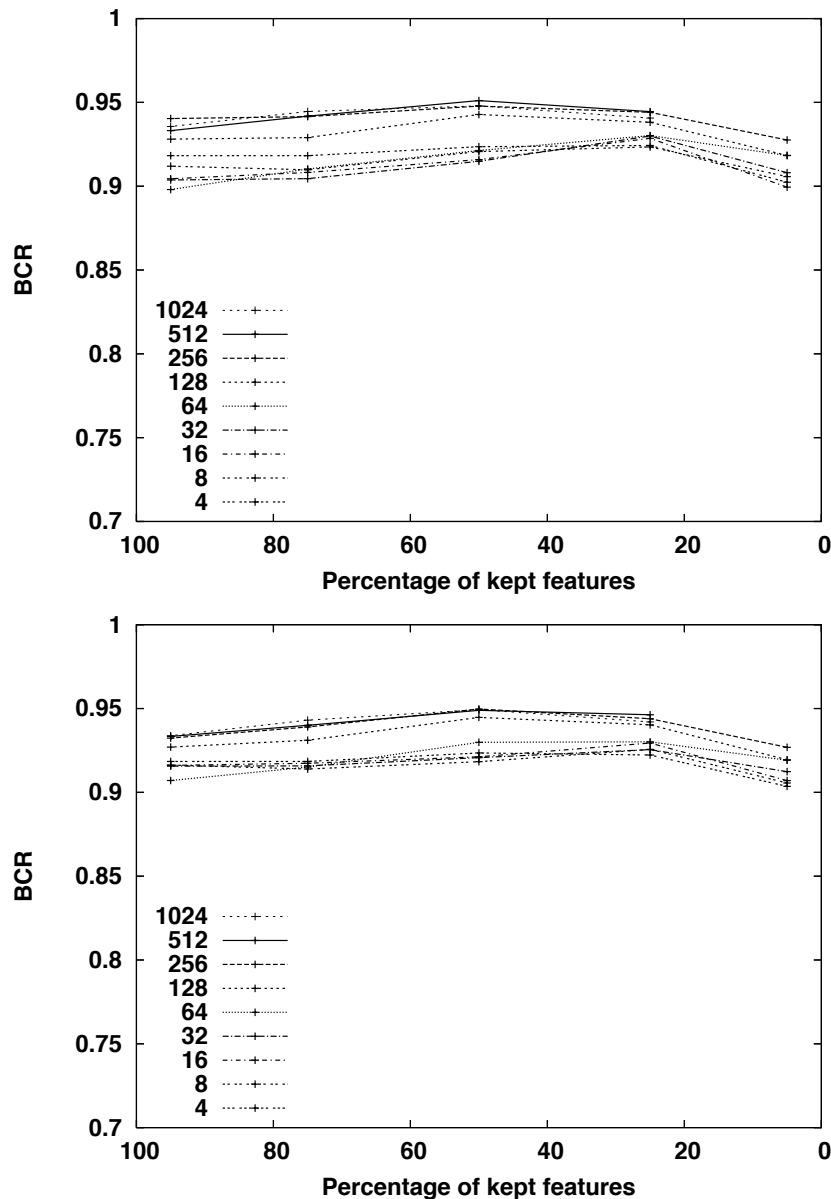


Figure 6.3 Non-specific filtering on PRO based on features variance. Top: t-Test used as a second selection method. Bottom: Golub index used as a second selection method

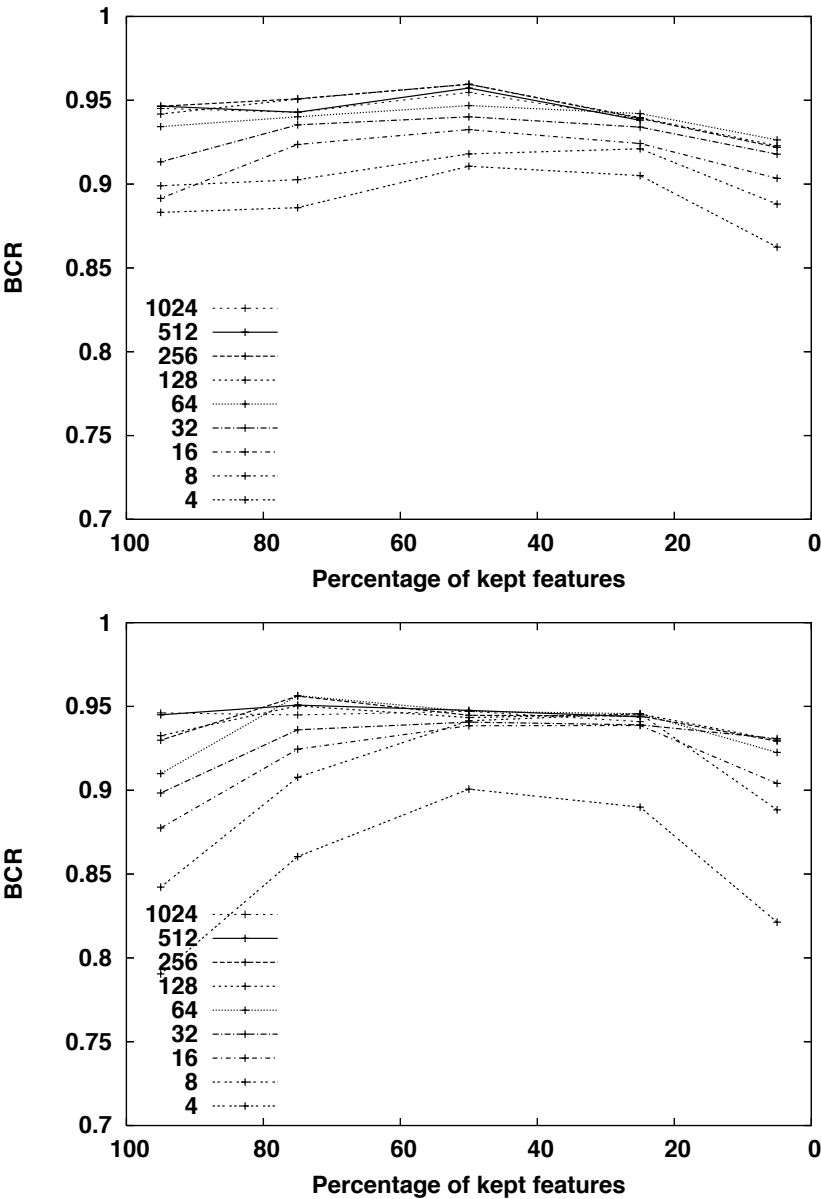


Figure 6.4 Non-specific filtering on PRO based on features variance. Top: RFE used as a second selection method. Bottom: L_2 -AROM used as a second selection method

Part III

Inductive Biases for High-Dimensional Learning

Ensemble Methods for Feature Selection

7

This chapter introduces the first main contribution of this thesis, consisting of an ensemble method for feature selection on high dimensional data with few samples. Ensemble methods have originally been developed to enhance classification performance (Diettrich, 2000). The general idea of that family of techniques consists in combining lots of different models in a global, likely more robust model. The different models are typically built on different sub-samplings of the original data set. Ensemble method ideas have recently been ported to high dimensional data with few samples (Long and Vega, 2003; Dettling, 2004) and in particular to feature selection (Saeys *et al.*, 2008). Loscalzo *et al.* (2009) propose an approach which is a consensus-version of the DRAGS algorithm for feature selection presented in Chapter 3 (Yu *et al.*, 2008). The notion of ensemble is there not applied to selection itself, but to robustly build groups of features before a group-selection step is applied. This approach suffers from a computational complexity depending on p^2 but shows encouraging results in favor of ensemble methods.

Different feature selection techniques may result in different rankings of the features, inducing stability issues, as detailed in Chapter 4. The same feature selection technique may produce drastically different results depending on the chosen setting of the parameters of the method. Most of the currently available genomic datasets fall in the small- n -big- p setting, which may lead to even more variation in the final ranking of the features. In the particular context of genomic data, a stable feature selection technique is desirable. Selection of relevant genes for a given pathology on different sub-samplings of the patients should produce nearly the same results since the biological process generating the data is assumed to be largely common for all patients,

at least without confounding factors.

Surprisingly, the analysis of the stability or robustness of biomarker selection techniques is only a topic of recent interest, and has not yet made it into the mainstream methodology for biomarker discovery. Therefore, this chapter also proposes a general experimental setup for stability analysis that can be easily included in any biomarker identification pipeline, based on elements introduced in previous chapters. This experimental setup is reused in the subsequent chapters.

The ensemble method for feature selection developed in this chapter is shown to increase the stability of the final set of selected biomarkers, using the recently introduced concept of ensemble feature selection (Saeys *et al.*, 2008). The technique is based on linear support vector machines classifiers (Chapter 2). These classifiers are interesting because the number of parameters to be estimated essentially depends on the number of samples rather than the number of features, which is particularly relevant with very small sample-to-feature ratios. Such models also offer state-of-the-art classification performance on a wide range of applications (Schölkopf and Smola, 2001). Moreover, SVMs have been extended to form embedded feature selection methods. A prominent approach among them is the RFE selection method (Chapter 3). A linear SVM is a classification model for which the influence of each dimension, here a specific gene, is explicitly available. RFE precisely uses this property to remove the least important features and iteratively re-estimates a classifier on the remaining features.

Experiments show that the robustness of SVMs for biomarker discovery can be substantially increased by using ensemble feature selection techniques, while at the same time improving upon classification performances. The proposed methodology is evaluated on four microarray data sets showing increases of up to almost 30% in robustness of the selected biomarkers, along with an improvement of about 15% in classification performance. The stability improvement with ensemble methods is particularly noticeable for small signature sizes (a few tens of genes), which is most relevant for the design of a diagnosis or prognosis model from a gene signature. The relative performances of various ways of combining the individual signatures are also rigorously assessed.

Section 7.1 details how the Recursive Feature Elimination technique can be used to produce a ranking over the full set of original features. Section 7.2 really introduces the ensemble method for feature selection, based on rankings produced by RFE on many subsamplings of the data. Section 7.3 proceeds to experiments on four cancer diagnosis microarray datasets showing the benefits of the proposed method in terms of classification performances and stability. The chapter ends in Section 7.4 with conclusions and ideas for further developments.

Table 7.1 Publications related to this chapter

Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., and Saeys, Y. (2009). Robust biomarker identification for cancer diagnosis using ensemble feature selection methods. In *Machine Learning in Systems Biology*

Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., and Saeys, Y. (2010). Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics (Oxford, England)*, **26**(3), 392–398

Louahed, J., Gaulis, S., Helleputte, T., Dupont, P., Gruselle, O., Spatz, A., Kruit, W., Dreno, B., Lehmann, F., and Brichard, V. (2008). Clinical response to the mage-3 immunotherapeutic in metastatic melanoma patients is associated with a specific gene profile present prior to treatment. In *33th European Society for Medical Oncology (ESMO) Congress*

Dupont, P., Gaulis, S., Helleputte, T. (Inventors), GlaxoSmithKline Biologicals S.A., Dupont, P., Gaulis, S., and Helleputte, T. (Applicants) (2010). Method for classifying a cancer patient as responder or non-responder to immunotherapy. Patent, WO/2010/029174

7.1 Selection within SVM

The reference classifier in this chapter is a linear SVM. SVMs are known to scale well to high-dimensional spaces, and have shown state-of-the-art performance in many problems in computational biology (Ben-Hur et al., 2008). Furthermore, a linear SVM offers the additional advantage that it contains an embedded capability for feature selection. The recursive feature

elimination algorithm (RFE) exploits this capability.

In the RFE algorithm, a recursive procedure is started that adopts a backward elimination strategy to iteratively remove features. Starting from the full feature set, a linear SVM is estimated from the training samples and features are sorted according to the absolute value of their weight in the hyperplane. Subsequently the least important features are eliminated and a linear SVM is re-estimated on the same samples but restricted to the remaining set of features. This process is iterated until all features have been removed or a desired number of features is reached. In this original formulation, RFE outputs a set of selected features. It is however possible to make RFE produce a ranking on the full feature set. This is explained next.

An internal parameter of the RFE method is the fraction E of features to eliminate at each step, which greatly influences the computational complexity of the method. Decreasing E increases the computational cost since less features are dropped at each iteration but possibly offers a more refined selection. In the experiments of this chapter, we choose to drop $E = 20\%$ features at each iteration by default. An additional sensitivity analysis is reported in Section 7.3.4 to check the precise influence of this parameter. When setting $E = 100\%$, RFE reduces to a single SVM estimation which ranks all the features in one step. When setting $E < 100\%$, the overall ranking of features is constructed iteratively from worst to best features. Whenever features are removed, they are sorted according to their absolute weight value in the hyperplane at the iteration of removal and added at the top of the current ranking.

Much simpler selection methods do exist (Guyon and Elisseeff, 2003) but RFE is chosen here as a baseline because it is known to provide state-of-the-art classification performance and was originally applied precisely in the context of gene selection for cancer classification. Besides, like most feature selection methods embedded with a classifier estimation, RFE is intrinsically multivariate in the sense that it evaluates the relevance of several features considered jointly. Multivariate approaches are thus more refined from a data analysis viewpoint and also biologically more relevant, because genes are known to interact in many ways and are often co-regulated.

Another internal parameter of the SVM learning procedure is the so-called

regularization parameter C . An SVM aims at classifying correctly training samples with a certain margin, which is not always possible when the training samples cannot be perfectly separated in two classes by a hyperplane. The C parameter precisely controls how strong margin errors (samples which cannot be classified with the prescribed margin) are penalized. We use the simple default value $C = 1$ in our reported experiments. We also investigated how to optimize C using a grid search with an internal 5-fold cross-validation on the training part but the additional computational complexity only offered marginal performance gains.

7.2 Ensemble Feature Selection

In the following, we introduce the ensemble feature selection approach which relies on different sub-samplings of the original data to build different signatures. We detail two different aggregation methods to build a consensus from the various signatures.

To increase the stability of feature selection algorithms, we further elaborate on the recently introduced concept of ensemble feature selection techniques (Saeys *et al.*, 2008). Ensemble feature selection techniques use an idea similar to ensemble learning for classification (Diettrich, 2000): in a first step, a number of different feature selectors are used, and in a final phase the output of these separate selectors is aggregated and returned as the final (ensemble) result. We focus on the analysis of ensemble feature selection techniques using linear SVMs and RFE as the feature selection mechanism.

Starting from a particular training set our aim is now to generate a diverse set of RFE feature selections. To generate diversity in the selection, RFE is run on different training sub-samples, because the RFE procedure is deterministic. To this end, we make use of the bootstrapping method, a well-established technique in statistics to reduce variance (Chapter 5). By drawing (with replacement) different bootstrap samples of the training data, we can apply RFE to each of these bootstrap samples, and thus obtain a diverse set of feature rankings. More formally, we take an ensemble EFS consisting of t feature selectors, $\text{EFS} = \{F_1, F_2, \dots, F_t\}$, then we assume each F_i provides a feature ranking $\mathbf{r}_i = (r_i^1, \dots, r_i^p)$, where r_i^j denotes the rank of feature j in bootstrap i . The best feature is assigned rank 1, and the worst one rank p .

To aggregate the different rankings, obtained by bootstrapping the training data, into a final signature we propose two aggregation schemes. These aggregation schemes differ in the way the aggregation (sum) of the individual rankings is calculated. A general formulation for the ensemble ranking $\mathbf{r}$, obtained by summing the ranks over all bootstrap samples is as follows:

$$\mathbf{r} = (\sum_{i=1}^t w_i r_i^1, \dots, \sum_{i=1}^t w_i r_i^p), \quad (7.1)$$

where w_i denotes a bootstrap dependent weight.

7.2.1 Complete Linear Aggregation (CLA)

This method uses the complete ranking of all the features to create the ensemble result. The ensemble ranking $\mathbf{r}$ is then obtained by just summing the ranks over all bootstrap samples. In the general formulation of Equation (7.1), this amounts to setting all weights w_i equal to 1. To select the final set of features for a signature of size s , the s features with the lowest summed rank are selected from $\mathbf{r}$.

7.2.2 Complete Weighted Linear Aggregation (CWA)

This method is a variation on the previous method. It no longer just sums all scores, but additionally weighs the scores of each bootstrap ranking. The weight assigned to a bootstrap ranking is the AUC obtained by a linear SVM, trained on the bootstrap samples and evaluated on the out-of-bag (OOB) samples. In Equation (7.1), this amounts to setting $w_i = \text{AUC}_{\text{OOB}_i}$.

7.3 Experiments and Results

Our experimental data consists of 4 cancer diagnosis microarrays data sets which are first briefly described. Next, we propose a general evaluation protocol to evaluate both the stability and the classification performance of gene selection methods.

7.3.1 Microarray Datasets

The main characteristics of the microarray datasets used in this chapter are summarized in Table 7.2. The samples/dimensions ratio is very low, which

is typical of microarray data. Those datasets are more extensively described in Appendix D. The original studies on these data sets stress the limitations of univariate methods, which look at the influence of each gene individually, and the issue of signature robustness. They trigger the interest of considering a multivariate technique, like RFE, and of trying to improve its robustness.

Table 7.2 Datasets used for ensemble feature selection validation. Columns respectively show the dataset name, the number of samples for each class and the number of features.

Name	Class 1	Class 2	# Features
LEU	47	25	7,129
COL	40	22	2,000
LYM	22	23	4,026
PRO	52	50	6,033

7.3.2 Biomarker Evaluation Protocol

In order to analyze the stability of a biomarker selection algorithm, we propose to generate slight variations of the original data set, and compare the outcome of the marker selection algorithm across these different variations. The rationale behind this is that, for a stable marker selection algorithm, small changes in the training set should not yield big changes in the set of finally selected markers. This is consistent with what a domain expert would expect from a marker selection algorithm: adding or deleting a few samples should not drastically modify the top-ranked markers identified by the algorithm.

To implement the strategy of generating slight variations of the data set, a sub-sampling approach is proposed, based on the N-Resamplings protocol detailed in Chapter 5. A large number J of datasets, for example $J = 500$, can be generated by subsampling the original data set without replacement. As microarray datasets generally contain few samples, we suggest to generate subsamplings containing 90% of the samples of the original data set.

We argue that stability matters, but stability alone is not a good quality measure. As discussed in Chapter 4, stability needs to be assessed together with classification performance. We propose to use the same subsamplings - each containing 90% of the original data set - as training sets to select features

and estimate the performance of a classifier. The remaining 10% of the data can be used each time as an independent test set to evaluate classification performance without falling under the selection bias. Since we are considering typically $J = 500$ independent partitionings into 90% training and 10% test, we reduce the risk of over-optimistic results of traditional cross-validation experiments on small sample domains (Braga-Neto and Dougherty, 2004).

According to Chapter 6, the robust IQR feature normalization procedure of Section 6.1.3 is included at each of the J iterations of the protocol. The normalization parameters for each feature are always estimated from the training samples only and applied subsequently to the validation samples, avoiding the normalization bias. After the normalization step, feature selection is performed on each of the J subsamplings, and a marker set, a signature, of a given size is selected. Here, following (Kalousis *et al.*, 2007), we take a similarity-based approach where feature stability is measured by comparing the signatures selected on each of the J subsamplings. The more similar all signatures are, the higher the stability measure will be. The overall stability $\mathcal{K}(\{\mathbf{S}_1, \dots, \mathbf{S}_J\})$ measured by the Kuncheva index in Equation (4.1) can then be defined as the average over all pairwise similarity comparisons between all signatures on the J subsamplings, as in Equation (4.2). In the sequel, the overall stability $\mathcal{K}(\{\mathbf{S}_1, \dots, \mathbf{S}_J\})$ is simply denoted $\mathcal{K}$.

To compare the classification performance of different methods, we use the area under the Receiver Operator Characteristic curve (ROC curve, Chapter 4), further abbreviated as AUC. This area is defined by a function of sensitivity and specificity, frequently used in clinical settings.

7.3.3 Ensemble Feature selection Methods Improve Classification Performance and Biomarker Stability

We report here the experimental evaluations on the four cancer diagnosis microarray data sets considered in this chapter. The reference method serving as a baseline is the traditional RFE approach used as a ranking technique. It already offers state-of-the-art classification performance and a multivariate selection mechanism for evaluating the combined relevance of sets of markers. The results show that the stability of biomarker selection with RFE, as well as the classification performance can be significantly improved with the proposed ensemble methodology. The relative performance of the different ways of building a consensus from distinct individual signatures is carefully assessed. Results on the variance of the pairwise stability between two signatures are also discussed, as well as the sensitivity to the setting of internal parameters of the marker selection methods.

As a first experiment, we compare the newly proposed ensemble RFE method to the traditional RFE setting, analyzing the stability and classification performance for each of the four cancer datasets. Figure 7.1 displays the results for the dataset PRO, using a default configuration where the number of bootstrap rounds to create the ensemble is set to 40, and RFE was applied eliminating $E = 20\%$ of the features at each iteration. Results for the other datasets can be found in Appendix F (Figures F.1 and F.2). The default configuration was based on earlier work (Saeys *et al.*, 2008), where it was shown that these parameter settings yielded a good default. Robustness of the selected signatures (marker sets) was measured by the Kuncheva index ($\mathcal{K}$), and the AUC was used to measure the classification performance.

It can be observed that the ensemble methods CLA and CWA clearly improve upon the baseline, both in terms of stability and classification performance. Moreover, the gains increase as signature sizes get smaller. In three out of four datasets, the ensemble methods even perform better with fewer than with all features (see Appendix F), thus showing that ensemble methods are better capable of eliminating noisy and irrelevant dimensions.

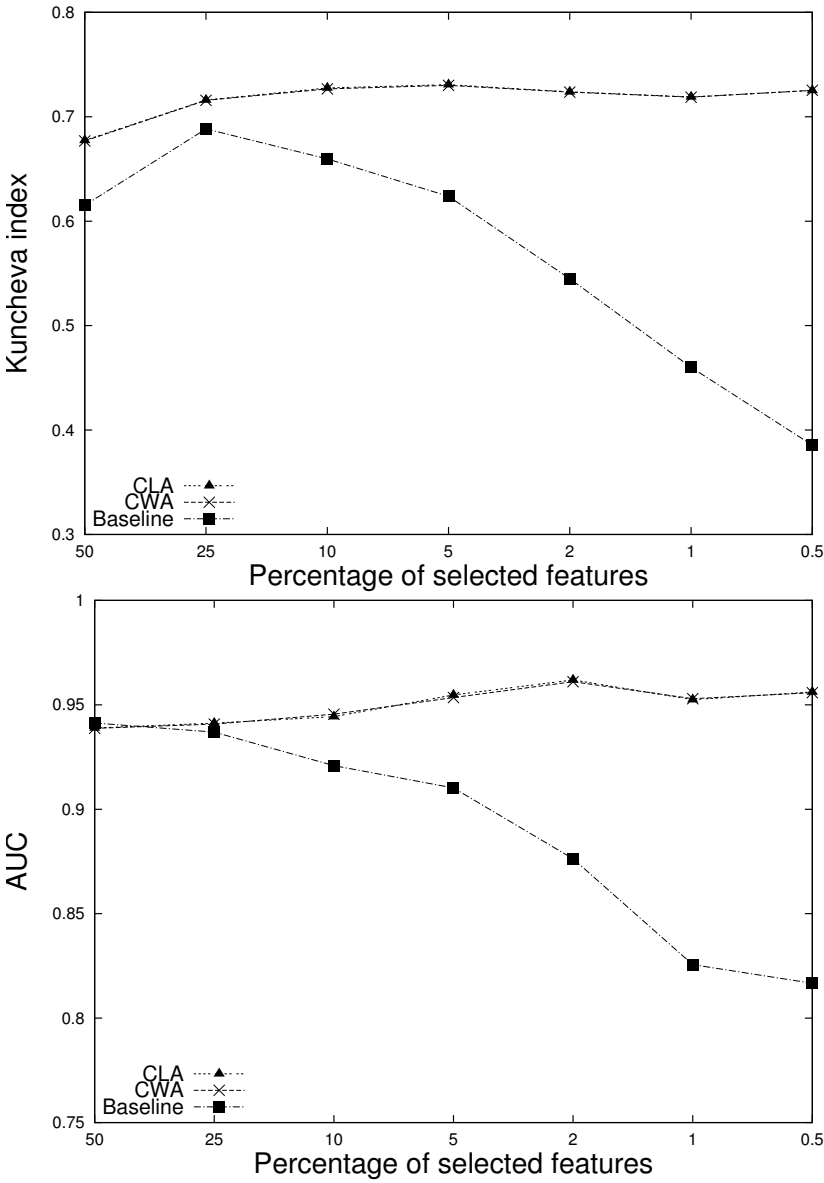


Figure 7.1 Stability (top) and classification performances (bottom) of the baseline method (original RFE) and the ensemble methods for PRO. We used 40 bootstraps and used RFE with $E = 20\%$.

In the above analysis we reported average classification and stability performances over 500 distinct subsamplings from each original data set. The pro-

posed stability metric is an average over all pairwise comparisons of two signatures built using different subsamplings. Figure 7.2 details this analysis by reporting the histogram of stability values across pairwise comparisons. We observed that the ensemble CLA method improves the average stability over the baseline, since there is a systematic shift of the histograms with respect to the baseline, with no influence on the variance of the stability since the respective histograms have the same spread.

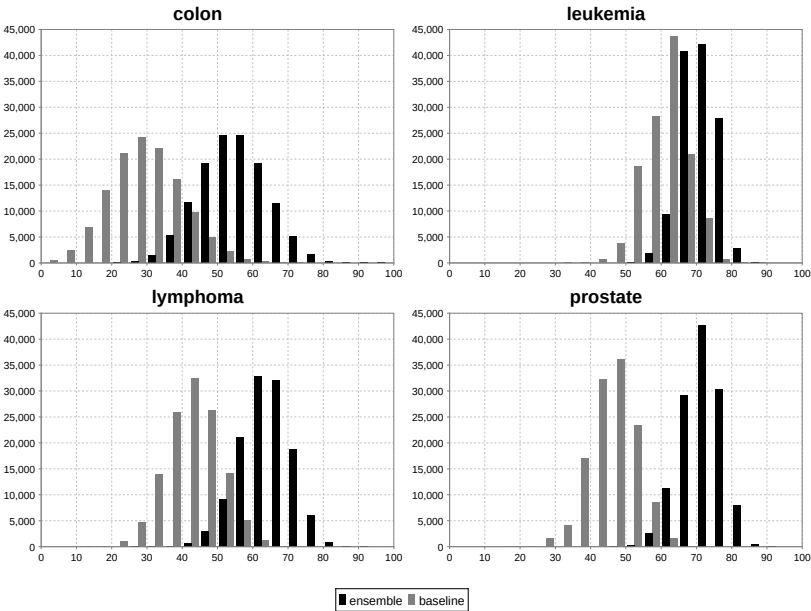


Figure 7.2 Distribution plots of the pairwise stabilities for the four data sets. We used 40 bootstraps, eliminated $E = 20\%$ features at each iteration of RFE, used a signature size of 1% and chose the CLA aggregation model. We used fixed-width bins of 5%.

7.3.4 Sensitivity Analysis with Respect to Internal Parameters

Up to now, performance metrics were computed using default parameters which were proven useful in the preliminary study. We further assess the influence of two internal parameters that may influence the results: the number of bootstrap rounds used by the ensemble methods and the fraction of features discarded at each iteration of RFE (baseline or ensemble versions).

Sensitivity with Regard to the Number of Bootstrap Samples

Figure 7.3 shows the stability and classification performance with regard to the number of bootstrap samples used to construct the ensemble. As the number of bootstraps used increases, so does the stability of the ensemble. The default value of 40 bootstrap rounds is suitable since increasing it to 60 rounds only marginally increases the stability while requiring a 50% increase of the computational effort. On the other hand, the number of bootstraps does not seem to have an effect on the classification performance. Results for the other data sets can be found in Appendix F (Figures F.3 and F.4).

Sensitivity with Regard to the Number of Features to Eliminate

The number of features to eliminate does not have an effect on the ensemble methods, neither regarding stability nor classification performance (Figure 7.4). For the baseline model on the other hand, ranking the features in one step, i.e. removing 100%, yields the best results, both in terms of stability and classification performance. While the effect is only marginal regarding classification performance, the effect has a clear impact on the stability: a single SVM classifier used as a feature ranker largely outperforms the RFE variants that eliminate 20% or 50% of the features. Ensemble methods nevertheless show better performance both in terms of stability and classification, as compared to a single SVM run.

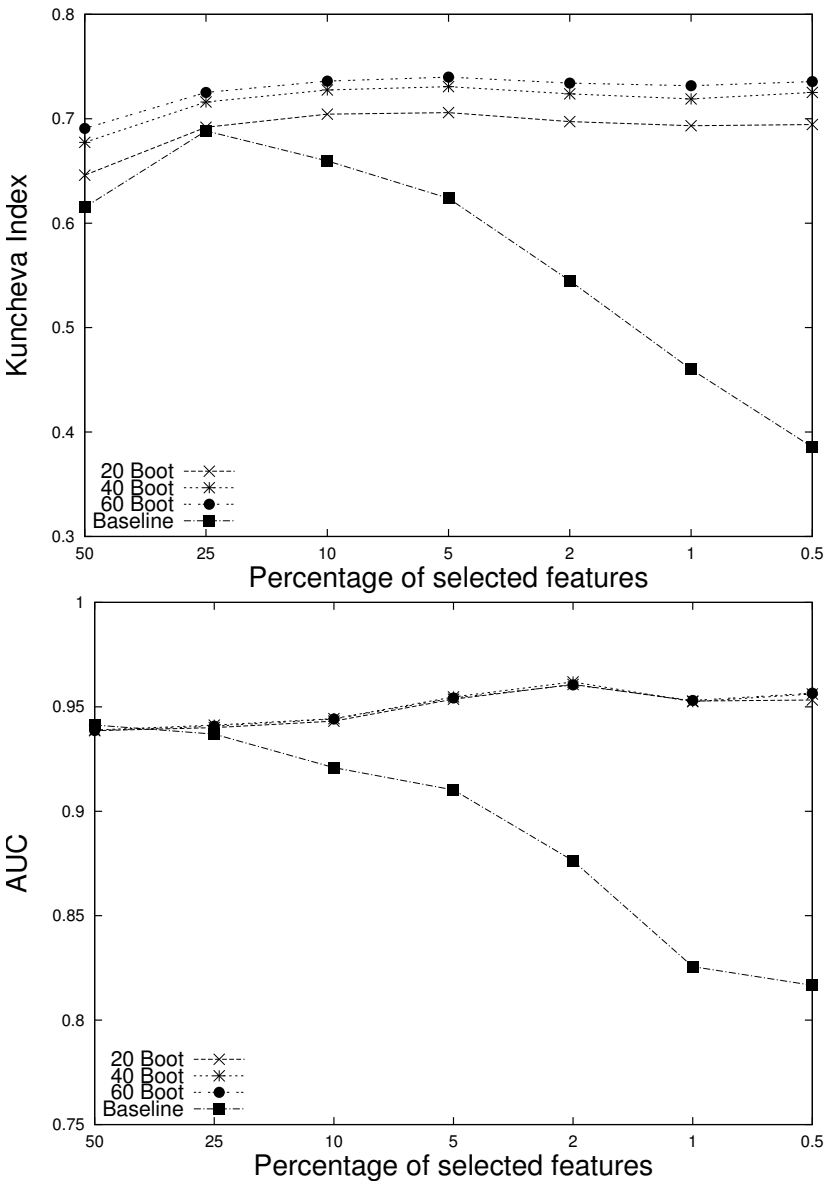


Figure 7.3 Stability (top) and classification performances (bottom) for several numbers of bootstrap rounds for the construction of an ensemble signature for PRO. We used the CLA aggregation method and eliminated 20% of the features at each iteration of RFE. The baseline is the original RFE on the full training sets without bootstrap.

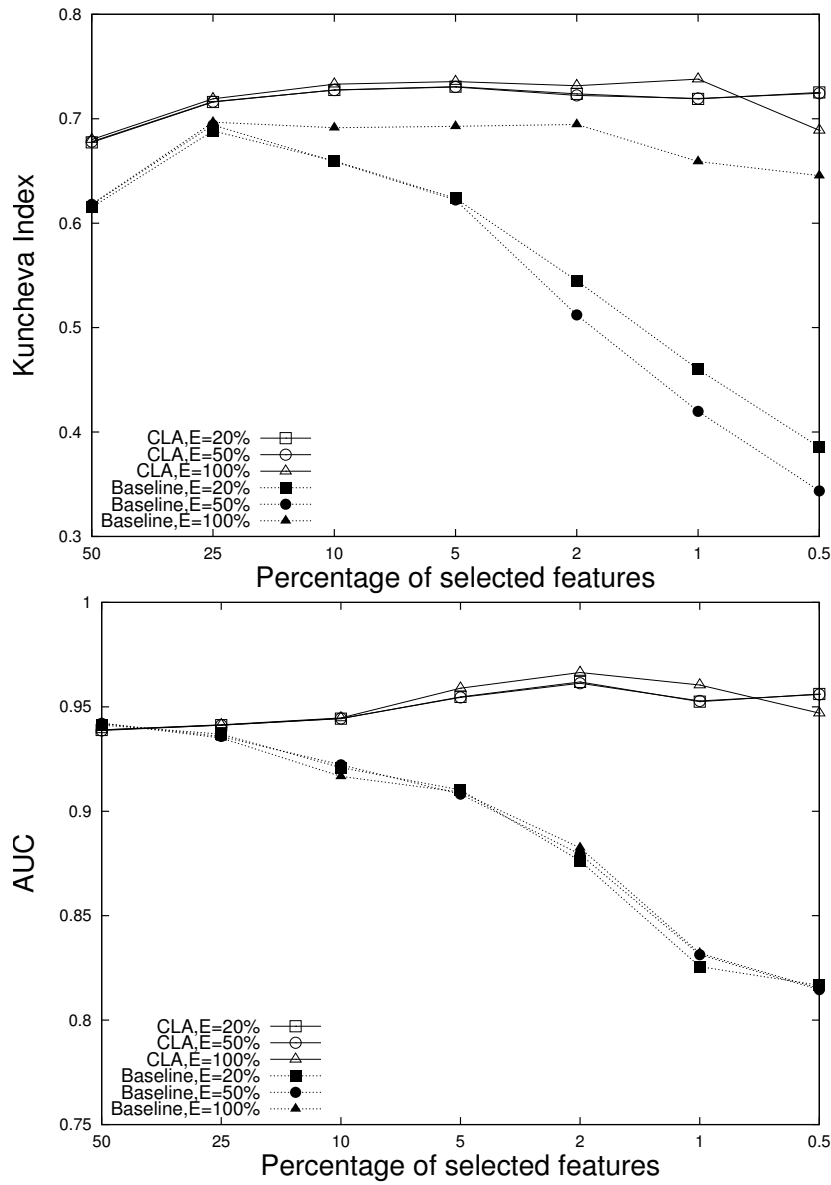


Figure 7.4 Stability (top) and classification performance (bottom) with regard to a varying number of features to eliminate during RFE. Results represent the CLA aggregation method and were constructed using 40 bootstrap samples.

7.4 Discussion

The present chapter discusses the robustness of biomarker identification with a concrete focus on microarray experiments on four cancer diagnosis data sets. We stress the importance of the markers stability with respect to sample variation. Such a stability is desirable both for the reproducibility and the easiness of the biological validation of the extracted signature. Our underlying hypothesis is that small changes in the data sampling should not yield dramatic changes in the set of finally selected markers. Stability alone is however not a good quality criterion since it is straightforward to increase stability by always considering some fixed set of markers. The resulting predictive model would however likely be poor at classifying new samples.

The first contribution is a set of ensemble feature selection methods improving biomarker stability and classification performance. RFE is used as a baseline method because it already offers state-of-the-art classification performance and it is intrinsically multivariate, that is measuring the joined relevance of sets of markers. When decreasing the number of selected features, the stability of RFE tends to degrade while ensemble methods offer significantly better stability. Stability and classification performance are particularly improved for signature sizes as small as 0.5% of the initial feature set. This is particularly convenient since it corresponds to sizes of practical interest (a few tens of genes), for instance, for the design of a diagnosis model.

The second contribution is an experimental methodology to assess the stability of biomarker lists combined with the predictive performance of classification models built from them. Such an experimental protocol repetitively considers some samples both to select markers and to estimate classifiers from independent samples used to estimate classification performance. In particular it avoids a common optimistic bias of the selection process as well as the normalization bias ([Ambroise and McLachlan, 2002](#)). This protocol also relies on the area under the ROC curve (AUC) and the Kuncheva stability index ($\mathcal{K}$). AUC is a more convenient metric than classification accuracy to evaluate the predictive performance on data sets with unbalanced class proportions, a common situation for microarray experiments. The balanced classification rate could be considered as well. $\mathcal{K}$ measures to which extent several signatures, typically extracted from different sampling of the data, share features in common while including a correction of such a common selection purely by chance.

A short comment on the similarities and differences between the results presented in this chapter and the results presented in the closely related work of [Saeys *et al.* \(2008\)](#) can be made. Most importantly, the messages of both works are qualitatively coherent: ensemble methods for feature selection in a context of classification provide most of the time a gain in signature stability and in classification performance. Quantitatively, conclusions are a bit stronger in the present work. This can be explained by several factors. First, the work of [Saeys *et al.* \(2008\)](#) uses accuracy as a classification performance measure, while the current work uses AUC. The considered datasets are unbalanced and a high accuracy might thus simply reflect the fact that the classifier deals well with the majority class. The impact of poor results on the minority class are then just dwarfed in the average. AUC is in that respect a finer tool of analysis, because it gives equivalent importance to both classes, just like BCR does. The number of resamplings (10 in ([Saeys *et al.*, 2008](#)) vs. 500 here) might also clearly affect the results. It is likely that redoing all the experiment of ([Saeys *et al.*, 2008](#)) with a different set of 10 splits would produce numerically different results, while it is less likely for a 500-resampling scheme. Finally, we stress that the most significant gain provided by ensemble methods for selection occurs for signatures not larger than 0.5% of the feature set, a range of size not covered by ([Saeys *et al.*, 2008](#)).

The present work proposes ensemble methods to improve biomarker stability. Since this is an important issue, it would be worth developing additional alternatives to further increase stability. In particular, the ensemble method presented here increases stability by building a consensus signature a posteriori. This requires to compute a lot of different signatures. For computational matters, it would be interesting to favor stability a priori. Several new techniques pursuing such objectives are considered in the following chapters either by incorporating prior knowledge in the feature selection or by estimating models across several related datasets. The following chapters reuse the experimental protocol detailed here.

Partially Supervised Feature Selection

8

This chapter introduces another technique improving feature selection quality in a binary classification context. It works by incorporating prior knowledge about some of the available features to guide the selection.

Semi-supervised learning for classification deals with problems for which only a fraction of the learning examples have known class labels ([Chapelle et al., 2006](#)), and semi-supervised feature selection methods have been recently proposed ([Zhao and Liu, 2007](#); [Cheng et al., 2008](#)). We use here a different kind of partial supervision, namely on the *dimensions* of a feature selection procedure. For instance in the case of microarray data classification, a molecular biologist may know or guess that some genes are likely to be more discriminant. This knowledge is usually only partial, as the purpose of biomarker selection is often to discover new gene signatures, or even inaccurate, as gene expression may be influenced by several factors not related with the outcome. The technique presented in this chapter makes use of such prior knowledge to guide feature selection while letting the final selection depart from it if necessary to optimize the classification objective.

Our method extends the embedded AROM methods (Chapter 3), by adding a partial supervision on the dimensions to be selected, in a simple yet efficient way.

We show in the present study that the use of prior knowledge on relevant genes effectively induces a large gain in stability with improved classification performances in most cases.

The rest of the chapter is organized as follows. Section 8.1 details how to extend the AROM methods with partial supervision on the selected features. Practical experiments on various microarray datasets are reported in Section 8.2. Conclusions and perspectives are discussed in Section 8.3.

Table 8.1 Publications related to this chapter

Helleputte, T. and Dupont, P. (2009c). Partially supervised feature selection with regularized linear models. In *International Conference on Machine Learning (ICML)*

8.1 Partially Supervised AROM

Whenever some prior knowledge on the relative importance of each feature is available, the L_1 -AROM objective can be modified by adding a prior relevance vector $\beta = [\beta_1, \dots, \beta_p]^t$ defined over the input dimensions. Let $\beta_j \geq 1$ denote the relative prior relevance of the j^{th} feature, the higher its value the more relevant the corresponding feature is *a priori* assumed. If no information is available about a given feature prior relevance, it is fixed to the default value $\beta_j = 1$. The original L_1 -AROM optimization problem in Equation (3.11) is modified to penalize less the dimensions which are assumed *a priori* more relevant:

$$\min_{\mathbf{w}} \sum_{j=1}^p \frac{1}{\beta_j} \ln(|w_j| + \epsilon), \quad (8.1)$$

subject to

$$y_i \langle \mathbf{x}_i, \mathbf{w} \rangle \geq 1 \quad \forall i \in \{1, \dots, n\}. \quad (8.2)$$

This is the PS- L_1 -AROM problem. Note that, just like L_1 -AROM, it can fit in the loss & regularizer pattern where the loss would be the hinge loss, and the regularizer would be now a modified version of the log- L_1 -approximation of the zero-pseudo-norm in Table 2.2. The prior knowledge thus plays on the regularization. Figure 8.1 illustrates the original L_1 -AROM objective and what happens if it is modified by introducing prior relevance on dimensions, as in Problem (8.1). The objective function is nearly flat everywhere but the gradient is now even smaller along a dimension corresponding to a larger β_j . Following the same line of reasoning as in (Weston *et al.*, 2003), we derive below an iterative algorithm to solve Problem (8.1). An equivalent problem is

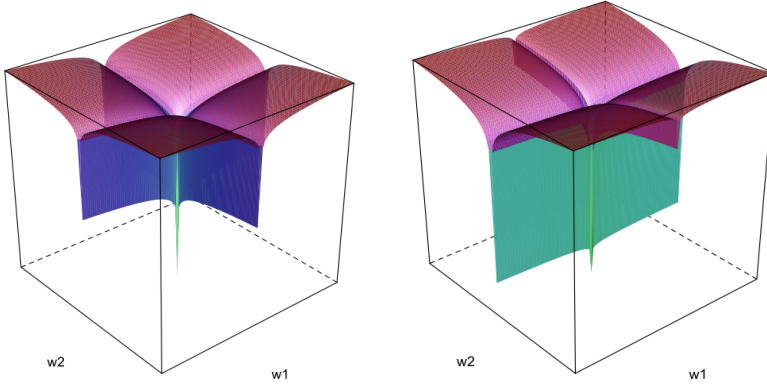


Figure 8.1 2D-representation of the zero-pseudo-norm approximation by $\sum_j \frac{1}{\beta_j} \ln |w_j|$. Left: without prior relevance ($\beta = [1, 1]^t$). Right: with prior relevance ($\beta = [5, 1]^t$).

obtained after introducing auxiliary variables v_j 's:

$$\min_{\mathbf{v}} \sum_{j=1}^p \frac{1}{\beta_j} \ln(v_j + \epsilon), \quad (8.3)$$

subject to

$$\begin{cases} y_i \langle \mathbf{x}_i, \mathbf{w} \rangle \geq 1 \quad \forall i \in \{1, \dots, n\}, \\ v_j \geq w_j \quad \forall j \in \{1, \dots, p\}, \\ v_j \geq -w_j \quad \forall j \in \{1, \dots, p\}. \end{cases} \quad (8.4)$$

Next, Problem (8.3) is solved using an iterative constrained gradient descent technique due to (Frank and Wolfe, 1956):

1. Find the steepest descent direction of the objective function that is consistent with the constraints:

$$\min_{\mathbf{v}, \mathbf{w}} \nabla h(\mathbf{v}_k) \cdot (\mathbf{v} - \mathbf{v}_k), \quad (8.5)$$

subject to

$$\begin{cases} y_i \langle \mathbf{x}_i, \mathbf{w} \rangle \geq 1, \\ v_j \geq w_j, \\ v_j \geq -w_j, \end{cases} \quad (8.6)$$

where $h(\mathbf{v}_k)$ is the value of the objective function at step k . Let $(\bar{\mathbf{v}}, \bar{\mathbf{w}})$ be the optimum value of this problem.

2. Optimize along that steepest descent direction: compute λ such that $h(\mathbf{v}_k + \lambda(\bar{\mathbf{v}} - \mathbf{v}_k))$ is minimal.

At step k , the objective function is approximately given by

$$h(\mathbf{v}_k) \approx \sum_{j=1}^p \frac{1}{\beta_j} \ln(v_{kj}). \quad (8.7)$$

It follows that:

$$\frac{\partial h(\cdot)}{\partial v_{kj}} = \frac{1}{\beta_j v_{kj}}. \quad (8.8)$$

Since the steepest descent is given by:

$$\nabla h(\mathbf{v}_k) \cdot (\mathbf{v} - \mathbf{v}_k) = \sum_{j=1}^p \frac{v_j - v_{kj}}{\beta_j v_{kj}}, \quad (8.9)$$

Problem (8.5) becomes

$$\min_{\mathbf{v}, \mathbf{w}} \sum_{j=1}^p \frac{v_j - v_{kj}}{\beta_j v_{kj}}, \quad (8.10)$$

subject to

$$\begin{cases} y_i \langle \mathbf{x}_i, \mathbf{w} \rangle \geq 1, \\ v_j \geq w_j, \\ v_j \geq -w_j. \end{cases} \quad (8.11)$$

By introducing new variables $v'_j = \frac{v_j}{\beta_j v_{kj}}$, it can be rewritten as:

$$\min_{\mathbf{v}', \mathbf{w}} \sum_{j=1}^p v'_j - \sum_{j=1}^p \frac{1}{\beta_j} = \min_{\mathbf{v}', \mathbf{w}} \sum_{j=1}^p v'_j, \quad (8.12)$$

subject to

$$\begin{cases} y_i \langle \mathbf{x}_i, \mathbf{w} \rangle \geq 1, \\ v'_j \geq \frac{w_j}{\beta_j v_{kj}}, \\ v'_j \geq \frac{-w_j}{\beta_j v_{kj}} \end{cases} \quad (8.13)$$

By defining $w'_j = \frac{w_j}{\beta_j v_{kj}}$ and given that $|w_{kj}| = |v_{kj}|$, the two last constraints can be rewritten to obtain $w_j = w'_j w_{kj} \beta_j$. Hence Problem (8.12) can be reformulated as a 1-norm optimization with margin constraints on rescaled inputs:

$$\min_{\mathbf{w}'} \sum_{j=1}^p |w'_j|, \quad (8.14)$$

subject to

$$y_i \langle (\mathbf{x}_i * \mathbf{w}_k * \boldsymbol{\beta}), \mathbf{w}' \rangle \geq 1, \quad (8.15)$$

where $*$ denotes the component-wise product¹.

The second step of the iterative Franke and Wolfe's method aims at finding λ such that $h(\mathbf{w}_k + \lambda(\bar{\mathbf{w}} - \mathbf{w}_k))$ is minimal (with $\bar{\mathbf{w}}$ being the optimal solution to Problem (8.14)). Since h is a weighted positive sum of logarithms, it is concave. Consequently, at each iteration, λ is either 0, in which case $\mathbf{w}_{k+1} = \mathbf{w}_k$ and a local optimum is reached, or 1, in which case $\mathbf{w}_{k+1} = \bar{\mathbf{w}} * \mathbf{w}_k * \boldsymbol{\beta}$, and the process is iterated till convergence.

It is shown in Chapter 3 how L_1 -AROM can be approximated by L_2 -AROM. Similarly, Problem (8.14) can be approximated by replacing the 1-norm by the 2-norm. This formulation reduces to an iterative algorithm using hard-margin linear SVMs with rescaled margin constraints (a soft-margin variant is straightforward). The original L_2 -AROM method is obtained when $\beta_j = 1 \ \forall j \in \{1, \dots, p\}$, in other words, without prior preference between the input features.

The RFE approach proposed by Guyon *et al.* (2002) is an iterative procedure where a linear SVM is trained at each iteration and features corresponding to the smallest absolute weights are discarded. It has a different initial motivation but is algorithmically very similar to L_2 -AROM. RFE can be seen as an iterative thresholding of the vector $\mathbf{w}_k$, masking some features at each iteration: each dimension is either multiplied by 1 (kept) or 0 (discarded) resulting in a backward selection process. L_2 -AROM performs a smoother selection at each step. We show here how some prior knowledge $\boldsymbol{\beta}$ can weight the smooth selection mask $\mathbf{w}_k$.

Some *a priori* less relevant features may appear in the final solution to Problem (8.14), or its 2-norm approximation, since all β_j 's are strictly positive. This observation, confirmed in our practical experiments reported in Section 8.2 and in Chapter 12, illustrates why our feature selection procedure is only *partially* (and softly) supervised.

¹Slight abuse of notation: we assume previously absent $\beta_0 = w_{0k} = 1$ corresponding to the bias term.

8.2 Experiments and Results

We report here practical experiments with the feature selection method proposed in this chapter. These experiments are conducted with the partially supervised L_2 -AROM approach (PS- L_2 -AROM for short) because of its computational efficiency. This choice is also motivated by the results reported in (Weston *et al.*, 2003) which show that classification performances of the original L_1 -AROM and L_2 -AROM methods do not significantly differ while the computational time is in favor of the latter. This method is applied to four microarray data sets described in Section 8.2.1. The experiments reported in Sections 8.2.2 and 8.2.3 illustrate that partial supervision leads to an increased stability with improved classification performance in most cases. Comparative results with random supervision also show the soundness of the proposed approach.

8.2.1 Microarray Datasets

Table 8.2 summarizes the main characteristics of the four data sets used in the present chapter, namely the number of samples of each class and the initial dimension of the input space. Each dimension corresponds to the expression value of a particular gene. The samples/dimensions ratio is very low, which is typical of microarray data. Those datasets are more extensively described in Appendix D. In all cases, the classification tasks are binary.

Table 8.2 Datasets used for partially supervised feature selection validation. Columns respectively show the dataset name, the number of samples for each class and the number of features.

Name	Class 1	Class 2	# Features
DLB	57	20	7,129
LEU	47	25	7,129
PRO	52	50	6,033
COL	40	22	2,000

8.2.2 Partial Supervision from Prior Biological Knowledge

Shipp *et al.* (2002) mention two genes previously known as clinical markers to discriminate the two classes of the DLB dataset. Those two genes are Transferrin Receptor (TR) and Lactate Dehydrogenase A (LDHA).

Our first experiment with PS- L_2 -AROM favors those two dimensions to build a signature of 30 genes (the same signature size as in (Shipp *et al.*, 2002)). We define the prior relevance vector β as follows: $\beta_{j \in [\text{TR}, \text{LDHA}]} = 10$, $\beta_{j \notin [\text{TR}, \text{LDHA}]} = 1$. The relevance value for genes assumed more relevant is arbitrarily assigned to 10. Additional experiments reported in Chapter 9 illustrate that results presented in this section depend only marginally on this choice.

We report here comparative results with no prior preference between genes ($\beta_j = 1 \ \forall j \in \{1, \dots, p\}$, in which case PS- L_2 -AROM reduces to the original L_2 -AROM approach). This experiment is run on the whole DLB dataset (77 samples). Without prior preference (L_2 -AROM), LDHA and TR are not ranked within the top 30 genes selected as signature. In contrast, they correspond to the two largest components of the weight vector $\mathbf{w}$ with non uniform prior relevance (PS- L_2 -AROM). The number of genes differing between signatures generated with and without prior relevance is greater than just the number of favored genes. Only 6 genes are shared between both signatures. This illustrates the multivariate nature of the selection.

Shipp *et al.* (2002) report a leave-one-out (LOO) accuracy of 91% on the 77 samples with their 30 genes signature. Their classifier is a linear model with weighted voting, where the weights measure the correlation with class labels. Their evaluation does not look completely sound. First, it relies on accuracy while class priors are unequal. More importantly, it includes a selection bias as the signature is built on the whole dataset before evaluating several classifiers with LOO. With the same biased protocol, a linear SVM built on the 30 genes produced by L_2 -AROM (respectively PS- L_2 -AROM) has 93 % (resp. 92 %) LOO-accuracy. Such a protocol includes an optimistic performance bias (Ambroise and McLachlan, 2002). The additional experiments detailed below avoid such a bias and aim at evaluating both stability and classification performances.

We consider the same protocol as in previous chapter, where J independent sub-samplings without replacement of the DLB dataset with arbitrary splits into 90% training - 10% test are drawn (here, $J = 200$). The robust IQR normalization described in Chapter 6 is applied, with parameters computed only on the training part, and applied on the test part.

Figure 8.2 reports the average stability measured by the Kuncheva index and classification performances measured by the balanced classification rate (BCR) obtained with PS- L_2 -AROM and L_2 -AROM for several signature sizes². The RANDOM approach refers to PS- L_2 -AROM when the partial supervision relies on two genes picked at random instead of TR and LDHA. In practice, we perform in this case 10 independent random selection of genes and 20 random partition into 90% training and 10% test for a total of 200 independent runs on which average results are reported. We also report results obtained with the related approach RFE and the Golub's ratio (Chapter 3). For all methods, soft-margin linear SVMs are built on selected features on the training sets and evaluated on the test sets.

The comparison between L_2 -AROM and PS- L_2 -AROM shows that a partial supervision on only 2 genes improves drastically the stability of gene signatures with 64 or fewer genes. There is also an important gain in classification performance for very small signatures (≤ 8 genes). It is expected that those effects would be even stronger, or also observed for larger signatures, if additional biological knowledge were available to favor more genes (see Section 8.2.3). We also note that the stability of a 2-genes signature computed with PS- L_2 -AROM is significantly below 1 showing that the 2 genes *a priori* favored are not systematically present in the final signature. Partial supervision with randomly chosen genes increases the stability with respect to L_2 -AROM, because they are favored through PS- L_2 -AROM, but not at all the classification performance. This illustrates that, if the partial supervision is based on (likely) irrelevant dimensions, the PS- L_2 -AROM may depart from those but without improving prediction. RFE offers intermediate classification performance but a lower stability for small signatures, while Golub index filtering offers good classification results but a drop in stability when fewer than 64 genes are selected.

Golub *et al.* (1999) mention three genes, CD11c, CD33 and MB-1, as clinical markers to distinguish between AML and ALL leukemia subtypes. We repeat the same experiments on the related leukemia dataset LEU with $J = 200$ random splits in 90% training and 10% test. Figure 8.3 reports stability and classification performances. The conclusions are even stronger than those obtained on DLB, with substantial improvements both in stability and BCR for

²Refer to Chapter 4 for a discussion on those metrics

small signatures. Partial supervision with 3 randomly selected genes offers no benefit, neither in stability nor in BCR, as compared to no supervision. RFE performances are comparatively worse on this dataset both in stability and BCR, while Golub index offers BCR results equivalent to PS- L_2 -AROM with a drop in stability for small signatures.

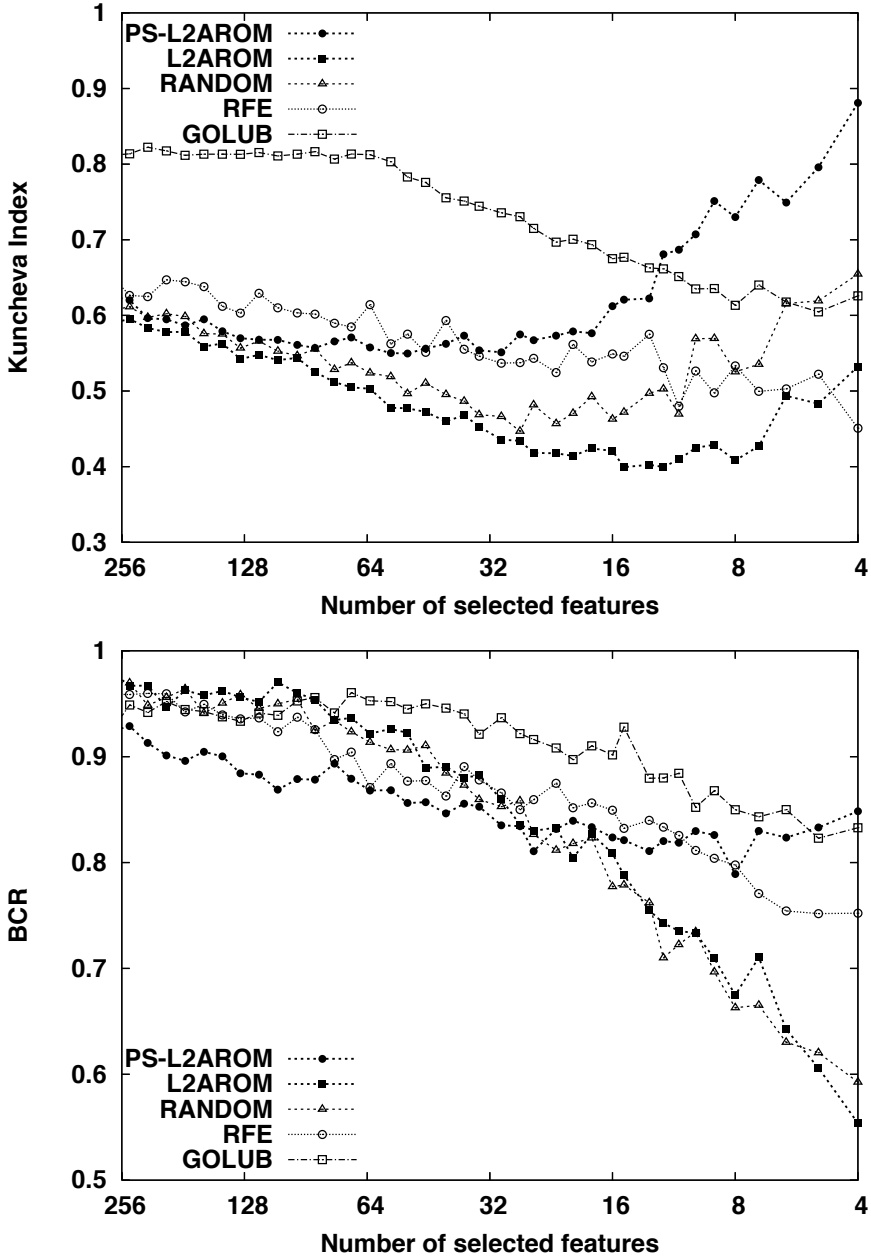


Figure 8.2 Signature stability (Kuncheva index) and classification performance (BCR) of PS- L_2 -AROM (with $\beta_{j \in [TR, LHDA]} = 10$), L_2 -AROM, RANDOM, RFE and Golub index filtering on DLB. Average results over 200 runs (90 % training - 10% test).

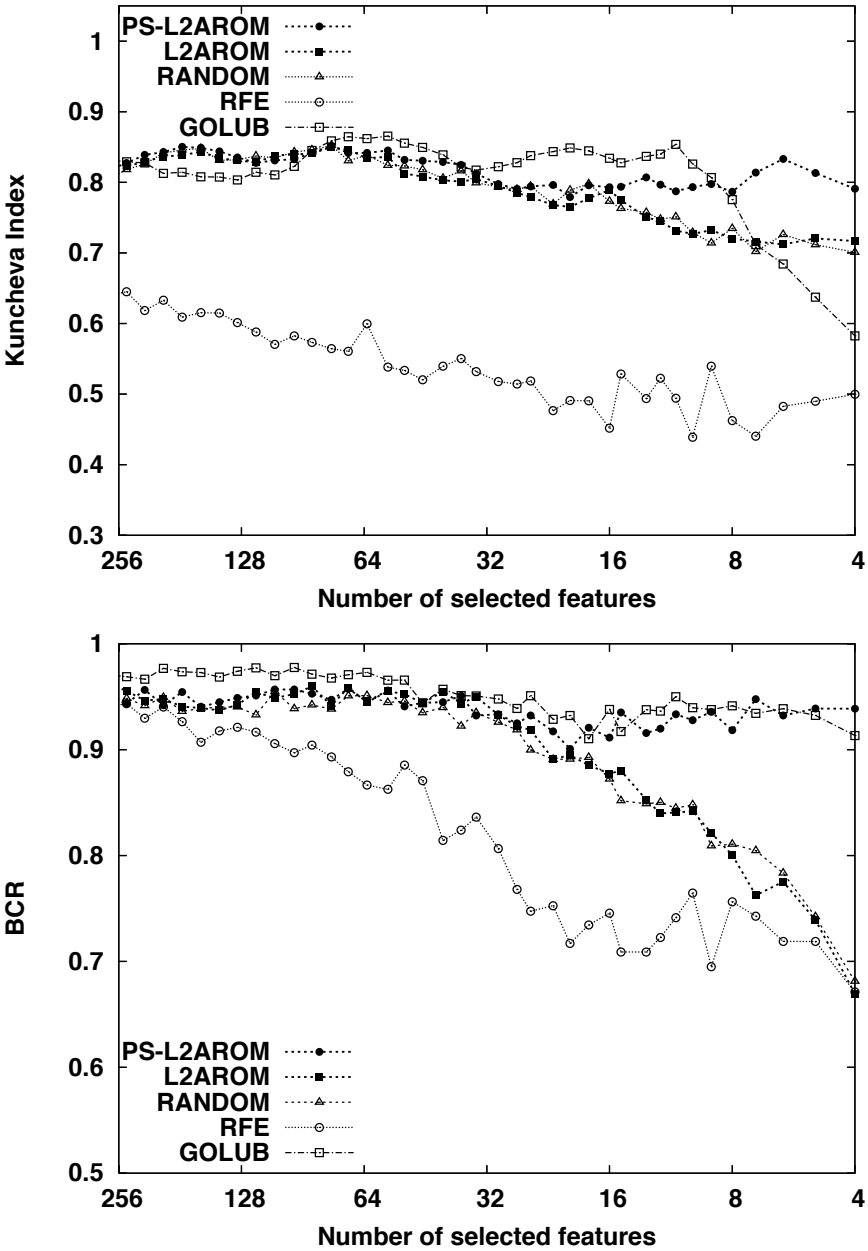


Figure 8.3 Signature stability (Kuncheva index) and classification performance (BCR) of PS- L_2 -AROM (with $\beta_{j \in [CD11c, CD33, MB-1]} = 10$), L_2 -AROM, RANDOM, RFE and Golub index filtering on LEU. Average results over 200 runs (90 % training - 10% test).

8.2.3 Partial Supervision from Data Partitioning

It is interesting to compare the various selection methods on additional data sets while checking the influence of a partial supervision on a larger number of genes. However such a biological knowledge on public databases is not always available. Our experimental protocol is consequently adapted as follows. Each dataset is first randomly split into two stratified folds representing respectively 20% and 80% of the whole data. Prior knowledge is simulated by computing on the 20% partition a signature $\mathbf{S}_{prior}$ made of the 50 most differentially expressed genes according to the Golub's ratio. The 80% partition is subsequently randomly partitioned into 90% training and 10% test sets. The 50 dimensions in $\mathbf{S}_{prior}$ are favored with PS- L_2 -AROM to select features on the 90% training set on which a linear SVM is built. BCR performances are estimated on the 10% test set. We report average results over a total of 200 runs: 10 random external splits (20%-80%) and ($J = 20$) internal splits (90%-10%).

Figures 8.4 and 8.5 report stability and classification performances on DLB and LEU with $\beta_{j \in \mathbf{S}_{prior}} = 10$, $\beta_{j \notin \mathbf{S}_{prior}} = 1$ for PS- L_2 -AROM. The partial supervision greatly increases both the stability of the selected gene lists and BCR with respect to L_2 -AROM. An improved stability is noticeable even when significantly more than 50 genes are selected. Results with RFE are globally worse especially in terms of stability. BCR results are equivalent between the Golub's ratio and PS- L_2 -AROM while the latter generally offers a better stability for small signatures. Figures 8.6 and 8.7 show significant stability improvements with PS- L_2 -AROM on PRO and COL. If we combine stability and BCR performances in a single metric (for instance, by computing the geometric average between both measures), PS- L_2 -AROM offers improved results over L_2 -AROM in all our experiments. The closest competitor to PS- L_2 -AROM is the Golub's ratio combined with a linear SVM classifier but this is likely related with the fact that the prior knowledge was precisely simulated with the Golub's ratio.

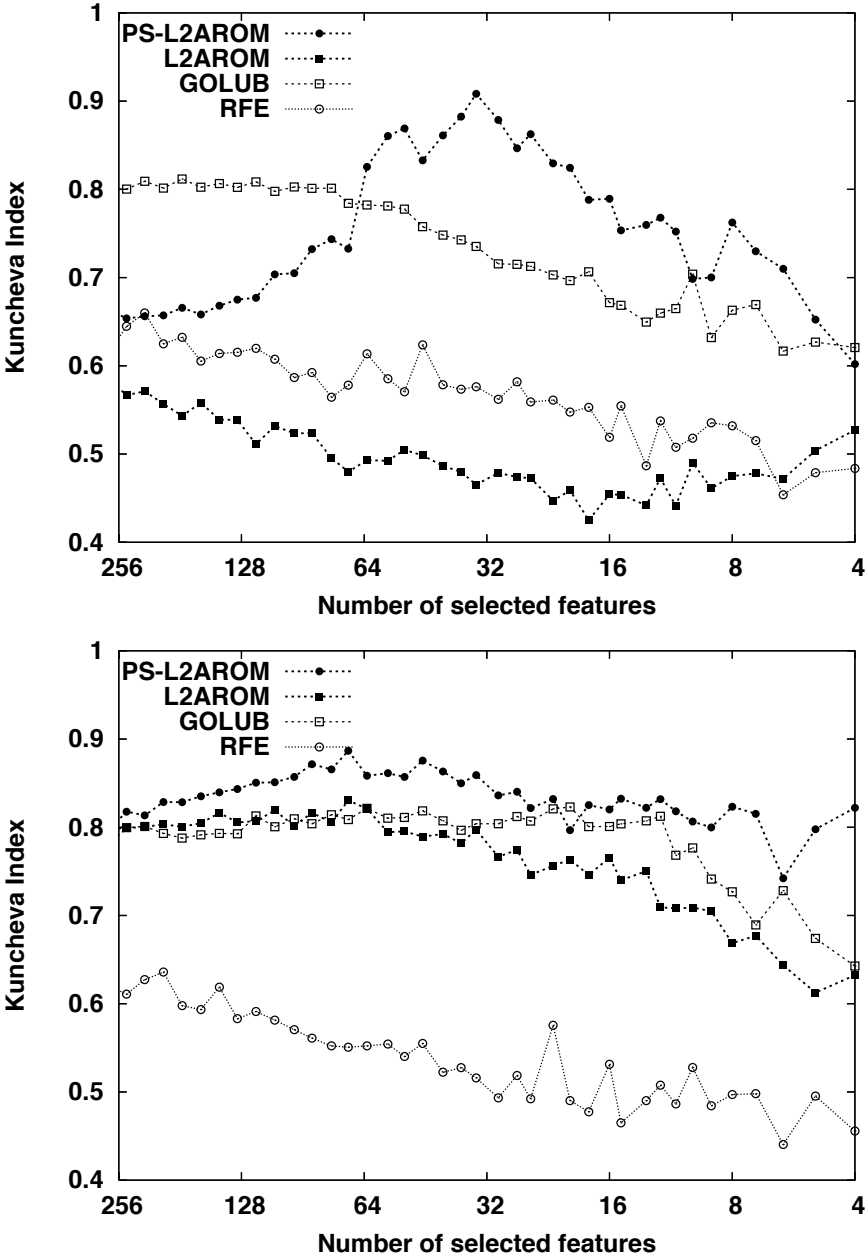


Figure 8.4 Signature stability (Kuncheva index) on DLB (top) and LEU (bottom). Partial supervision for PS-L2-AROM with 50 genes selected on an independent partition (20%) with Golub index.

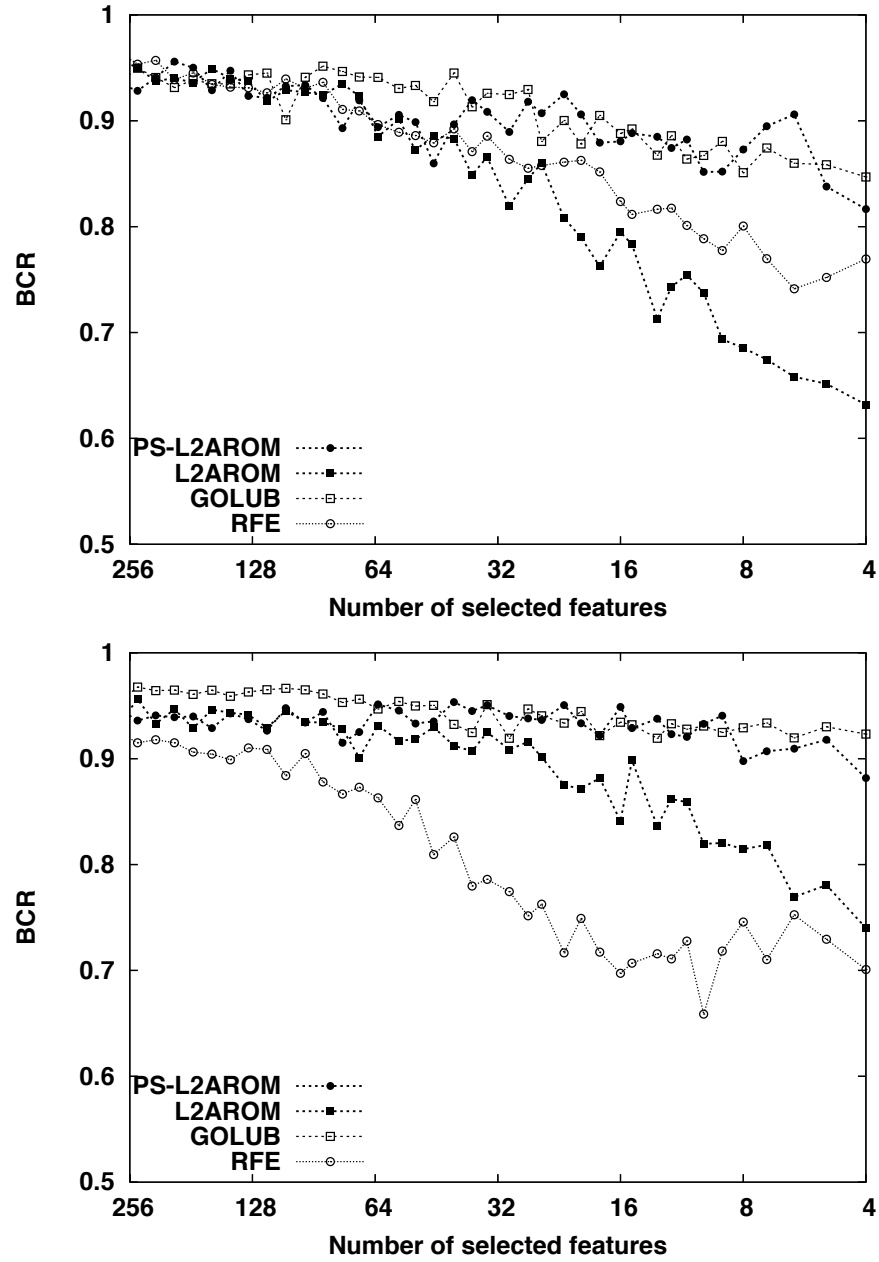


Figure 8.5 Classification performances (BCR) on DLB (top) and LEU (bottom). Partial supervision for PS- L_2 -AROM with 50 genes selected on an independent partition (20%) with Golub index.

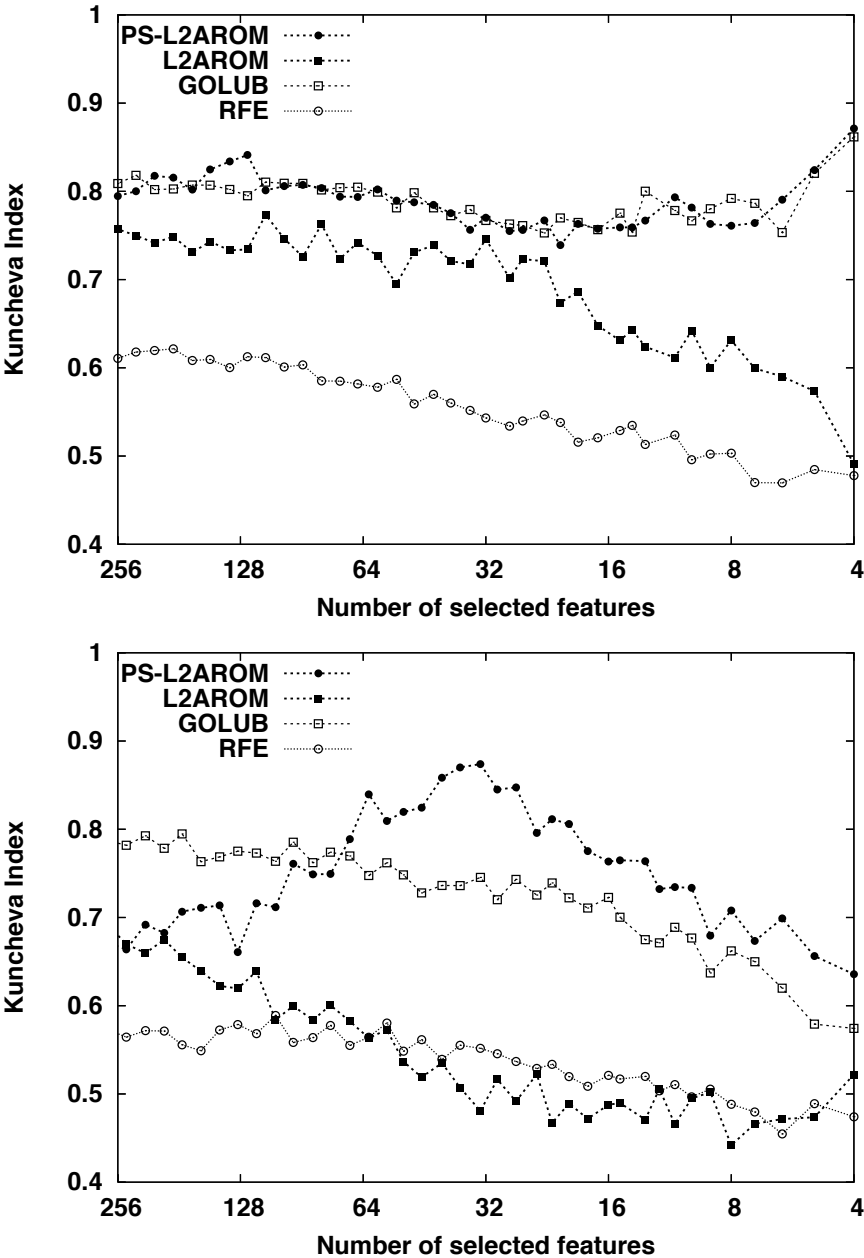


Figure 8.6 Signature stability (Kuncheva index) on PRO (top) and COL (bottom). Partial supervision for PS-L₂-AROM with 50 genes selected on an independent partition (20%) with Golub index.

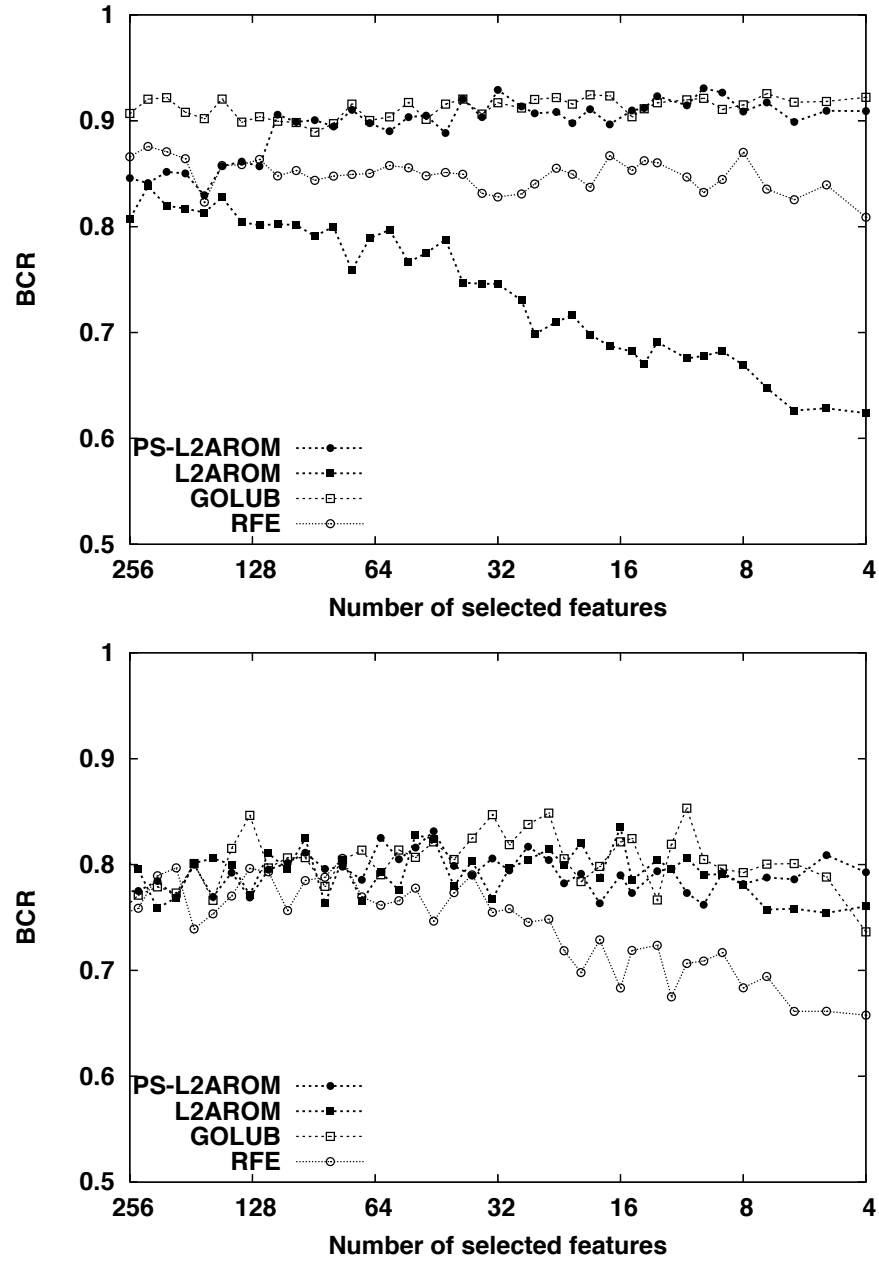


Figure 8.7 Classification performances (BCR) on PRO (top) and COL (bottom). Partial supervision for PS- L_2 -AROM with 50 genes selected on an independent partition (20%) with Golub’s S/N ratio.

8.3 Discussion

We propose a new feature selection method based on regularized linear models. This approach uses a partial supervision on the features *a priori* assumed to be more relevant. It naturally extends the AROM methods due to (Weston *et al.*, 2003). Several experiments on microarray datasets show that the partial supervision largely improves the stability of the selected gene lists, with respect to variation in data sampling. Classification performances are also improved in most cases. Since the selection algorithm is multivariate, partial supervision of a few dimensions may influence the other selected features. The dimensions *a priori* favored may also be discarded if necessary to optimize the classification objective. The iterative learning algorithm uses a L_1 -norm regularization. This objective function can be subsequently approximated with a L_2 -norm. This approximation is very efficient in practice for small- n -big- p settings, even if potentially less sparse. The algorithm then reduces to linear SVM learning with iterative rescaling of the inputs. The scaling factors directly depend on the prior relevance defined on each dimension. Approximating L_1 -AROM by L_2 -AROM has no significant influence on the practical results of Weston *et al.* (2003). It would be worth to confirm this observation when partial supervision is added to the feature selection. Theoretical arguments support that L_1 regularization is likely to fail in the case of high between-feature correlation (Zhao and Yu, 2006), as it is the case with microarray data. In this case, it has been shown that L_2 regularization could be more efficient with respect to classification performances than L_1 regularization (Bach, 2009).

This chapter deals with the introduction of prior knowledge within a learning problem. Prior knowledge is a frequently used notion in the Bayesian framework. We already pointed that the L_1 regularization is an equivalent to the Laplace prior in Chapter 3. Bayesian approaches use various priors to induce sparsity: the Student's t distribution, the Laplace (Seeger, 2008) or the *spike-and-slab* prior (George and McCulloch, 1997; Hernández-Lobato *et al.*, 2010). An advantage of the latter over the previous ones is to let the user define the desired fraction of non-zero coefficients. Nevertheless, those types of priors are not specific enough to suit the context considered in this chapter. The type of knowledge they allow to introduce merely translates information about the kind of distribution the model weights should follow. They do not let the user include real expert knowledge, for example about *which* dimensions are more likely to be included in the final solution. Making

possible the incorporation of similar specific prior knowledge in the Bayesian framework would be a welcome contribution.

Our approach in this chapter is motivated by microarray data analysis but it can be used in principle in any domain with some prior preference on the relevant features. The weights to favor some dimensions could also depend on the degree of certainty of the prior knowledge. The proposed approach could also be used to perform transfer learning across tasks, acquiring prior knowledge on one dataset and using it as partial supervision on others. This is the topic of the next chapter.

Transfer Learning for Feature Selection

9

The number of microarray experiments needed to obtain robust models is generally orders of magnitude higher than the actual size of most datasets ([Eindor *et al.*, 2006](#)). The number of available datasets is however continuously rising. Large databases like the NCBI's Gene Expression Omnibus (GEO) ([Edgar and Barrett, 2006](#)) or the EBI's ArrayExpress ([Parkinson *et al.*, 2009](#)) offer tens of thousand microarray samples which are well formatted and documented. The construction of a large microarray dataset consisting of the simple juxtaposition of independent smaller datasets (pooling) would be difficult or irrelevant due to biological, technical or experimental differences.

Source and Target Domains

Transfer learning techniques have been designed to overcome situations where too few samples for a task are available, but where experience from slightly different tasks is available ([Silver and Bennett, 2008](#); [Pan and Yang, 2008](#)). Knowledge is extracted from previous experience (source domains) to help solving the new problem (target domain). Samples of source and target domains are supposed to be drawn from similar but different distributions. This setting contrasts with semi-supervised learning in which samples used to build the model are supposed to be drawn from the same distribution but comprise some unlabeled samples ([Chapelle *et al.*, 2006](#)).

Transfer Learning Ontology

A transfer learning ontology is based on the labels (un)availability. Inductive transfer requires labeled data both for the source and target domains ([Argyriou *et al.*, 2006](#)). The target domain examples are unlabeled for transductive

transfer (Daumé III and Marcu, 2007). Fully unsupervised transfer techniques have been designed as well (Wang *et al.*, 2008). Transfer learning addresses several questions, independently of the labels availability. The main ones are: *What* and *How to transfer*. Transfer learning is not limited to a feature selection context, though some approaches of feature selection use transfer learning. The transfer of knowledge between source and target domains may concern the examples (Liao *et al.*, 2005; Huang *et al.*, 2007; Dai *et al.*, 2008), model parameters (Evgeniou and Pontil, 2004; Lawrence and Platt, 2004) or, as in this chapter, the feature space. For example, Multi-task Feature Learning (Argyriou *et al.*, 2006) learns a common feature representation for several tasks at the same time. Multi-task is a particular type of transfer more extensively covered by the next chapter. Structural correspondence learning (Blitzer *et al.*, 2006) uses features supposed to be relevant for several domains to generate new features for the target domain. Another approach compares learning tasks by measuring a distance between relevance weights on a set of common features (Mierswa and Wurst, 2005).

Negative Transfer

The third important question raised by transfer learning is *When to transfer*. This addresses the relevance of making the transfer given the source and the target domains, potentially raising a non-i.i.d. setting. The answer can be a hard *Transfer or not transfer at all*, but can also be more subtle, like *With what strength to transfer*. Indeed, if the source and target domains differ too much, it happens that the transfer could degrade the performances instead of improving them. This is called negative transfer and should be avoided. It can be done either by carefully selecting the source domain data, or by including built-in devices in the learning algorithm to ensure that, in the worst case, performances are not worse than they would be without transfer. However, many transfer learning methods do not address this issue, and yet seem to work well in practice.

Bounds for Transfer

Recently, a few researchers tried to formally explain why, if and when such settings can be beneficial. Related to the similarity of data distributions, the work of Kifer *et al.* (2004) introduces bounds and algorithms to detect changes in data streams. Ben-David and Schuller Borbely (2008) propose a bound in a multiple transfer setting. Finally, an upper bound on generalization error in single transfer learning has also been introduced by Ben-David *et al.* (2007). It

bounds the generalization error on a target domain with no labeled samples available, based on a related source domain which samples are labeled. However, these bounds are either non-practical or too loose to be directly phrased as optimization algorithms. For example, several of these bounds use a measure of divergence only relevant when the number of samples n is larger than the VC dimension of the considered class of models. In the case of linear models, it implies that n should be larger than $p + 1$. This is clearly not suitable in our context.

In contrast to the previously mentioned approaches for transferring feature representation, this chapter proposes a transfer learning method in which the transferred knowledge can be automatically partly or fully dropped whenever it does not help to optimize the classification objective on the target domain. This somehow circumvents the need for a priori estimating the similarity between source and target distributions. As a result, the specific choice of source datasets is not too critical. Benefits of transfer learning have been reported mainly as a gain in classification performances. The proposed approach moreover improves the stability of the selected features.

In the present chapter we rely on the selection method embedded in linear models called L_1 -AROM (Chapter 3). This specific choice is motivated by the possibility of extending this approach in a simple yet efficient way to perform transfer learning by biasing the optimization procedure towards some dimensions. The previous chapter proposes such a *partially supervised* (PS) extension but there, the favored dimensions are defined from prior knowledge. In the context of microarray data, molecular biologists may indeed sometimes guess that a few genes should be considered *a priori* more relevant. In the present work, we do not use such prior knowledge but rather related datasets, hence performing inductive transfer learning at the feature level. The additional benefits are a fully automated feature selection procedure and the possibility to choose the number of features to be transferred independently of some expert knowledge. A practical approximation of this technique reduces to learn linear SVMs with iterative rescaling of the inputs. The rescaling factors depend here on previously selected features from existing datasets.

This initial feature selection on source domains is performed using a simple univariate t -test ranking while the final iterative selection is intrinsi-

cally multivariate. Using an initial univariate selection on the source domains is both computationally efficient and arguably a relevant starting point before transferring to a distinct target domain. As shown in our experiments this choice results in significant stability and classification performance improvements.

The rest of the chapter is organized as follows. Section 9.1 details how to use the partially supervised feature selection technique PS- L_2 -AROM introduced in Chapter 8 to perform transfer learning. Experiments on three microarray datasets are reported in Section 9.2. Conclusions and future perspectives are discussed in Section 9.3.

Table 9.1 Publications related to this chapter

Helleputte, T. and Dupont, P. (2009a). Biomarker selection by transfer learning with linear regularized models. In *Machine Learning in Systems Biology*

Helleputte, T. and Dupont, P. (2009b). Feature selection by transfer learning with linear regularized models. In *European Conference on Machine Learning (ECML)*, number 5781 in Lecture Notes in Artificial Intelligence, pages 533–547

9.1 Transfer Learning with PS- L_2 -AROM

We discuss here how to use the PS- L_2 -AROM method for transfer learning. Let the *Target Domain* D_T be a set of samples $\mathbf{x}_i \in \mathbb{R}^p$, generated according to a distribution $P_T(\mathcal{X})$, and associated class labels $y_i \in \{\pm 1\}$ following $P_T(\mathcal{X}, \mathcal{Y})$, $i = 1, \dots, n$. The task is to build a robust classification model $g_T(\mathbf{x})$ and to identify a discriminative signature $\mathbf{S}_T$ for D_T . It is generally possible to find related datasets, called *Source Domains* D_S , for which $P_S(\mathcal{X}, \mathcal{Y}) \approx P_T(\mathcal{X}, \mathcal{Y})$ but $\hat{P}_S(\mathcal{X}) \neq \hat{P}_T(\mathcal{X})$. The notation $\hat{P}$ represents the finite training set made of samples drawn from P . For example, several microarray datasets are available on GEO (Edgar and Barrett, 2006) for which the class labels correspond to the same concepts, *cancer tissue* or *normal tissue*, but for which the gene expression distributions differ. There are many sources of divergence such as the type of biological samples, the RNA extraction protocol, the normalization steps applied to raw data, etc.

It is possible to use a partially supervised feature selection method as a transfer learning technique. The proposed approach is an inductive transfer learning technique since the class labels are known both for D_S and D_T . We assume that the data from all domains share a sufficiently large set of p features and, without loss of generality, that feature index j in D_S maps to the same index in D_T . The proposed approach simply uses any convenient feature selection on D_S to build an initial signature $\mathbf{S}_S$. A prior relevance vector $\boldsymbol{\beta} \in \mathbb{R}^p$ is defined from $\mathbf{S}_S$ to favor the actual selection of some dimensions on D_T :

$$\beta_j = \begin{cases} B & \forall j \in \mathbf{S}_S, \\ 1 & \forall j \notin \mathbf{S}_S, \end{cases} \quad (9.1)$$

where $B > 1$ corresponds to the weight to favor features belonging to $\mathbf{S}_S$. The choice of B is arbitrary but experiments reported in Section 9.2.5 illustrate that the proposed method is not sensitive to a specific choice for a large range of possible values. The vector $\boldsymbol{\beta}$ is used to bias the selection of features on D_T via PS- L_2 -AROM to obtain a final signature $\mathbf{S}_T$. The selection on D_T is influenced by the knowledge extracted from D_S , i.e. a set of indexes of relevant features. Those transferred features are assumed relevant for D_S but not necessarily highly discriminative on D_T , since $\hat{P}_S(\mathcal{X}) \neq \hat{P}_T(\mathcal{X})$. Our modeling assumption is however that the features extracted from similar tasks provide useful information as compared to selecting features only from a single domain D_T . This assumption is confirmed by our practical experiments reported in Section 9.2.

Any standard feature selection method can be used for the initial selection on D_S . We recommend in particular the use of a simple univariate technique such as a t -test ranking. It is computationally efficient and this initial selection is not meant to be highly accurate on D_S but to guide the detailed selection to be performed eventually on D_T . Moreover, t -test ranking might favor stability since it penalizes features showing large within-class variance. This is not explicitly the case of PS- L_2 -AROM. On the other hand, t -test is not multivariate, but PS- L_2 -AROM is. By combining the t -test and PS- L_2 -AROM, we expect to inherit the advantages of both methods.

9.2 Experiments and Results

This section describes and evaluates several practical ways of performing transfer learning based on the partially supervised feature selection implemented by PS- L_2 -AROM. Three prostate cancer microarray datasets are presented in Section 9.2.1. Baseline results are obtained with no transfer, that is by applying the L_2 -AROM feature selection technique on a given dataset. Improved stability and classification performances are obtained with a *single transfer*. This first protocol uses one related dataset as source domain to guide the selection on the target domain (Section 9.2.2). Further improvements can be obtained with *multiple transfer* which combines several source domains (Section 9.2.3). Experimental results are presented in Section 9.2.4. Finally, the sensitivity to a specific choice of the prior weight value (the B parameter) is analyzed in Section 9.2.5.

In a nutshell those experimental results show that transfer learning based on partially supervised feature selection always leads to a gain in stability as well as a systematic gain in classification performance for signature sizes of interest.

9.2.1 Microarray Data

Table 9.2 presents the main characteristics of the three prostate cancer microarray datasets used in this experimental section: PRO1 (Singh *et al.*, 2002), PRO2 (Chandran *et al.*, 2007) and PRO3 (Welsh *et al.*, 2001). In the original publications, the task is almost the same for the three datasets: binary classification between tumor and normal tissues. In PRO2, tumor samples are of two types: primary tumor and metastatic tumor. Tumor samples in PRO3 correspond to 24 primary tumors and 1 lymph node metastasis. No precision is made about the type of tumor tissue in PRO1. The microarray technology used to produce those datasets is the same for PRO1 and PRO2, but is a bit older for PRO3 (see Table 9.2). Consequently, features (genes) present on each type of chip differ very slightly. Samples and RNA extraction were also performed according to different protocols. The internal normalization to produce one value for each feature also differ from set to set. All these differences make their simple combination in a larger dataset irrelevant. The three datasets are here reduced to the set of 12,600 features they have in common. More details on the datasets can be found in Appendix D.

Table 9.2 Datasets used for transfer learning for feature selection validation. Columns respectively show the dataset name, the number of normal samples, the number of tumor samples, the original number of features and the type of Affymetrix chips used.

Name	Normal	Tumor	# Features	Chip
PRO1	50	52	12,625	HGU95Av2
PRO2	18	86	12,625	HGU95Av2
PRO3	9	25	12,626	HGU95A

9.2.2 Single Transfer

The experimental protocol applied in this chapter is inspired from the N-Resamplings strategy described in Chapter 5, as in Chapters 7 and 8. Our first experimental protocol uses $J = 200$ random 90%-10% samplings from the target domain D_T and is depicted in Figure 9.1. Each 90% fraction forms a training set. The features of each training set are first normalized to zero median and unit standard deviation according to the robust version of the z-score normalization (IQR normalization, Chapter 6). Features are then selected via PS- L_2 -AROM and a linear soft-margin SVM is built on the selected dimensions. Each 10% fraction forms the associated test samples that are pre-processed according to the training normalization parameters. The vector β used for PS- L_2 -AROM is set by selecting a signature $\mathbf{S}_S$ on D_S with a t -test ranking. The 50 top ranked features define $\mathbf{S}_S$, which is a common default signature size for biomarker selection. The feature selection is performed on D_T while favoring the genes from $\mathbf{S}_S$ according to:

$$\beta_j = \begin{cases} 10 & \forall j \in \mathbf{S}_S, \\ 1 & \forall j \notin \mathbf{S}_S. \end{cases} \quad (9.2)$$

Here D_S is a single dataset different from D_T . Given the three datasets available, six combinations of D_S and D_T are tested. Stability over the 200 samplings and averaged balanced classification rate performances are reported. Stability is measured by the Kuncheva index as detailed in Equation (4.2). For comparison purposes, the same protocol is performed with no transferred knowledge, i.e. with $\beta_j = 1, \forall j \in 1, \dots, p$.

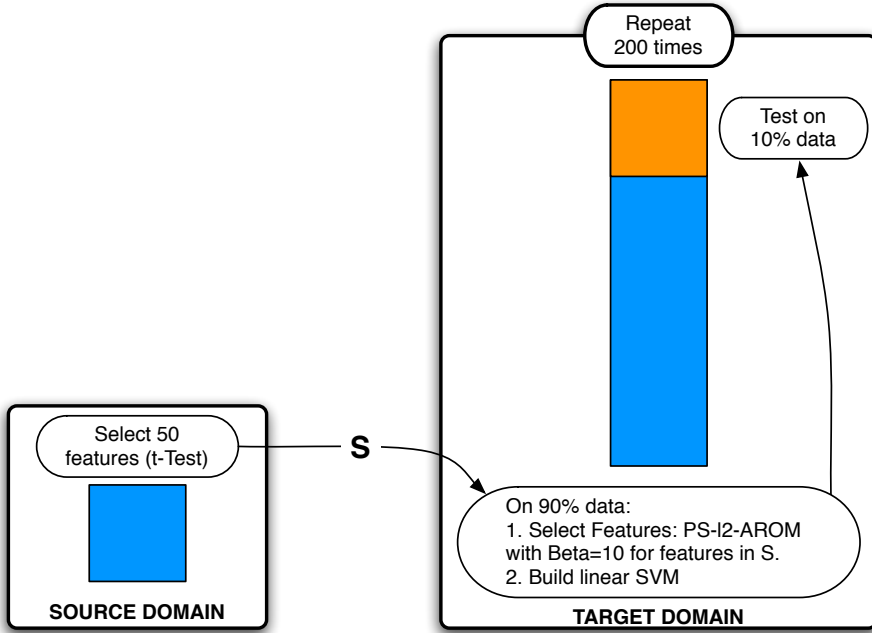


Figure 9.1 Single transfer protocol

9.2.3 Multiple Transfer

When several datasets are available as source domains it may be useful to combine the knowledge extracted from each of them to guide the feature selection on the target domain D_T . Such a multiple transfer protocol is described below with two source domains and illustrated in Figure 9.2. The extension to more than two source domains is straightforward.

A signature S_S of 50 features is extracted from the source domains D_{S_1, S_2} and applied to D_T via PS- L_2 -AROM. Features are ranked according to a t -test on each source dataset. The s best ranked features are selected from each source dataset and the intersection of those two signatures is computed. The parameter s is chosen such that the size of the intersection is 50. The rationale behind this choice is to transfer the same amount of knowledge in D_T as compared to the single transfer protocol. The values of s used for the three possible combinations of the available source datasets are 557 for $(PRO1 \cap PRO2)$, 275 for $(PRO3 \cap PRO1)$ and 385 for $(PRO3 \cap PRO2)$. Those differences result from the fact that some combinations have more top ranked features in

common than others. For example, the number of features needed to build an intersected signature of 50 genes is about twice as much for PRO1 and PRO2 as compared to PRO1 and PRO3. This could explain why using PRO1 rather than PRO2 as D_S for PRO3 as D_T gives better results in a single transfer protocol (see next section).

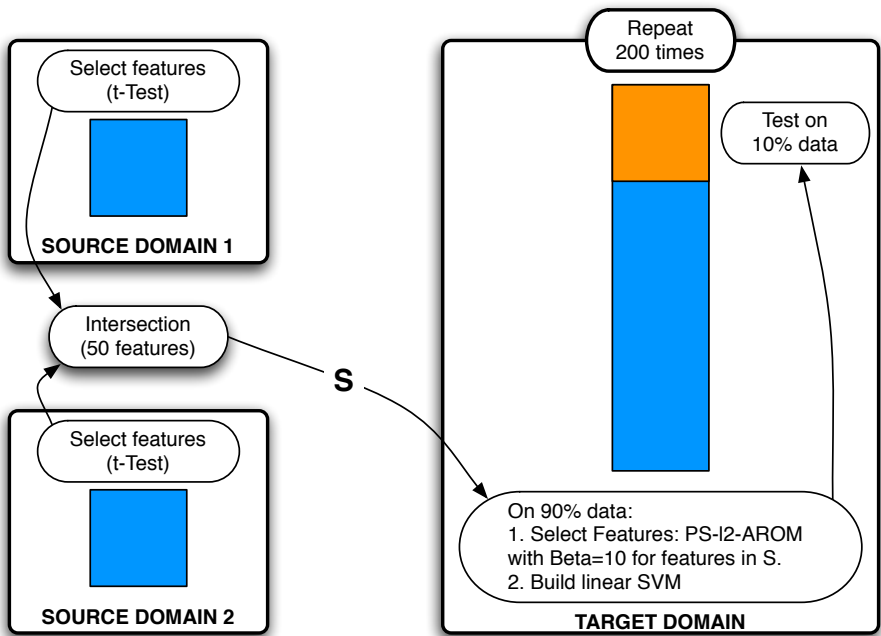


Figure 9.2 Multiple transfer protocol

9.2.4 Results

Figures 9.3 to 9.5 show the BCR and the stability results for various signature sizes $|S_T|$ for PRO1, PRO2 and PRO3, respectively. For signatures significantly larger than the transferred knowledge ($|S_S| = 50$) results are equivalent to baseline results (no transfer). In contrast, for signature sizes of practical interest (a few tens of genes), there is a large increase both in classification performance and stability.

For example, transferring knowledge from PRO1 to PRO3 (top of Figure 9.5) improves the average BCR from 79.6% (resp. 65.7%) for a signature

size of 16 (resp. 10) genes up to more than 99.2% (resp. 98.7%). Those differences are statistically significant according to the corrected resampled *t*-test described in Chapter 5. Such a test corrects for the fact that the various test sets are not independent since they may overlap. The BCR differences are significant with a (likely conservative) p -value $= 1.9 \times 10^{-2}$ for 16 genes and a p -value $= 2.6 \times 10^{-4}$ for 10 genes. BCR results on the PRO2 and PRO1 target datasets follow the same trends.

Multiple transfer may further improve the BCR performances. This is in particular the case on the PRO2 dataset (top of Figure 9.4) with transferred knowledge from $\text{PRO3} \cap \text{PRO1}$. For example, the BCR differences are statistically significant (p -value ≤ 0.025) between a single transfer and a multiple transfer with a final signature of 10 genes. In general, multiple transfer BCR results are always equivalent or better than single transfer results. Multiple transfer thus offers a more robust approach not requiring to carefully select which source domain should be considered for a given target domain.

Transfer learning always improves the stability of the selected gene lists as illustrated at the bottom of Figures 9.3 to 9.5. The maximal stability is often reached around 50 features. This is no surprise since precisely 50 genes are favored during the selection on the target domain. However this maximal stability does not reach 100% which illustrates that the selected genes are not just those belonging to S_5 .

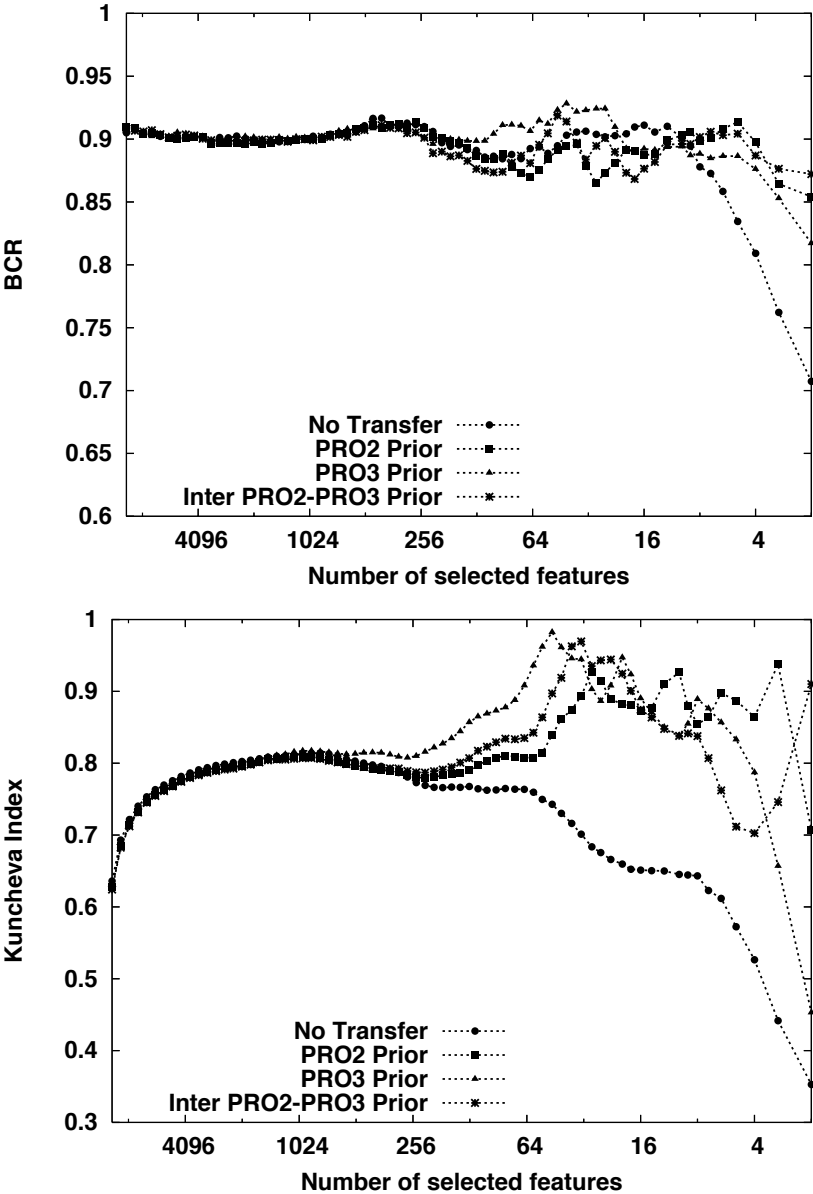


Figure 9.3 Transfer learning results on on PRO1. Top: Classification performances (BCR). Bottom: Signature stability (Kuncheva index). No Transfer is the baseline: features are selected on the target dataset without prior preference. The next two curves specify which dataset was used in a single transfer setting. The fourth curve refers to the multiple transfer setting.

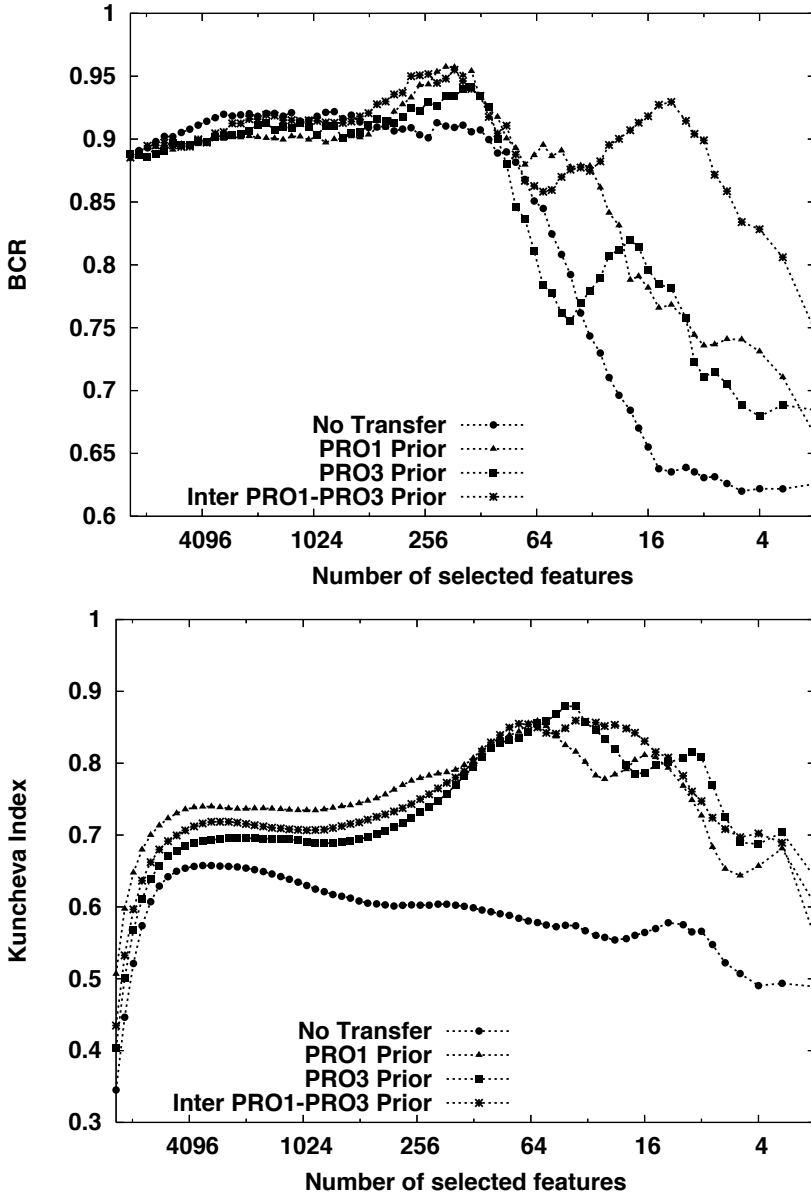


Figure 9.4 Transfer learning results on on PRO2. Top: Classification performances (BCR). Bottom: Signature stability (Kuncheva index). No Transfer is the baseline: features are selected on the target dataset without prior preference. The next two curves specify which dataset was used in a single transfer setting. The fourth curve refers to the multiple transfer setting.

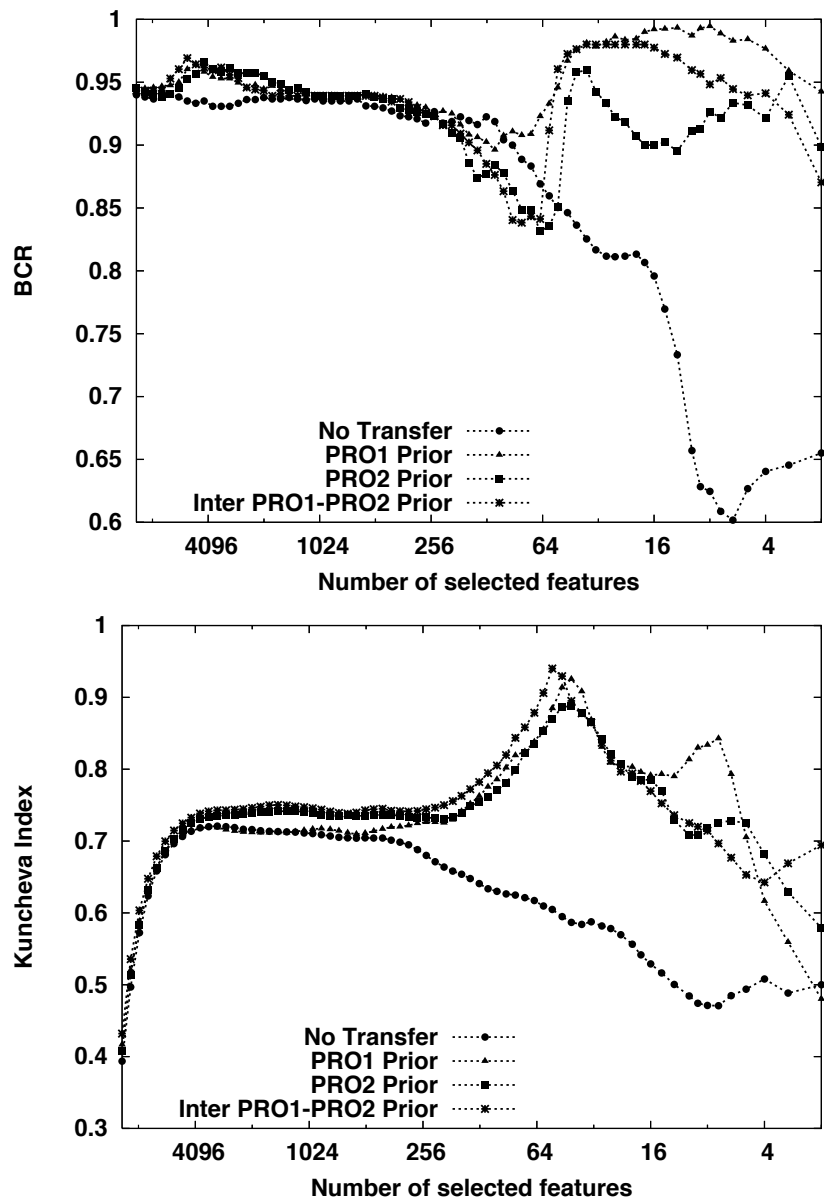


Figure 9.5 Transfer learning results on on PRO3. Top: Classification performances (BCR). Bottom: Signature stability (Kuncheva index). No Transfer is the baseline: features are selected on the target dataset without prior preference. The next two curves specify which dataset was used in a single transfer setting. The fourth curve refers to the multiple transfer setting.

9.2.5 Impact of Prior Relevance Weight

In the experiments described in Section 9.2.4, the value $B = 10$ was chosen to favor some dimensions via PS- L_2 -AROM. The same arbitrary configuration appears in experiments of Chapter 8. The influence of a specific choice of the B value on stability and classification performances is analyzed in this section. We detail experiments with multiple transfer since this approach offers the best results so far. Figure 9.6 displays stability and BCR results for $B = \{1, 2, 5, 10, 100, 1000\}$ on PRO3. The curves for $B = 1$ and $B = 10$ correspond to the previous settings respectively with no transfer and multiple transfer. Equivalent trends are observed on the other datasets (results not shown). The influence of the B value can be summarized as follows.

Results show that the higher the B value the stronger the stability peak around 50 features. This is a logical consequence of the design of PS- L_2 -AROM. The stability is not influenced for signature sizes $|\mathbf{S}_T|$ significantly larger than 50 features except for a very large $B = 1000$. For signature sizes smaller than 50 a better stability is obtained with B in the range $[10, 100]$. BCR results show a positive effect of the partial supervision as soon as B is greater than 5. The proposed approach is not highly sensitive to a specific choice of B in the range $[5, 100]$. The default value of $B = 10$ offers a reasonable choice overall. Hence the proposed approach does not require to carefully optimize the meta-parameter B in a nested validation loop.

9.3 Discussion

We address in this chapter the problem of transfer learning for feature selection and classification of high-dimensional data, such as those produced by microarray experiments. We propose a feature selection method on a *target domain* that can be partially supervised (PS) from features previously extracted from related *source domains*. Such knowledge can be acquired from public databases like GEO (Edgar and Barrett, 2006) or ArrayExpress (Parkinson et al., 2009). The initial feature selection on the source domains is typically performed with a fast univariate technique. The purpose of this initial selection is to guide the selection process on the target domain.

We rely here on the PS- L_2 -AROM method, the partially supervised feature selection approach embedded within the estimation of a regularized linear model introduced in Chapter 8. This algorithm reduces to linear

SVM learning with iterative rescaling of the input features. The scaling factors depend here on the selected dimensions on the source domains. The proposed optimization procedure smoothly favors the pre-selected features but the finally selected dimensions may depart from those to optimize the classification objective under rescaled margin constraints.

Practical experiments on several microarray datasets illustrate that the proposed approach not only increases classification performances, a usual benefit of a sound transfer learning scheme, but also the stability of the selected dimensions with respect to sampling variation. We also show how a multiple transfer from various source domains can bring further improvements.

The proposed approach relies on a meta-parameter defining the prior weight of the favored dimensions during the partially supervised feature selection. We show experimentally that this method is not sensitive to a specific choice of this parameter for a large range of possible values. Distinct weight values for different features could also be considered in the future. One could for instance define those weights as a function of the p -values of the initial t -test. Here the t -test was applied on the source domain(s) but it could also be interesting to compute the t -test on the target domain itself, hence without transfer. The combination of a simple feature ranking method and the partially supervised feature selection could already improve stability and/or classification performances on a given dataset. This configuration is tested in the context of the Cristall project reported in Chapter 12. In this case, stability is improved, but not as much as it is with expert prior knowledge.

We rely on a simple univariate selection on the source domain(s). The purpose of this initial selection is indeed not to be highly accurate since the final and more refined selection is performed on a distinct target domain. Our current choice is a simple t -test ranking for this initial selection. Several alternatives could however be considered, including a multivariate embedded selection method such as L_2 -AROM. More importantly, the size of the initial signature extracted from the source domain(s) is currently fixed to 50 genes. This is a common default value for biomarker selection and it offers very good performances. It would however be interesting to investigate further the influence of this size on the quality of the final selection. A related issue would be the automatic selection of the best source domain(s) for a given

target domain and the number of features to be extracted from each of them. The design of similarity measures between various datasets could help in this regard, but seems to be a difficult task, namely because of the very low samples/features ratio in our applicative context.

Our partially supervised feature selection is a general approach which does not depend, at least in principle, on how the favored dimensions are initially selected. The present chapter relies on other related datasets while Chapter 8 uses real *prior knowledge* from field experts. A further and natural extension would combine the transferred knowledge with such prior knowledge whenever available.

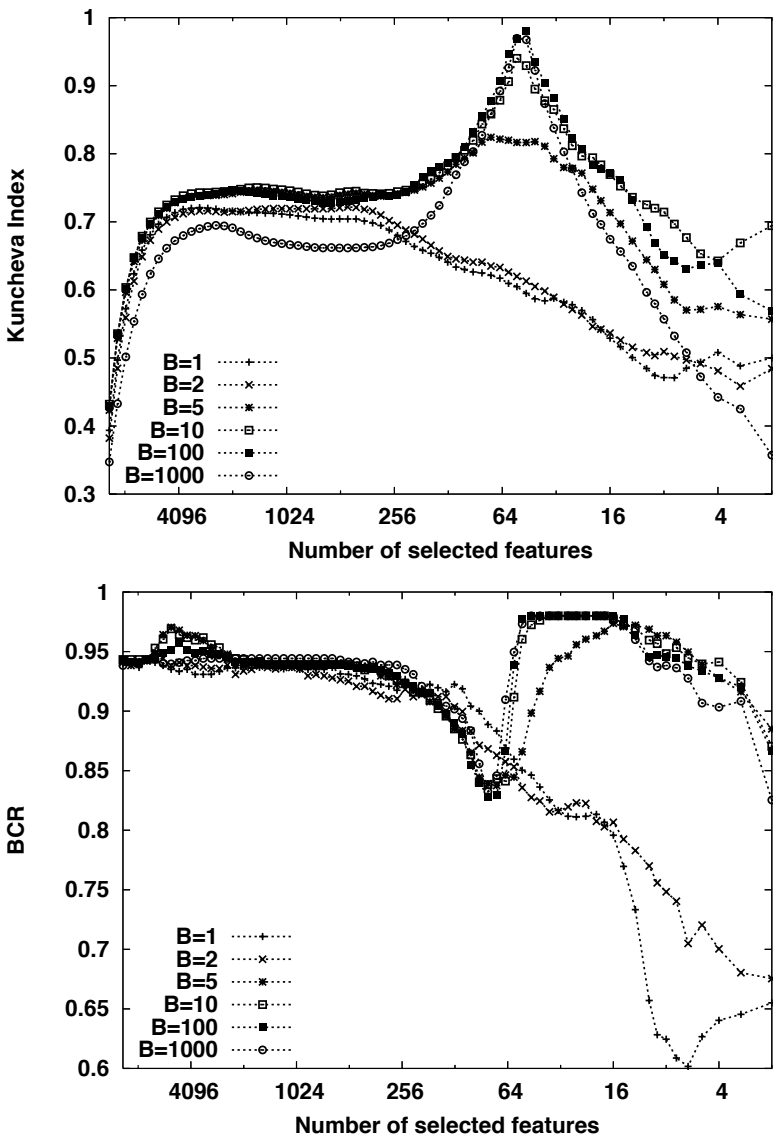


Figure 9.6 Impact of the prior relevance weight on signature stability (Kuncheva index) and classification performances (balanced classification rate) on PRO3.

Feature Selection for Multi-Task Classification

10

In the previous chapter, several datasets about related tasks are used together in a transfer learning setting. The goal is to optimize the classification and selection performances on a target domain with the help of potentially several source domains. The performances are thus not optimized for those source domains. What if we want to jointly enhance the classification and selection processes on all the related domains at the same time? This is what this chapter is about.

More precisely, this chapter focuses on feature selection in a context of binary classification over multiple tasks for which a small number of samples (some dozens) are available, each of them being described by tens of thousands of features. The same features are assumed to be defined for each task, or equivalently the intersection of commonly defined features is considered, but a specific, and possibly small, subset of features is assumed to be relevant for each task. Most biomolecular technologies (gene expression microarray, SNP, CNV, spectrometry, *etc.*) are representative examples. Large databases like GEO ([Edgar and Barrett, 2006](#)) offer hundreds of datasets of distinct but related tasks, such as discriminating between cancer and non-cancer gene expression profiles. Specific samples, RNA extraction protocols, microarray platforms and tissue types however differ from one dataset to the other. This situation pleads for multi-task learning instead of simply pooling the data from various tasks.

Chapters 8 and 9 showed that stability of the feature selection in small- n -big- p settings could benefit from some original inductive biases, such as the

incorporation of prior knowledge or the transfer learning from related tasks with equal, or sometimes better, classification performance.

On the other hand, single-task classification has been extended to multi-task learning (Caruana, 1997). The various approaches differ in the way task dependence is modeled. Evgeniou and Pontil (2004) combine a model component shared across tasks and a task-specific component while Obozinski *et al.* (2009) use a mixed-norm regularization. Both approaches fit in the loss & regularizer pattern.

The first approach controls, through regularization constants, the amount of assumed similarity between the supports for each task. The similarity between supports is the proportion of features that are commonly selected across different tasks. Such measure can be regarded as a between-tasks stability in a multi-task setting. In the case of strong sharing between tasks, not only the support but also the model parameters values are shared. This case becomes equivalent to the pooling of the samples from all tasks considered.

The mixed-norm approach performs a dimensionality reduction embedded in a multi-task classification problem. As a special case of this procedure, multi-task feature selection is performed while looking for similar supports across tasks but possibly distinct model parameters for each task. In various multi-task applications, it is sound to consider a fully shared support with distinct model parameters corresponding to the commonly chosen features. In the context of gene expression data for instance, such a situation typically corresponds to the design of a unique diagnosis kit for several pathologies. Each task amounts to predict whether a patient is affected by a given pathology. The number of genes to be measured via this diagnosis kit could be minimized for cost and standardization purposes. An optimal solution in such a case could be a set of genes such that each of them is used for every task considered. However the values of the model parameters corresponding to those genes need not be the same across all tasks.

The contributions in this chapter are the followings. We address multi-task feature selection with (generalized) linear models for classification. This approach combines a sum of logistic losses over each task with a mixed-norm regularizer. Six different sparsity-enforcing regularizers $L_{i,j}$ are considered,

where i stands for the regularization over all features and j stands for the regularization over the same feature for every task. We consider specifically $i \in \{0, 1\}$ and $j \in \{1, 2, \infty\}$. The $L_{1,1}$ regularization is a natural extension to the lasso penalty but actually reduces to single-task learning, repeated over several tasks. The $L_{1,2}$ regularizer favors some sharing between model supports while the $L_{1,\infty}$ mixed-norm favors almost full sharing. Mixed-norms extension to the pseudo-norm L_0 are also proposed.

We also propose a dedicated solver to estimate efficiently the parameters of a sparse multi-task penalized logistic regression. Each optimization step has a linear time complexity both in terms of the number of features and the number of samples, hence offering scalability to large problems. The six regularizers are evaluated experimentally on several microarray datasets according to several quality criteria: the number of selected features, within- and between-task stability, classification performances and convergence speed. Multi-task penalized logistic regression is also compared with an approach for which the model parameters combine a weight vector shared across tasks and a task-specific weight vector (Evgeniou and Pontil, 2004).

The rest of this chapter is organized as follows. Section 10.1 discusses several methods to perform multi-task learning of generalized linear models, including several alternatives for dimensionality reduction and feature selection. We focus on embedded approaches and hence on several possible regularizers of the respective optimization problems. Section 10.2 further details the sparse multi-task penalized logistic regression proposed here and stresses the links to related works including the group lasso (Yuan and Lin, 2006) and sparse SVM for grouped variables (Zou and Yuan, 2008). Section 10.3 presents an efficient solver for estimating the parameters of a multi-task penalized logistic regression. It discusses the solver time complexity, its convergence and convexity issues. Section 10.4 analyzes the relative qualities of the six different regularizers considered here. Comparisons are also reported with the method of Evgeniou and Pontil (2004). Section 10.5 summarizes the contributions of this chapter and presents some perspectives of this work. Finally, it is shown in Section 10.6 how the sparse multi-task penalized logistic regression proposed in this chapter can also be used to tackle multi-class problems, and what benefits it can bring.

10.1 Multi-Task Regularizers for Generalized Linear Models

We define a set of K binary classification tasks $T_k = (\mathbf{X}_k, \mathbf{y}_k)$, each made of a set of n_k samples in p dimensions and a set of corresponding labels $\mathbf{y}_k$. A sample i for task k is of the form $\mathbf{x}_{ik} \in \mathbb{R}^p$, where $\mathbf{x}_{ik}$ is the i^{th} row of the matrix $\mathbf{X}_k$, and its corresponding label is $y_{ik} \in \{-1, 1\}$. Remember that in our context, $n_k \ll p$. A linear model f_k of parameters $\mathbf{w}_k$, defined for each task, maps the inputs to the response variable:

$$f_k(\mathbf{X}_k) = \text{sign}(\langle \mathbf{X}_k, \mathbf{w}_k \rangle). \quad (10.1)$$

In practice, $\mathbf{X}_k$ is complemented with a first column of ones to benefit from a bias term w_{0k} for each task, while keeping simpler notations.

In the context of multi-task learning, a joint regularization is frequently considered where model parameters take the form of a $K \times p$ matrix $\mathbf{W}$ with one row $\mathbf{w}_k$ per task.

We discuss here several multi-task methods which can be phrased as optimizing a convex combination between a loss term $\mathcal{L}$ and a regularizer $\mathcal{R}$, the loss & regularizer pattern of Equation (2.20). The loss term typically corresponds to a square loss, a hinge loss or a logistic loss but we keep here the loss notation generic to stress the links between the various regularizers. [Evgeniou and Pontil \(2004\)](#) propose a multi-task approach that defines the weights $\mathbf{w}_k = \mathbf{w} + \mathbf{v}_k$ of several linear models as a sum of a shared weight vector $\mathbf{w}$ and a task-specific vector $\mathbf{v}_k$. The learning aims at solving the following optimization problem with two regularization terms:

$$\min_{\mathbf{W}} \sum_{k=1}^K \sum_{i=1}^{n_k} \mathcal{L}(y_{ik}, \langle \mathbf{w}_k, \mathbf{x}_{ik} \rangle) + \frac{\lambda_1}{K} \sum_{k=1}^K \|\mathbf{v}_k\|_2^2 + \lambda_2 \|\mathbf{w}\|_2^2. \quad (10.2)$$

Setting the regularization constants λ_1 and λ_2 is done via cross-validation. Their ratio influences the level of sharing between models. If $\mu = \lambda_1/\lambda_2$ is high, the task-specific part is more penalized and $\mathbf{w}_k \rightarrow \mathbf{w}$. In this case, the various models not only share a common (and possibly non-sparse) support but also common parameter values. This setting leads to pooling data from different tasks together. Conversely, if μ is low, the models are task-specific, but nothing guarantees a common support since $\mathbf{w}_k \rightarrow \mathbf{v}_k$. This case reduces to single-task learning, repeated independently on several

tasks. The regularization constants also need to be tuned with respect to the influence of the loss function. This method does not favor sparse supports and hence does not perform feature selection.

Our objective is to produce models for multiple tasks having mostly similar and small supports, but with possibly distinct parameter values for each task. A generic form of this objective is represented by the following optimization problem:

$$\min_{\mathbf{W}} \sum_{k=1}^K \sum_{i=1}^{n_k} \mathcal{L}(y_{ik}, \langle \mathbf{w}_k, \mathbf{x}_{ik} \rangle) + \lambda \|\mathbf{W}\|_{i,j}, \quad (10.3)$$

where $\|\mathbf{W}\|_{i,j}$ is a mixed-norm regularizer. The $L_{1,1}$ norm is a natural extension to the lasso penalty. However, this regularizer actually applies to each task separately:

$$\sum_{i=1}^p \sum_{k=1}^K |w_{ki}| = \sum_{k=1}^K \sum_{i=1}^p |w_{ki}| = \sum_{i=1}^p |w_{1i}| + \dots + \sum_{i=1}^p |w_{Ki}|. \quad (10.4)$$

Consequently the $L_{1,1}$ norm favors sparse models but no dependence between the tasks, and in particular no common support across tasks. Group-regularizers should be preferred in this case, each group corresponding to the same feature across different tasks. The $L_{1,2}$ group-regularizer provides some sharing across tasks. This specific approach has been recently proposed by [Obozinski et al. \(2009\)](#). The objective function becomes:

$$\min_{\mathbf{W}} \sum_{k=1}^K \sum_{i=1}^{n_k} \mathcal{L}(y_{ik}, \langle \mathbf{w}_k, \mathbf{x}_{ik} \rangle) + \lambda \|\mathbf{W}\|_{1,2}. \quad (10.5)$$

This regularization favors small and similar supports across tasks. The authors compare to other settings such as pooling the samples from several tasks into a unique dataset or repeated single-task learning. As the number of tasks and/or dimensions grows, there is an increasing benefit of considering an $L_{1,2}$ regularizer in terms of classification performance. Problem (10.5) is a special case of the more general form (10.3). It is also a special case of the more general problem introduced by [Argyriou et al. \(2008\)](#):

$$\min_{\mathbf{W}, \mathbf{U}} \sum_{k=1}^K \sum_{i=1}^{n_k} \mathcal{L}(y_{ik}, \langle \mathbf{w}_k, \mathbf{U}^T \mathbf{x}_{ik} \rangle) + \lambda \|\mathbf{W}\|_{1,2}^2, \quad (10.6)$$

where $\mathbf{U}$ is an orthogonal matrix projecting $\mathbf{X}$ to a new feature space. This method performs feature extraction rather than selection. Problem (10.6) is

generally not convex and the authors propose an alternated convex optimization to estimate the optimum. The solutions for each task may be sparse in the new feature space, but these combinations are unlikely to be sparse in the original feature space. Problem (10.6) naturally reduces to Problem (10.5) when $\mathbf{U}$ is fixed equal to the identity matrix rather than being optimized.

We also consider the $L_{1,\infty}$ group-regularizer as a specific instance of Problem (10.3) since we aim at producing models for each task that have a maximally common support across tasks while letting the model parameters to be specific for each task. In the context of multi-task regression, this regularizer has been used by [Liu et al. \(2009\)](#) in combination with a sum of square losses over each task to form the multi-task lasso. The optimization procedure proposed in that work is quite complex but presents interesting ideas. For example, it uses the concept of *warm start*: The learning algorithm first solves the problem with $\lambda = 0$, which comes to solve a simple regression on each task separately. The model weights found this way are then used as initial values for other values of λ .

An alternative approach to perform feature selection in a single-task setting has been proposed by [Weston et al. \(2003\)](#) and is detailed in Chapter 3. It uses a log-1-norm approximation to the minimization of the pseudo-norm L_0 :

$$\min_{\mathbf{w}} \sum_{j=1}^p \ln(|w_j| + \epsilon), \quad (10.7)$$

subject to

$$y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1, \quad \forall \mathbf{x}_i \text{ in a learning sample}, \quad (10.8)$$

where $0 < \epsilon \leq 1$ is a small constant preventing the objective to be ill-defined when some $|w_j|$ vanishes. Problem (10.7) is extended to perform partially supervised feature selection in Chapters 8 and 9. In its original form, it actually minimizes a hinge loss penalized by an approximation to the zero-pseudo-norm. We propose to consider as well the extension of such a regularizer to a multi-task setting.

Table 10.1 summarizes the 6 regularizers considered in this chapter, each one defining a specific instance of the general problem (10.3). This table complements Table 2.2 for single-task regularizers.

Table 10.1 Mixed-Norms. Top three: 1-norms. Bottom three: approximations to zero-pseudo-norms. The first index stands for the regularization over all features while the second index stands for regularization across tasks for a given feature.

$L_{1,1} =$	$ \mathbf{W} _{1,1}$	$=$	$\sum_{i=1}^p \sum_{k=1}^K w_{ki} $
$L_{1,2} =$	$ \mathbf{W} _{1,2}$	$=$	$\sum_{i=1}^p \sqrt{\sum_{k=1}^K w_{ki}^2}$
$L_{1,\infty} =$	$ \mathbf{W} _{1,\infty}$	$=$	$\sum_{i=1}^p \max w_i $
$L_{0,1} =$	$ \mathbf{W} _{0,1}$	$\approx$	$\sum_{i=1}^p \ln(\sum_{k=1}^K w_{ki} + \epsilon)$
$L_{0,2} =$	$ \mathbf{W} _{0,2}$	$\approx$	$\sum_{i=1}^p \ln(\sqrt{\sum_{k=1}^K w_{ki}^2} + \epsilon)$
$L_{0,\infty} =$	$ \mathbf{W} _{0,\infty}$	$\approx$	$\sum_{i=1}^p \ln(\max w_i + \epsilon)$

10.2 Multi-Task Penalized Logistic Regression

The previous section presents various regularizers that can be used to perform feature selection in a multi-task setting. We propose to apply such a regularizer to a set of logistic regression models, with one model per task. Logistic regression offers a probabilistic interpretation of the response variable which can be used as a confidence level of the prediction (Chapter 2). Besides, we propose in Section 10.3 a dedicated solver to estimate the parameters of such a multi-task logistic regression. The loss term $\mathcal{L}_M$ in multi-task is naturally defined as the sum of the losses over each task, with the corresponding objective function:

$$\begin{aligned} \min_{\mathbf{W}} \mathcal{L}_M(\mathbf{y}, \mathbf{W}, \mathbf{X}) + \lambda ||\mathbf{W}||_{i,j} \\ = \min_{\mathbf{W}} \sum_{k=1}^K \sum_{i=1}^{n_k} \log \left(1 + \exp^{-y_{ik} \langle \mathbf{x}_{ik}, \mathbf{w}_k \rangle} \right) + \lambda ||\mathbf{W}||_{i,j}. \end{aligned} \quad (10.9)$$

Problem (10.9) with $||\mathbf{W}||_{1,2}$ is related to the group-lasso for single-task learning (Yuan and Lin, 2006):

$$\min_{\mathbf{w}} \sum_{i=1}^n \mathcal{L}(y_i, \langle \mathbf{w}, \mathbf{x}_i \rangle) + \lambda ||\mathbf{w}||_{1,2}, \quad (10.10)$$

with the 2-norm part of the regularizer defined on subsets of dimensions which are not necessarily related to several tasks but to some predefined groups according to the applicative context. Typical examples are the grouping of dimensions corresponding to a range of frequencies in spectrometry

or of dummy binary variables resulting from the transformation of a multi-value categorical feature. The group-lasso favors the common selection or elimination of all the features within a group. The main difference between Problems (10.9) and (10.10) is that the loss term applies to several models in the former case but a single model in the latter case. Group-lasso is also traditionally used in regression with a square loss rather than a logistic loss for classification. The proposed approach also has a link to kernel learning since multiple kernel learning was shown equivalent to a specific form of group-lasso (Bach, 2008).

Multi-task penalized logistic regression, as formulated in Problem (10.9) with the $L_{1,\infty}$ norm, is similar to the F_∞ -norm Support Vector Machine for single-task with grouped variables (Zou and Yuan, 2008):

$$\min_{\mathbf{w}} \sum_{i=1}^n \mathcal{L}(y_i, \langle \mathbf{w}, \mathbf{x}_i \rangle) + \lambda \|\mathbf{w}\|_{1,\infty}. \quad (10.11)$$

Problem (10.9) reduces to Problem (10.11) by substituting the multi-task sum of losses with a single-task loss (a hinge loss in this specific case), by rewriting tasks input matrices in a block-diagonal form, and by concatenating respectively the response vectors $\mathbf{y}_k$ and the model parameters $\mathbf{w}_k$ for each task in single vectors:

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{X}_K \end{pmatrix}, \mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_K \end{pmatrix}, \mathbf{w} = \begin{pmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_K \end{pmatrix}. \quad (10.12)$$

This relationship holds provided that each group corresponds to a given feature across all tasks.

10.3 Optimization of a Sparse Multi-Task Logistic Regression

This section introduces an efficient solver for Problem (10.9) as an extension to a solver proposed by Shevade and Keerthi (2003) for a single-task penalized logistic regression, already mentioned in Chapter 3. Section 10.3.1 describes this single-task solver and Section 10.3.2 introduces our multi-task extension. Section 10.3.3 discusses the algorithm convergence and convexity issues. The solver computational complexity is analyzed in Section 10.3.4.

10.3.1 A Solver for a L_1 -Norm Penalized Logistic Regression

The solver introduced by Shevade and Keerthi (2003) is asymptotically convergent. It is based on defining the first order optimality conditions with respect to each dimension of the optimization problem. Since the L_1 regularizer is not differentiable with respect to w_j in 0, sub-gradients are used to define the optimality conditions. This solver maintains a set of active dimensions $\mathcal{A}$ containing the non-zero dimensions of the optimization problem, and a set of non-active dimensions $\overline{\mathcal{A}}$ containing the dimensions that are equal to zero. Initially, all dimensions are set to zero (i.e. $\mathcal{A} = \emptyset$ and $\overline{\mathcal{A}} = \{1, \dots, p\}$). A double loop optimization algorithm is implemented. The outer loop iterates until there are no dimensions violating the optimality conditions in $\overline{\mathcal{A}}$. In practice, the optimization is stopped when there are no dimensions violating the optimality conditions by more than a small tolerance value τ . In this loop, the dimension maximally violating the optimality conditions is removed from $\overline{\mathcal{A}}$ and included in $\mathcal{A}$. Next, a coordinate-wise strategy is applied in the inner loop. This strategy chooses from the active set the dimension maximally violating the optimality conditions and optimizes the problem with respect to that dimension. This non-linear and unconstrained optimization process is based on a mix of a gradient descent Newton method and a bisection technique that iteratively splits a closed interval in two sub-intervals of the same size. The sub-interval in which the optimization process performs the search is selected depending upon the sign of the derivative of the objective function evaluated at the mid-point. The inner loop ends when there are no more violating dimensions in the active set.

10.3.2 A Solver for a Sparse Multi-Task Logistic Regression

We extend the solver described in Section 10.3.1 to handle a multi-task penalized logistic regression with each of the six regularizers proposed. The pseudo-code of this solver is illustrated in Algorithm 3. The main modification with respect to the single-task solver previously described consists in considering the multi-task logistic loss in Problem (10.9) rather than a single logistic loss. Furthermore, the first order optimality conditions are those of Problem (10.9). Some additional modifications will be described next. We introduce first the optimality conditions and the corresponding measures of violation associated to the six possible regularizers.

Optimality Conditions and Violations for $\mathcal{R} = L_{1,1}$

In the case of the $L_{1,1}$ regularizer, the optimality conditions are as follows.

$$\frac{\partial \mathcal{L}_M}{\partial w_{0k}} = 0 \quad (10.13)$$

$$\lambda \geq \frac{\partial \mathcal{L}_M}{\partial w_{jk}} \geq -\lambda \quad w_{jk} = 0 \quad (10.14)$$

$$\frac{\partial \mathcal{L}_M}{\partial w_{jk}} = -\lambda \text{sign}(w_{jk}) \quad w_{jk} \neq 0 \quad (10.15)$$

The double inequality in the second condition comes from the non-differentiability of the 1-norm, i.e. the absolute value function, in case its argument is null. In this case, the sub-gradient is used. The sub-gradient is defined as the interval bounded by the derivatives at the limit of each side of a non-differentiability (Boyd and Vandenberghe, 2004). In the case of the absolute value function, the limits of the sub-gradient at $x = 0$ are defined as:

$$\text{sub}_{0-}(f) = \lim_{\delta \rightarrow 0-} \frac{f(\delta) - f(0)}{\delta} = -1 \quad \delta < 0, \quad (10.16)$$

$$\text{sub}_{0+}(f) = \lim_{\delta \rightarrow 0+} \frac{f(\delta) - f(0)}{\delta} = 1 \quad \delta > 0, \quad (10.17)$$

and the sub-gradient of f is defined as:

$$j \mid \{j \in [\text{sub}_{0-}(f); \text{sub}_{0+}(f)]\}. \quad (10.18)$$

The violations corresponding to the optimality conditions are defined as follows.

$$\left| \frac{\partial \mathcal{L}_M}{\partial w_{0k}} - 0 \right| \quad (10.19)$$

$$\max \left(\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \lambda, -\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \lambda, 0 \right) \quad \text{if } w_{jk} = 0 \quad (10.20)$$

$$\left| \frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \lambda \text{sign}(w_{jk}) \right| \quad w_{jk} \neq 0 \quad (10.21)$$

Optimality Conditions and Violations for $\mathcal{R} = L_{1,2}$

In the case of the $L_{1,2}$ regularizer, the optimality conditions are:

$$\frac{\partial \mathcal{L}_M}{\partial w_{0k}} = 0 \quad (10.22)$$

$$\lambda \geq \frac{\partial \mathcal{L}_M}{\partial w_{jk}} \geq -\lambda \quad \text{if } w_{jk} = 0 \quad (10.23)$$

$$\frac{\partial \mathcal{L}_M}{\partial w_{jk}} = -\lambda \frac{w_{jk}}{\sqrt{\sum_{l=1}^K w_{jl}^2}} \quad \text{otherwise} \quad (10.24)$$

and their violations are defined as:

$$\left| \frac{\partial \mathcal{L}_M}{\partial w_{0k}} - 0 \right| \quad (10.25)$$

$$\max \left(\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \lambda, -\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \lambda, 0 \right) \quad \text{if } w_{jk} = 0 \quad (10.26)$$

$$\left| \frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \lambda \frac{w_{jk}}{\sqrt{\sum_{l=1}^K w_{jl}^2}} \right| \quad \text{otherwise} \quad (10.27)$$

Optimality Conditions and Violations for $\mathcal{R} = L_{1,\infty}$

In the case of the $L_{1,\infty}$ regularizer the optimality conditions are:

$$\frac{\partial \mathcal{L}_M}{\partial w_{0k}} = 0 \quad (10.28)$$

$$\lambda \geq \frac{\partial \mathcal{L}_M}{\partial w_{jk}} \geq -\lambda \quad \text{if } w_{jk} = 0 \quad (10.29)$$

$$\frac{\partial \mathcal{L}_M}{\partial w_{jk}} = -\lambda \text{sign}(w_{jk}) \quad |w_{jk}| \text{ is single max} \quad (10.30)$$

$$0 \geq \frac{\partial \mathcal{L}_M}{\partial w_{jk}} \geq -\lambda \quad w_{jk} > 0 \wedge |w_{jk}| \text{ is multimax} \quad (10.31)$$

$$\lambda \geq \frac{\partial \mathcal{L}_M}{\partial w_{jk}} \geq 0 \quad w_{jk} < 0 \wedge |w_{jk}| \text{ is multimax} \quad (10.32)$$

$$\frac{\partial \mathcal{L}_M}{\partial w_{jk}} = 0 \quad |w_{jk}| \text{ is not max} \quad (10.33)$$

The larger number of conditions is explained by the fact that $L_{1,\infty}$ is non-differentiable not only when a weight is null, but also when one or several weights have the maximum value. The corresponding violations are defined as:

$$\left| \frac{\partial \mathcal{L}_M}{\partial w_{0k}} - 0 \right| \quad (10.34)$$

$$\max \left(\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \lambda, -\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \lambda, 0 \right) \quad \text{if } w_{jk} = 0 \quad (10.35)$$

$$\left| \frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \lambda \text{sign}(w_{jk}) \right| \quad |w_{jk}| \text{ is single max} \quad (10.36)$$

$$\max \left(-\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \lambda, \frac{\partial \mathcal{L}_M}{\partial w_{jk}}, 0 \right) \quad w_{jk} > 0 \wedge |w_{jk}| \text{ is multimax} \quad (10.37)$$

$$\max \left(\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \lambda, -\frac{\partial \mathcal{L}_M}{\partial w_{jk}}, 0 \right) \quad w_{jk} < 0 \wedge |w_{jk}| \text{ is multimax} \quad (10.38)$$

$$\left| \frac{\partial \mathcal{L}_M}{\partial w_{jk}} - 0 \right| \quad |w_{jk}| \text{ is not max} \quad (10.39)$$

Optimality Conditions and Violations for $\mathcal{R} = L_{0,1}$

In the case of the $L_{0,1}$ regularizer the optimality conditions are:

$$\frac{\partial \mathcal{L}_M}{\partial w_{0k}} = 0 \quad (10.40)$$

$$\frac{\lambda}{\epsilon + \sum_{l=1}^K |w_{jl}|} \geq \frac{\partial \mathcal{L}_M}{\partial w_{jk}} \geq -\frac{\lambda}{\epsilon + \sum_{l=1}^K |w_{jl}|} \quad w_{jk} = 0 \quad (10.41)$$

$$\frac{\partial \mathcal{L}_M}{\partial w_{jk}} = \frac{-\lambda}{\epsilon + \sum_{l=1}^K |w_{jl}|} \text{sign}(w_{jk}) \quad w_{jk} \neq 0 \quad (10.42)$$

and their violations are defined as:

$$\left| \frac{\partial \mathcal{L}_M}{\partial w_{0k}} - 0 \right| \quad (10.43)$$

$$\max \left(\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \frac{\lambda}{\epsilon + \sum_{l=1}^K |w_{jl}|}, -\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \frac{\lambda}{\epsilon + \sum_{l=1}^K |w_{jl}|}, 0 \right) \quad \text{if } w_{jk} = 0 \quad (10.44)$$

$$\left| \frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \frac{-\lambda}{\epsilon + \sum_{l=1}^K |w_{jl}|} \text{sign}(w_{jk}) \right| \quad w_{jk} \neq 0 \quad (10.45)$$

Optimality Conditions and Violations for $\mathcal{R} = L_{0,2}$

In the case of the $L_{0,2}$ regularizer the optimality conditions are:

$$\frac{\partial \mathcal{L}_M}{\partial w_{0k}} = 0 \quad (10.46)$$

$$\frac{\lambda}{\epsilon} \geq \frac{\partial \mathcal{L}_M}{\partial w_{jk}} \geq -\frac{\lambda}{\epsilon} \quad w_{jk} = 0 \quad (10.47)$$

$$\frac{\partial \mathcal{L}_M}{\partial w_{jk}} = \frac{-\lambda}{\epsilon + \sqrt{\sum_{l=1}^K w_{jl}^2}} \frac{w_{jk}}{\sqrt{\sum_{l=1}^K w_{jl}^2}} \quad \text{otherwise} \quad (10.48)$$

and their violations are defined as:

$$\left| \frac{\partial \mathcal{L}_M}{\partial w_{0k}} - 0 \right| \quad (10.49)$$

$$\max \left(\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \frac{\lambda}{\epsilon}, -\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \frac{\lambda}{\epsilon}, 0 \right) \quad \text{if } w_{jk} = 0 \quad (10.50)$$

$$\left| \frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \frac{-\lambda}{\epsilon + \sqrt{\sum_{l=1}^K w_{jl}^2}} \frac{w_{jk}}{\sqrt{\sum_{l=1}^K w_{jl}^2}} \right| \quad \text{otherwise} \quad (10.51)$$

Optimality Conditions and Violations for $\mathcal{R} = L_{0,\infty}$

In the case of the $L_{0,\infty}$, the situation with respect to non-differentiability is similar to the one of $L_{1,\infty}$. The optimality conditions are:

$$\frac{\partial \mathcal{L}_M}{\partial w_{0k}} = 0 \quad (10.52)$$

$$\frac{\lambda}{\epsilon} \geq \frac{\partial \mathcal{L}_M}{\partial w_{jk}} \geq -\frac{\lambda}{\epsilon} \quad w_{jk} = 0 \quad (10.53)$$

$$\frac{\partial \mathcal{L}_M}{\partial w_{jk}} = \frac{-\lambda}{|w_{jk}|+\epsilon} \text{sign}(w_{jk}) \quad |w_{jk}| \text{ is single max} \quad (10.54)$$

$$0 \geq \frac{\partial \mathcal{L}_M}{\partial w_{jk}} \geq \frac{-\lambda}{|w_{jk}|+\epsilon} \quad w_{jk} > 0 \wedge |w_{jk}| \text{ is multimax} \quad (10.55)$$

$$\frac{\lambda}{|w_{jk}|+\epsilon} \geq \frac{\partial \mathcal{L}_M}{\partial w_{jk}} \geq 0 \quad w_{jk} < 0 \wedge |w_{jk}| \text{ is multimax} \quad (10.56)$$

$$\frac{\partial \mathcal{L}_M}{\partial w_{jk}} = 0 \quad |w_{jk}| \text{ is not max} \quad (10.57)$$

and their violations are defined as:

$$\left| \frac{\partial \mathcal{L}_M}{\partial w_{0k}} - 0 \right| \quad (10.58)$$

$$\max \left(\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \frac{\lambda}{\epsilon}, -\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \frac{\lambda}{\epsilon}, 0 \right) \quad \text{if } w_{jk} = 0 \quad (10.59)$$

$$\left| \frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \frac{-\lambda}{|w_{jk}|+\epsilon} \text{sign}(w_{jk}) \right| \quad |w_{jk}| \text{ is single max} \quad (10.60)$$

$$\max \left(-\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \frac{\lambda}{|w_{jk}|+\epsilon}, \frac{\partial \mathcal{L}_M}{\partial w_{jk}}, 0 \right) \quad w_{jk} > 0 \wedge |w_{jk}| \text{ is multimax} \quad (10.61)$$

$$\max \left(\frac{\partial \mathcal{L}_M}{\partial w_{jk}} - \frac{\lambda}{|w_{jk}|+\epsilon}, -\frac{\partial \mathcal{L}_M}{\partial w_{jk}}, 0 \right) \quad w_{jk} < 0 \wedge |w_{jk}| \text{ is multimax} \quad (10.62)$$

$$\left| \frac{\partial \mathcal{L}_M}{\partial w_{jk}} - 0 \right| \quad |w_{jk}| \text{ is not max} \quad (10.63)$$

10.3.3 Convergence and Convexity Issues

Based on the optimality conditions, some additional particularities have to be considered for convergence purposes. Firstly, since the max function in $L_{i,\infty}$ is not differentiable not only in 0, but also when one or several weights have the maximum value, the coordinate descent of the inner optimization step is in those cases preceded by an additional steepest descent to still guarantee convergence to a global minimum. After this steepest descent, the original coordinate descent is not strictly required but showed to produce better solutions in practice. The rest of the reasoning follows the one of [Shevade and Keerthi \(2003\)](#) reviewed in Section 10.3.1.

Secondly, Problem (10.9) is the sum of a convex and differentiable loss function and a mixed-norm regularizer L_{ij} . However, as mentioned in Chapter 2, the logistic loss tends to behave in practice as a non-strictly convex function. When $i = 1$, the regularizers are also convex and Problem (10.9) is convex with a global optimum. The unicity of that global optimum is however not guaranteed. When $i = 0$, the regularizers are not convex and a unique solution for the optimum cannot be guaranteed, although the solution is still guaranteed to satisfy the optimality conditions.

10.3.4 Computational Complexity

The pseudo-code of our solver is illustrated in Algorithm 3. If the maximal number of iterations of the inner loop and of the steepest descent are bounded (in practice, to 1,000 iterations), the complexity of this solver is $\mathcal{O}(d p n)$ where $n = \sum_k n_k$, and n_k is the number of samples of task T_k , p is the number of features and d is the number of non-zero dimensions in the solution. The maximum value of d depends on the regularization constant: the larger the regularization constant, the smaller d . In a binary classification setting, d is at most $n - 1$ since, with (generalized) linear models, $n - 1$ dimensions are sufficient to shatter n (unaligned) points. Consequently the worst case complexity is $\mathcal{O}(p n^2)$, which is efficient since $n \ll p$ in our context.

Our algorithm presents similarities with the work on multi-task lasso (Liu *et al.*, 2009). First, our solver makes use of a warm start as well, but at the other possible extreme, i.e. when $\lambda \rightarrow \infty$, implying that all weights are equal to zero. This nuance is important since it makes possible in our case to deal only with a very small number of dimensions in the inner loop of our algorithm. Second, both algorithms adopt a coordinate-wise strategy. The global complexity of the solver proposed by Liu *et al.* (2009) is however not provided and seems at least as high as our.

10.4 Experiments and Results

Experimental evaluations of the proposed methods are conducted on the datasets described in Section 10.4.1. Section 10.4.2 presents various performance metrics to compare multi-task feature selection methods. Section 10.4.3 details the experimental protocols and analyzes the results. Section 10.4.3 evaluates specifically the multi-task penalized logistic regression with each of

Algorithm 3 Multi-Task Penalized Logistic Regression Solver**Input:** Tasks $T_k, k \in \{1, \dots, K\}$ **Input:** A choice for the regularizer $\mathcal{R}$ **Output:** W **Initialization:** Set $w_{jk} = 0, \forall j, k$, let $\overline{\mathcal{A}}$ be the set of all dimensions and $\mathcal{A} = \emptyset$.**while** Optimality violator exists in $\overline{\mathcal{A}}$ **do** Find the maximum violator $v \in \overline{\mathcal{A}}$. Update: $\overline{\mathcal{A}} = \overline{\mathcal{A}} \setminus \{v\}$ and $\mathcal{A} = \mathcal{A} \cup \{v\}$. Optimize (10.9) w.r.t the dimensions in $\mathcal{A}$. **if** $\mathcal{R}$ is $L_{1,\infty}$ or $L_{0,\infty}$ **then** Steepest descent on (10.9) w.r.t $w_v, \forall v \in \mathcal{A}$ **end if** **while** Optimality violator exists in $\mathcal{A}$ **do** Find the maximum violator $v \in \mathcal{A}$ Optimize (10.9) w.r.t w_v **end while****end while**

the six regularizers. Section 10.4.3 compares those results with the multivariate technique proposed by [Evgeniou and Pontil \(2004\)](#).

10.4.1 Datasets

We consider three gene expression microarray datasets related to binary classification of prostate cancer, the same as in previous chapter. The characteristics of the datasets are displayed in Table 9.2. More information on these datasets are available in Chapter D. The medical objective is to discriminate tumor from normal tissues. The first dataset (PRO1) contains data produced with the Affymetrix HGU95Av2 technology from normal and primary tumor tissues ([Singh et al., 2002](#)). The second dataset (PRO2) contains samples produced with HGU95Av2 from normal tissues as well as primary and metastatic tumor tissues ([Chandran et al., 2007](#)). The last dataset (PRO3) contains samples produced with Affymetrix HGU95A, a slightly older technology, from normal tissues, primary and metastatic tumor tissues ([Welsh et al., 2001](#)). Since two different microarray platforms are used, those datasets were reduced to the set of their 12,600 commonly defined features (genes).

10.4.2 Evaluation Metrics

Four metrics are used to assess the performances of multi-task classification and feature selection. Stability measures to which extent some sets of selected features (gene signatures) share common elements. Those sets can typically be produced by running a feature selection method on different samplings of the data. The stability between two sets of selected features $\mathbf{S}_1$ and $\mathbf{S}_2$ is measured by the Jaccard stability index $\mathcal{J}(\mathbf{S}_1, \mathbf{S}_2)$ as defined by Equation (4.3). The greater the index value the larger the proportion of commonly selected features in the various sets. The Jaccard index is well defined even when $\mathbf{S}_1$ and $\mathbf{S}_2$ do not have the same cardinality. This property motivates our choice for this particular index since there is no exact control on the number of features selected by the methods we evaluate, unlike in the three previous chapters. We consider two types of stability: between-tasks stability ($\mathcal{J}_{BTS}$), which measures how similar signatures are across several tasks, and within-task stability ($\mathcal{J}_{WTS}$), which measures how similar signatures are on different sub-samplings of the data for a given task. The third criterion is the classification performance, assessed with the balanced classification rate (BCR) on each task. The fourth criterion is the sparsity of the selected signatures, as measured by the number of features they contain.

10.4.3 Evaluation Protocols and Results

This section presents the protocols used for the various experiments reported, in the continuity of protocols used in previous chapters. Section 10.4.3 presents the main evaluation of our penalized logistic regression in a multi-task setting with each of the six regularizers. The case $L_{1,1}$ is actually a set of independent single-task problems and serves as a baseline to assess the benefits of real multi-task approaches. Section 10.4.3 compares those results with the multi-task technique due to [Evgeniou and Pontil \(2004\)](#).

The original data described in Table 9.2 is preprocessed as follows. The Affymetrix datasets are normalized according to the RMA summarization process. The objective of this preprocessing is to make all gene expression profiles within a dataset comparable and to get rid of experimental bias such as microarray scanner variability ([Irizarry et al., 2003](#)).

An additional normalization is performed to center each feature on a zero median through an IQR-normalization procedure. The expression of this

normalization step is modified to suit to a multi-task setting, in contrast with Equation (6.1). The normalized expression value x'_{ijk} is defined as follows:

$$x'_{ijk} \leftarrow \frac{x_{ijk} - \mu(\mathbf{x}_{jk})}{IQR(\mathbf{x}_{jk})/1.349},$$

where x_{ijk} is the original expression value of dimension j from sample i in task k , $\mu(\mathbf{x}_{jk})$ is the median of expression of this feature over all training samples of task k , and $IQR(\mathbf{x}_{jk})$ stands for the feature specific interquartile range (Tukey, 1977). The 1.349 factor is explained in Chapter 6.

Multi-Task Penalized Logistic Regression

Two hundred random subsamplings, containing 90% of the samples, are generated (without replacement) from each dataset to define training sets. The 10% remaining samples serve as test sets. For each subsampling, IQR feature normalization is run on the training set and normalization parameters are applied to the test samples.

Feature selection is performed on each training set by solving Problem (10.9) with a given regularization constant λ . Once features are selected by keeping only the non-zero coefficients of the respective weight vectors, a logistic ridge regression¹ model is estimated on each task separately. The purpose of this step is to build a final model on each task from the features selected across tasks. A logistic regression using a L_2 penalization is chosen here because sparsity has already been enforced in the previous step but some regularization may still be useful to prevent over-fitting. The regularization constant λ_* of the logistic ridge regression models is optimized for each task via cross-validation on the training set, a fast procedure on the few selected dimensions. The BCR of the final model for each task is then evaluated on the corresponding test sets. An average BCR over the 200 resamplings and over the three tasks is reported.

The two types of stabilities ($\mathcal{J}_{BTS}$ and $\mathcal{J}_{WTS}$) are estimated on the signatures produced from the 200 resamplings. For each of the 200 resamplings, $\mathcal{J}_{BTS}$ is evaluated from the signatures from the 3 pairs of tasks considered and averaged. The reported $\mathcal{J}_{BTS}$ are averages over the 200 resamplings. $\mathcal{J}_{WTS}$ is measured for every single task over all pairwise combinations of the 200 signatures of a given task via the Jaccard index. The reported $\mathcal{J}_{WTS}$ are

¹logistic regression with an L_2 penalty

averages over the 3 tasks.

Problem (10.9) is solved for $\lambda = \{0.01, 0.1, 1, 10\}$. We report results for different values of λ in order to compare the sparsity, stability and prediction performance for various settings. The solver convergence is set to $\tau = 10^{-5}$ and the $L_{0,j}$ regularizers use $\epsilon = 1$. Results for the prostate datasets with $\lambda = \{0.01, 0.1, 1, 10\}$ are graphically displayed in Figure 10.1 and more detailed in Tables 10.2 to 10.5. All regularizers produce sparse solutions.

The $L_{1,2}$ regularization offers the best BCR overall but tends to produce less sparse solutions than the other approaches. By choosing an appropriate regularization constant, it can however be forced to produce solutions that are as sparse as those produced using $L_{1,\infty}$ (see in particular the plot of BCR versus sparsity in Figure 10.1 corresponding to $\lambda = 10$). In this case, the achieved BCR are similar with both regularizers.

A high between-task stability is obtained with $L_{1,\infty}$ closely followed by $L_{1,2}$. However a higher stability is easier to obtain for less sparse solutions. Hence we further compare the between-task stability with signatures restricted to the top-30 features of each signature produced with $L_{1,\infty}$ and $L_{1,2}$, which is the typical size of the sparsest solutions obtained with both regularizers. Results reports in Table 10.6 clearly shows the benefit of the $L_{1,\infty}$ norm in this regard.

The $L_{0,j}$ mixed-norms produce the sparsest solutions and offer the fastest solver convergence (up to three times faster) at the price of lower predictive performance (BCR). This reduced time complexity is likely due to the higher sparsity achieved.

The $L_{1,1}$ regularization achieves the best within-task stability but logically offers the lowest between-task stability overall. The situation is more subtle for $L_{0,1}$. High values of λ lead in this case to a very low between-task stability, like with $L_{1,1}$, while low values of λ lead to a better between-task stability, although still much below those obtained with the other regularizers.

Table 10.2 Average classification performance (BCR averaged over 3 tasks).

	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
$L_{1,1}$	86.84 ± 4.24	87.70 ± 3.77	87.52 ± 3.96	83.41 ± 8.44
$L_{1,2}$	90.22 ± 0.49	90.72 ± 1.35	91.13 ± 0.93	85.93 ± 4.54
$L_{1,\infty}$	85.75 ± 1.70	84.69 ± 1.40	86.82 ± 1.89	85.74 ± 2.46
$L_{0,1}$	80.24 ± 0.90	83.51 ± 0.48	81.66 ± 6.15	82.59 ± 7.36
$L_{0,2}$	80.78 ± 1.27	82.04 ± 1.26	84.27 ± 1.29	82.08 ± 6.28
$L_{0,\infty}$	79.48 ± 0.54	81.80 ± 0.72	84.81 ± 1.47	83.04 ± 4.36

Table 10.3 Average sparsity (number of selected features averaged over 3 tasks).

	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
$L_{1,1}$	31.75 ± 15.51	30.14 ± 14.3	26.76 ± 11.95	15.12 ± 2.06
$L_{1,2}$	67.20 ± 15.75	63.21 ± 0.19	54.30 ± 3.06	32.39 ± 0.03
$L_{1,\infty}$	39.90 ± 1.91	38.22 ± 0.22	37.80 ± 0.00	26.31 ± 0.003
$L_{0,1}$	12.69 ± 2.60	11.40 ± 1.96	11.16 ± 1.19	8.85 ± 1.71
$L_{0,2}$	19.09 ± 0.53	23.15 ± 0.31	25.36 ± 0.33	21.80 ± 0.06
$L_{0,\infty}$	17.92 ± 0.07	22.27 ± 0.02	28.31 ± 0.03	22.74 ± 0.008

Comparative Results

We report in this section comparative results with the multi-task approach proposed by [Evgeniou and Pontil \(2004\)](#) solving Problem (10.2). This approach is tested with the same experimental protocol as described in Section 10.4.3. This method requires two regularization constants λ_1 and λ_2 , controlling the level of model sharing across tasks. Experiments reported in Table 10.7 with λ_1 and $\lambda_2 \in \{0.01, 0.1, 1, 10\}$ show that BCR results are essentially the same for a fixed value of the ratio $\mu = \lambda_1 / \lambda_2$. High predictive performance (BCR) values are observed but this technique does not perform any feature selection. Since the 12,600 features are always used, the stability is trivially maximal but the models are not sparse at all.

We assess the performance of this approach by further selecting the top 30 most significant features obtained from each resampling of each task independently. This feature ranking is based on the absolute value of the w_{jk} 's estimated after solving Problem (10.2). A logistic ridge regression is then

Table 10.4 Average between-task stability ($\mathcal{J}_{BTS}$ averaged over 200 resamplings).

	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
$L_{1,1}$	0.53 ± 0.55	0.55 ± 0.60	0.61 ± 0.71	0.48 ± 0.69
$L_{1,2}$	99.09 ± 1.56	99.10 ± 0.25	99.41 ± 0.06	99.61 ± 0.04
$L_{1,\infty}$	94.96 ± 4.34	99.34 ± 0.56	100 ± 0.00	99.87 ± 0.02
$L_{0,1}$	45.12 ± 9.40	23.48 ± 4.87	11.98 ± 1.34	1.47 ± 0.14
$L_{0,2}$	95.95 ± 2.72	97.02 ± 1.21	96.36 ± 1.12	98.91 ± 0.16
$L_{0,\infty}$	99.12 ± 2.81	99.03 ± 0.16	99.01 ± 0.10	99.59 ± 0.03

Table 10.5 Average within-task stability ($\mathcal{J}_{WTS}$ averaged over 3 tasks).

	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
$L_{1,1}$	35.66 ± 4.95	36.79 ± 4.89	38.81 ± 4.92	46.25 ± 3.91
$L_{1,2}$	34.29 ± 0.05	35.67 ± 0.01	37.66 ± 0.05	46.56 ± 0.11
$L_{1,\infty}$	21.52 ± 0.30	21.98 ± 0.05	25.73 ± 0.00	41.34 ± 0.04
$L_{0,1}$	15.55 ± 0.98	12.66 ± 1.66	16.19 ± 2.86	40.71 ± 2.11
$L_{0,2}$	20.10 ± 0.35	16.65 ± 0.06	19.02 ± 1.22	37.74 ± 0.15
$L_{0,\infty}$	27.35 ± 0.07	21.13 ± 0.05	20.75 ± 0.00	40.57 ± 0.04

built for each task on the selected 30 features, with a regularization constant λ_* estimated via cross-validation. Such a protocol let us report comparative performances at the same levels of sparsity as those reported in Section 10.4.3. The BCR, stability ($\mathcal{J}_{BTS}$ and $\mathcal{J}_{WTS}$) obtained with this method are displayed in Figure 10.2 and detailed in Tables 10.8 to 10.10.

The observed BCRs are much lower than when all features are used (approximately by 8% to 14% in absolute terms). The specific choices of meta-parameters values also have more impact. The BCR is least degraded when $\mu = 1$ or $\mu = 10$. The between-task stabilities ($\mathcal{J}_{BTS}$) are generally low, except when μ is high as theoretically expected. The top performing setting for maximizing the between task stability also contradicts the optimal values chosen to optimize BCR or the within-task stability ($\mathcal{J}_{WTS}$).

Without feature selection, the method of Evgeniou and Pontil (2004) produces higher BCR than any solution provided by the multi-task penalized

Table 10.6 Average between-task stability on restricted signatures. Those results are obtained by restricting signatures obtained through $L_{1,2}$ and $L_{1,\infty}$ regularizations to the top 30 features of each signature. These top 30 features correspond to the 30 largest absolute weight values.

	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
$L_{1,2}$	26.16	29.09	37.15	86.21
$L_{1,\infty}$	64.10	70.40	70.95	99.68

Table 10.7 Average classification performance (BCR) with models obtained according to (Evgeniou and Pontil, 2004). Missing values correspond to very high values of the $\mu = \lambda_1/\lambda_2$ ratio leading to convergence problems.

	$\lambda_1 = 0.01$	$\lambda_1 = 0.1$	$\lambda_1 = 1$	$\lambda_1 = 10$	$\lambda_1 = 100$
$\lambda_2 = 0.01$	93.12±3.36	93.05±2.48	92.98±2.54	–	–
$\lambda_2 = 0.1$	92.60±3.51	93.12±3.36	93.05±2.48	92.98±2.54	–
$\lambda_2 = 1$	92.44±3.50	92.60±3.51	93.12±3.36	93.05±2.48	92.98±2.54
$\lambda_2 = 10$	92.46±3.51	92.44±3.50	92.60±3.51	93.12±3.36	93.05±2.48
$\lambda_2 = 100$	92.42±3.49	92.46±3.51	92.44±3.50	92.60±3.51	93.12±3.36

logistic regression. At the same level of sparsity (i.e. with 30 features) however all the $L_{1,j}$ variants of the multi-task penalized logistic regression achieve comparable and mainly better BCR performances. The $L_{i,2}$ and $L_{i,\infty}$ regularizers always offer higher values of between-task stability ($\mathcal{J}_{BTS}$).

Table 10.8 Selection of top 30 features after solving the problem of Evgeniou and Pontil (2004). Average classification performance (BCR averaged over 3 tasks).

	$\lambda_1 = 0.01$	$\lambda_1 = 0.1$	$\lambda_1 = 1$	$\lambda_1 = 10$	$\lambda_1 = 100$
$\lambda_2 = 0.01$	85.33±3.31	85.44±2.27	83.92±2.88	–	–
$\lambda_2 = 0.1$	79.12±9.29	85.33±3.31	85.44±2.27	83.92±2.88	–
$\lambda_2 = 1$	78.79±8.01	79.12±9.29	85.33±3.31	85.44±2.27	83.92±2.88
$\lambda_2 = 10$	78.36±8.82	78.79±8.01	79.12±9.29	85.33±3.31	85.44±2.27
$\lambda_2 = 100$	78.51±8.63	78.36±8.82	78.79±8.01	79.12±9.29	85.33±3.31

Table 10.9 Selection of top 30 features after solving the problem of [Evgeniou and Pontil \(2004\)](#). Average between-task stability ($\mathcal{J}_{BTS}$ averaged over 200 resamplings).

	$\lambda_1 = 0.01$	$\lambda_1 = 0.1$	$\lambda_1 = 1$	$\lambda_1 = 10$	$\lambda_1 = 100$
$\lambda_2 = 0.01$	18.73±14.56	69.03±10.00	94.72±2.01	–	–
$\lambda_2 = 0.1$	2.71±2.66	18.73±14.56	69.03±10.00	94.72±2.01	–
$\lambda_2 = 1$	2.16±2.27	2.71±2.66	18.73±14.56	69.03±10.00	94.72±2.01
$\lambda_2 = 10$	2.16±2.24	2.16±2.27	2.71±2.66	18.73±14.56	69.03±10.00
$\lambda_2 = 100$	2.16±2.24	2.16±2.24	2.16±2.27	2.71±2.66	18.73±14.56

Table 10.10 Selection of top 30 features after solving the problem of [Evgeniou and Pontil \(2004\)](#). Average within-task stability ($\mathcal{J}_{WTS}$ averaged over 3 tasks).

	$\lambda_1 = 0.01$	$\lambda_1 = 0.1$	$\lambda_1 = 1$	$\lambda_1 = 10$	$\lambda_1 = 100$
$\lambda_2 = 0.01$	44.33±3.13	39.30±1.14	39.02±0.14	–	–
$\lambda_2 = 0.1$	43.89±7.31	44.33±3.13	39.30±1.14	39.02±0.14	–
$\lambda_2 = 1$	45.08±7.13	43.89±7.31	44.33±3.13	39.30±1.14	39.02±0.14
$\lambda_2 = 10$	45.07±7.12	45.08±7.13	43.89±7.31	44.33±3.13	39.30±1.14
$\lambda_2 = 100$	45.06±7.16	45.07±7.12	45.08±7.13	43.89±7.31	44.33±3.13

10.5 Discussion

This chapter presents a method for feature selection in the context of multi-task classification. The general objective is to estimate (generalized) linear models for each task while favoring a common sparse support across tasks. This problem is formulated as the optimization of a sum of logistic losses combined with a sparsity-enforcing regularizer. We study 6 mixed-norm regularizers $L_{i,j}$, where $i \in \{0,1\}$ and $j \in \{1,2,\infty\}$, respectively over the input features and the tasks. An efficient solver for all variants of this problem is proposed. This solver has a time complexity in $\mathcal{O}(pn^2)$, with p the initial number of features and n the number of samples. This is particularly interesting when $n \ll p$, a frequent situation in several applicative contexts such as the analysis of genomic data.

The evaluation of our method is done through a joint analysis of classification performance and feature selection stability (between and within tasks), which is not covered by most related work on multi-task learning. Experiments reported on several microarray datasets show that the $L_{1,2}$ regularization offers the best classification performance but the $L_{1,\infty}$ regularizer

produces sparser solutions. When regularization constants are chosen such that the same number of features is obtained with both regularizers, they offer the same classification performance but the $L_{1,\infty}$ regularizer is better at favoring the selection of common features across tasks. The $L_{0,j}$ mixed-norms offer the sparsest solutions and are faster to optimize but often at the cost of lower classification performances.

In the present work, the loss term for each task is proportional to the number of samples in each task but alternative weighting between the various loss terms could also be considered. More fundamentally, the current approach relies on a specific choice of a regularizer to favor more or less a common support across tasks while letting the model parameters be specific to each task. It would be worth investigating further approaches to model explicitly the dependence between tasks and to review the optimization problem accordingly. This issue is related to the non-i.i.d nature of multi-task learning. Theoretical works along those lines have been proposed in (Baxter, 2000; Ben-David and Schuller Borbely, 2008) but they require further extensions to formulate and to solve adequate optimization problems in very large feature spaces with few samples, as discussed in Chapter 9.

Multi-task penalized logistic regression can also be used to achieve feature selection in a multi-class setting. This is the topic of the next section.

10.6 Feature Selection for Multi-Class Classification

So far, only binary classification problems have been touched by this thesis. In most cases, binary classification techniques are able to tackle multi-class problems as well. By multi-class we mean more than two classes. In practice, one has two approaches to do so:

- Transform a k -class problem in k binary classification problems according to the *one-against-the-rest* paradigm. In this paradigm, each class is opposed to the rest of the classes together, constituting the second class.
- Transform a k -class problem in $k \times (k - 1) / 2$ binary classification problems according to the *one-against-one* paradigm. In this paradigm, each class is opposed to another class, one at a time, and all combination are considered.

The benefit of the first approach is to limit the number of discrimination tasks to k . The drawback of that first approach is to be likely to consider strongly unbalanced datasets, since the second class is compounded of $k - 1$ classes. As far as just classification is concerned, those approaches are fine in practice. When feature selection is considered, another issue is raised: It is likely there will not be a single signature for the k classes, but well k or $k \times (k - 1)/2$ different signatures for each binary sub-task, according to which paradigm is chosen.

10.6.1 Multi-Task Penalized Logistic Regression for Feature Selection in Multi-Class Settings

The end of this chapter evaluates the use of the multi-task penalized logistic regression as a way to address feature selection for a multi-class problem. We consider here the lymphoma dataset (LYM1) containing three classes corresponding to various tissue types. This dataset was described, used and pre-processed in (Diaz-Uriarte and de Andres, 2006). Its main characteristics are displayed in Table 10.11. Three sub-tasks are defined to discriminate one class versus the rest:

- Sub-task 1: class 1 (42 samples) vs. class 2 and class 3 (20 samples),
- Sub-task 2: class 2 (9 samples) vs. class 1 and class 3 (53 samples),
- Sub-task 3: class 3 (11 samples) vs. class 1 and class 2 (51 samples).

Table 10.11 Dataset used for multi-class feature selection validation. Columns respectively show the dataset name, the number of samples for each class and the number of features.

Name	Class 1	Class 2	Class3	# Features
LYM1	42	9	11	4,026

The experimental protocol and metrics are the same as the one described in Section 10.4. Features are selected for each sub-task based on the model estimated by a multi-task penalized logistic regression solving Problem (10.9). A logistic ridge regression is estimated next on each task separately and restricted to the features selected across sub-tasks. Figure 10.3 and Tables 10.12 to 10.15 report the performances obtained in terms of $\mathcal{J}_{BTS}$, $\mathcal{J}_{WTS}$, BCR and sparsity. The $L_{1,1}$ version is equivalent to the learning of a different model for

each sub-task.

Those results show that a multi-task logistic regression can indeed be used to address a multi-class problem (here on a single task) while selecting a unique subset of features for all sub-tasks (especially with the $L_{i,2}$ or $L_{i,\infty}$ regularizers). Such a stability between sub-tasks is clearly not optimized with $L_{1,1}$. The results obtained with the $L_{0,1}$ regularizer depends much on the specific regularization constant chosen. The within-task stability is optimal with $L_{1,1}$ on average (i.e. over all the sub-tasks) but with a significantly larger variance than with the other regularizers (cf. Table 10.14). The sparsest solutions are obtained with the $L_{0,j}$ regularizers as already observed in Section 10.4.3 in the context of multi-task learning.

Table 10.12 Average classification performance (BCR averaged over 3 sub-tasks).

	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
$L_{1,1}$	87.58±7.37	87.40±6.95	87.10±6.30	88.28±5.91
$L_{1,2}$	88.52±5.43	88.07±4.82	88.33±5.03	89.51±5.90
$L_{1,\infty}$	86.38±3.99	87.65±3.75	87.62±4.99	88.42±3.45
$L_{0,1}$	86.08±3.33	85.79±4.06	88.74±4.11	83.53±1.39
$L_{0,2}$	85.98±3.04	86.14±2.86	87.64±4.30	86.78±4.80
$L_{0,\infty}$	85.73±3.82	85.52±4.13	86.55±5.35	85.55±3.72

Table 10.13 Average between-task Stability ($\mathcal{J}_{BTS}$ averaged over 200 resamplings).

	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
$L_{1,1}$	5.29±4.25	4.76±4.50	4.50±4.47	4.50±3.95
$L_{1,2}$	99.12±0.21	99.26±0.07	99.38±0.20	100.00±0.00
$L_{1,\infty}$	99.95±0.02	100.00±0.00	100.00±0.00	100.00±0.00
$L_{0,1}$	72.92±3.25	68.61±4.30	53.12±7.93	32.62±19.03
$L_{0,2}$	99.31±0.32	99.28±0.26	98.94±0.19	99.87±0.06
$L_{0,\infty}$	99.92±0.04	100.00±0.00	100.00±0.00	100.00±0.00

Table 10.14 Average within-task stability ($\mathcal{J}_{WTS}$ averaged over 3 sub-tasks).

	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
$L_{1,1}$	31.76 $\pm$ 4.14	32.36 $\pm$ 4.43	32.66 $\pm$ 5.43	39.96 $\pm$ 14.56
$L_{1,2}$	26.15 $\pm$ 0.05	27.21 $\pm$ 0.00	30.40 $\pm$ 0.03	38.88 $\pm$ 0.00
$L_{1,\infty}$	16.40 $\pm$ 0.00	17.27 $\pm$ 0.00	22.70 $\pm$ 0.00	35.54 $\pm$ 0.00
$L_{0,1}$	13.35 $\pm$ 1.88	13.23 $\pm$ 2.24	14.82 $\pm$ 4.97	23.81 $\pm$ 7.63
$L_{0,2}$	15.69 $\pm$ 0.04	15.99 $\pm$ 0.00	18.62 $\pm$ 0.06	28.55 $\pm$ 0.00
$L_{0,\infty}$	16.26 $\pm$ 0.02	16.59 $\pm$ 0.00	20.13 $\pm$ 0.00	31.32 $\pm$ 0.00

Table 10.15 Sparsity (number of selected features averaged over 3 sub-tasks).

	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
$L_{1,1}$	21.51 $\pm$ 8.39	19.76 $\pm$ 8.01	16.48 $\pm$ 7.24	7.91 $\pm$ 4.34
$L_{1,2}$	22.55 $\pm$ 0.06	20.03 $\pm$ 0.06	16.04 $\pm$ 0.05	7.81 $\pm$ 0.00
$L_{1,\infty}$	15.89 $\pm$ 0.00	15.57 $\pm$ 0.00	13.00 $\pm$ 0.00	7.10 $\pm$ 0.00
$L_{0,1}$	6.82 $\pm$ 0.19	6.43 $\pm$ 0.26	5.31 $\pm$ 0.77	3.07 $\pm$ 0.79
$L_{0,2}$	8.16 $\pm$ 0.03	8.12 $\pm$ 0.02	7.83 $\pm$ 0.03	4.98 $\pm$ 0.00
$L_{0,\infty}$	9.19 $\pm$ 0.00	9.12 $\pm$ 0.00	9.04 $\pm$ 0.00	5.44 $\pm$ 0.00

10.6.2 Discussion

The multi-task penalized logistic regression proposed in this chapter is shown to be also able to perform feature selection in a multi-class setting with the benefit of relying on the same feature subset to discriminate between any class versus the others, especially with the $L_{i,2}$ or $L_{i,\infty}$ regularizers. This is not offered by the traditional single-task *one-against-the-rest* paradigm, here evaluated via the $L_{1,1}$ regularizer. The application of the multi-task penalized logistic regression to multi-class problems is straightforward, and does not require any change to the objective function nor to the solver.

Perspectives

The application of our multi-task approach to multi-class problems is evaluated on a single task with several classes. It could also be used in a multi-task and multi-class setting without changing anything.

Several issues are not covered by this preliminary study of the multi-task penalized logistic regression applied to the multi-class setting.

First, what transformation paradigm would be the best in general: one-against-one or one-against-the-rest? This raises the question of the behavior of our multi-task approach as a function of the number of tasks, and the number of samples by tasks.

Second, how this use of the multi-task penalized logistic regression does deal with conflict resolution? Indeed, it might be the case, for example, that a sample truly belonging to class 1 is correctly classified as belonging to class 1 in the task *class-1-versus-classes-2-and-3*, but classified as belonging to class 2 in the task *class-2-versus-classes-1-and-3*. What is then the final label that will be affected to that sample within our model? This preliminary study does not address this issue. But several comments should be made in that respect:

1. In the training phase, the optimization of the objective function in Equation (10.9) tends to avoid conflicts. If the classification over two tasks differ for a given sample, one out of the two is wrong, and the loss is thus increased. It is true however that conflicts might still arise.
2. We stress that the main objective of this approach is to perform feature selection leading to a single signature for multiple classes. In our experimental setting, the classification phase that follows this selection step does not take conflict resolution into account. It is however easy to overcome this limitation. A first possibility is to use the probabilistic output of the logistic regression and finally assign the problematic sample to the most probable class. Another solution is to use a multi-class logistic regression (Bishop, 2006) on the initial multi-class dataset restricted to the set of features selected with our method.

Those questions should be addressed by further experiments.

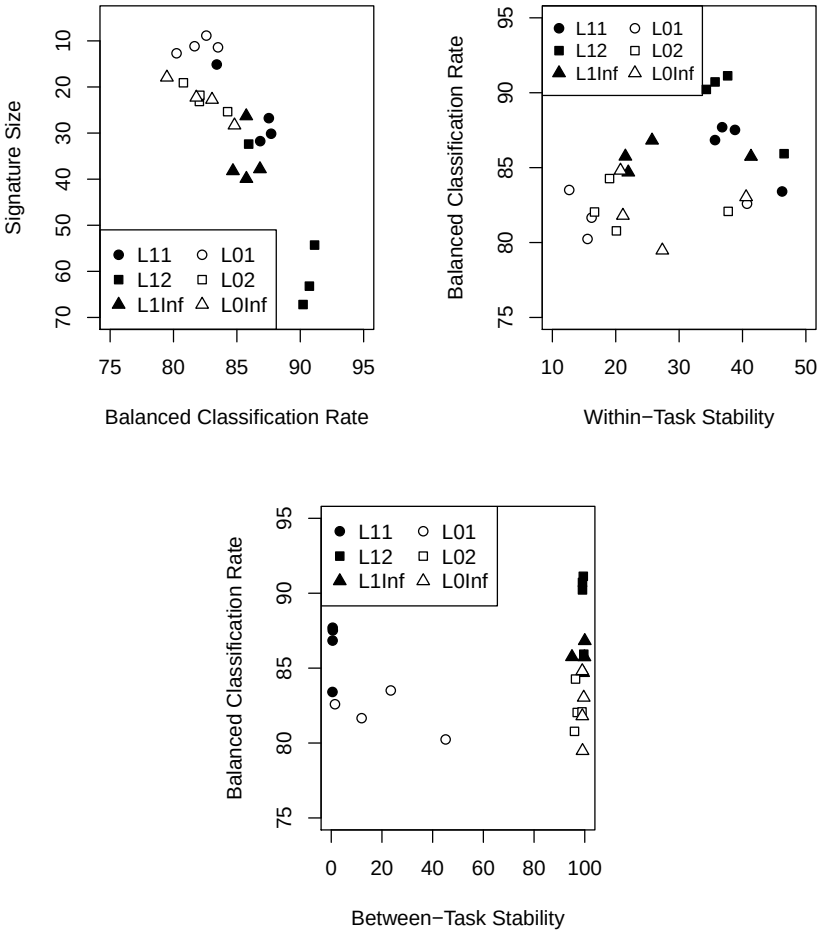


Figure 10.1 Performance of multi-task penalized logistic regression with 6 different sparsity-enforcing regularizers along 4 different criteria for $\lambda = \{0.01, 0.1, 1, 10\}$. Specific performances including standard deviations for each λ are further detailed in Tables 10.2 to 10.5. Each plot compares 2 criteria. Each point is an average over the 3 tasks. The axes are chosen such that the optimal performance always corresponds to the up-right corner of each plot.

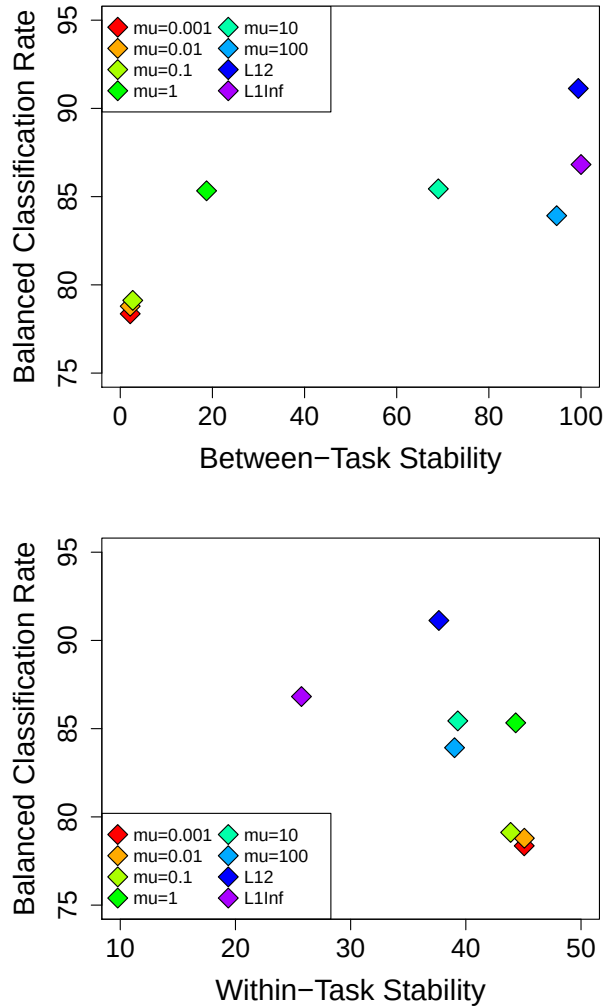


Figure 10.2 Selection of top 30 features after solving Problem (10.2). Performance of the method of [Evgeniou and Pontil \(2004\)](#) with 7 different $\mu = \lambda_1/\lambda_2$ ratios along 3 different criteria for $\mu = \{0.0001, 0.001, 0.01, 0.1, 1, 10, 100\}$. Each plot compares 2 criteria. Each point is an average over the 3 tasks. Results obtained with the multi-task logistic regression regularized with $L_{1,2}$ or $L_{1,\infty}$ are reported for $\lambda = 1$. The axes are chosen such that the optimal performance always corresponds to the up-right corner of each plot.

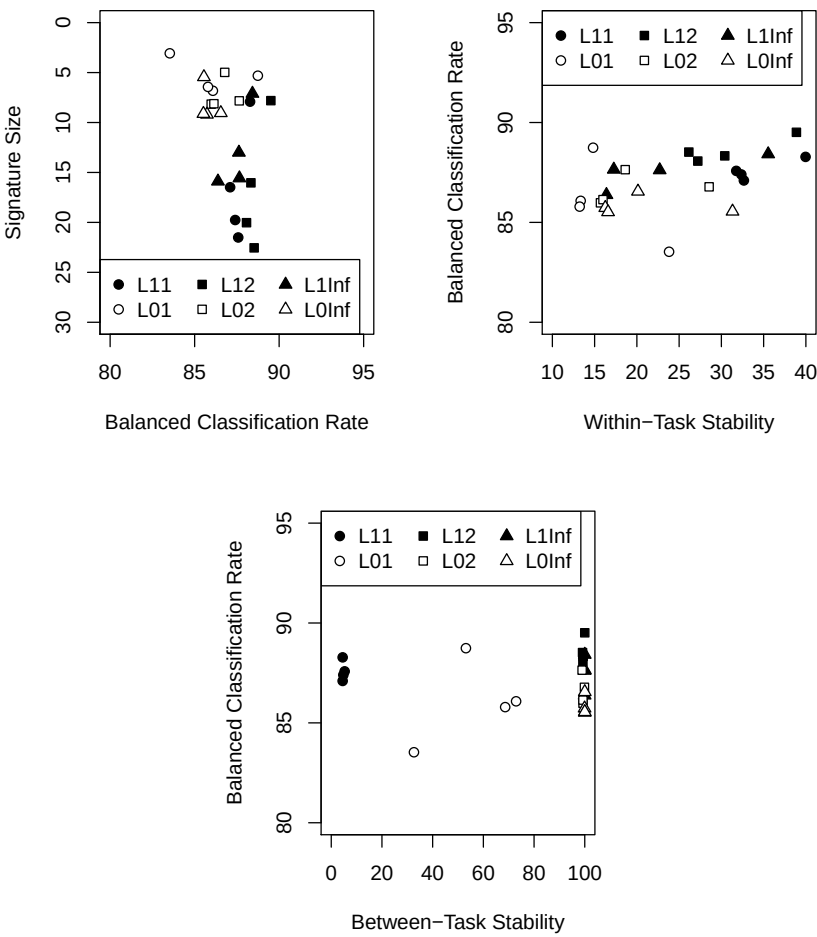


Figure 10.3 Performance of multi-task Penalized Logistic Regression with 6 different sparsity-enforcing regularizers along 4 different criteria for $\lambda_1 = \{0.01, 0.1, 1, 10\}$. Each plot compares 2 criteria. Each point is an average over the 3 sub-tasks. The axes are chosen such that the optimal performance always corresponds to the up-right corner of each plot.

Conclusion

11

11.1 Contributions

This thesis proposes several original methods to improve the quality of highly sparse feature selection and classification with linear models for binary discrimination tasks in a context where only a few high dimensional points are available. It reaches those objectives via different means, summarized below.

Ensemble methods for feature selection are inspired from ensemble methods for classification. This approach is based on an aggregation of feature selections embedded in linear SVM. It leads to higher stability and classification performances, at the cost of computing more models in order to build the consensus.

Ensemble methods is a way to get better results without adding any information to a given dataset. The other approaches proposed in this thesis do make use of extra information, in various ways. The first proposed method uses expert prior knowledge in the form of a guess (maybe inaccurate, maybe incomplete) over the relevance of the features. If some dimensions are suspected to play a role in the data generative process, the selection of those dimensions is favored (but not forced). This prior knowledge is incorporated in an iterative selection process embedded in a linear model under discriminative constraints. The prior knowledge is used to rescale the constraints input. This approach leads to a gain in stability, and sometimes to a gain in classification performances. It has been shown to be able to depart from the prior knowledge if it does not help to improve the quality of the selection and of the classification. This technique also allows biologists

and analysts to have more interactions. This method named PS- L_2 -AROM can be used in another context, where prior knowledge may be lacking, but where similar datasets are available. In this case, these similar datasets are used to generate knowledge about the feature relevance, and this knowledge is transferred to the dataset of interest via PS- L_2 -AROM. The application of this technique leads to an improvement both in stability and classification performances.

Transfer learning is not the only framework where several datasets can be used together. The next move in that direction is multi-task learning. In that setting, selection and classification are performed simultaneously on several related datasets. Differences between the considered datasets, such as the method of production or some experimental conditions, may prevent the simple pooling of the datasets in one larger dataset. This thesis proposes a multi-task learning approach to select features that are relevant for several tasks at once. It does so by using a model compounded of a multi-task logistic loss function and a group-regularizer in the form of a matrix-norm $L_{i,j}$. Six versions of such a regularizer are considered where $i \in \{0, 1\}$ and $j \in \{1, 2, \infty\}$. We introduce a competitive optimization algorithm which can be used with any of those regularizers. Its time complexity is linear in terms of the number of dimensions. In the case of multi-task learning, two types of stabilities may be defined: between-tasks and within-task stability. Sparsity and classification performances are the two other criteria considered. Experiments show that $L_{1,2}$ regularization offers the best classification performances while the $L_{1,\infty}$ regularizer leads to sparser solutions. When the same number of features are selected, the $L_{1,\infty}$ norm produces sets of features that are significantly more similar across tasks with equivalent prediction performances. The L_0 mixed-pseudo-norm regularizers tend to produce the sparsest solutions at the cost of decreased classification performance. The $L_{1,1}$ regularization serves for comparison purposes since it comes to a single-task generalized lasso problem. The benefit of the multi-task approach over the single task approach is assessed.

The proposed multi-task approach can also be used in a context of feature selection for multi-class problems by generating sub-tasks according to the one-against-one or the one-against-the-rest pattern. It is particularly noticeable that the use of the $L_{1,\infty}$ group regularizer offers the benefit of relying on the same feature subset to discriminate between any class versus

the others, which is generally not achieved by other feature selection methods.

As a side contribution, the evaluation of a fast and efficient feature pre-filtering technique is also proposed. This approach does not use the class labels, is fast to compute, and allows to drop up to 75% of the features. It either induces a slight gain in classification performance or at least does not degrade those performances on the considered datasets.

All the proposed approaches are evaluated over many microarray datasets containing hundreds of samples and thousands of dimensions. The methodological aspects of the experimental parts of this thesis have received a sustained attention. Several validation protocols are discussed, along with relevant significance tests, and even some developments are proposed in that matter, for example regarding an extension to the bootstrap validation procedure. Normalization and selection biases are discussed and avoided. Metrics for relevant and sound evaluation of the proposed methods are discussed. Many comparisons with other baseline approaches are also delivered.

To conclude, feature selection in a context of classification of high dimensional data is a difficult challenge. It can be addressed by using strong inductive biases based either on data only, with ensemble methods, or on external extra information. This extra information can consist of prior knowledge incorporation, transfer learning or multi-task learning. Those biases generally improve classification and selection performances.

11.2 Perspectives

Many tracks for further developments are already identified and are briefly discussed below.

Generally, objective functions for classification involve a loss function corresponding to the accuracy of a model. In other words, what is optimized is the capability of classifying correctly the largest possible number of samples. As discussed in this thesis, the evaluation criterion of binary classification performances in cases where the class priors are largely unbalanced should be done through the balanced classification rate, not the accuracy. There is thus a mismatch between what is optimized and what we are really after. Accuracy and BCR are equivalent only when classification is perfect.

Otherwise, optimizing accuracy does not necessarily optimize BCR. All the techniques proposed in this thesis, as well as the baseline approaches from the literature, could be modified in order to optimize BCR instead of accuracy. This also holds for the feature selection method based on random forests.

Another perspective is about regularizers. The L_1 norm is known to produce sparser solutions than the L_2 norm and is also known to require less samples than the L_2 to provide quality solutions (Ng, 2004). However, it is also known to be possibly less stable (Zhao and Yu, 2006). A stabilizer of the L_1 regularizer is proposed in the form of a combination of both L_1 and L_2 penalties, for example combined with a loss in a weighted sum with two regularization constants. This is the case for example of the elastic net (Zou and Hastie, 2005) or the hybrid huberized SVM (Wang *et al.*, 2007). Such elastic penalty could also be applied to the multi-task and multi-class approaches. This could lead to a gain in both between- and within-task stability, whereas there is now sometimes a tradeoff between those two criteria.

Methods similar to the multi-task approaches developed in this thesis are currently being ported to the bayesian framework (Hernández-Lobato *et al.*, 2010). In this approach, the spike-and-slab prior discussed namely in Chapter 8 is generalized to a multi-task context.

A crucial issue in both transfer learning and multi-task learning not yet fully addressed is the ability of a priori measuring the adequacy between several datasets to combine them together in one or the other framework. Such development could lead to a more rigorous validation of those methods without requiring a subjective appreciation of the datasets similarity. Some solutions are already proposed for pairwise comparisons (Ben-David *et al.*, 2007) but it can be used only if the number of dimensions is smaller than the sum of points available in the two datasets. For now, in small- n -big- p contexts, only an a posteriori use of this metric can be used, to check if the two datasets are similar *after* the selection process.

Another perspective can be sketched, more related to the particular applicative case of microarray data analysis. In Appendix C.2, a method for bypassing an arbitrary step of summarization (the standard Affymetrix microarray preprocessing procedure) is proposed, but not tested. It proposes to use the expression measurements at the probe level (gene sequence sub-parts)

instead of taking their median and work at the probeset (gene) level. A group of features can be defined for each set of probes and a group-lasso technique (Yuan and Lin, 2006) or a sparse SVM for grouped variables (Zou and Yuan, 2008) could be applied on those groups. It would have as effect to either fully drop or fully select a group of probes together, but allowing the model weights to differ from probe to probe within a selected probe group. The importance of each probe could be driven by the data instead of being a priori fixed. It can be argued that this procedure would lead to even more overfitting than in the regular case since it multiplies the number of features roughly by ten. On the other hand, two arguments in favor of such an approach can be formulated. First, the number of features is already about hundred times higher than the number of samples. It is probable that beyond that point, adding more features would not make the problem much more difficult. Second, the genetic sequences corresponding to the probes of a probeset are spread all over a gene sequence. The two ends of a gene sequence do not have an equal physical robustness and it is known that one of the two ends is more rapidly degraded than the other. Probes from different regions of the genes could then receive a different importance in the global gene expression computation.

On a higher level, a method such as the group-lasso (Yuan and Lin, 2006) could be applied on group of genes to further structure the input feature space, and either select or drop groups of genes together. We know that many dimensions in genomic applications are highly correlated. In the case of ensemble methods for selection, this could lead to the following problematic situation. Suppose two variables are equally involved in the generative process and highly correlated. Through a feature selection technique, we expect to select at least one of those two, or both, depending on the level of sparsity requested. In fact, it may happen with ensemble methods that none of those are selected. Indeed, it may happen that one of the two features is included only in half of the selections used to build the consensus selection. In the other half of selections, the other, highly correlated feature is selected. It comes to divide by two the number of time each feature is selected, because they are somehow a substitute for each other. In the final consensus, it may happen then that none of them is selected. By imposing a group structure, involving that both are together either selected or dropped, we might avoid this situation. The structure, i.e. the groups of variables, could be defined by specifying a threshold correlation θ between pairs of features. Features

may be represented as the vertices of a graph in which edges are added only between features which correlation is higher than θ . Methods exist to create unconnected cliques (i.e. distinct groups of fully connected features) from such a graph (Jones and Pevzner, 2004). Once groups are defined, the group-lasso or the SVM for grouped variables can be simply applied, with an expected gain both in classification and stability performances.

In Chapter 3, the TV-norm has been introduced. It consists of a L_1 regularization on the derivatives of a function rather than on its parameter vector. It is not directly obvious how to apply this regularization scheme outside of the image processing field, because it requires a structure between the samples. In the case of image processing, this structure is the location of the pixels with respect to each others. In the genomic field, what can be the distance between patients? An equivalent of the TV-norm for graphs has been proposed (Peyré *et al.*, 2008). Graphs can be easily defined over the patients, in which edges could be defined according to clinical factors. TV-norm penalizes the variation between close points. Its application to genomic data is thus likely to provide an embedded stability-enforcing regularization.

We sketched briefly some links between sparse learning and compressed sensing in Chapter 3. In this thesis, we try to build the best models possible on the few points available, based on high dimensional measurements and we do not intend to have an impact on how the data are generated. In the compressed sensing field, researchers try to embed the sparsity in the measuring devices, to explicitly avoid the high dimensional setting. It makes possible to lower the minimal number of samples required to correctly model a signal. This way of thinking can have an impact on the design of the sensors used to make measurements. A new type of microarray inspired by compressed sensing has been recently proposed (Sheikh *et al.*, 2007). This technology has much less probes than the standard microarray technology, and each probe measures a combination of several expressed sequences. This drastically reduces the number of chips required to correctly estimate a model.

Finally, it is also worth discussing the applicability of the methods discussed in this thesis to other applicative fields¹. In principle, there is not much preventing the application of the proposed methods to other domains offering few points described by many variables. A preliminary study shows

¹Of course, summarization techniques are specific to Affymetrix microarrays

that the filters considered in this thesis (namely the Golub's ratio and the t -test) could be applied on text data (Branders and Paul, 2010). This kind of data is also described by a generally high number of variables (large vocabularies). Unlike microarray data however, they often contain many samples. This preliminary work precisely studies situations where only a small subset of the available points are used, with satisfactory results. Other domains could be tested as well. The application to other domains offering more points could suffer from the fact that most solvers considered in this thesis have a computational complexity much more depending on the number of samples than on the number of features. This holds both for the SVM and for the multi-task penalized logistic regression of Chapter 10. This limitation could be overcome in some cases, at least for methods based on a linear SVM, since an efficient solver for large-scale problems has been recently developed (Hsieh *et al.*, 2008).

Part IV

Microarray Case Study

Cristall: Allergy Risk Factors Identification in Newborns

12

This chapter proposes an extended microarray data analysis case study, in the context of the Cristall project. The Cristall project aims to identify allergy risk factors in newborns. The main goal of the gene expression analysis part of the project is to predict the allergic status of children at 18 months based on their gene expression data at 6 months. We are also interested in the identification of a gene signature making this prognosis possible.

The chapter contains five parts. First, the project is described in Section [12.1](#). Microarray data generated in the context of the project are described and analyzed in Section [12.2](#). The third part of the chapter reports the experiments led on the data in order to obtain a prognosis model as well as a list of biomarkers (Section [12.3](#)). The chapters goes on in Section [12.4](#) with an illustration of the concept of confounding factor, a frequent issue in microarray analysis. In this case, the precise age of the patients at the time of blood collection may be a confounding factor. A final discussion on the Cristall project is proposed in Section [12.5](#).

Our analysis fails to identify a set of biomarkers making possible the estimation of a prognosis model with high classification performances. We try to provide a sufficient amount of elements and experimental results to justify this outcome.

Although the current advances in this project do not offer satisfying findings on the mechanisms of allergy in the early childhood, it illustrates many aspects discussed in the previous parts of this thesis. It also discusses

other practical aspects not covered so far such as data visualization, outlier detection, confounding factor detection, etc. More importantly, it confirms the advantages of some of the original contributions of this thesis, such as an increase of stability with partially supervised feature selection or ensemble methods for feature selection.

In this chapter, we most of the time write *patient* instead of *sample* and *gene*, *biomarker* or *probeset* instead of *feature* or *dimension*.

12.1 Project Description

Context

As described in (DGTRE, 2006), allergy is a steadily increasing disease. A significant number of children have allergic symptoms that tend to evolve and persist into adulthood, a phenomenon known as the atopic march. The atopic disease significantly impacts the quality of life of affected individuals as well as their families and raises socioeconomic issues regarding their support. In this context, the hypothesis of an early prevention of allergy is becoming more attractive. The mechanisms of allergy have been widely studied but knowledge about the early stages of allergy development is still missing. The risk factors of developing allergy remain undetermined, preventing the establishment of a prevention strategy. The discovery of some major players in the immune system regulation has opened new perspectives in the understanding of immunological factors involved in the development of allergy. Regulatory T-cells, antigen-presenting cells and mesenchymal cells seem to play an important role in the development of tolerance¹.

Aims

The project's main objective is to study and characterize the immunological factors associated with developing allergy, in order to individually and early identify patients at risk. On this basis, the underlying objective is to define and test a strategy for prevention of allergy in children.

It is very difficult to output a clinical diagnosis for allergy before a child is about eighteen months old because of the constantly evolving immune

¹Tolerance is a lack of immune response to stimuli (such as antigens, viruses) from outside the body that should normally be fought and eliminated.

system of young children. Even at that age, the allergic status may still change but is more or less stabilized. In that context, 300 children are being followed from their birth for a duration of five years. This is the first study on allergy following so many children.

Measurements

Blood samples are taken at zero, two, six, twelve and eighteen months, two, three and five years. Many other factors are being monitored aside from blood samples analysis, such as the family antecedents (do the parents show allergic symptoms?), some of the parents practices (smokers?), socioeconomic factors (does the child live in the country or in a city?), nutrition facts (mother milk?), etc. More than 70 characteristics are measured. Blood samples are themselves used to perform other measurements such as the hemoglobin rate, flow cytometry, and also gene expression microarray analysis. This chapter reports the microarray study.

Gene Expression

The specific aim of this part of the study is to:

- design a prognosis model with the highest prediction performances possible, able to predict children allergic status at 18 months based on the gene expression values from their blood samples at 6 months.
- identify the smallest set of biomarkers on which these predictions can be based

The second objective is itself motivated by two subsequent goals, which are respectively the possibility of designing an affordable prognosis kit (or a cheaper microarray chip for further experiments), and the understanding of biological processes related to allergy.

Partners

This project is a collaboration between the University of Louvain (pediatric research institute), the Walloon region and GlaxoSmithKline Biologicals. The project is mainly directed by Professor Etienne Sokal. Other teams also take part to the project, such as the Machine Learning Group of the University of Louvain and the Center for Applied Molecular Technology of the University of Louvain. The former is in charge of the microarray data analysis under the direction of Professor Pierre Dupont, while the latter is in charge of

microarray data production under the responsibility of Professor Jean-Luc Gala.

Extra information on microarray technology can be found in Appendices [A](#) and [C](#).

12.2 Data Description

This section gives a description of the data used in this part of the Cristall project, and of the experimental setting. It first proceeds to the description of the cohort of patients in Section [12.2.1](#). The microarray experimental setting is detailed in Section [12.2.2](#). We then report a first analysis of the microarray data in order to detect potential outliers in Section [12.2.3](#). This section ends with the brief description of a list of genes mentioned by the medical team as potential markers in Section [12.2.5](#). These potential markers constitute expert knowledge that is incorporated as prior feature relevance in some of the experiments reported in Section [12.3](#).

12.2.1 Cohort

The Cristall project cohort contains 300 children followed-up from their birth for five years. This is a world premiere. Following a cohort of this size through five years is a huge endeavor subject to major issues. First of all, there can be dropouts. They can be caused by the move of a family far from the hospital related to the study, or by parents changing their minds as time goes by, etc. For the Cristall project, 80 dropouts have been recorded at the time of writing (the project is still running). The extraction of blood is also difficult with so young children, namely in order to get sufficient amounts of blood when children get six months old to perform all the required analyses. For the gene expression part, the long and complex experimental protocol assumes as a prerequisite that the RNA extracted be of sufficient quality, as measured by the 28s:18s ratio. Finally, to be able to output a prognosis by applying the methods described in this thesis, the clinical status of the children has to be clearly assessed. At the time of writing, 140 children show non-allergic tendency, 43 children show allergic tendency, 25 children are in a grey zone in which the allergic status is not clear and 12 children still need to be diagnosed, but are still too young (i.e. <18 months). To get more chance of getting a clear signal in the microarray data, the 25 children whose diagnosis is not clear-cut are not considered for microarray data experiments.

Combining all of these constraints, only 77 children have the RNA from their blood samples hybridized on microarray chips. It is on that basis that the current chapter performs a detailed analysis. Getting this amount of data took several years, and is a progressive process. A first preliminary study on 33 patients has been led in fall 2009 and is the topic of Section [12.4](#).

Each patient has been given an ID, and their medical records have been made anonymous by replacing their name by their ID. Among the 77 patients who underwent a microarray experiment, 51 have a non-allergic status and 26 have an allergic status at 18 months. This is a typical 2:1 ratio for microarray data class priors. Table [12.1](#) summarizes the allergic status of all the patients at 18 months.

Table 12.1 Clinical status of the 77 patients considered for microarray experiments. Columns respectively show the patient ID and the allergic status (−1 codes for non-allergic, 1 codes for allergic).

ID	Status	ID	Status	ID	Status
3	-1	79	-1	183	-1
5	-1	81	-1	187	-1
7	-1	82	-1	188	-1
9	1	86	-1	203	-1
18	1	93	-1	205	1
19	-1	94	1	219	-1
31	-1	95	1	220	1
34	-1	100	-1	230	-1
35	-1	105	1	231	1
36	-1	106	-1	238	1
40	1	108	1	240	-1
43	-1	113	-1	247	-1
44	-1	120	1	248	-1
46	1	131	-1	255	1
47	-1	132	1	256	1
50	-1	144	-1	259	1
54	-1	146	-1	260	-1
55	-1	149	-1	268	1
63	-1	152	1	269	-1
65	-1	159	-1	271	-1
68	1	170	1	280	-1
70	1	172	-1	283	1
71	-1	174	-1	285	-1
72	-1	176	-1	287	1
74	-1	180	1	302	1
75	-1	182	-1		

12.2.2 Experimental Setting

The gene expression analysis part of the Cristall project is a perfect example of prognosis. Blood samples are taken at 6 months and the target variable to predict is the binary allergic status at 18 months.

At the age of 6 months, a 5ml sample of blood is extracted from each

child. The actual quantity varies from child to child. The collected cells are stimulated by an allergen called nDer p 1 (Dermatophagoides pteronyssinus allergen 1), which is an extract of mite. The motivation for stimulation is to increase the chances of getting different signals between allergic and non-allergic children. The particular choice of nDer p 1 is a consequence of the fact that it is one of the most common allergen for adults as well as for children. Samples are stimulated for 16 hours. Cells are then collected, washed, and finally put in a preserving medium. The samples are finally stored at -80°C until the RNA extraction for microarray experiments.

RNA is extracted from the stimulated blood samples. It is processed² and hybridized on Affymetrix Genechips Human Genome U133 Plus 2.0 (HGU133Plus2.0). These microarray chips have 54,675 probesets, among which 62 control probesets. Once produced, the expression intensities have to be preprocessed. This procedure is globally known as summarization and is more extensively described in Appendix C. This preprocessing aims first at getting rid of experimental noise, namely appearing during the hybridization and scanning steps. The second objective of summarization is to normalize the global expression intensity distribution for each microarray chip, to make all patients profiles comparable. The third and last objective of summarization is to produce a unique expression value for each probeset, based on the different probe intensities of each probeset. Summarization protocols are various and there is no consensus on which is the most relevant or adequate. In this study, we perform both RMA and GC-RMA protocols (see Appendix C for more information about summarization). Qualitatively, the conclusions of this study are the same with both protocols. RMA seems to produce numerically slightly better performances in the next sections. That is why only results obtained on RMA-summarized data are reported here.

12.2.3 Outlier Detection

It is difficult to visualize data described by several tens of thousands of dimensions. Nevertheless we would like to get an idea of the homogeneity of the patients expression profiles distribution.

²The details of that procedure are described in a 200+ pages document provided by Affymetrix, and are beyond the scope of this thesis

Principal Component Analysis

Principal component analysis has been criticized in Chapter 3 as a mean of dimensionality reduction when the aim is the identification of biomarkers. It remains however a convenient visualization tool. PCA is a procedure that transforms a number of possibly correlated features into a smaller number of uncorrelated new features called principal components (Pearson, 1901). PCA is said to capture most of the data variance. Without entering the details of its computation, the intuition in a simple case is given next. In an initial space in two dimensions, PCA will find a projection on a single component by minimizing the distances between all points (here, patients) and the candidate first principal components, as illustrated on Figure 12.1. The same figure (right) illustrates on a simple example why PCA is not favoring classes separability.

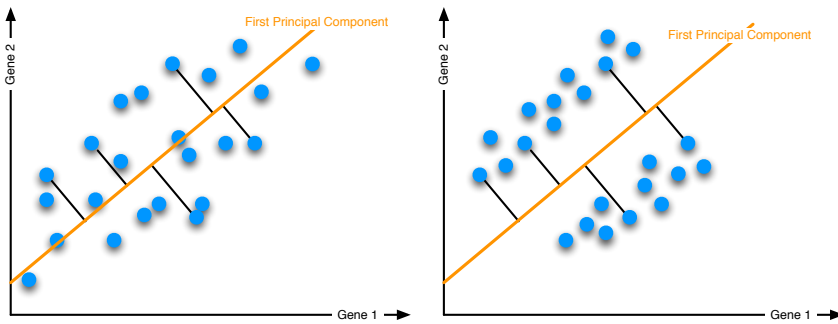


Figure 12.1 The computation of the first principal component (orange line) is equivalent to finding the line minimizing the sum of the distances between this line and all points. Right: PCA is not intended to enhance separability between classes.

Figure 12.2 (top) displays the data projected on the two first principal components of a PCA computed on all the 77 Cristall patients. The numerical values on the figure do not matter much. What is more important is the density of the cloud of patients and irregularities within it. It appears clearly on this picture that some patients seem to be outliers with respect to the cloud. In particular, samples 113 and 280 corresponding to non-allergic patients, and samples 9 and 268 corresponding to allergic patients seem to be clear outliers.

We decide to exclude those four patients from our dataset. Small datasets are sensitive to strong outliers. From now, we will thus consider a dataset of 73 samples with 24 allergic patients and 49 non-allergic patients. The same

PCA experiment is led on the new dataset, and the projection on the first two principal components is illustrated in Figure 12.2 (bottom). The cloud is much more regular. The two first principal components in this last case capture 53.1% of the total data variance.

To support our argument according to which PCA does not favor class separation, Figure 12.3 displays the same projections as in Figure 12.2, but with a different color for both classes. It is clear that no separation can easily be found. The clouds of the two classes are tightly interleaved.

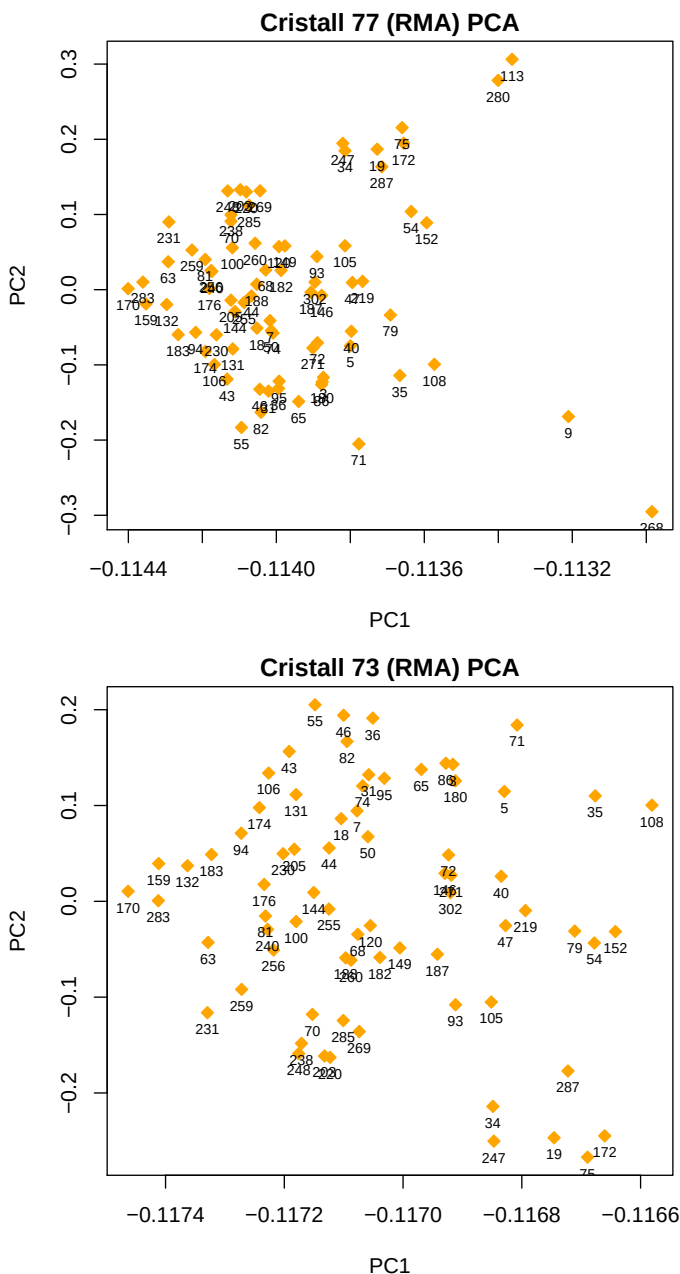


Figure 12.2 Top: The 77 patients projected on the two first principal components of a PCA. Bottom: The 73 patients projected on the two first principal components of a PCA (53.1% of the data variance).

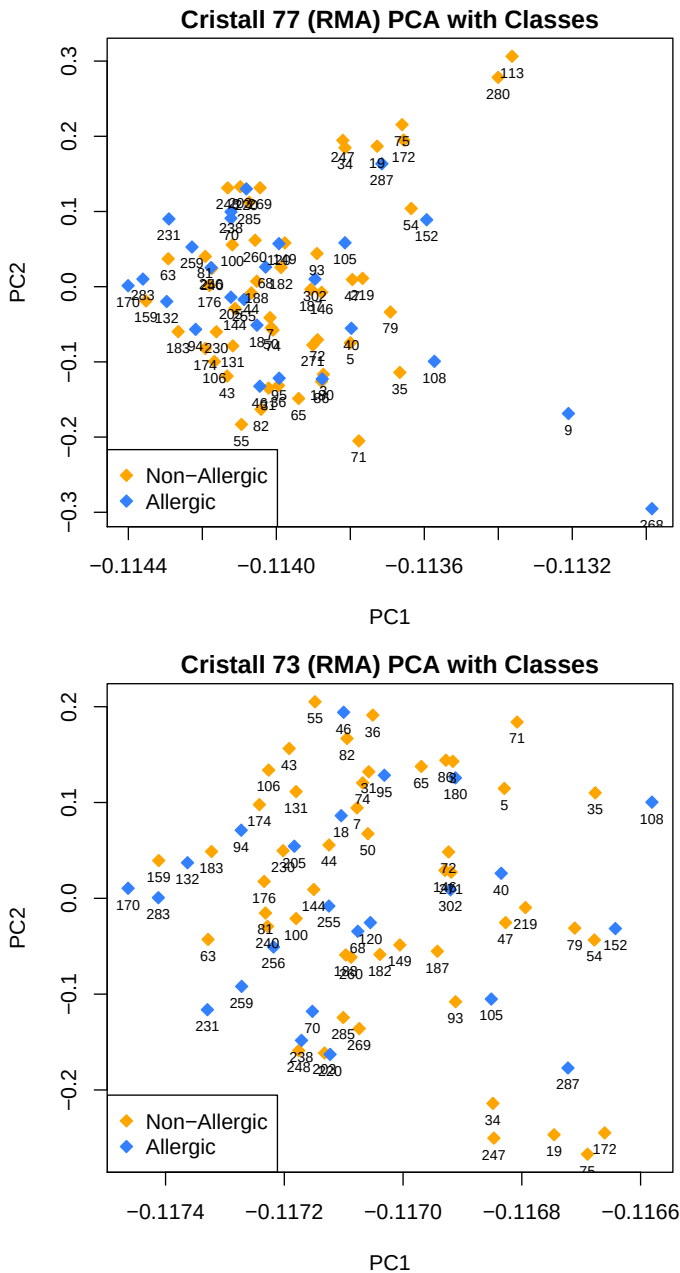


Figure 12.3 The 77 patients (top) and the 73 patients (bottom) projected on the two first principal components of a PCA. A distinction is made between allergic and non-allergic samples.

12.2.4 Discriminative Power

Before getting to the core analysis of the data, we would like to adopt the point of view of descriptive statistics. We can, without any bias, assess the discriminative power of each probeset in the sense of a t -test. We apply the hypothesis test described in Chapter 3 on the full dataset. No selection bias occurs here, even if we compute this test on the whole dataset, since the current test results are not used for the selection of biomarkers, nor to output predictive performances in the next section. It is just a way to evaluate the separability of the data. Results for the 20 probesets with the smallest p -values are reported in Table 12.2 (left).

Table 12.2 (left) may lead to think that many probesets are significantly discriminative. It is in fact not so. First, our experience with microarray data make us think that those p -values are not sufficiently small to offer a good discriminative power. As a comparison, Table 12.2 (right) reports the results of the same experiment for the dataset LEU, counting a similar number of samples (72). The p -values are about 7 orders of magnitude smaller than the Cristall ones. We can visually feel the difference by plotting the samples from both datasets in the space defined by their two most discriminative probesets (Figure 12.4). Classes are much more separated in the case of LEU than in the case of the Cristall dataset. The same results can be obtained on other datasets than LEU. Second, the p -values reported are not corrected for multiple testing. The Benjamini-Hochberg correction makes all of the Cristall data probeset insignificant.

This descriptive analysis should not make us conclude that discrimination between the allergic and non-allergic patients is impossible, but at least it is likely to be difficult to obtain as good classification performances as on other datasets. The t -test is univariate and may miss important interactions between genes potentially making possible a good classification. This analysis is the topic of Section 12.3.

Table 12.2 Left: p-values of the 20 most discriminative probesets computed on the 73 Cristall patients. Right: p-values of the 20 most discriminative probesets computed on the 72 LEU samples.

Cristall Dataset		LEU Dataset	
Probeset ID	p-Value	Probeset ID	p-Value
227453_at	0.000079	X59417_at	2.958823e-12
211502_s_at	0.000084	M92287_at	6.432665e-12
226772_s_at	0.000135	M31523_at	1.179739e-11
235732_at	0.000268	M31211_s_at	1.974663e-11
214826_at	0.000275	U05259_rna1_at	4.119695e-11
204942_s_at	0.000322	M84371_rna1_s_at	4.726905e-11
206512_at	0.000330	J05243_at	8.406608e-11
1566032_at	0.000374	M11722_at	2.394580e-10
215693_x_at	0.000397	M89957_at	2.673149e-10
223148_at	0.000403	S50223_at	2.708239e-10
1564299_at	0.000437	M28170_at	4.856048e-10
226917_s_at	0.000451	HG1612-HT1612_at	4.890888e-10
1562446_at	0.000508	U22376_cds2_s_at	5.210564e-10
1554982_a_at	0.000540	Y08612_at	5.318582e-10
1555600_s_at	0.000566	U29175_at	8.502406e-10
213979_s_at	0.000629	X97267_rna1_s_at	9.253862e-10
1570100_at	0.000640	D88270_at	2.666022e-09
208425_s_at	0.000645	M33680_at	2.818187e-09
207117_at	0.000662	M31303_rna1_at	3.057414e-09
227582_at	0.000670	X17042_at	3.162247e-09

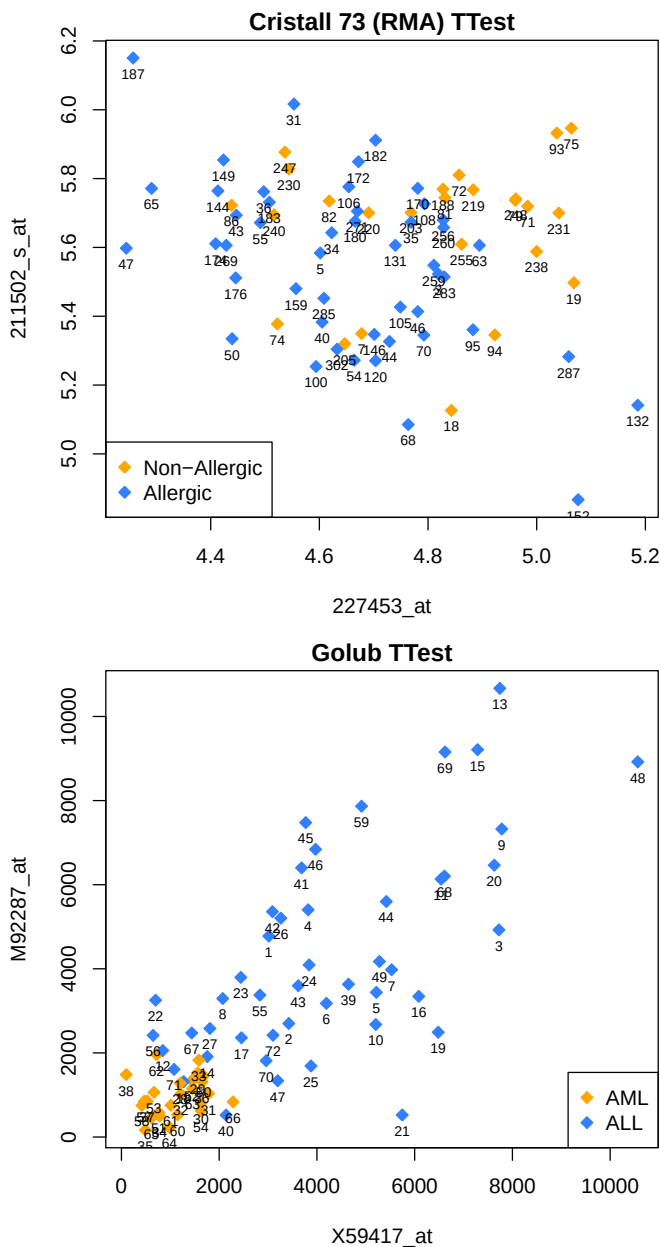


Figure 12.4 Top: The 73 Cristall patients in the space of the two most discriminative probesets: 227453_at (UNC13A) and 211502_s_at (PTFK1). Bottom: The 72 LEU samples in the space of the two most discriminative probesets: X59417_at (PSMA6) and M92287_at (CCND3).

12.2.5 Prior Knowledge

In Chapters 8 to 10, we address the issue of the selected signatures stability by making use of external information. The transfer learning and the multi-task approaches cannot be considered here. The objective of the Cristall project is innovative enough to make impossible the acquisition of other public datasets on allergy with more than a ten of samples. However, the partially supervised technique proposed in Chapter 8 can be used provided that we have an a priori guess of higher relevance for at least some probesets.

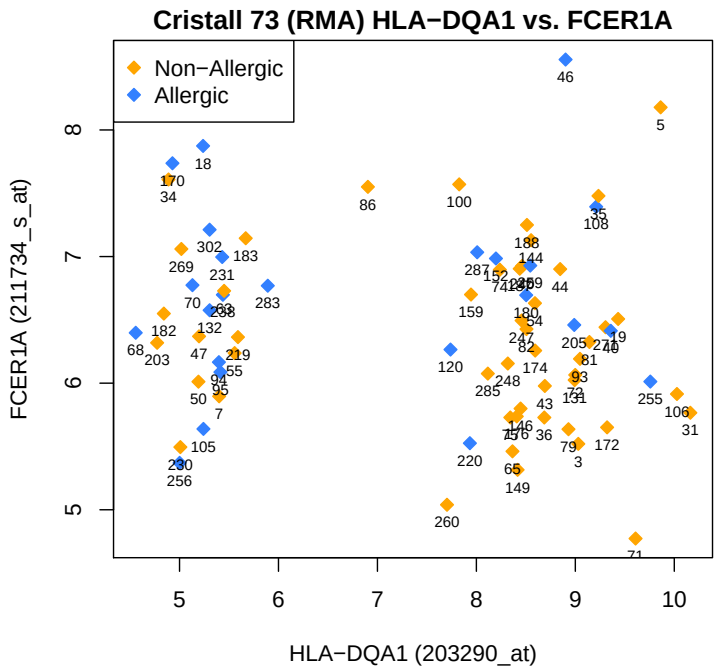


Figure 12.5 Plot of the 73 Cristall patients according to the two most discriminative probesets among the a priori relevant ones.

The team of Professor Sokal listed 43 genes and micro-RNA mentioned in the literature as potential influence factors of allergy. Those elements are listed in tables of Appendix H, with their 102 corresponding probesets on the Affymetrix HGU133Plus2.0 chips and some extra information. This list of probesets is used in some of the experiments of the following section to bias the selection with our PS- L_2 -AROM technique introduced in Chapter 8.

Before getting to that, we can already study the discriminative power of the recommended probesets. To do so, we refer to the *t*-test results computed in the previous section and look for the p-values of the current list of probesets to favor. Second, we compute the positions of those probesets p-values within the ranking of all the dataset probesets. Both the p-values and the overall ranking are shown in Tables 12.3 to 12.5. The results show that most a priori relevant features have a really low discriminative power in the sense of a *t*-test. The four most discriminative probesets are 203290_at (HLA-DQA1), 211734_s.a (FCER1A), 213831_at (HLA-DQA1 again) and 201042_at (TGM2). Those four probesets are the only ones having a p-value < 0.1. HLA-DQA1 is the only gene in the list containing a probeset having a p-value < 0.05, i.e. significant with 95% confidence. This is without any correction for multiple testing. The median ranking for these probesets is 32,427. This means that half of those probesets are ranked beyond that point. The four most discriminative probesets are ranked respectively at positions 669, 2465, 2481 and 3726. Figure 12.5 shows the 73 samples in the space of the two most discriminative probesets. The two classes seem strongly interleaved.

Table 12.3 *t*-test results on the 102 probesets corresponding to the a priori relevant elements. Symbol: gene symbol, PS ID: probeset name, PS Index: index of the probeset on the chip, Ranking: rank of p-value according to the ranking over all 54,675 probesets, p-Value: result of the *t*-test.

	Symbol	PS ID	PS Index	Ranking	p-Value
1	IL1B	205067_at	14515	50108	0.92356
2	IL1B	39402_at	54330	41516	0.77895
3	IL3	207906_at	17343	42521	0.79610
4	IL4	207538_at	16980	16916	0.35176
5	IL4	207539_s_at	16981	17686	0.36527
6	IL5	207952_at	17386	51018	0.93842
7	IL8	202859_x_at	12308	19091	0.39085
8	IL8	211506_s_at	20834	30820	0.59649
9	IL9	208193_at	17620	43094	0.80549
10	IL10	207433_at	16876	17173	0.35649
11	IL13	207844_at	17284	22320	0.44902
12	IL17A	208402_at	17824	6980	0.16081
13	IL17A	216876_s_at	26167	10933	0.23855
14	STAT3	208991_at	18406	38800	0.73317
15	STAT3	208992_s_at	18407	37523	0.71138
16	STAT3	225289_at	34547	39120	0.73876
17	STAT3	243213_at	52463	11701	0.25384
18	RORC	206419_at	15866	8878	0.19820
19	RORC	228806_at	38061	11883	0.25733
20	IFNG	210354_at	19753	43621	0.81504
21	CSF2	210228_at	19630	53502	0.97984
22	CSF2	210229_s_at	19631	39875	0.75194
23	TGFB1	203084_at	12534	39017	0.73700
24	TGFB1	203085_s_at	12535	29035	0.56521
25	TNF	207113_s_at	16560	44115	0.82291
26	STAT6	201331_s_at	10780	33655	0.64541
27	STAT6	201332_s_at	10781	54211	0.99212
28	TGM2	201042_at	10491	3726	0.09459
29	TGM2	211003_x_at	20371	46986	0.87146
30	TGM2	211573_x_at	20899	25271	0.50102
31	TGM2	216183_at	25476	30369	0.58828
32	NR3C1	201865_x_at	11314	37881	0.71763

Table 12.4 Table 12.3 continued.

	Symbol	PS ID	PS Index	Ranking	p-Value
33	NR3C1	201866_s.at	11315	33065	0.63587
34	NR3C1	211671_s.at	20994	10781	0.23562
35	NR3C1	216321_s.at	25614	42058	0.78794
36	GATA3	209602_s.at	19013	44780	0.83461
37	GATA3	209603.at	19014	17894	0.36892
38	GATA3	209604_s.at	19015	6925	0.15952
39	HSPD1	200806_s.at	10255	41259	0.77430
40	HSPD1	200807_s.at	10256	24025	0.47897
41	HSPD1	241716.at	50966	36857	0.70064
42	HSPD1	243372.at	52622	28912	0.56316
43	HSPD1	243845.at	53095	35283	0.67344
44	CCL11	210133.at	19538	4525	0.11121
45	FOXP3	221333.at	30617	52090	0.95573
46	FOXP3	221334_s.at	30618	37093	0.70448
47	FOXP3	224211.at	33483	20281	0.41262
48	TLR2	204924.at	14372	35939	0.68493
49	TLR4	1552798.a.at	388	32004	0.61824
50	TLR4	221060_s.at	30344	38050	0.72043
51	TLR4	224341_x.at	33609	36934	0.70157
52	TLR4	232068_s.at	41323	41036	0.77037
53	TLR9	223903.at	33180	40891	0.76818
54	FCER1A	1562537.at	6653	7681	0.17420
55	FCER1A	211734_s.at	21055	2465	0.06681
56	IDO1	210029.at	19435	28479	0.55524
57	CXCL2	1569203.at	9088	35661	0.67995
58	CXCL2	209774_x.at	19183	43732	0.81688
59	CXCL2	230101.at	39356	35506	0.67750
60	CXCL3	207850.at	17290	40765	0.76625
61	PTGS2	1554997.a.at	1992	29031	0.56514
62	PTGS2	204748.at	14196	30408	0.58915
63	NFKB1	209239.at	18654	45670	0.84867
64	NFKB2	207535_s.at	16977	4405	0.10901
65	NFKB2	209636.at	19046	9975	0.21940
66	NFKB2	211524.at	20851	15510	0.32518
67	FGFR1	207822.at	17262	45521	0.84597

Table 12.5 Table 12.4 continued.

	Symbol	PS ID	PS Index	Ranking	p-Value
68	FGFR1	207937_x.at	17374	44519	0.83020
69	FGFR1	210973_s.at	20341	39381	0.74336
70	FGFR1	211535_s.at	20862	38380	0.72612
71	FGFR1	215404_x.at	24699	10073	0.22134
72	FLG	215704.at	24998	33504	0.64318
73	FGFR1	222164.at	31445	44801	0.83492
74	FGFR1	226705.at	35961	53706	0.98335
75	HLA-A	213932_x.at	23232	51098	0.93968
76	HLA-A	215313_x.at	24608	39810	0.75071
77	HLA-A	217436_x.at	26722	22737	0.45623
78	HLA-B	208729_x.at	18145	21918	0.44204
79	HLA-B	209140_x.at	18555	29165	0.56741
80	HLA-B	211911_x.at	21220	11280	0.24543
81	HLA-DPA1	211990.at	21298	48523	0.89667
82	HLA-DPA1	211991_s.at	21299	32850	0.63216
83	HLA-DPA1	213537.at	22839	38403	0.72660
84	HLA-DPB1	201137_s.at	10586	47895	0.88678
85	HLA-DQA1	203290.at	12738	629	0.02013
86	HLA-DQA1	212671_s.at	21977	42945	0.80311
87	HLA-DQA1	213831.at	23132	2481	0.06717
88	HLA-DQA1	236203.at	45453	16777	0.34899
89	HLA-DQB1	209480.at	18894	10260	0.22496
90	HLA-DQB1	209823_x.at	19230	4276	0.10634
91	HLA-DQB1	210747.at	20129	38233	0.72360
92	HLA-DQB1	211654_x.at	20979	19201	0.39302
93	HLA-DQB1	211656_x.at	20981	14139	0.29998
94	HLA-DQB1	212998_x.at	22302	30439	0.58960
95	HLA-DRA	208894.at	18310	22505	0.45202
96	HLA-DRA	210982_s.at	20350	26352	0.52006
97	HLA-DRB1	209312_x.at	18727	5951	0.13945
98	HLA-DRB1	215193_x.at	24488	12243	0.26408
99	HLA-G	210514_x.at	19905	11858	0.25683
100	HLA-G	211528_x.at	20855	35015	0.66885
101	HLA-G	211529_x.at	20856	15725	0.32947
102	HLA-G	211530_x.at	20857	9125	0.20302

12.3 Prognosis Model and Biomarkers Identification

This section reports the core analysis in order to get a prognosis model along with a list of candidate biomarkers. Section 12.3.1 details the protocol applied to the 73 Cristall patients as well as the considered selection and classification methods. Section 12.3.2 reports and analyzes the results.

12.3.1 Experimental Protocol and Metrics

Let us first describe the protocol considered. It is highly similar to the one used in the other chapters of this thesis. First, the features corresponding to the 62 control probesets are removed from the data. The data has now 54,613 probesets left. For a given biomarker selection technique F , an iterative procedure repeats 100 times the following steps:

- The patients are split into 90% (the training set) and 10% (the test set)
- On the training set:
 1. If F is not PS- L_2 -AROM, a non-specific pre-filtering step is performed (see Chapter 6), removing about 75% of the probesets. About 13,650 probesets remain.
 2. An IQR normalization procedure is applied (see Chapter 6). Parameters are stored for future use on the test set.
 3. The selection technique F is applied and a signature of size s is computed.
 4. A linear SVM (Chapter 2) is trained on the training set only on the s selected biomarkers. The C parameter of the SVM is tuned according to the Joachims heuristic of Equation (2.38).
- On the test set:
 1. The test set is restricted to the s biomarkers selected on the training set.
 2. Those remaining biomarkers are normalized via the application of the IQR normalization parameters computed on the training set.
 3. The linear SVM is applied on the test set of patients. A classification decision is obtained for each of them.
 4. The balanced classification rate is computed on those decisions.

After the 100 iterations, the average BCR is computed. The 100 signatures are also pairwise compared through the Kuncheva Index (see Chapter 4), computing the stability of method F . An average stability index is finally computed. This procedure is run for s ranging from 9,999 to 2 probesets with a decrease of about 10% between each. The fact that non-specific filtering is not used in combination with PS- L_2 -AROM is easily explained. Most of the probesets to favor have a poor discriminative ability, as assessed in Section 12.2.5. This is likely to be due to a very low variability. The filtering thus drops most of those probesets to favor. Since we want to study their impact and potentially benefit from them, they have to be kept.

The aim of such a protocol is not the identification of a single signature of biomarkers. It rather allows us to identify a candidate size for an ideal signature on the data and to study the differences between families of selection methods, namely in terms of stability and classification performances.

What selection method do we consider for F in this chapter?

Methods from the literature and described in Chapter 3 are first considered. Univariate filters such as the t -test and the Golub's ratio are applied. Multivariate embedded methods are also considered. This is the case of RFE, L_2 -AROM and the feature selection based on the random forests.

We are naturally interested in using the methods proposed in this thesis. The ensemble selection based on the SVM and described in Chapter 7 is thus also retained. The partially supervised feature selection approach implemented in PS- L_2 -AROM is also applied, in two different ways. First, just as in Chapter 8, we favor the probesets corresponding to genes mentioned as potentially playing a role in allergy. This gene list has been constituted by the team of Professor Sokal and is detailed in Appendix H. A summary is available in Tables 12.3 to 12.5. As in Chapter 8, we choose to set the prior weight $B = 10$ for those probesets, while setting $B = 1$ for the others. The second way we propose to use PS- L_2 -AROM is to create a list of 100 potentially relevant probesets in terms of a t -test. This approach is motivated by the general good behavior of the t -test in terms of stability and the fact that we notice in Section 12.2.5 that the a priori relevant probesets do not exhibit a highly discriminative power. The choice of 100 is arbitrary but similar to the amount of favored features from prior knowledge.

We did not find any consequent dataset on allergy in public databases. It is thus not possible to apply the transfer learning or the multi-task learning approaches of Chapters 9 and 10. The following section reports and discusses the results of those experiments.

12.3.2 Results

Figure 12.6 displays the results of the previously described experiments. We first analyze the results in the applicative perspective. Second, we re-analyze the results from the point of view of this thesis contributions.

The data seem to tell us that the considered broad range of techniques considered is not able to provide a prognosis model for the allergic status of a patient at 18 months based on the nDer p 1 stimulated blood samples extracted at 6 months. The balanced classification rate does not get higher than 60% in the best cases, but it is globally around 50% for most techniques. A BCR of 50% is what a random classifier would achieve. It is also the score of a classifier putting all the samples in one of the two classes. The worst classification performances, not by much, are the one produced by the partially supervised selection when expert knowledge is used. The stability of the models is not extremely high neither. The best performances are achieved this time by the partially-supervised technique with expert prior knowledge. The worst performance is obtained by the selection based on the random forests. The best (or least worse) combination of classification and stability performances seems to be offered by the ensemble selection method around 32 features. At this point it is second in terms of stability and first in terms of classification performances with a BCR of slightly more than 60%.

From the point of view of this thesis contributions, we see that the proposed methods increase the stability. The top two methods in this respect are the partially supervised feature selection technique $PS-L_2$ -AROM and the ensemble selection method based on linear SVM. It is interesting to compare the three versions of selection inspired by the AROM technique on Figure 12.6, L_2 -AROM, $PS-L_2$ -AROM with expert knowledge and $PS-L_2$ -AROM with prior knowledge generated by a t -test. The least stable is L_2 -AROM, without surprise. Favoring some dimensions determined by a t -test do offer a gain in stability. As in Chapter 8, the increase tendency starts before the signature size reaches the number of favored features. The two profiles are

similar until signatures become smaller than around 400 features, then the partially supervised approach is better than the non supervised one. This gain is not strong compared to the one obtained with expert knowledge. The t -test generated knowledge seems to lead to models which are among the best with respect to classification. The ensemble method also increases the stability of the selection, and to produce among the best predictions. We can thus confirm that the methods proposed in this thesis offer advantages over baseline approaches.

The classification results estimated by the balanced classification rate are globally poor. We stressed the fact that most classification models optimize an objective corresponding to the accuracy rather than the balanced classification rate in the perspectives of Chapter 11. To further illustrate that fact, we display accuracy results obtained by the same experiments in Figure 12.7. The results confirm our observations. The accuracy is indeed higher than the BCR by ten to twenty percent. However, the combination of a BCR around 50% and an accuracy close the majority class prior make us think that accuracy is indeed deceiving and that classification models in this case tend to classify all samples in the majority class.

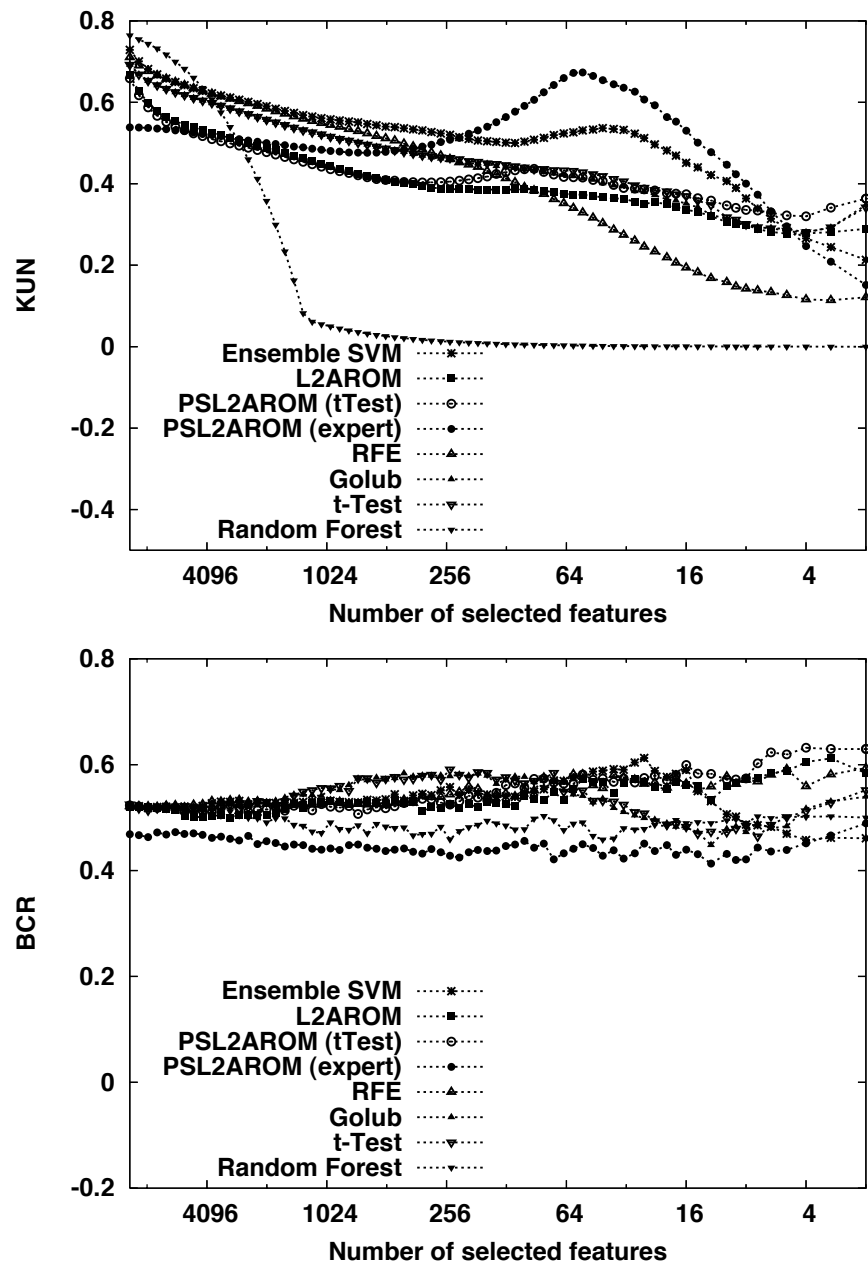


Figure 12.6 Stability (top) as measured by the Kuncheva Index and balanced classification rate (bottom) for the 73 Cristall patients for different selection techniques and different signature sizes.

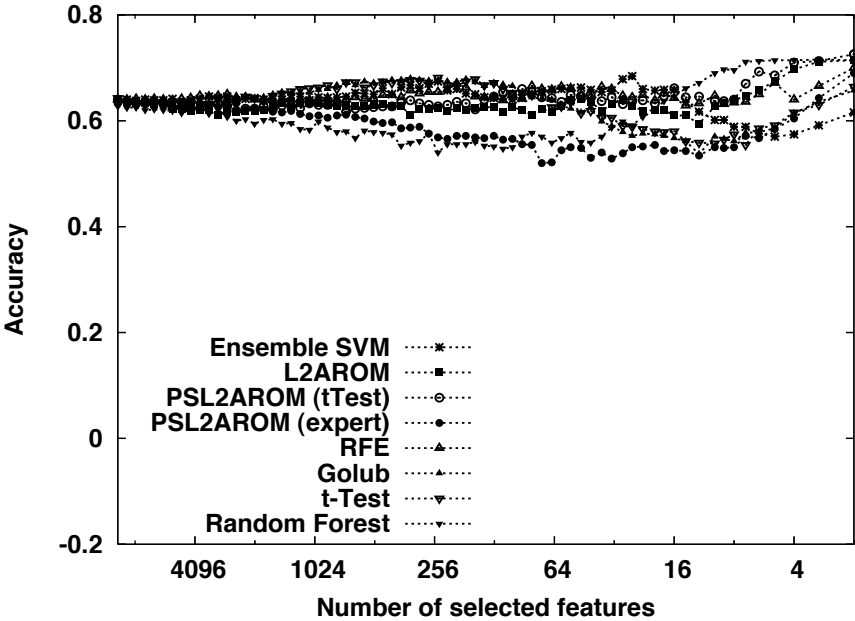


Figure 12.7 Accuracy for the 73 Cristall patients for different selection techniques and different signature sizes.

Complementary Experiments

We propose two extra experiments. The first one further studies the impact of the prior knowledge incorporation. A single signature of 100 probesets is generated on the full 73 dataset with PS- L_2 -AROM where the 102 probesets mentioned as a priori relevant are favored ($B=10$). The same experiment is reproduced with L_2 -AROM. The two signatures of 100 probesets are compared. Their intersection is only of 21 probesets, reported in Table H.1 of Appendix H. The signature generated with L_2 -AROM contains no probeset listed as a priori relevant. The signature generated with PS- L_2 -AROM contains 61 of the 102 probesets listed as a priori relevant. This means first that the 21 probesets shared by L_2 -AROM and PS- L_2 -AROM are not contained in the 102 a priori relevant probesets. This also shows that, as expected, PS- L_2 -AROM is able to depart from some favored probesets when necessary. The list of the 61 probesets among the 102 that are kept by the partially supervised selection technique is detailed in Table H.2 of Appendix H. Among the 43 genes or miRNA corresponding to the 102 probesets, 10 have no probeset in the PS- L_2 -AROM signature: IL3, IL5, IL8, IL13, CSF2, TNF, TLR9, PTGS2, NFKB1,

HLA-DPA1. Conversely, the 33 other genes appear with at least one probeset in the 100 probesets signature selected via PS- L_2 -AROM. It should be further investigated if this makes sense from a biological point of view.

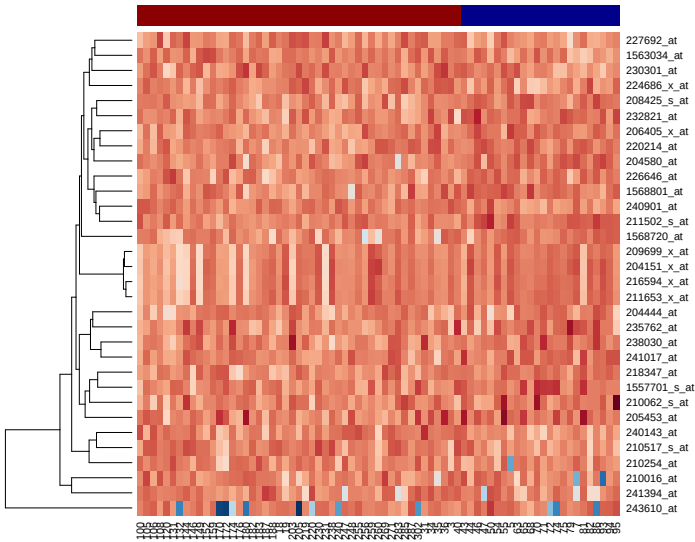


Figure 12.8 Heatmap obtained on a 32 probesets signature extracted on the 73 patients with ensemble SVM. Within the heatmap, red is low expression, blue is high expression. Expressions have been IQR-normalized. The color bar at the top of the figure corresponds to sample labels: red is for non-allergic patients, blue is for allergic patients. Probesets are clustered (complete linkage, euclidean distance).

The second extra experiment is intended to extract a single signature with ensemble selection based on linear SVM since it seems to be the best performing approach on this dataset. The best size seems to be around 32 features. Since we extract this signature on the 73 patients of the available cohort, we do not report any performance evaluation for this signature. Doing so would produce optimistic performances due to a selection bias. The only performance we have is the average performance around 32 probesets for 100 different signatures ($\text{BCR} \approx 60\%$, $\text{accuracy} \approx 67\%$, $\text{Kuncheva index} \approx 53\%$). Two heatmaps are produced with this new signature of 32 probesets. They are available in Figures 12.8 and 12.9. Both of them picture the IQR-normalized intensities of the 32 probesets for the 73 patients. In both cases, the probesets are reordered

according to a hierarchical clustering³ (Felsenstein, 2004). Such a method is another example of unsupervised learning: it does not make use of the information contained in patients status. Clustering is however a convenient visualization tools. The heatmap of Figure 12.9 also displays a clustering on the samples, while Figure 12.8 does not. It is clear that the signature is unlikely to be predictive, given the small color variations. Sorting the patients according to their allergic status as in Figure 12.8 should make appear two distinct color blocks. It is not the case. Sorting the patients according to a clustering as in Figure 12.9 should ideally lead to the same patients ordering. It is not the case neither. This tends to show that clustering (at least in this configuration) is not able to discover the groups of allergic and non-allergic patients.

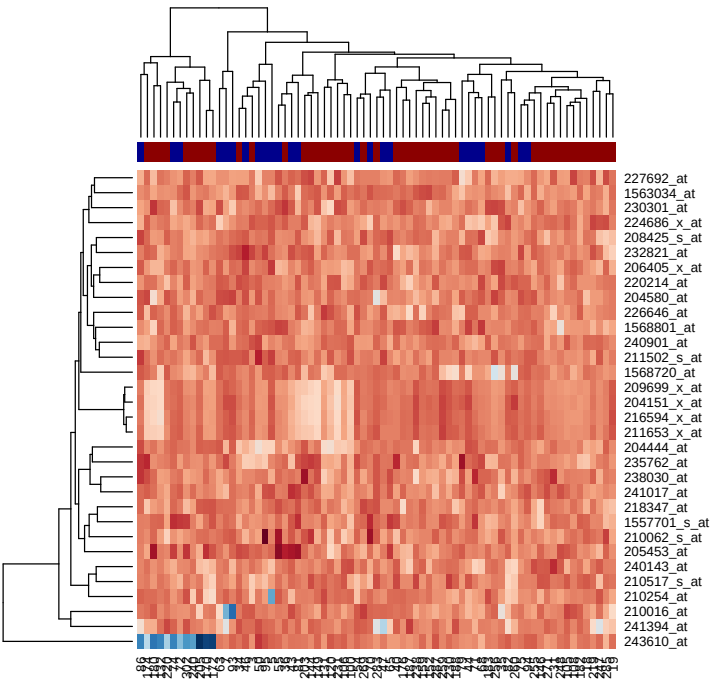


Figure 12.9 Heatmap obtained on a 32 probesets signature extracted on the 73 patients with ensemble SVM. Within the heatmap, red is low expression, blue is high expression. Expressions have been IQR-normalized. The color bar at the top of the figure corresponds to sample labels: red is for non-allergic patients, blue is for allergic patients. Probesets and patients are clustered (complete linkage, euclidean distance).

³Default R parameters: complete linkage and euclidean distance.

12.4 Confounding Factor: An Illustration

The introduction of this thesis mentions the phenomenon of *confounding factor* as a likely issue of gene expression analysis. A confounding factor can be seen as a variable y' correlated with the true variable to predict y . When such a factor exists, a biomarker selection process may select probesets that are in fact predictive of the confounding factor instead of the true target variable. The difficulty relies in the fact that y' could be anything. It could very well be something that is even not recorded in the clinical data of the cohort. The result is a false conclusion that biomarkers for y have been identified and that such a signature can be highly predictive, while in fact it does predict something else.

The topic of this section is to report a confounding factor in the context of the Cristall project. For the purpose of illustration, we put ourselves in a time perspective and get back to a point where only a small subset of the Cristall data was available. This preliminary cohort is described in Section 12.4.1. We report experiments on this cohort leading to a candidate signature for allergic status prediction in Section 12.4.2. This analysis led to several similar signatures of 14 probesets, described and analyzed in Section 12.4.3. The confounding factor is identified as being the true age of the patients at the time of blood collection. This is discussed in Section 12.4.4 where a solution is proposed to avoid the age bias in the experiments on the whole cohort.

12.4.1 Preliminary Cohort

By the fall of 2009, 33 microarray experiments had been led and made available after some quality controls. The experimental setting is exactly the same as the one described in Section 12.2.2. The version of the Cristall dataset at that time is reported in Table 12.6. Among the 33 samples, 19 are from non-allergic children and 14 from allergic children. The diagnosis had been clinically assessed when children got 18 months. As for the full cohort, no patients with a non-clear clinical diagnosis had been included in the set of hybridized samples.

12.4.2 Signature Identification

This section reports the first experiments led to get an idea of the performances of four baseline methods of selection for classification. The analysis

Table 12.6 Clinical status of the 33 patients considered for microarray experiments in the preliminary cohort. Columns respectively show the patient ID and the allergic status (−1 codes for non-allergic, 1 codes for allergic).

ID	Status	ID	Status	ID	Status
3	-1	50	-1	105	1
7	-1	55	-1	113	-1
9	1	68	1	120	1
18	1	70	1	131	-1
19	-1	71	-1	132	1
31	-1	75	-1	146	-1
34	-1	82	-1	152	1
40	1	86	-1	180	1
44	-1	94	1	182	-1
46	1	95	1	183	-1
47	-1	100	-1	220	1

protocol is the same as previously, excepted the fact that we run 200 iterations. The selection methods are the Golub’s ratio, a t -test, RFE and L_2 -AROM. Data are preprocessed both with RMA and GC-RMA and no significant difference has been noticed. Figure 12.10 shows the results for the RMA processed data.

Several conclusions can be drawn. First, the global performances of all models, with respect to both classification and stability are much higher than on the whole 73 patients cohort. This was of course not known at the time of this preliminary analysis. Second, univariate methods (t -test and Golub’s ratio) clearly outperform multivariate methods. With so few samples, it is likely that iterative methods like RFE and L_2 -AROM overfit the training sets. The analysis of the number of biomarkers interestingly reveals a clear stability peak around 14 biomarkers for univariate methods. That peak at 14 probesets is doubled with top classification performances for the same signature size (BCR around 80%). Finally, in any case, classification performances improve as the number of selected probesets decreases.

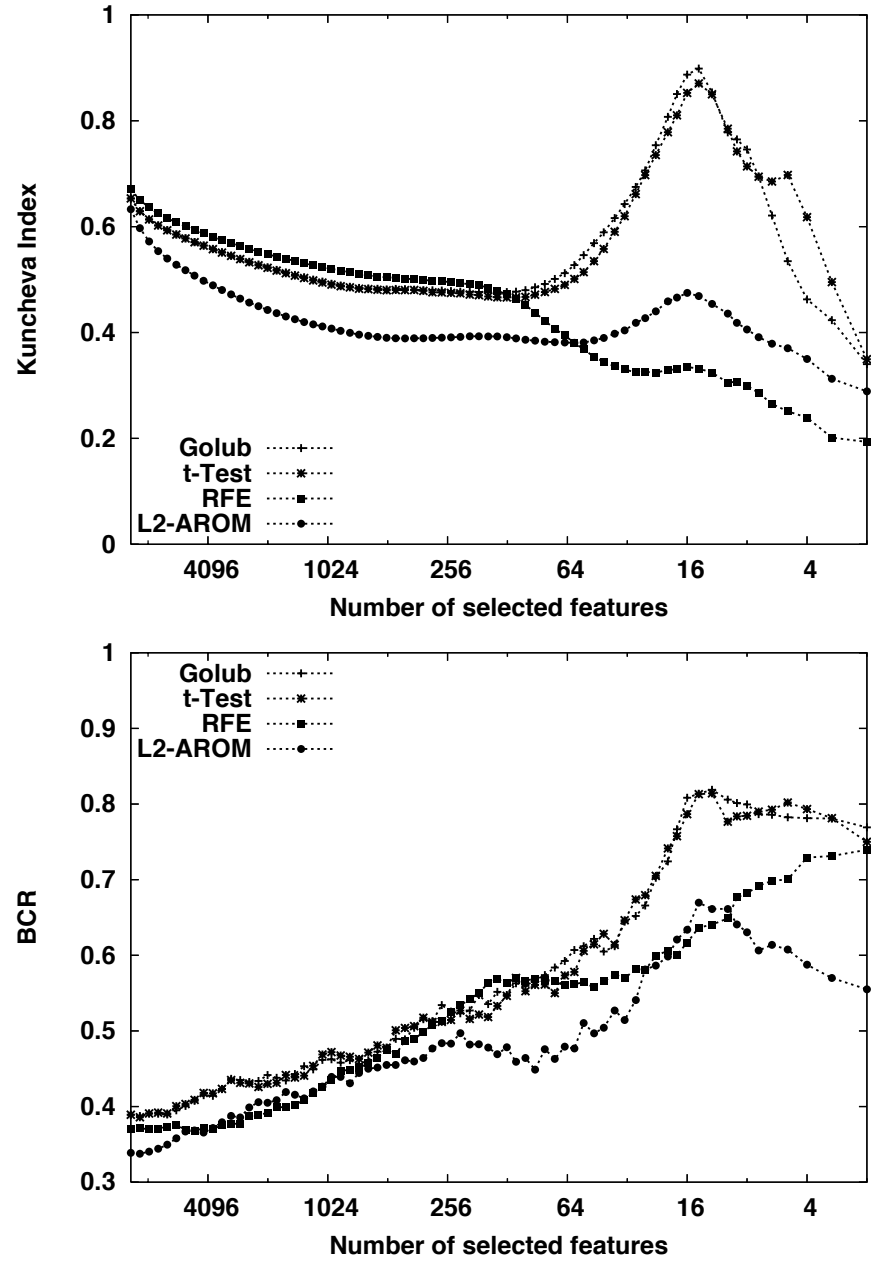


Figure 12.10 Stability (top) as measured by the Kuncheva Index and balanced classification rate (bottom) for the 33 Cristall samples of the preliminary cohort for different selection techniques and different signature sizes.

A unique signature of 14 biomarkers is first extracted and its content analyzed, since there seems to be a peak both in stability and classification performances around that size for univariate methods. This is done on the entire preliminary cohort, with RMA, GC-RMA, both combined with *t*-test and Golub Index, after a non-specific pre-filtering step. Those combinations lead to 4 signatures, detailed in Figure 12.7. The probesets are ranked from top to bottom by decreasing importance. With GC-RMA, both signatures contain the same 14 probesets, but in a different order. With RMA, 13 out of the 14 probesets are shared. Between the RMA/*t*-test signature and the GC-RMA signatures, 2 probesets are shared (211696_x_at and 204419_x_at). Between the RMA/Golub signature and the GC-RMA signatures, 3 probesets are shared (204419_x_at, 211696_x_at and 206834_at).

Table 12.8 shows the probeset ids, the GenBank ids, and alternative gene names corresponding to the GC-RMA signatures. Tables 12.9 and 12.10 show the same information for the RMA/*t*-test and RMA/Golub signatures respectively. Table 12.11 shows information about the probesets shared among the four signatures. Tables 12.8 to 12.10 suggest that it is mainly hemoglobin that leads to high predictive power with respect to allergic diagnosis. Most probesets of the different signatures relate in fact to a small set of common genes.

Table 12.7 Signatures obtained by combination of GC-RMA, RMA, Golub index and a *t*-test on the preliminary cohort.

Rank	GC-RMA/ <i>t</i> -test	GC-RMA/Golub	RMA/ <i>t</i> -test	RMA/Golub
1	217232_x_at	213515_x_at	209116_x_at	204848_x_at
2	214414_x_at	217232_x_at	217232_x_at	213515_x_at
3	209116_x_at	204419_x_at	214414_x_at	204419_x_at
4	211696_x_at	204848_x_at	211696_x_at	209116_x_at
5	221992_at	209116_x_at	221992_at	217232_x_at
6	211745_x_at	214414_x_at	209458_x_at	211696_x_at
7	209458_x_at	211696_x_at	217414_x_at	214414_x_at
8	204018_x_at	221992_at	204018_x_at	221992_at
9	213515_x_at	209458_x_at	211745_x_at	209458_x_at
10	217414_x_at	211745_x_at	211699_x_at	217414_x_at
11	211699_x_at	204018_x_at	213515_x_at	211699_x_at
12	204419_x_at	217414_x_at	204848_x_at	204018_x_at
13	204848_x_at	211699_x_at	240946_at	211745_x_at
14	206834_at	206834_at	204419_x_at	206834_at

Table 12.8 GC-RMA signatures ordered according to the *t*-test.

Affymetrix ID	Genbank	Gene name
217232_x.at	AF059180	hemoglobin, beta
214414_x.at	T50399	hemoglobin, alpha 1
209116_x.at	M25079	hemoglobin, beta
211696_x.at	AF349114	hemoglobin, beta
221992.at	AI925734	similar to nuclear pore complex interacting protein
211745_x.at	BC005931	hemoglobin, alpha 1
209458_x.at	AF105974	hemoglobin, alpha 1
204018_x.at	NM.000558	hemoglobin, alpha 1
213515_x.at	AI133353	hemoglobin, gamma G
217414_x.at	V00489	hemoglobin, alpha 1
211699_x.at	AF349571	hemoglobin, alpha 1
204419_x.at	NM.000184	hemoglobin, gamma G
204848_x.at	NM.000559	hemoglobin, gamma A
206834.at	NM.000519	hemoglobin, delta

Table 12.9 RMA/*t*-test signature.

Affymetrix ID	Genbank	Gene name
209116_x.at	M25079	hemoglobin, beta
217232_x.at	AF059180	hemoglobin, beta
214414_x.at	T50399	hemoglobin, alpha 1
211696_x.at	AF349114	hemoglobin, beta
221992.at	AI925734	similar to nuclear pore complex interacting protein
209458_x.at	AF105974	hemoglobin, alpha 1
217414_x.at	V00489	hemoglobin, alpha 1
204018_x.at	NM.000558	hemoglobin, alpha 1
211745_x.at	BC005931	hemoglobin, alpha 1
211699_x.at	AF349571	hemoglobin, alpha 1
213515_x.at	AI133353	hemoglobin, gamma G
204848_x.at	NM.000559	hemoglobin, gamma A
240946.at	AA778828	NA
204419_x.at	NM.000184	hemoglobin, gamma G

Table 12.10 RMA/Golub signature.

Affymetrix ID	Genbank	Gene name
204848_x.at	NM.000559	hemoglobin, gamma A
213515_x.at	AI133353	hemoglobin, gamma G
204419_x.at	NM.000184	hemoglobin, gamma G
209116_x.at	M25079	hemoglobin, beta
217232_x.at	AF059180	hemoglobin, beta
211696_x.at	AF349114	hemoglobin, beta
214414_x.at	T50399	hemoglobin, alpha 1
221992.at	AI925734	similar to nuclear pore complex interacting protein
209458_x.at	AF105974	hemoglobin, alpha 1
217414_x.at	V00489	hemoglobin, alpha 1
211699_x.at	AF349571	hemoglobin, alpha 1
204018_x.at	NM.000558	hemoglobin, alpha 1
211745_x.at	BC005931	hemoglobin, alpha 1
206834.at	NM.000519	hemoglobin, delta

Table 12.11 Probesets shared between GC-RMA and RMA signatures. Rows with a * at the end corresponds to probesets shared among all four signatures. The last row is only shared by three out of four signatures (all but the RMA/*t*-test signature).

Affymetrix ID	Genbank	Gene name	
204419_x.at	NM.000184	hemoglobin, gamma G	*
211696_x.at	AF349114	hemoglobin, beta	*
206834.at	NM.000519	hemoglobin, delta	

12.4.3 Signature Validation and Interpretation

Hemoglobin (*Hb*) is a molecule which role is to carry oxygen from lungs to places where it is needed in the body through blood vessels. Hemoglobin is made of several sub-compounds, which vary through an individual’s life. Adult hemoglobin is made of two hemoglobin- α and two hemoglobin- β , and also in a smaller proportion of hemoglobin- δ (97% is $Hb - \alpha_2\beta_2$, and only 3% is $Hb - \alpha_2\delta_2$). Fetal hemoglobin has hemoglobin- γ instead of hemoglobin- β ($Hb - \alpha_2\gamma_2$). Fetal hemoglobin has a higher attractive power with respect to

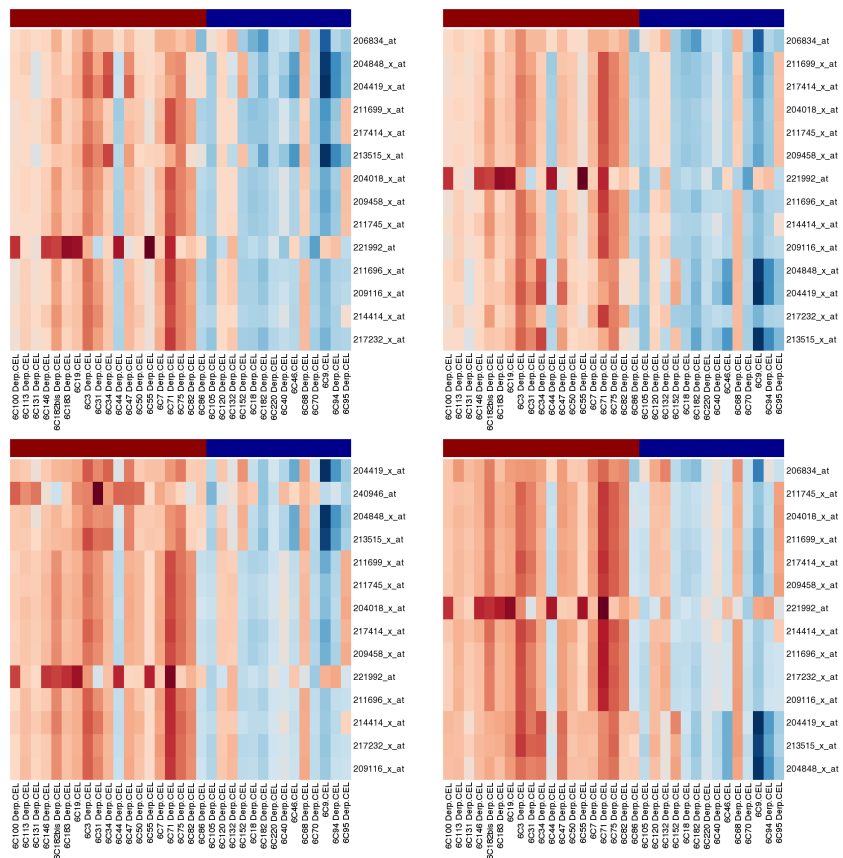
oxygen than adult hemoglobin. This makes fetus in the womb able to attract oxygen from the mother's blood. Normally, the production of hemoglobin- γ becomes inactive few weeks after birth and is replaced by hemoglobin- β . What the current experiments show is that 18-months-allergic children and 18-months-non-allergic children show different gene expression profiles at 6 months in terms of transcription rate of genes responsible for hemoglobin- γ .

Another component appearing in both RMA and GC-RMA signatures is a gene related to Nuclear Pore Complex (NPC). NPC regulates the transport of water-soluble molecules (including RNA and ribosomes) outside the cell nucleus, and proteins, carbohydrates, signal molecules and lipids inside the nucleus. A gene corresponding to a NPC interacting protein seems generally strongly up-regulated in allergic samples with respect to its expression level in non-allergic samples.

Those considerations can be observed on expression profiles in Figures [H.1](#) to [H.3](#) of Appendix [H](#). This choice of display has been made to show some coherence with other works projects⁴. However, this representation of expression profiles is somehow deceiving because it may suggest that everything below 0 is *down-regulated* and everything above 0 is *up-regulated*, which is not necessarily the case. Another way of viewing data according to signature is heatmaps. Figure [12.11](#) shows heatmaps for the four signatures.

⁴namely with GSK Biologicals

Figure 12.11 Heatmaps of the four signatures on the 33 samples. Top row: GC-RMA signatures. Bottom row: RMA signatures. Left column: *t*-test signatures. Right column: Golub signatures. The color bar at the top of each figure corresponds to sample labels: red is for non-allergic patients, blue is for allergic patients.



12.4.4 Age: A Confounding Factor?

The hemoglobin composition (α , β , γ and δ proportions) is known to change as the children get older. In particular, hemoglobin γ is supposed to be supplanted by hemoglobin β precisely after the first few months of life. The age of the patients is thus a potential confounding factor with respect to the identified signatures. The current section analyses the potential link between the exact age of the patients when blood samples were collected, and the

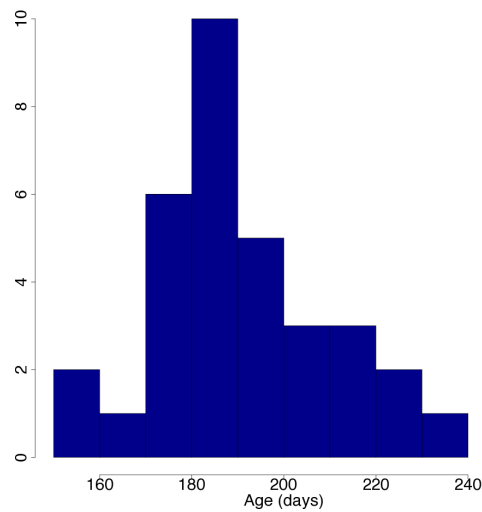
allergic status or the expression of genes in the potential signatures. All hybridized samples are supposed to be collected at 6 months but in practice, the precise moment when the collection was performed may vary. On the 33 patients cohort, the average age is 192.89 days, or ≈ 6.41 months⁵, with a standard deviation of 17.65 days. Figure 12.12 illustrates the ages distribution. The distribution is far from being narrow around the mean. The minimum age of collection is 159 days (5.2 months) and the maximum age is 235 days (7.7 months). The age range is thus of 2.5 months which is a variation of 41.6% of the expected age of 6 months.

We measure the correlation between the age (in days) and the allergic/non-allergic status. The linear correlation (Pearson correlation) between the age and the allergic/non-allergic status (respectively assumed to be 1 and -1) is -0.3186. Some anti-correlation is thus observed between age of blood sample collection and the allergic status. Figure 12.13 (left) illustrates the link between age of sampling and allergic status. Otherwise said, allergic children in the preliminary cohort tend to have been sampled globally a bit earlier than non-allergic patients. This is pure chance, since their status could not have been known at the time of sampling. This could be the source of a confounding factor. A *t*-test to assess the significance of the difference between means of the allergic and non-allergic vectors of ages outputs a *p*-value of 0.067. We can thus say with 93% confidence that the difference in ages between the two groups is significant.

The linear correlation between a binary vector and a continuous values vector is questionable, because linear correlation is not intended to be applied in such context. Correlation between age and the expression of each signature gene reveals that all genes in the four signatures are anti-correlated with age, though in a moderate way. All those correlations fall in the interval $[-0.3351; -0.0915]$. The 14 probesets in signatures are also correlated to the allergic/non-allergic status: $[0.5796; 0.6815]$.

⁵Supposing average months of 30.5 days

Figure 12.12 Ages distribution (in days) for the preliminary 33 patients cohort.



Although there is no strong proof, the correlation between age and the allergic status is a warning against a confounding factor. Another argument sustaining such hypothesis is clearly the interpretation of the signatures, related mainly to hemoglobin. Hemoglobin is strongly modified during the first months of life. To get rid of that potential bias, subsequent hybridizations have been done on samples from non-allergic children whose blood has been collected earlier, and samples from allergic children whose blood has been collected later. Figure 12.13 (right) illustrates the link between age of sampling and allergic status of the 73 patients final cohort. The situation looks more similar in the two groups than previously. The correlation between age and the allergic status in the larger cohort is -0.0721 instead of -0.3186 in the preliminary cohort. The hypothesis of the age as a confounding factor in the larger cohort seems much less likely than in the preliminary cohort. A new t -test on the age of the patients for the full 73 patients cohort with respect to the allergic status exhibits a p -value of 0.546 , thus much higher than the value obtained on the preliminary cohort (0.067). With such a p -value, we cannot say anymore that the ages differ from one status to the other in the larger cohort (rigorously, we cannot say that they are equivalent neither).

Subsequently, the four hemoglobin-related signatures have been tested on the full 73 patients cohort. Five hundred subsamplings of 90% of the data have been drawn with the data restricted to one of the four signatures. On each subsampling, a linear SVM is built with the C constant defined with the Joachims heuristic. Normalization is performed via IQR-Normalization. The results are shown in Table 12.12. It appears that the signatures are indeed an artifact of the confounding factor. They are unable to classify correctly the full cohort. All samples are classified in the majority class.

Figure 12.13 Ages with respect to non-allergic and allergic status. Left: preliminary 33 patients cohort. Right: final 73 patients cohort.

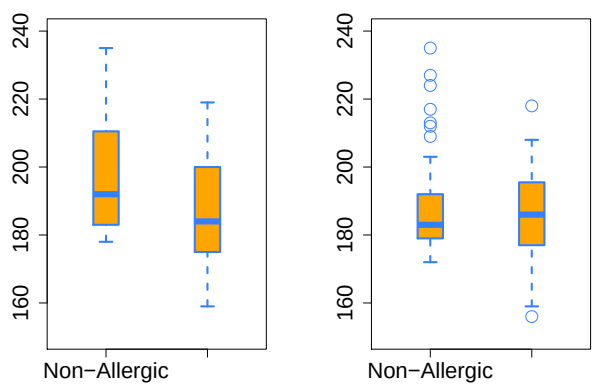


Table 12.12 Performances of the four hemoglobin/NPC signatures on the 73 patients cohort. Average of 500 runs.

Signature	BCR	ACC
GC-RMA/ <i>t</i> -test	0.498±0.02	0.711±0.03
GC-RMA/S2N	0.498±0.02	0.711±0.03
RMA/ <i>t</i> -test	0.498±0.02	0.713±0.02
RMA/S2N	0.498±0.02	0.711±0.03

12.5 Discussion

This chapter extensively describes the gene expression data analysis of the Cristall project. The main aim of this part of the project is to identify the smallest possible number of biomarkers on which a prognosis model with high prediction performances can be built. The prognosis task is to predict the allergic status of children at 18 months based on their gene expression data at 6 months. The samples have been stimulated with an allergen (nDer p1).

First, we clean the data by excluding some potential outliers. The final cohort contains data from 73 patients. A first descriptive analysis outlines that discriminating the allergic patients from the non-allergic patients might not be easy, based on a univariate t -test on every dimension. A comparison with another dataset makes this assumption stronger. A list of 43 genes and micro-RNA identified as potentially relevant in the context of allergy is studied. Its discriminative power seems again poor. Experiments are led to obtain new biomarker signatures. Seven feature selection methods described and/or proposed in this thesis are tested: t -test, Golub's ratio, RFE, L_2 -AROM, PS- L_2 -AROM, ensemble selection based on linear SVM and feature selection based on random forests. The PS- L_2 -AROM approach makes possible the incorporation of the prior knowledge (the 43 genes/micro-RNA). Once the biomarkers are selected, a linear SVM is built and tested. The obtained performances are poor on both the prediction and the signatures stability point of views.

This negative results is confirmed by an unsupervised clustering approach. It fails at discovering coherent groups of patients related to the allergic status, even based on a subset of the original probesets selected by the seemingly best technique on this dataset (ensemble SVM for selection, described in Chapter 7).

About a quarter of the list of the a priori relevant markers is dropped by the partially supervised feature selection technique. It should be investigated if the rest of the list is indeed relevant in the context of allergy. The use of prior knowledge leads to the best stability results. This pleads for its relevance.

This chapter also proposes positive results. Some experiments stress the benefits of the methods proposed in this thesis, such as an improvement of

the stability with PS- L_2 -AROM and the ensemble selection based on linear SVM. Another setting is tested in which the genes to favor via PS- L_2 -AROM are defined as the most discriminative probesets based on a t -test. A stability gain is observed in this case, though not as strong as the one obtained with expert knowledge. As in Chapter 8, it is shown that PS- L_2 -AROM is able to partially depart from the prior knowledge.

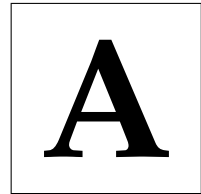
This chapter is also an opportunity to illustrate several practical aspects of microarray data analysis. Visualization tools are proposed (PCA, heatmaps and clustering). In all cases, it is clear that those tools should not be used as predictive tools. They however make possible the identification of potential outliers. The issue of avoiding normalization and selection biases is also discussed. On top of that, this chapter provides an illustration of the concept of confounding factor. In this case, the precise age of the patients is likely to be a bias in the preliminary cohort since the mean of ages for allergic and non-allergic patients differ with a significance 93% in this cohort. It led to the discovery of signatures mainly related to the hemoglobin composition. This was a surprising result, and led to more caution in subsequent experiments. The age balance in both classes has been corrected. This stresses the importance of signature interpretation by experts. When the age bias is corrected, the hemoglobin signatures are not effective anymore.

In order to get more results with respect to the Cristall project objectives, several tracks can be considered. The number of microarray experiments could be increased, but a significant improvement is not likely to occur with the number of available blood samples satisfying the quality and quantity criteria. Another eventuality is that there is indeed no signal to discover at 6 months predictive of the allergic status at 18 months. It can be the case that allergy is not visible at this age, or that it is not visible in expression. Another time point could be considered, for example 12 months. Knowing if the allergic status can even be diagnosed at 18 months based on gene expression only is still an open question. Another approach could be considered, for example based on single nucleotide polymorphism analysis, or via an epigenetic study (see Appendix A). Working on both microarray and clinical records at the same time should be considered. Decision tree-based approaches such as random forests and extra trees make possible to work on heterogeneous data.

Part V

Appendix

Molecular Biology for Dummies (and Computer Scientists)



The first years of bioinformatics have been mainly ruled by biologists faced to difficult problems. They learned to program and to develop or apply algorithms driven by necessity, most of the time without a deep background in computing sciences. When bioinformatics became more popular, the field evolved and attracted more computer scientists and mathematicians, and is nowadays more often referred to as computational biology. What is true in one way is also true in the other: people from applied sciences most of the time lack a biological background. This chapter intends to deliver, in a nutshell, some elements of molecular biology. It is of course not exhaustive (many books on molecular biology count several thousands pages) and the concepts it introduces are not fully detailed. The aim is just to provide the basics of the field in order to better understand the issues related to genomic applications. This chapter is mainly inspired by the books of [Hunter \(2009\)](#); [Lane \(2005\)](#); [Jones and Pevzner \(2004\)](#); [Cristianini and Hahn \(2006\)](#); [Kessler \(2006\)](#).

The chapter begins by describing the building blocks of complex life in Section [A.1](#). Section [A.2](#) introduces some of the most advanced molecular biology tools (for proteomics, sequencing, expression analysis, etc.). The chapter ends with a discussion on the perspectives of the application of molecular biology tools in Section [A.3](#), namely with the rise of next-generation ultra-high-throughput sequencing.

A.1 The Central Dogma

The main claim of the DNA structure discoverers, [Watson and Crick \(1953\)](#), is that information about how to make proteins is encoded in DNA and that DNA can be copied to make more DNA or transcribed into RNA, which codes for proteins. Roughly said, the flow of information is always one of the followings:

- From DNA to DNA
- From DNA to RNA to protein

This is the *Central Dogma* of molecular biology. But what are DNA, RNA and proteins? And what are their functions? This is the topic of the next sections.

A.1.1 The Cell

The cell is the simplest unit that supports independent life, both on a structural and functional point of view. The structure of a cell varies according to the considered form of life. Most elements described below are illustrated in [Figure A.1](#).

Structure

Whatever the cell type considered, cells are delimited by a tough but permeable cell wall. The very concept of living individual is related to the presence of a membrane to distinguish the inside of the individual from the rest of the universe. Depending on the type of cell considered, this wall, or membrane, may take different forms. The shape of the cell is maintained by a network of fibers together constituting the cytoskeleton. This structure is dynamic, allowing cells to move and change shape.

Nucleus

All multicellular organisms are made of particular cells named eukaryotes. All eukaryotic cells have a membrane-enclosed nucleus (as opposed to the simpler prokaryotes). The area within the cell but outside the nucleus is filled with a fluid called the cytosol. The nucleus contains the blueprints in order to build every functional and structural units of the cells. These blueprints are encoded in large molecules called DNA (further described in [Section A.1.2](#)). Those molecules are tightly packaged into chromosomes with the help of

some structural proteins such as histones. All of the contents of a eukaryotic cell that are not the nucleus or the cell wall forms the cytoplasm.

DNA & RNA

DNA is read by a family of devices named RNA Polymerases¹. Those transcribe inert DNA into RNA. The roles and structure of RNA are detailed in Section A.1.3. Once encoded, RNA is sent outside the nucleus through nuclear pores (a kind of gates in the nucleus membrane). There in the cytoplasm, RNA is used to produce proteins (Section A.1.4).

Ribosome

In addition to the nucleus, complex cells have other subunits constituting the cellular machinery, floating in the cytoplasm. Those subunits are called organelles and many of them are also delimited by a membrane. Among those organelles, the ribosomes assume a particular function: synthesize the proteins, based on an RNA molecule. Ribosomes are themselves made of proteins and of a particular kind of RNA. Ribosomes are built by another organelle found in the nucleus and called nucleolus.

Mitochondria

Mitochondria are another type of organelle only found in eukaryotes and intrinsically related to the existence of complex forms of life. Mitochondria are suspected to be the remaining of an ancient parasite engulfed by archaic simpler cells, for the benefit of both the host and the parasite. As a reminder of this distant past, the mitochondria have their own genetic material, noted mtDNA. On top of its surprising origin, mitochondria have three particularities. First, mitochondria are responsible for the production of the cell energy. To be more precise, they do not *create* energy, for the sake of Lavoisier, but they capture energy of elements² floating in the cytoplasm and transform it into the universal cell energy currency: the ATP³. Mitochondria may be seen as powerful hoovers, storing energy (in the form of ATP) behind their double membrane, and releasing it on demand. The break of a phosphate bond in ATP leads to ADP⁴ and phosphate group. This reaction is beneficial from a thermodynamic point of view. The resulting energy is used to power most of

¹A polymerase is composed of several subunits that favors the creation of a polymer

²Those elements mainly come from nutrition

³ATP: Adenosine TriPhosphate

⁴ADP: Adenosine DiPhosphate

the biochemical work in the cell, from muscular contraction to protein synthesis. This function of mitochondria is known as cellular respiration, as it is the oxidation of foodstuffs to produce (store) energy. A regular human cell counts several thousands mitochondria. Second, mitochondria are responsible for the programmed cell death: the apoptosis. Cells in a multicellular organism have to be frequently replaced when they become either unnecessary or damaged. The failure of apoptosis in a cell may lead to cancer when cells begin to proliferate instead of just die and being replaced. The third particularity of mitochondria is to be transmitted only from the mother to its offspring. Fathers do not pass their mitochondria over. The mitochondrial DNA is thus not recombined and that is why it is used in phylogeny to trace back lineage through many generations: it is more stable. Mitochondria are a fascinating topic of life sciences. They even inspired the concept of medichlorians in Star Wars movies.

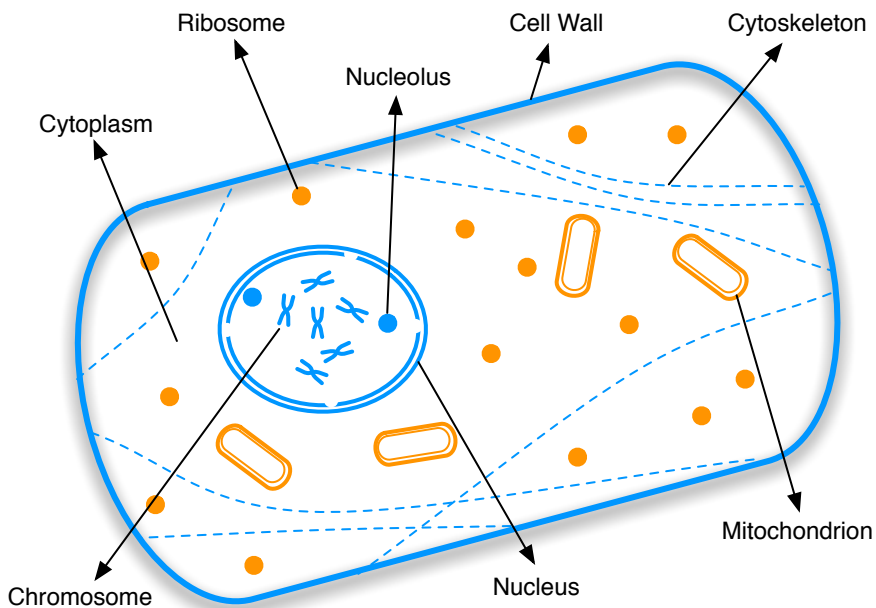


Figure A.1 A schematic view of an eukaryotic cell

A.1.2 DNA

DNA Structure

DNA is the acronym of DeoxyriboNucleic Acid and is the molecule responsible for heredity. Its role is to encode genetic information, and it is composed of a double helix. Both strands of the helix are composed of a sequence of bases. Those bases are deoxyribonucleotides, generally called nucleotides for short. There are four such nucleotides: adenine (A), cytosine (C), guanine (G) and thymine (T). The two strands of the helix are complementary. Adenine in one strand always pairs with thymine with a double hydrogen bond. Cytosine in one strand always pairs with guanine with a triple hydrogen bond. The energy required for the breakage of a G-C pair is thus higher than the one required for an A-T pair. Two separate but complementary sequences will hybridize in presence of each other to form a double-stranded DNA molecule. This phenomenon of hybridization is the key for most genomic tools. The DNA structure is pictured in Figure A.2. The length of a DNA double stranded sequence is often counted in term of its number of *base pairs*. The DNA molecules are tightly packed around other molecules called histones. They together form the chromatin, which constitutes the chromosomes. Figure A.7 offers an illustration of the packaging of DNA.

Genes

The information contained in DNA is encoded in a particular language. This language is based on a four-letters alphabet: the four nucleotides (A, C, G and T). Nucleotides form words of three characters named codons. Each codon corresponds to some building block of the proteins called an amino acid. A subsequence of DNA containing the information to produce a protein is a gene. Genes are the smallest unit of inheritance. The creation of a gene product from a gene is the expression of a gene. To be more complete, a gene might also code for another end product than a protein: a functional mRNA. More on it in the next section. DNA is made of coding parts, exons, and non-coding parts, introns. A mnemonic is that exons are expressed. Both introns and exons are interleaved, so that a gene generally contains several exons and introns. It may happen that a given gene is copied several times in a DNA molecule.

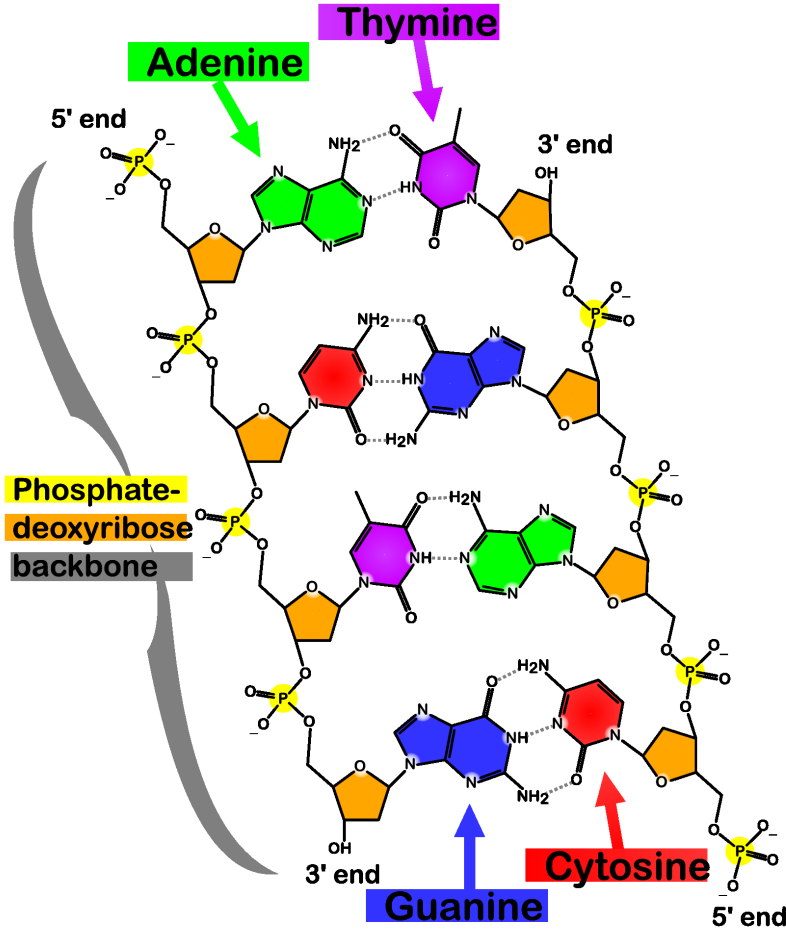


Figure A.2 Close look at the DNA double-stranded molecule structure. Every nucleotide is composed of a sugar (orange) and a phosphate (yellow), forming the strands backbone. Only the radicals change. Hydrogen bonds are shown between A and T (2 bonds) and between C and G (3 bonds).

Genotype vs. Phenotype

The phenotype of an individual is the entirety of its observable characteristics. Those characteristics result from the use the organism makes of some of its genes. The entire set of the genes of an organism constitutes the genotype, including genes that do not influence observable traits (i.e. the phenotype). More generally, the genome is all the physical hereditary material contained

in an organism.

DNA Damage

DNA molecules can get damaged. This may have terrible effect on the cell, stopping the production of important elements, or producing corrupted elements damaging the cell in turn. The cell has a machinery most of the time able to repair damaged DNA. In some cases, the DNA cannot be repaired. Such cases generally triggers the cell suicide, the apoptosis.

Replication

According to the central dogma, DNA can be replicated. This is done with the help of an enzyme⁵ called the DNA-directed DNA polymerase. It creates a new DNA molecule that has a complementary sequence to the existing molecule. DNA can also be copied into RNA. This is done by a similar enzyme named DNA-directed RNA polymerase. It creates an RNA molecule that has a complementary sequence to the DNA molecule.

Mitochondrial DNA

The mitochondrial DNA (mtDNA) takes the form of a circular chromosome, unlike the nuclear DNA which is straight. This form of chromosome is typical of bacteria, which is another reminder of the origin of mitochondria. Five to ten copies of this circular chromosome can be found in each mitochondrion. Human mtDNA contains 13 protein-encoding genes and some extra genes coding for RNA used to produce proteins on site.

A.1.3 RNA

DNA is stored safely in the nucleus and contains the instructions required to produce proteins. Proteins are produced outside the nucleus by the ribosomes. The mean of transmission of information from the nucleus to the ribosomes is the RiboNucleic Acid, or RNA. It is a linear polymer of nucleotides whose biological functions include transmitting information, catalysis and regulation of gene expression. The building blocks of RNA are ribonucleotides, generally called nucleotides, as in the case of DNA nucleotides. And just like DNA, RNA is a sequence of four different nucleotides. Those are the

⁵An enzyme is a biological molecule favoring a particular chemical reaction. It is generally a protein, but can also be RNA in some cases.

same as the ones making DNA, except that thymine (T) is replaced by uracil (U).

Types of RNA

Several types of RNA exist and maintain different functions.

- The RNA referred to in the previous paragraph is the messenger RNA (mRNA). It is the transcript of a DNA sequence, carrying protein building instructions from the nucleus to the ribosomes. DNA is first transcribed into pre-mRNA. In a second step, the portions of code corresponding to introns are removed from the pre-mRNA to form mature mRNA.
- The ribosomes are themselves mainly made of a particular type of RNA, the ribosomal RNA (rRNA).
- Another form of RNA is the transfer RNA (tRNA). At one end, tRNA bonds to a specific amino acid. The other end of a tRNA hybridizes to the corresponding codon on an mRNA. Transfer RNA are recruited by the ribosomes at the time of mRNA reading. Transfer RNA is thus the physical realization of the genetic code.
- Micro-RNA (miRNA) is a single-stranded form of RNA short sequences playing a role in the regulation of protein production.
- RNA is normally found in a single stranded form. A very particular type of RNA is the double-stranded RNA, generally 20 to 25 nucleotides long and named small interfering RNA (siRNA). A given siRNA sequence triggers the destruction of single-stranded RNA containing that sequence. As a consequence, the expression of the corresponding end product is strongly (or totally) repressed. This mechanism is known as RNA interference (RNAi).
- RNA molecules that do not act as messenger RNA, such as rRNA and siRNA for example are sometimes referred to as non-coding RNA (ncRNA).

Alternative Splicing

The transcription of a DNA sequence into an mRNA sequence is made of several edition steps. The initial RNA transcript can be further processed

into different mRNA molecules including some exons and excluding others. As a result, the same initial DNA sequence codes for different end products (generally proteins). This phenomenon is known as alternative splicing.

From an evolutionary point of view, it is believed that the origin of life is strongly related to RNA and that the earliest forms of life stored their genetic information into RNA, not DNA. Some organisms still live this way, but it is rare. It is the case of some viruses such as HIV, the virus causing AIDS. Such viruses are therefore named retro-viruses. It is an exception to the central dogma.

RNA to cDNA

Based on a given messenger RNA molecule, it is possible to synthesize the complementary DNA (cDNA) molecule in a laboratory by using the enzyme reverse transcriptase. It is another exception to the central dogma but is artificial. Note that cDNA is only made of coding nucleotides since mRNA does not contain code portions corresponding to introns.

A.1.4 Proteins

Proteins are large biological molecules made of a linear polymer of amino acids. Because of their size, they are called macromolecules (this is also the case of DNA and RNA). Proteins are in charge of nearly everything within a cell: they can be enzymes, hormones, antibodies, they can contribute to the structure of the cell, they can regulate gene expression (transcription factors) and are even in charge of communication between cells.

Protein Production

As a given mRNA encounters a ribosome, the ribosome starts reading the mRNA sequence by groups of three nucleotides (codon). Each group corresponds to a particular tRNA, itself carrying a particular amino acid. It is the job of ribosomes to recruit the tRNA corresponding to the codon being read. Once the tRNA has been recruited, the ribosome goes to the next codon, and so on. Each amino acid is added to the previous one, finally forming the generally long protein sequence. There are 20 varieties of such amino acids.

Protein Structure

The structure of proteins can be seen from four levels, respectively called the primary, secondary, tertiary and quaternary structures. The primary structure of a protein is simply the sequence of amino acids making the protein. A protein sequence, although linear, does not live in the cell in a straight shape. Proteins form complex structures which can be divided in simple substructures which make the secondary structure. This second level can be described in terms of alpha helices and beta strands. An alpha helix is a set of amino acids taking a corkscrew shape, while a beta strand is a set of amino acids taking an extended, rodlike shape. This secondary structure is useful for the study of proteins, allowing researchers to take a higher view than the amino acids sequence level. However, most of a protein function and behavior is dictated by its tertiary structure, which is how alpha helices and beta strands organize themselves to form complex molecular machines. While primary and secondary structure can be described as sequences, the tertiary structure is intrinsically a three-dimensional description of the molecule, thus more difficult to analyze. It can be described either by the precise three dimensional position of every atom within the protein, or by a set of angles between the elements of the secondary structure. Several subparts of the three-dimensional structure of a protein have particular roles, such subunits are called sites. Sites might be active or not, depending on the context. It is noticeable that proteins can change their three-dimensional shape according to their environment. It is also remarkable that proteins do not carry only chemical actions. Some of them are able to perform physical work on other products. This is the case for example of the ATP synthase, a protein working surprisingly similarly to an electrical turbine. It is driven by an electric current, gets into rotation, and acts mechanically on other molecules to synthesize the ATP. Hemoglobin is another example of shape variation depending on the context. Hemoglobin is in charge of oxygen transportation. It virtually *grasp* oxygen atoms, carries them through blood vessels, and release them at their desired location. Finally, the quaternary structure refers to the structure formed by an assembled set of proteins functioning together.

A.2 Genomic Technology

Now that the building blocks of life have been introduced, a discussion on modern molecular biology tools can take place. Some of these tools have been developed in the last 10 to 20 years and only become mature now. The first

complete sequencing of an independent organism (which was a bacteria) took place in 1995, and made us enter the post-genomic era. The human genome has been sequenced only in 2001. From an industrial point of view, the possibilities are vast and closely interleaved with academic research.

Bleed & Read

The endeavor of many researchers is to make real the concept of *Bleed and Read*. According to that principle, a simple blood drop should be sufficient to output a diagnosis on a large series of pathologies nearly instantly with a high sensitivity and specificity.

It is the proteins that should be *measured* within that blood drop, since they are in charge of the structure, the metabolism and the communication of the cells. The next section describes the set of techniques dedicated to the study of proteins, and what they effectively measure. It also discusses why they are not yet fully able to perform such analyses at a large scale. The subsequent sections present other technologies related to genome analysis and gene expression as alternatives to proteomics.

A.2.1 Proteomics

Proteomics is the set of techniques studying proteins. Many aspects of proteins can be studied.

Identification

To identify proteins in a sample, several techniques are available. Most of them are based on the mass and/or the charge of proteins. It is the case of centrifugation, mass spectrometry or electrophoresis. Figure [A.3](#) displays an example of spectrometry data. Each peak corresponds to the amount of particular components having a given mass and charge ratio.

Macromolecular Structure

The structure of the proteins are not easy to measure. The two main approaches to do so rely on X-ray crystallography and nuclear magnetic resonance (NMR). The X-ray crystallography for example analyzes the diffraction of X-rays through a crystal of a protein to discover its structure.

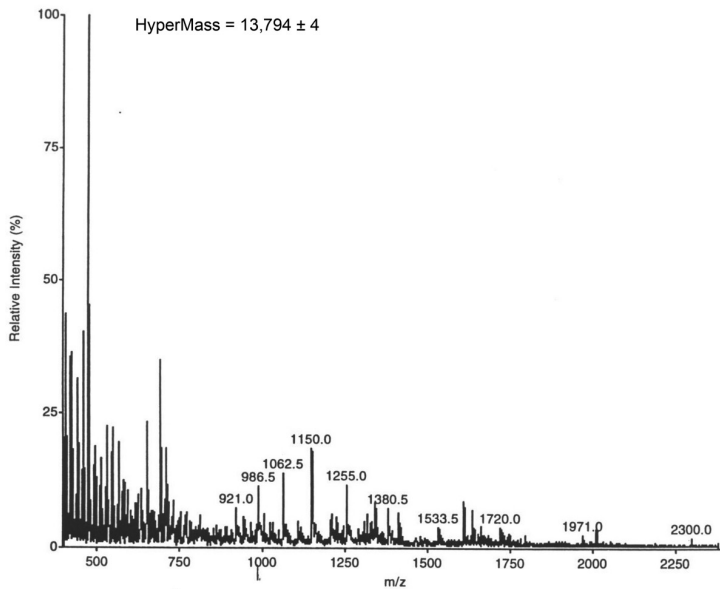


Figure A.3 An example of spectrometry data.

Molecular Activity

The identification of a protein and the knowledge of its structure is generally not sufficient to discover its function(s). An interesting phenomenon is the RNA interference described in the previous section (RNAi). This phenomenon can be artificially induced by introducing double-stranded small interfering RNA (siRNA). It will slow down, or totally stop the production of proteins which corresponding mRNA contains the sequence of the siRNA. This is called protein knockdown or knockout. Once the protein production is stopped, its function might not be ensured anymore, and its role can thus be deduced.

Protein Localization

Every cell of an individual contains the same genetic material. However, different cells can compose different tissue type, and activate different internal functions. This stresses that all the proteins a cell is able to produce are not produced everywhere, or at the same rate, in the organism. An interesting aspect of proteomics is the study of protein localization. The mainstream approach to do so is the fluorescent protein tagging. This technique consists in

artificially editing the genetic code of an individual (generally not humans) to insert a small code corresponding to the production of a fluorescent protein. This insertion takes place near the code of the protein of interest. When the gene corresponding to this protein is expressed, the fluorescent protein gene is also expressed. The protein can be detected with light of an adequate wavelength. The most used protein is the green fluorescent protein (GFP), a jellyfish protein, but other colors exist as well, namely red (RFP), blue (BFP) and yellow (YFP). Figure A.4 shows two examples of the use of fluorescent proteins. One is in the context of bone cancer using two markers, green and red fluorescent proteins (Gunkel *et al.*, 2009), the other is an artistic use of GFP by Eduardo Kac's, an artist who has produced two exhibits based on GFP technology. An excellent resource on GFP can be found on the website of the chemistry department of the Connecticut College, at <http://www.conncoll.edu/ccacad/zimmer/GFP-ww/GFP-1.htm>.

Protein Docking

The design of new drugs often requires a physical match between a protein and the active component of the drug, so that it can *dock* on a specific protein. It is also important to determine if the drug would be active on non-targeted proteins, which should be avoided in order to limit undesired side-effects. Manually scanning whole protein databases is too much time-consuming, if not impossible. Promising developments have been recently proposed in the field of proteomics to automatically detect such docking sites on proteins surface in an automated way (Giard, 2010). The proposed approach relies on machine learning tools.

Limitations

Proteomics suffer from three main limitations, although it is a very active field of research. First, these techniques are generally not able to process many proteins at a time. There is by now no satisfactory technology able to perform a screening at the protein level, as microarrays do for gene expression. And this is due to the difficulty of designing such arrays given the three-dimensional structure of proteins. If an important site is used to bind the protein to a glass array, it will not be available for detection. Furthermore, proteins do not have an hybridization capability as RNA or DNA. This also makes their detection uneasy. The second limitation is precisely the difficulty of measuring the three dimensional structure in an efficient and convenient way. On top of that, this conformation is likely to change according to the context. The third limitation

of these techniques is related to a lack of exhaustivity and/or specificity. For example, the RNAi technique might slow down the production of a protein. A visible effect might be noted, but it is likely that other effects will be missed, either because those effects should not occur in the particular condition of the experiment, or because other similar proteins start to ensure partially the role of the deficient protein.

Since proteomics are not currently totally satisfactory, what can be measured instead? This is the topic of the following sections.

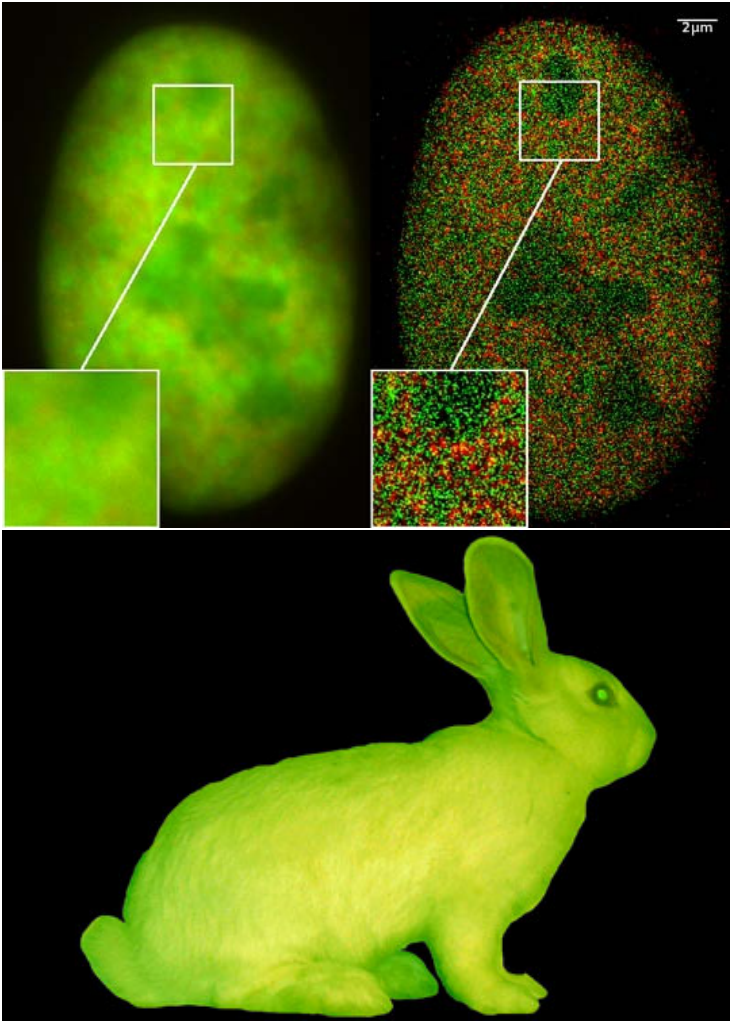


Figure A.4 Top: View of a nucleus of a bone cancer cell: a normal high resolution fluorescence microscopy is not able to distinguish details of its structure (top left). Using the two Color Localization Microscopy 2CLM (top right) it is possible to localize 70,000 histone molecules (red: RFP-H2A) and 50,000 chromatin remodeling proteins (green: GPF-Snf2H) in a field of view of $470\mu m^2$ with an optical depth of $600\mu m$. Bottom: Alba, a genetically modified green fluorescent rabbit. In normal light, it seems totally normal. When UV light is turned on, every cell of it starts glowing.

A.2.2 Genome Analysis

Sequencing

Nowadays genomes can be fully sequenced at a cost continuously decreasing. However, having access to the genome of a single individual is not so useful. What is interesting is to have access to plenty of genomes of the same species. Speaking about THE human genome, one has to realize that there are as many genomes as there are individuals. The human genome is thus some kind of average genome. This average genome can be represented as a long sequence in which each position has a probability of appearance of every nucleotide. Some parts are highly conserved across individuals, others are highly variable. Two main measures are done on genomes: single nucleotide polymorphisms (SNP) and copy number variations (CNV).

SNP

A *single nucleotide polymorphism*, also called a *snip* is a given coordinate in the genome where a particular nucleotide is expected for a vast majority of individuals, and where the presence of another nucleotide has a known impact on the individual's metabolism. For example, people with (C,C) genotype at position rs4988235 are more likely to be lactose intolerant w.r.t (C,T) or (T,T) ones. The pair of nucleotides represent the same position on two alleles. The list of SNP with known effects is steadily growing. An open source repository of such snips is available on SNPedia at <http://snpedia.org>. It is continuously updated.

CNV

Some portions of DNA can be copied several times. Several studies try to infer the impact of the number of copies of each gene on physical traits. This *copy number variation* study follows the same motivations as snips discovery.

Applications

Several companies began in the mid 2000's to offer particular genomic services. Unlike the rest of the genomic business, clients of those services can be anyone, not just laboratories or pharmaceutical companies. It typically works as follows: A client orders a kit on the web, and receives it at home few days later. The kit consists of a device collecting some biological material. For example, one of those kits is inspired by the bleed & read principle, but could

be renamed spit & read. The kit is then sent back to the company. A few days later, about 600,000 positions of the genome corresponding to potential snips have been analyzed and the results displayed on a web page, accessible only to the concerned client. CNV are also analyzed. These services come in several flavors, for a cost between 300 and 1,000 dollars. 23AndMe, DecodeMe and Navigenics are three companies active in that business and mainly focused on medical risks related to the analysis of the genomes. 23AndMe is a company mainly funded by Google. GeneBase is more oriented towards ancestry links and phylogenies. Refunds are provided for whole-family subscriptions. Other companies like DNA¹¹ proposes paintings created on the basis of the analysis of a given genome. SNPedia also offers a free and open source software tool to re-analyze a user's genome in a text file format. Such files can be obtained via 23AndMe, DecodeMe or Navigenics.

A.2.3 Gene Expression

Gene expression can easily be measured with microarrays. Gene expression is the closest proxy to the proteins production. Microarrays measure the amount of given mRNA sequences present in a sample under a given set of conditions. They make the assumption that the measured amounts correspond to the proteins production rate. The main trait of microarrays is to be able to measure the expression of up to tens of thousands of genes in a single experiment.

The measure itself is based on the capability of single-stranded DNA and RNA to hybridize with their complementary sequence. Predefined single stranded sequences (called probes) are fixed on a small glass chip. RNA is extracted from the sample to analyze and is marked with a small component (either a radioactive or a fluorescent marker). Marked RNA sequences from the sample hybridize with the corresponding probe on the chip. A scanner is then used to measure which probes (of known sequence) have been hybridized. Since each probe has millions of copies of itself on a single chip, the signal is not just binary. The intensity measured for a given (set of copies of a) probe really reflects the amount of RNA corresponding to that probe in the sample under study. The composition of the probes, their length, the type of marker, etc., depend on the manufacturer.

The expression of the genes is variable through time, conditions, environment and are thus a mean of fine analysis. This strength also makes their

analysis more complex, as discussed in this thesis. They are subject to both experimental and biological noises.

Density

Microarrays can be divided in two broad families: high-density and low-density microarrays. High-density arrays can typically measure several tens of thousands of genes. This screening capability is useful when no set of genes is readily identified for further analyses. They are generally manufacturer-designed. Each manufacturer sells chips with a predefined set of genes on it. For example, they may have one chip for humans, one for mice, one for a given bacteria, etc. The high-density technology is expensive. The production platform typically costs several hundreds thousands euros and each microarray chip experiment costs more or less 1,000 euros. This explains why most microarray datasets are so small in terms of the number of chips hybridized. Low-density arrays measure from tens to hundreds of genes. They are really useful in refinement studies on a given set of genes. Most of the time, they can be user-defined, unlike high-density arrays. The list of probes to be spotted on the chip can be specified by the client. They are generally one order of magnitude cheaper than high-density arrays. Some laboratories are even able to produce their own hand-made chips.

Two-color vs Single Chips

Microarray chips can be two-color or single chips. In two-color chips⁶, two samples are hybridized on the same chip. Those two samples are supposed to be in different conditions. For example, one might be in a hypoxia condition, and the other in normoxia condition. The material extracted from one sample (the *control sample*) is marked with a given dye (typically Cy3, green). The material from the other sample (the *experimental sample*) is marked with another dye (typically Cy5, red). Two-color chips are illustrated in Figure A.5. What is analyzed is not only the intensity, but the color itself as a measure of the relative expression of the test sample with respect to the control sample. In single chips, only one marker is used, and each chip corresponds only to one sample. What matters in this case is the luminous intensity. Figure A.6 shows a single-color chip. Two-color chips undergo several drawbacks. First it has been proved that for a given RNA concentration, the intensities produced by Cy3 and Cy5 dyes are not equal. The expression appears to be different while

⁶Sometimes, those chips are also referred to as Dual Chips

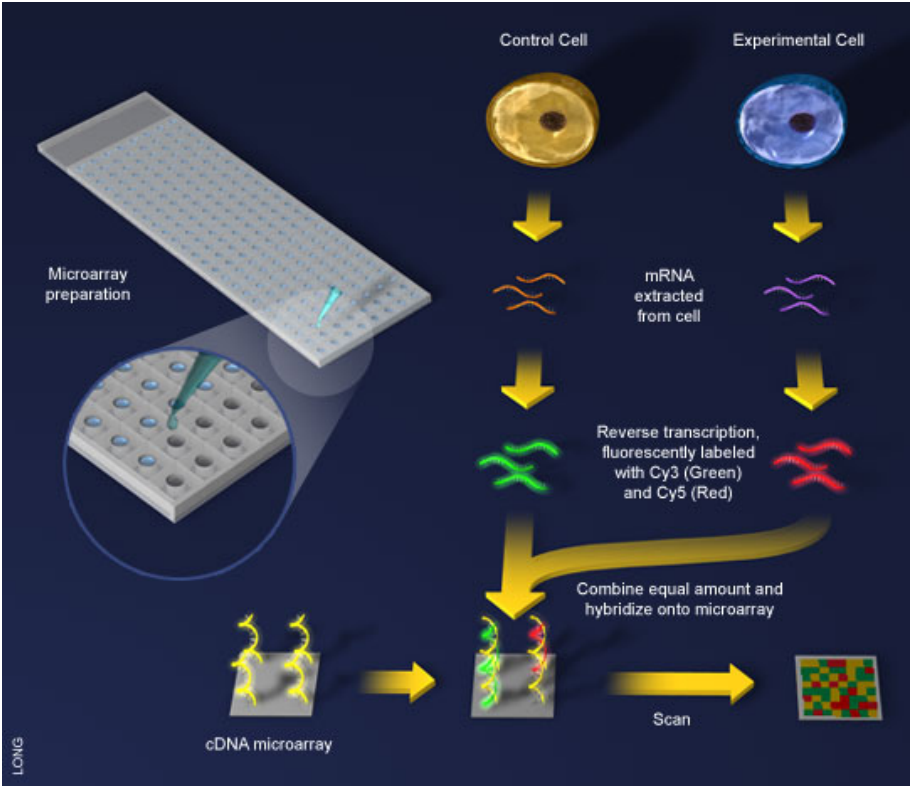


Figure A.5 Two-color microarray chip.

it might not be the case. To overcome this limitation, the dye-swap manipulation has been proposed. It comes to redo the experiments while interchanging the dyes among control and experimental samples. It doubles the global cost of the experimental campaign. On top of that, this technology assumes paired samples. This means that control and experimental samples are supposed to be identical in all points excepted the phenomenon under study. It is not always possible to provide such pairing. A given patient cannot be at the same time diseased and healthy. The most widely used gene expression microarray technology is the Affymetrix Genechip technology, described in more details in Appendix C.

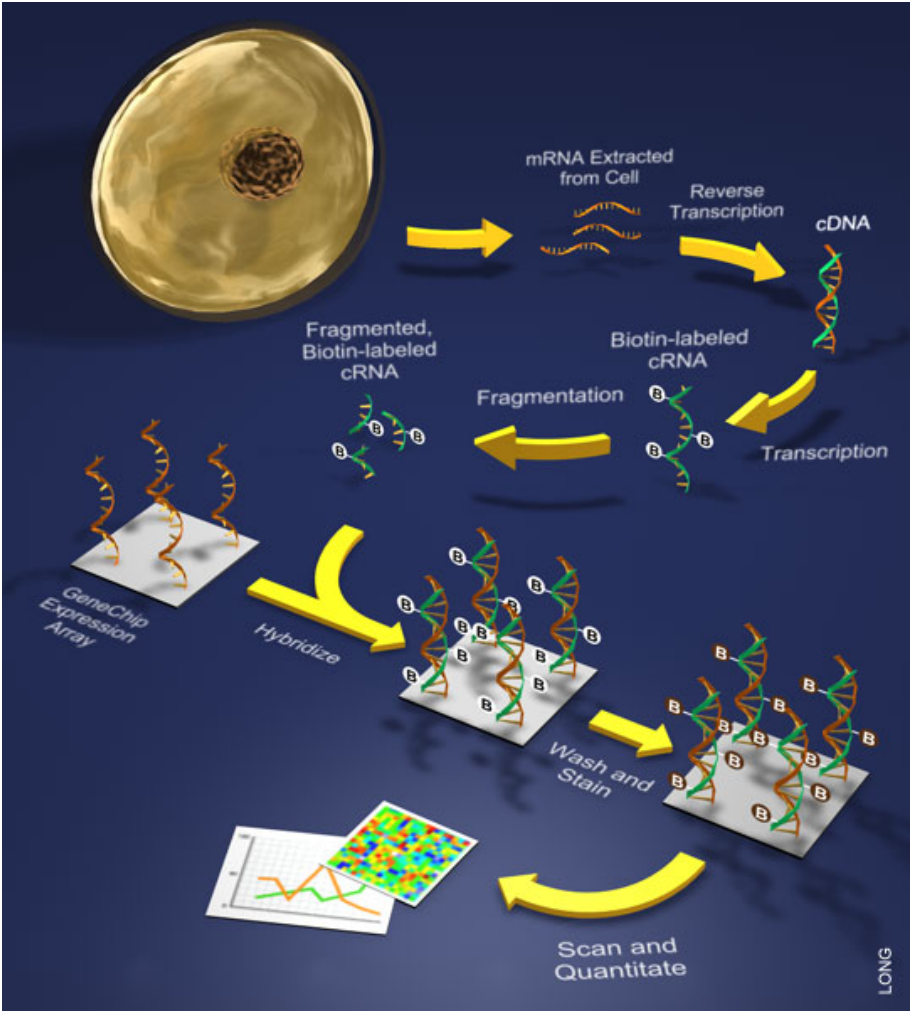


Figure A.6 Single-color microarray chip.

A.2.4 Epigenetics

Epigenetics is a rather novel field of investigation in the world of genomics. It tends to measure the conformation of the DNA molecules at a given time, under a given condition (Qiu, 2006). Didn't we say that DNA was inert and wasn't changing through time? Yes, the sequence is not modified, whatever the considered condition (except rare mutations). But the packaging of DNA molecules into chromosomes may vary. The way histones organize does change through time, in order to make some parts of the DNA readable. By getting information on the conformation at a given time, one also gets information of which part of the genetic code is available for expression. The hidden parts are not accessible for RNA polymerase. Another phenomenon, called DNA methylation, also plays a role in the availability of portions of DNA. Epigenetics can be used in parallel with DNA sequencing and gene expression to provide global pictures of the genomic processes. Epigenetics mechanisms are illustrated in Figure A.7.

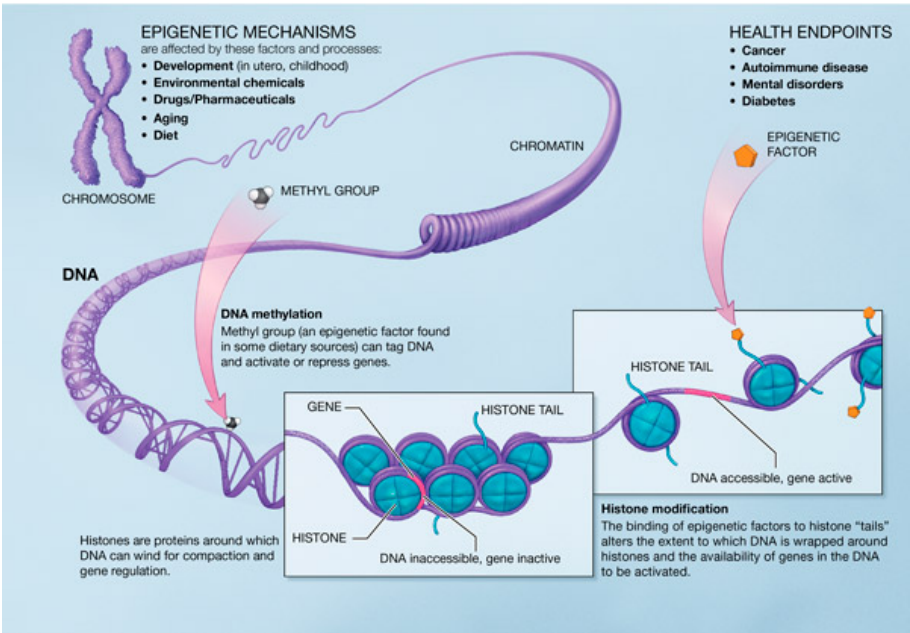


Figure A.7 Epigenetics mechanisms

A.3 Perspectives

Next-Gen Sequencing

Developments in the field of molecular biology technology led recently to next-generation sequencing (NGS) machines, such as the Illumina HiSeq2000. Such an equipment enables researchers to obtain the full sequence of two human genomes in a single run for less than 10,000 dollars per sample. The duration of a run is only around four days. And the costs and duration steadily lower. On top of that, next-generation sequencers are also capable of sequencing the whole transcriptome of an organism. This is a major advantage over probe-based expression microarrays. On microarrays, only the RNA corresponding to pre-defined probes can be measured. With these new tools, every piece of transcribed RNA can be measured. For a single transcriptome, ultra-high sequencing is even cheaper than microarray technology (less than 200 dollars per transcriptome).

Applications

Next-generation sequencing opens many possibilities for researcher: studying alternate splicing, discovering new SNPs, micro RNA profiling, de novo assemblies of bacterial and lower eukaryotic genomes, cataloguing the transcriptomes of cells, tissues and organisms, genome-wide profiling of epigenetic marks and chromatin structure in combination with other methods, and species classification and/or gene discovery by metagenomics studies ([Metzker, 2010](#)).

Challenges

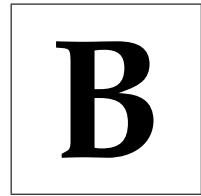
Nevertheless, these promising developments also raise many challenges. First of all, those sequencers produce in fact many short-length sequences (*reads*). Those sequences are overlapping. The average number of time that a particular position of the genome is sequenced is called the *coverage*. A typical value for the coverage is nowadays about 30. Once produced, all the reads have to be aligned to form the full genome. Alignment algorithms exist but typically have a complexity of $\mathcal{O}(l^2)$ where l is the length of a read. This only holds for pairwise alignments. There is so far no algorithm for optimal alignment in polynomial complexity ([Jones and Pevzner, 2004](#)). There is thus room for research in that field. Second, next-generation sequencers output more than 200Gb data per run. If the duration of a run is four days, it means that run-

ning once a week, a single machine produces 10.4Tb a year. The cost of storage devices is also steadily dropping, but the time required to analyze such an amount of data could be huge. As a comparison, microarray typical datasets can be stored on some tens of Mb, and still their analysis can take several days with the techniques described in this thesis. It is possible that the costs associated with downstream data handling and analysis could match or surpass the data-production costs (Metzker, 2010).

Ethics & Legislation

Beyond science and engineering challenges, genomic technologies raise many legal and ethics issues. The commercial applications that are currently offered based on SNPs and CNVs are questionable. First, they provide probabilities rather than real diagnosis. It is not because an individual has a snip showing that he is 70% more likely than others to develop some kind of cancer that he will indeed suffer from cancer. Second, such analyses output predictions about pathologies that are not curable yet. For example, Alzheimer's disease. How a patient without full understanding of probabilities and maybe a bit of biology or medicine can cope with those issues? Another issue is the Gattaca scenario (Niccol, 1997). In this sci-fi movie, every individual undergo a sequencing of his genome at birth. This data is made public and is used by others for example to give or refuse a job, to get informed before dating with someone, to define the price of an insurance policy, etc. Of course, in that movie, babies can be genetically engineered to correct some defects. With commercial services like 23AndMe and technologies such as next-generation sequencing, a Gattaca scenario becomes possible. Legislations may have an influence on that matter. And different countries in the world are taking opposite directions. In the USA, the guideline is that your own genetic material belongs to you and you should have access to the information it contains. European people are much more protected. For example in Belgium, an individual can only have access to its genetic data through an accredited physician. The american companies like Navigenics, 23AndMe or DecodeMe could be outlawed if they settle in Belgium, because they deliver genetic information just via the web. And even when this data is given to a patient via an accredited physician, it is forbidden for this patient to redistribute his own information to someone else. This is to prevent that a potential employer or an insurance company ask for genetic data. Society will face much more similar issues in the coming years.

Frequently Asked Questions (for Biologists)



This short appendix intends to outline the potential benefits of statistical learning and life sciences interactions, in a less mathematical phrasing than in the core chapters of this thesis. It is organized in the form of several questions.

I'm in Life Sciences. How Can I Benefit From Machine Learning?

The previous chapters are all examples of some possible added value provided by machine learning. We can summarize those here, and add some. Machine learning makes possible the identification of biomarkers in a rigorous and systematic way, with the extra advantage of offering tools to evaluate their quality. Biomarkers are helpful to guide further research on the mechanisms of some biological process. If a series of biomarkers is furthermore associated to a predictive model, it can lead to computer assisted diagnosis, or prognosis. Machine learning also makes possible the analysis of confounding factors, i.e. markers first believed as being relevant for a given biological process, but in fact related to another process. Automated diagnosis/prognosis is also useful for survival analysis, as it splits a cohort of patients in two or more groups. A survival curve can be plotted for each predicted group, and those curve can be compared.

The previous paragraph lists whole applications targeted by machine learning applied on biological data. But this field of research can do more, by easing the biologist everyday work or by proposing new ways of working. First, we showed in this thesis that expert prior knowledge not only could be incorporated in the building of our models, but that this way of proceeding

generally leads to an increase in model quality (see Chapter 8). This trait is really great, because it eases the interactions between biologists and engineers and makes their work more efficient as well. Some models provided by machine learning deliver a decision score when they make a prediction, not only a hard, categorical decision. This feature can help a physician to get more or less confident in its final decision. In clinical studies, we are given labels for each patient involved (for example, healthy vs. diseased). Those labels are generally not questioned. If a predictive model outputs very high classification performances, but a few patients always get misclassified, whatever the subset of the patients that is taken to build the model, it might lead to question this label. Are clinician sure of their decision? Were those patients difficult to label? Is it possible that something went wrong in the clinical process? It might be worth checking. If everything seems ok after check, then it might be that those patients show a rare or particular trait that may be of interest. Finally, machine learning, and statistics in general, offer plenty of tools for data visualization, and convenient presentation of results, as illustrated for example in Chapter 12.

Do I Need an Engineering Degree?

Should biologists read this thesis and other machine learning books and try to apply all the techniques themselves? Not necessarily. But is it possible to make a pure life scientist and a pure applied scientist collaborate efficiently? We stressed in the previous paragraph that it is desirable. We now claim that it is possible, and that is not needed that people working on the questions touched by this thesis have a full double education in biology and engineering. Of course, it makes collaboration easier if people from machine learning understand the vocabulary of biology (they might start reading Appendix A). What matters overall is that both sides be open-minded. It is important that people from machine learning understand the issues of the biologists, and that biologists understand some basic concepts of machine learning. It is however not required that life scientists get to dig into equations and implementations.

Will I Be Replaced by a Computer Someday?

Physicists and biologists are really not likely to be replaced by a computer scientist, let alone a computer. Computers are tools that can be useful for life sciences, just like telescopes are tools used by astronomers. However a telescope is not sufficient to do all the job, it is just a tool. But astronomers

would make far less discoveries without those powerful tools. The same reasoning applies for machine learning techniques used in biology. As we stated in Chapter 3, the ultimate quality criterion for biomarker identification is the biologist interpretation.

Will it (Always) Work?

Are there guarantees that a given machine learning technique will always work, whatever the data? Are there guarantees that on a given problem, there will be at least one machine learning technique that will work? There are no such guarantees. There is even a theorem proving that in order to learn something, a machine learning technique will fail on some data, on some task. But this is the price to pay to achieve good results on potentially many datasets and many tasks. This is the *No Free Lunch Theorem* (Wolpert, 1996).

Nevertheless, there is sufficient experience sustaining that machine learning tools are of primary interest in real-life applications. Outside the field of biology, examples are numerous. In everyday life, think about email spam detection, smile-detecting cameras, optical character recognition in scanners, etc. The DARPA project in the United States held a competition¹ for automated driving for several years (2004, 2005, 2007). Several vehicles showing good performances were partially based on machine learning tools. In the field of biology, and biotech industry, plenty of examples are available as well. The R2 ImageChecker² is a CAD (Computer Assisted Diagnosis) machine, helping radiologists read analog and digital mammographs. It identifies regions-of-interest on mammography images and brings them to the attention of the radiologist in order to decrease false negative readings. This algorithm is trained on thousands of biopsy proven cancers. It is FDA-approved, and has been on the market for 15 years. It is commercialized by the American company Hologic. HistoScanning³ is an ultrasound-based application using machine learning techniques to identify and visualise the position and extent of tissue suspected of being malignant. Histoscanning is commercialized by the Belgian company Advanced Medical Diagnostics. Finally, in the field of genomics, success stories can be identified as well. It is the case of the NKI study on breast cancer clinical outcome prediction based on microarray data (van 't Veer *et al.*, 2002). This study led to the creation

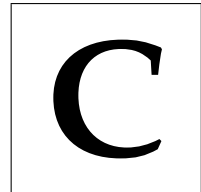
¹<http://www.darpa.mil/grandchallenge/index.asp>

²<http://www.hologic.com/en/breast-screening/imagechecker/>

³<http://www.histoscanning.com/>

of the MammaPrint diagnosis kit. It is used to assess the risk that a breast tumor will generate metastasis. MammaPrint is marketed by Agendia and is based on the NKI 70-gene breast cancer gene signature. In 2007, the FDA cleared the MammaPrint test for use in the United States. Another, more recent success, is the identification of a 33-genes signature predictive of the response to a candidate treatment against melanoma ([Dupont *et al.*, 2010](#)). This signature has also been identified on gene expression microarray data, based on a consensus feature selection method in the spirit of Chapter 7. Combined with a linear support vector machine (see Chapter 2), it predicts if a patient suffering from melanoma is likely to respond positively or not at all to the candidate treatment, before the patient receives the treatment.

Affymetrix Genechip Microarrays



This appendix covers in more details a particularly widespread microarray technology: the Affymetrix Genechip. Most of the datasets considered in this thesis have been produced with this technology. Section [C.1](#) explains how Genechips measure the expression of tens of thousands of genes. Section [C.2](#) introduces basics of how Affymetrix provides a single expression value for each probeset based on all the measurements provided by millions of probes. This is the summarization process, which is also in charge of signal normalization and denoising.

C.1 Measuring Expression

First, though microarray do indeed look as tiny electronic chips, there is nothing electronic about them. Microarray measure gene expression of several (tens of) thousands of genes in a single experiment. Gene expression is the rate of copy of a given gene sequence into RNA. The widely used single-fluorescence Affymetrix Genechip Human Genome U133 Plus 2.0 is pictured in Figure [C.1](#) (top-left). Affymetrix microarrays read mRNA transcripts as well as other functional RNA's, like miRNA's, depending on the particular chip type. A few years ago, only a few thousands probesets were available on each chip (Hu6000 series $\approx$ 6800 probesets). This number was raised to $\approx$ 12,600 with the Human Genome U95 A series (HGU95A). The HGU133Plus2.0 is one of the largest gene expression microarray chip with 54,675 probesets. The location of the probes on the gene sequence also varies from series to series. In the latest Affymetrix products, the Exon series, the probes are spread over a large number of exons, rather than concentrated at one sequence end. Probesets also contains more probes than previously.

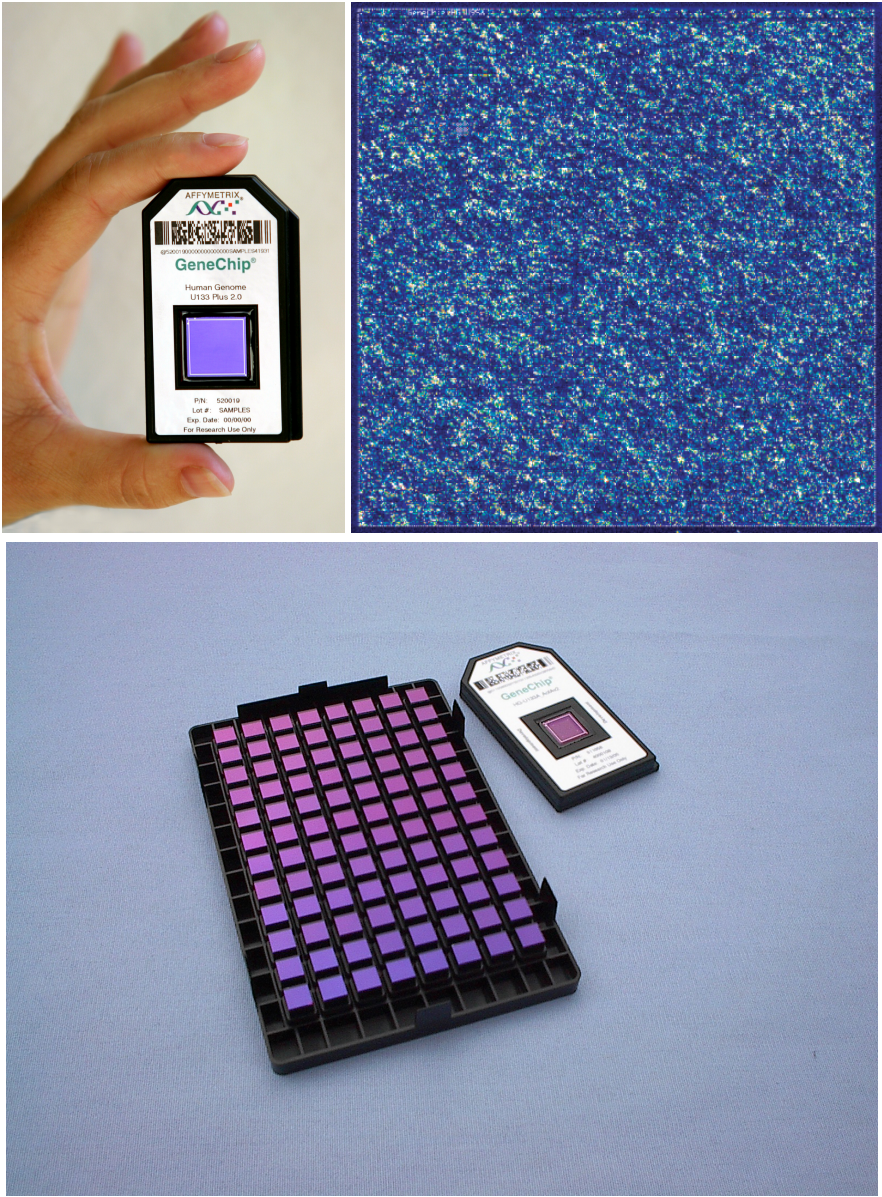


Figure C.1 Top-left: Human Genome U133 Plus 2.0 Array. Top-right: Data from an experiment showing the expression of thousands of genes on a single GeneChip probe array. Bottom: The Affymetrix high throughput microarray contains 96 individual arrays mounted onto a single plate. Each of the 96 arrays on the plate contain the same content as a cartridge-based GeneChip Human Genome U133A Array. Images courtesy of Affymetrix.

Design

On Genechip arrays, each gene is represented by a probeset on the glass array. A probeset is a set of 11 to 20 probes, depending on the precise chip type. As illustrated on Figures C.2, a probe is a single stranded 25 DNA nucleotides chain corresponding to a known sub-part of the genetic code of a gene transcript. Those sub-parts are spread over the gene sequence. They are chosen for their high specificity. The location of each probe on the array is known. If a single copy of each probe was fixed on the array, this technology would only be able to detect binary signals: presence or absence of the given transcripts. That is why each probe is in fact repeated a few million times.

Hybridization

The biological material to measure is tagged with a fluorescent label and washed over the array. This material bonds with the corresponding sequences, which is called the hybridization process, illustrated in Figure C.3. This process takes a few hours.

Scanning

The chip is scanned. Each probe shows its own luminous intensity level depending on the number of its copies having been hybridized. Probes with many hybridized copies of the tagged material will look bright. Probes which corresponding sequence was not present in the analyzed material will not shine when scanned. This step is illustrated in Figure C.4. The result of a single microarray experiment is an image with a given intensity for each probe, as pictured in Figure C.1 (top-right).

Perfect Match and MisMatch

It happens that some probes hybridize with other sequences than their target sequence. This is called non-specific binding (NSB). The result is a measured intensity larger than the expected one. To measure NSB, each probe is doubled by another probe differing only by the middle nucleotide (the 13th nucleotide). This modified probe is called a mismatch probe (*MM*), while the standard probe is the perfect match probe (*PM*). Ideally, no hybridization should occur on a mismatch probe. Other sets of probes are fixed on the arrays in order to assess the quality of each step of the experiments.

The intensities of all probes within a probeset have to be aggregated

into one intensity value by probeset. The NSB has also to be measured and removed from the signal of interest. Finally, all chips within an experiment campaign have to be comparable, and thus normalized. This operation is known as the summarization process, described in the next section.

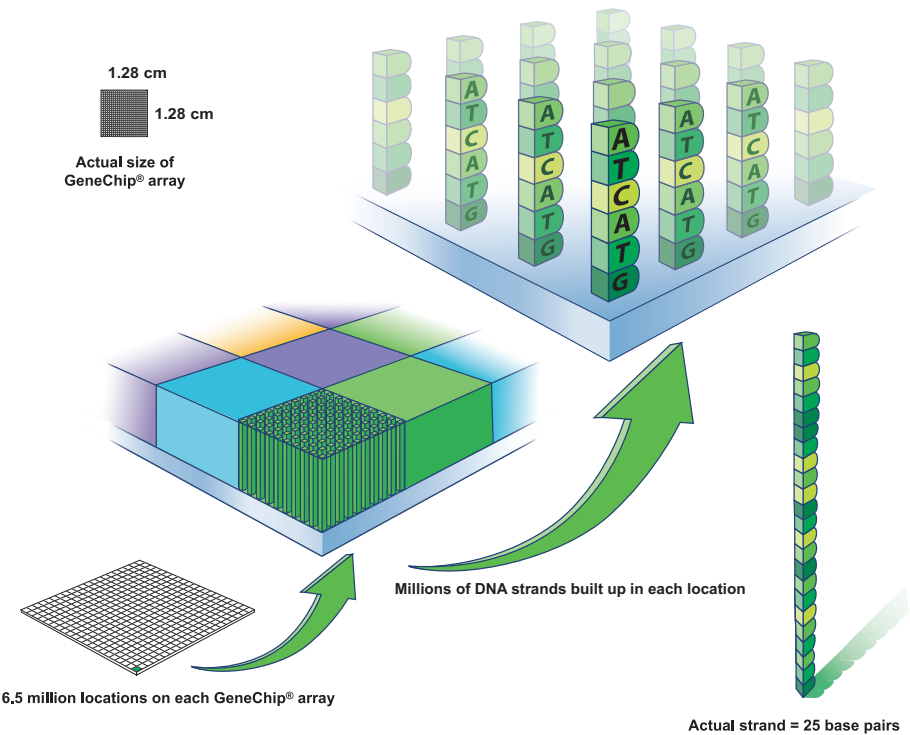


Figure C.2 A single feature on an Affymetrix GeneChip microarray. Image courtesy of Affymetrix.

C.2 Summarization

Summarization is the name given to the preprocessing of Affymetrix Genechips. This name is slightly abusive, since several steps are hidden behind summarization. Only the last step is effectively a summarizing one:

1. **Background adjustment:** Correction for NSB and optical noise introduced by the scanner.

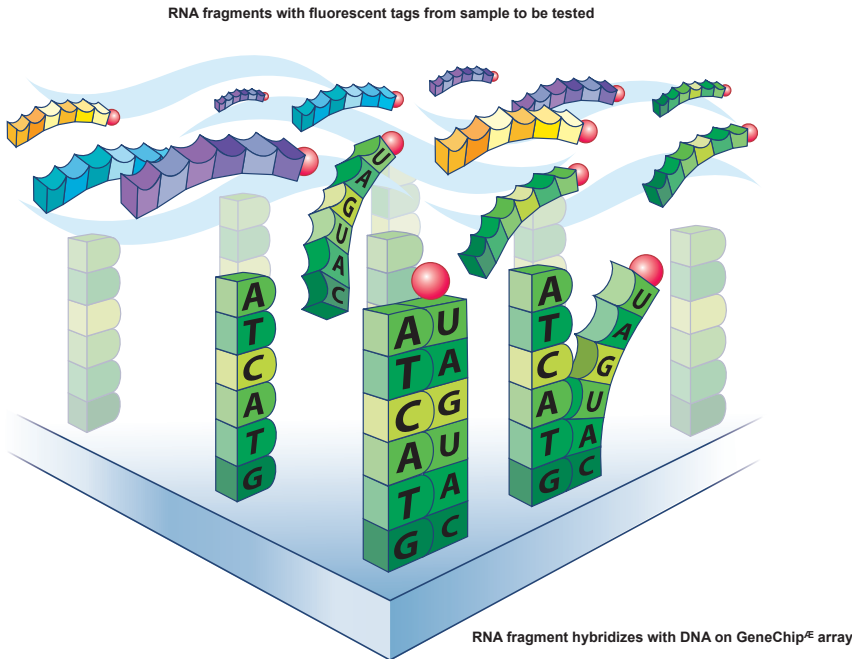


Figure C.3 Hybridization of tagged probes to Affymetrix GeneChip microarray. Image courtesy of Affymetrix.

2. **Normalization:** Correction for global intensity levels across several chips.
3. **Summarization:** Production of a unique expression value for each probeset based on the background-adjusted and normalized probe expression values.

This section covers three widely used summarization protocols: MAS, RMA and GC-RMA. In each section, the three steps of the process are described. The outline of a fourth, original, summarization technique is also proposed in Section C.2.4. The precise details of each method are not always publicly available, or clearly exposed. The following discussions about some sub-parts of the processes are sometimes conceptual.

C.2.1 MAS

MAS either stands for : **M**icro**A**rray **S**uite or **M**icroarray **A**nalysis **S**oftware. MAS has been proposed by Affymetrix collaborators. The current version is

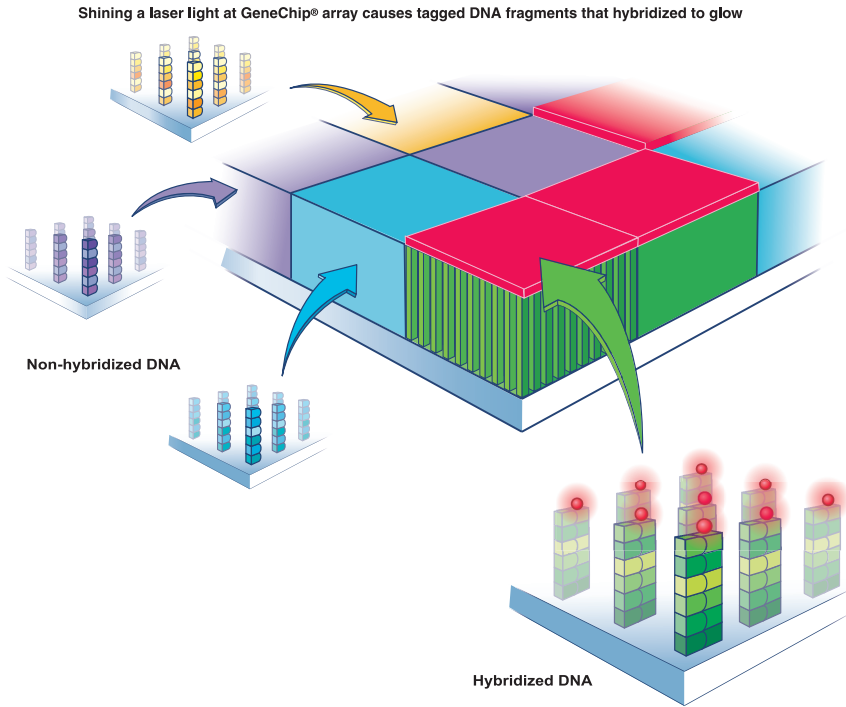


Figure C.4 Scanning of tagged and un-tagged probes on an Affymetrix GeneChip microarray. Image courtesy of Affymetrix.

MAS5.0 (Hubbell *et al.*, 2002).

Background

The background adjustment is mainly based on the *MM* and *PM* intensities. The basic solution offered by previous versions of MAS have been shown to be largely deficient. Let us define the corrected expression value of each probe j for a chip i as being PM'_{ij} . The initial solution to compute such a corrected value was:

$$PM'_{ij} \leftarrow PM_{ij} - MM_{ij}. \quad (\text{C.1})$$

The basic solution provided by Equation (C.1) for background adjustment has been strongly criticized because it leads to large variance and negative values (Li and Wong, 2001). A negative intensity is nonsense. This phenomenon seems to be due to the fact that *MM* probes also capture part of specific hybridization. Since MAS5.0 (the current version), this system has been cor-

rected and is now defined as:

$$PM'_{ij} \leftarrow \begin{cases} PM_{ij} - MM_{ij} & \text{if } MM_{ij} > PM_{ij}, \\ 0 & \text{otherwise.} \end{cases} \quad (\text{C.2})$$

This patch is not satisfying because it looks completely arbitrary. Moreover, the number of probes for which $MM_{ij} \leq PM_{ij}$ is not marginal at all. One third of the probes undergo this issue.

Normalization

Once probe intensities have been corrected for background noise, all the considered chips are normalized. This is done by subtracting a value such that the mean intensity of each chip is a constant.

Summarization

The summarization process provided by MAS is described as a robust median estimate of all probes within a probeset.

C.2.2 RMA: Robust Multi-array Average

RMA is a method developed by the academic community ([Irizarry et al., 2003](#)) but is now included in the software suite delivered with Affymetrix platforms.

Background

The main contribution of the RMA is the original background correction step. Authors of RMA consider the MM values provided by the arrays unreliable because they also detects specific binding, i.e. signal of interest. RMA offers an alternative which ignores the MM . It fits a model based on an exponential signal with gaussian noise, based on the PM intensities only. This improves the precision of summarization at the price of a small loss in accuracy with respect to MAS5.0.

Normalization

The RMA normalization is based on quantile normalization ([Bolstad et al., 2003](#)). It aims at making probe intensities distribution the same across a set of arrays. Suppose an experimental campaign with n microarray chips each having p probes. The resulting expression $n \times p$ matrix is denoted $\mathbf{X}$. The quantile normalization works as follows:

1. Sort each row of $\mathbf{X}$ to form $\mathbf{X}_s$.
2. Get $\mathbf{X}'_s$ by replacing each element $x_{i,j}$ of $\mathbf{X}_s$ by its column mean.
3. Get $\mathbf{X}_{norm}$ by rearranging each row of $\mathbf{X}'_s$ according to the initial ordering of $\mathbf{X}$.

Summarization

The RMA summarization is generally described as being a median polish estimate (Tukey, 1977). It is however unclear how this method is applied on microarray data since it provides a particular form of median for matrices, but a given probeset does not have an identical expression value across all chips after the summarization process.

C.2.3 GC-RMA: GC-Affinities for RMA

GC-RMA is an extension to RMA, proposing a different approach for the first preprocessing step, the background correction step (Wu *et al.*, 2004). The two other steps are identical to RMA.

Background

GC-RMA uses a model to estimate the true *MM* values instead of using the unreliable *MM* measurements from the arrays as in MAS5.0, or simply ignoring them as in RMA. This model takes into account the *affinity* of the probes. The affinity depends on the oligonucleotide sequence composition of each probe. Guanine (G) and Cytosine (C), two of the four possible nucleotides composing DNA, always bind together across two complementary strands. The two others, Adenine (A) and Thymine (T) always bind together. The difference between G-C and A-T linkages is in the number of hydrogen bonds they form, as illustrated in Figure A.2. Three bonds are formed between G-C linkages, whereas only two are formed for A-T linkages (Hunter, 2009). The probes containing more G and C nucleotides have a higher probability to hybridize than the ones containing more A and T nucleotides. Affinities predict NSB quite well. One advantage of the affinities over the *MM* is that they will not mix NSB with signal of interest, because they are precomputed numbers (Irizarry *et al.*, 2003). On top of the NSB correction, GC-RMA also includes a step for scanner optical noise correction.

Normalization and summarization steps are identical to those of RMA.

C.2.4 LASSUM: Group-Lasso-Based Summarization

All the above methods for background correction, normalization and summarization are more or less arbitrary. MAS has been shown to be a poor summarization technique, especially with respect to background correction. We outline here a new data-driven and regularized method for the last step, summarization itself. We suppose that background correction and normalization steps have already been performed, either through RMA or GC-RMA.

In this approach, we embed the summarization process within the classification and selection problems. We define an optimization problem for solving summarization, feature selection and classification at the same time. We thus have a p' -dimensional problem where p' is the number of probes, and p is still the number of probesets as it is the case in the rest of the thesis.

The technique we propose relies on the group-lasso, defined in Equation (3.7). We use the fact that probesets naturally define groups of probes. Depending on the precise technology, each probeset groups 10 to 20 probes. Those groups are the groups used in our summarization technique by the group-lasso. We thus restate the group-lasso objective applied to the p' dimensions of the problem here:

$$\min_{\mathbf{w}} \sum_{i=1}^n (y - \langle \mathbf{x}_i, \mathbf{w} \rangle)^2 + \lambda \|\mathbf{w}\|_{1,2}, \quad (\text{C.3})$$

with the $\|\mathbf{w}\|_{1,2}$ extensively rewritten as:

$$\|\mathbf{w}\|_{1,2} = \sum_{i=1}^p \sqrt{\sum_{j \in \mathcal{G}_i} w_j^2}. \quad (\text{C.4})$$

It corresponds to the sum for each probeset $\mathcal{G}_i \forall i \in \{1, \dots, p\}$ of the 2-norm over the weights of the probes j belonging to this probeset $\mathcal{G}_i$. Note that $\mathbf{w}$ is a p' -dimensional vector. The group-lasso defined this way favors the common selection or elimination of all the probes within a probeset. It allows however each probe to contribute differently to the model. It thus achieve a selection at the probeset level while performing a data-driven summarization based on probe level intensities.

The same idea could be implemented via a sparse SVM for grouped variables (Zou and Yuan, 2008) instead of a group-lasso. This procedure

could lead to even more overfitting than in the regular case since it multiplies the number of features by ten to twenty. Two arguments are however in favor of such an approach:

1. **Complexity:** The number of feature is already about hundred times higher than the number of samples. It is probable that beyond that point, adding more features would not make the problem much more difficult.
2. **Biology:** The genetic sequences corresponding to the probes of a probe-set are spread all over a gene sequence. RNA in unprotected conditions, namely outside a cell, is rapidly degraded by enzymes called Ribonucleases (RNAses for short). These enzymes are in charge of cell cleaning (removing RNA no longer required) and defense (first defense mechanism against RNA viruses). The two ends of a gene sequence do not have an equal physical robustness to RNAses. The two ends of RNA are called the 3' end and the 5' end. The 3' end is much more rapidly degraded than the 5'. Probes from different regions of the genes could then receive a different importance in the global gene expression computation.

The effective impact of this proposed LASSUM summarization approach should be assessed experimentally.

C.3 Discussion

Affymetrix Genechip microarrays have been massively used. Data corresponding to more than 400,000 hybridized chips are freely available on the Gene Expression Omnibus database. A vast majority of these data have been produced on Genechips. Recent developments show the effort done by Affymetrix to lower the cost of its technology as well as the time and complexity required to perform gene expression experiments. For example, a Genechip platform now exists that deals with 96 HGU133A in a single experiment. This device is pictured in Figure C.1 (bottom). It is much more efficient in terms of cost and time than its predecessor.

However, Affymetrix is likely to be supplanted in the next years by ultra-high throughput sequencing tools. These next-generation sequencing are not only able to perform sequencing of DNA, but are also able to sequence a whole transcriptome. This means that in addition of being more time and

cost efficient than traditional microarray technology, they are also more comprehensive: what is measured does not depend anymore on the definition of a finite set of probes. Everything that is expressed is measured. Nevertheless, microarrays will still be used for several years given the large number of platform that have been bought.

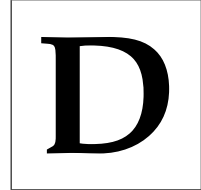
Regarding summarization, there is no consensus about which solution is the best. An extra level of complexity is introduced in this discussion since the Bioconductor community, dedicated to bioinformatics open source software packages in R language, distributes packages allowing combinations of the different steps from each method. One possibility to study the quality of summarization methods is to apply them on datasets where the expected probeset expressions is known in advance. Few such datasets exist: the *Latin Square* dataset at Affymetrix, and the *Dilution* dataset. However, they have been criticized because they were artificially hybridized with known quantities of RNA. They do not reflect the natural hybridization process of known RNA quantities.

Several questions remain open:

- Documentation on summarization technique is rare or vague. It is thus difficult to get a precise idea of precise protocols definition.
- How to evaluate separately the three steps of summarization?
- How to evaluate them jointly?
- How to evaluate the effect of combining steps from several methods together?
- Is the LASSUM summarization technique competitive?
- How to avoid a summarization bias?

The last question deserves a few comments. This thesis discusses the normalization bias and how to avoid it (see Chapter 6). Within the second step of summarization, a sample normalization occurs. If the summarization is done on the whole dataset (both training and test samples), a *summarization bias* is likely to occur. Parameters of the three summarization steps should be computed only on the training samples and then applied on the test samples. Note that most published datasets come already summarized. In such case, the summarization step cannot be tuned, nor studied.

Datasets



This section provides a short description for every dataset used in this thesis. Through the text, the datasets are named with code of three letters (generally referring to the tissue type or pathology considered) and eventually a number. Those codes are displayed in the margin. In addition, Table D.1 provides a summary of their characteristics.

COL

Published by Alon *et al.* (1999). The Colon Cancer dataset is made of samples from 40 tumor and 22 normal colon tissues probed by an Affymetrix microarray chip measuring more than 6,500 genes. The data set was published after a pre-filtering step and the resulting samples include only 2,000 genes. The task described in the original study is to determine if groups of patients could automatically be constructed by a clustering algorithm. Clustering separated cancerous from non-cancerous tissue and cell lines from in vivo tissues on the basis of subtle distributed patterns of genes even when expression of individual genes varied only slightly between the tissues. Since annotations of the samples are available, we address in this work the binary classification problem of predicting whether a sample corresponds to a tumor versus a normal colon tissue. Data available at <http://microarray.princeton.edu/oncology/affydata/index.html>

DLB

Published by Shipp *et al.* (2002). The classification task in DLBCL (standing for diffuse large B-cells) is the prediction of the tissue types. The two classes of this dataset are DLBCL tissues (57 patients) and Follicular Lymphomas (20

patients). It counts 7,129 features and consists of Affymetrix Hu6800 chips. In the original publication, authors applied a supervised learning prediction method to identify cured versus fatal or refractory disease. The algorithm classified two categories of patients with very different five-year overall survival rates. Data available at <http://www.broadinstitute.org/mpr/lymphoma/>.

LEU

Published by Golub *et al.* (1999). The Leukemia dataset was produced in a study aimed at building a model to discriminate between Acute Myeloid Leukemia (AML) and Acute Lymphoblastic Leukemia (ALL) tissues. A set of 50 genes was selected in a univariate way according to a correlation criterion with the class label. Those genes were combined afterwards to build a simple weighted vote classifier. The model was tested on an independent test set of 34 Leukemia samples, making correct predictions for 29 among them. Authors mention the difficulty of choosing the right set of informative genes, given that lots of them were highly correlated with the ALL-AML distinction, and stress the need of a stable selection approach. The type of chip used in this study is not fully specified, but the manufacturer is clearly identified as being Affymetrix. Given the year of publication and the number of considered probesets (7,129), we guess that the chips have to belong to the Hu6xxx series. Data available in R package golubEsets on <http://www.bioconductor.org>.

LYM

Published by Alizadeh *et al.* (2000). The LYM dataset comes from a study on Diffuse Large B-Cell Lymphoma. The task is here to discriminate between two types of Lymphoma (Germinal Center B-Like and Activated B-Like) based on gene expression measured by microarray technology. The authors originally performed a clustering analysis based on similarity measure related to correlation. This dataset has expression measurements for 4,026 genes. It has some missing values, which were replaced by the mean of values obtained by using a k-NN with $k = 5$ neighbors in a preprocessing step. The data used is a subset of the one provided by (Alizadeh *et al.*, 2000), such as provided at <http://www.upo.es/eps/aguilardatasets.html>.

LYM1

Published by Alizadeh *et al.* (2000), as LYM. This dataset contains 62 gene expression data samples about the three most prevalent adult lymphoid ma-

lignancies: 42 samples of diffuse large B-cell lymphoma, 9 observations of follicular lymphoma, and 11 cases of chronic lymphocytic leukemia. 4,026 genes are measured that are preferentially expressed in lymphoid cells or with known immunological or oncological importance. Data available at <http://ligarto.org/rdiaz/Papers/rfVS/randomForestVarSel.html>.

PRO

Published by [Singh et al. \(2002\)](#). This dataset comes from a prostate cancer study and counts 50 normal tissue samples and 52 tumor tissue samples. One of the tasks addressed by the authors is to build a model able to discriminate between normal and tumor prostate tissue, based on microarray data. No precision is made about the type of tumor tissue (primary? relapse?). A univariate selection criterion (Golub's Signal-to-Noise Ratio) is applied on each gene and its significance is assessed by a permutation test. The most significant genes are selected and a predictive model is built on this signature with a k-nearest neighbor classifier. The number of genes found to be significantly differentially expressed in the two classes according to the Golub's ratio is higher than 400. This result shows the difficulty of identifying small and stable signatures. Each selected gene has been individually compared with clinical factors but no significant correlation was found. The dataset PRO contains 6,033 features, corresponding to those kept after a prefiltering step performed by the authors of the original publication. Genes whose expression varied less than 5-fold between any two samples in any given experiment were removed. Data available at <https://array.nci.nih.gov/caarray/project/golub-00142>.

PRO1

Published by [Singh et al. \(2002\)](#). The dataset PRO1 comes from the same study as dataset PRO. It contains gene expression data for the same patients. There are two differences between PRO and PRO1. The number of features is 12,625 for PRO1 because it is based on the whole set of original features available on an Affymetrix HGU95Av2 chip, not only the features kept by authors of the original study. The second difference is that PRO1 has been summarized with the RMA protocol. Data available at <http://www.genome.wi.mit.edu/MPR/prostate>.

PRO2

Published by [Chandran et al. \(2007\)](#). This dataset comes from a prostate cancer study and counts 18 normal tissue samples and 86 tumor tissue samples. Tumor samples are of two types: primary tumor and metastatic tumor. In the original publications, the task is to discriminate between tumor and normal tissues. PRO2 has been produced with Affymetrix HGU95Av2 chips, and has been summarized with the RMA protocol. It has 12,625 features. Data available on Gene Expression Omnibus (GEO) at <http://www.ncbi.nlm.nih.gov/geo/>, with accession number GSE6919.

PRO3

Published by [Welsh et al. \(2001\)](#). This dataset comes from a prostate cancer study and counts 9 normal tissue samples and 25 tumor tissue samples. Tumor samples in this dataset correspond to 24 primary tumors and 1 lymph node metastasis. In the original publications, the task is to discriminate between tumor and normal tissues. PRO3 has been produced with Affymetrix HGU95A chips, and has been summarized with the RMA protocol. It has 12,626 features. Data available at <http://public.gnf.org/cancer/prostate/>

Table D.1 Microarray Datasets. **Name** is the code given to each dataset throughout the text. **C_x** is the number of samples in class *x*. **# Features** is the number of features in the dataset. **Technology** is the type of chip used for experiments, when available. **Chapter** is the number of the chapter(s) where a dataset is used.

Name	C ₁	C ₂	C ₃	# Features	Technology	Chapter
COL	40	22	-	2,000	Affymetrix Hu6000	7, 8
DLBCL	57	20	-	7,129	Affymetrix Hu6800	8
LEU	47	27	-	7,129	Affymetrix Hu6xxx	7, 8, 12
LYM	22	23	-	4,026	2-Color cDNA chips	7
LYM1	42	9	11	4,026	2-Color cDNA chips	10
PRO	50	52	-	6,033	Affymetrix HGU95Av2	7, 8
PRO1	50	52	-	12,625	Affymetrix HGU95Av2	9, 10
PRO2	18	86	-	12,625	Affymetrix HGU95Av2	9, 10
PRO3	9	25	-	12,626	Affymetrix HGU95A	9, 10

Non-Specific Filtering



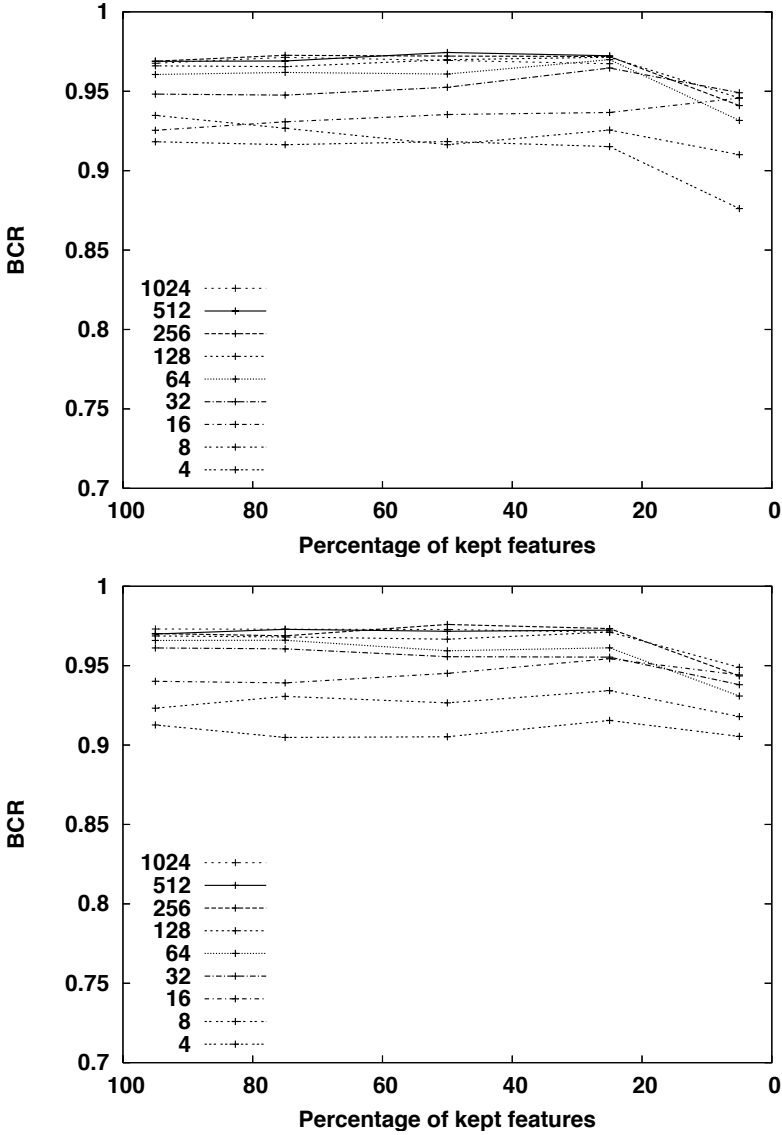


Figure E.1 Non-specific filtering on dataset LEU based on features minimal value. Top: t-Test used as a second selection method. Bottom: Golub index used as a second selection method

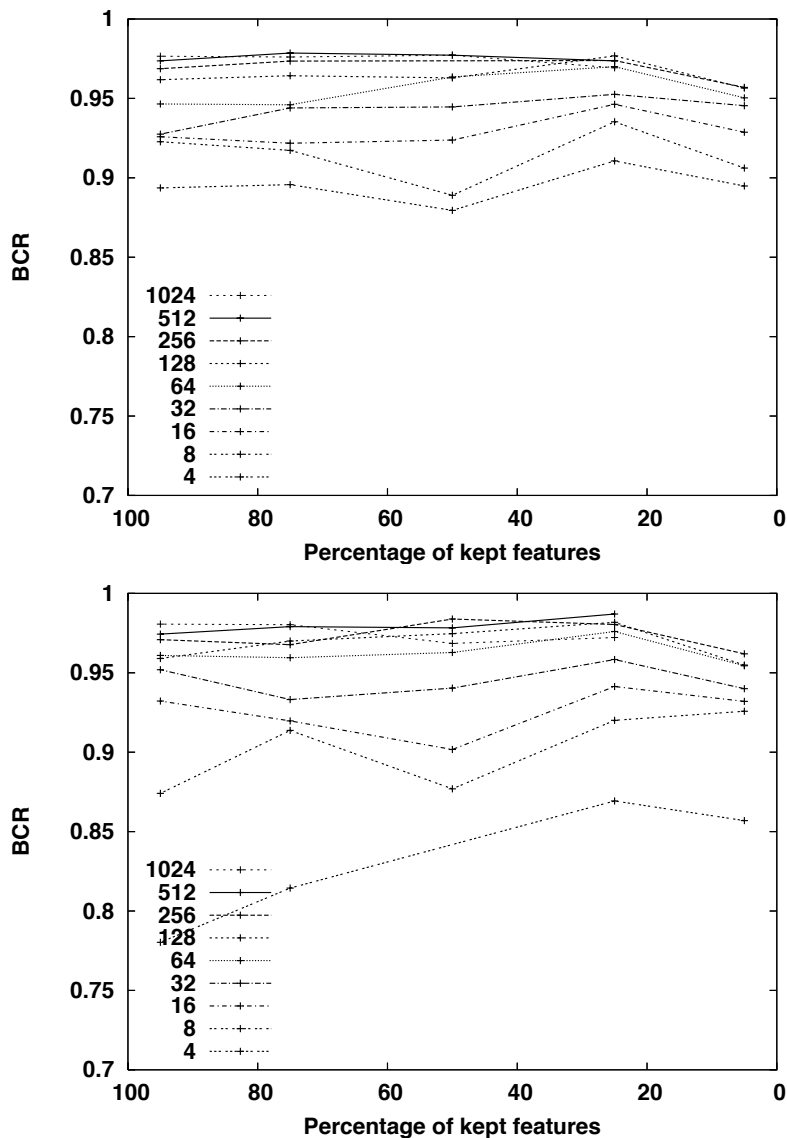


Figure E.2 Non-specific filtering on dataset LEU based on features minimal value. Top: RFE used as a second selection method. Bottom: L_2 -AROM used as a second selection method

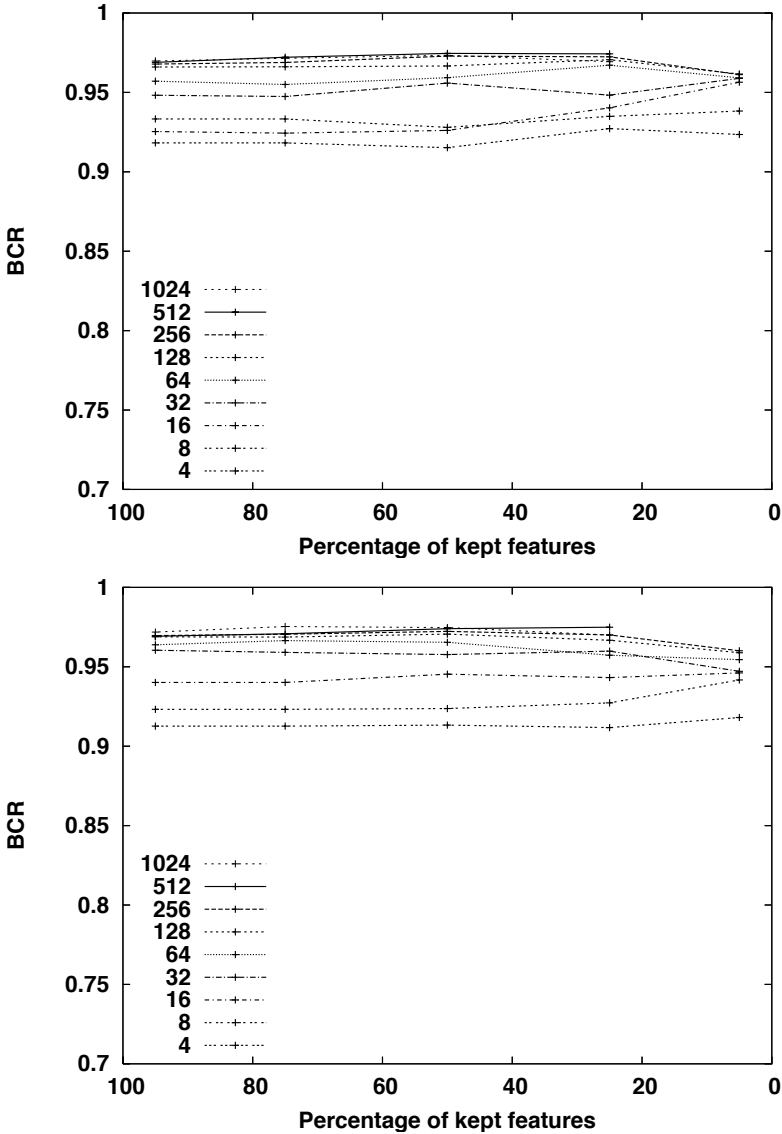


Figure E.3 Non-specific filtering on dataset LEU based on features variance. Top: t-Test used as a second selection method. Bottom: Golub index used as a second selection method

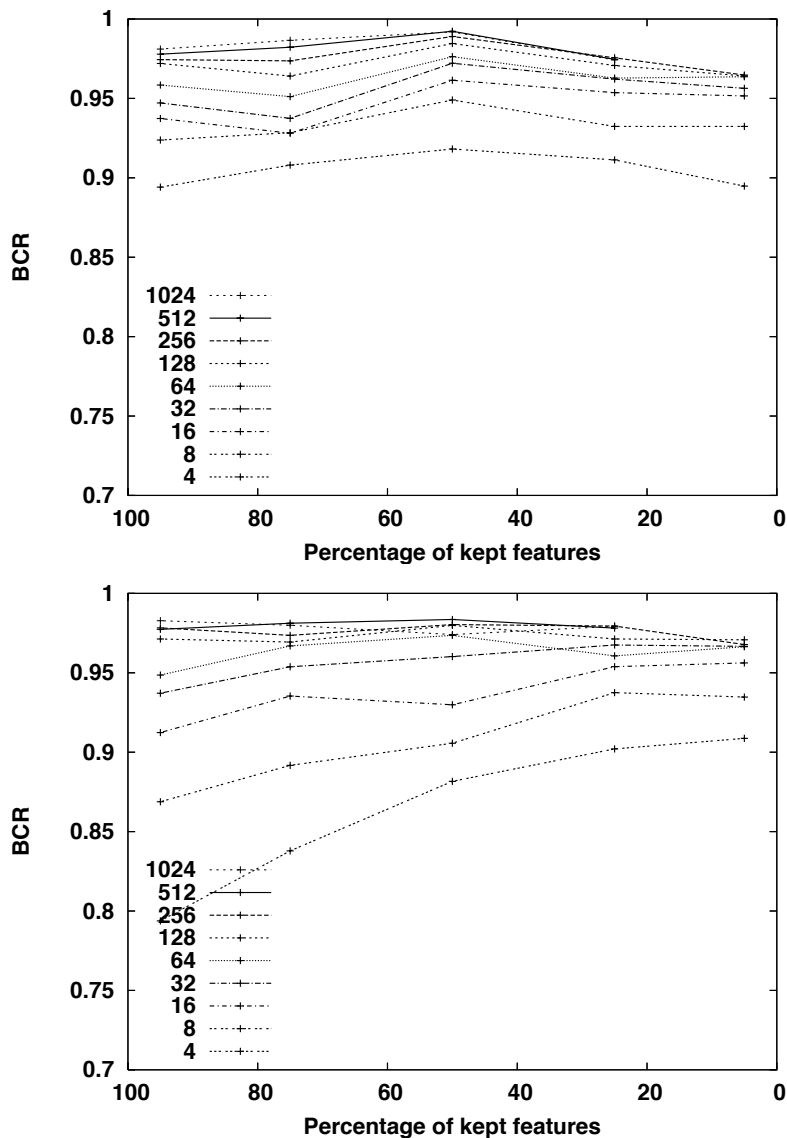


Figure E.4 Non-specific filtering on dataset LEU based on features variance. Top: RFE used as a second selection method. Bottom: L_2 -AROM used as a second selection method

Ensemble Selection

F

F.1 Ensemble Method Improves Selection Stability and Classification Performance

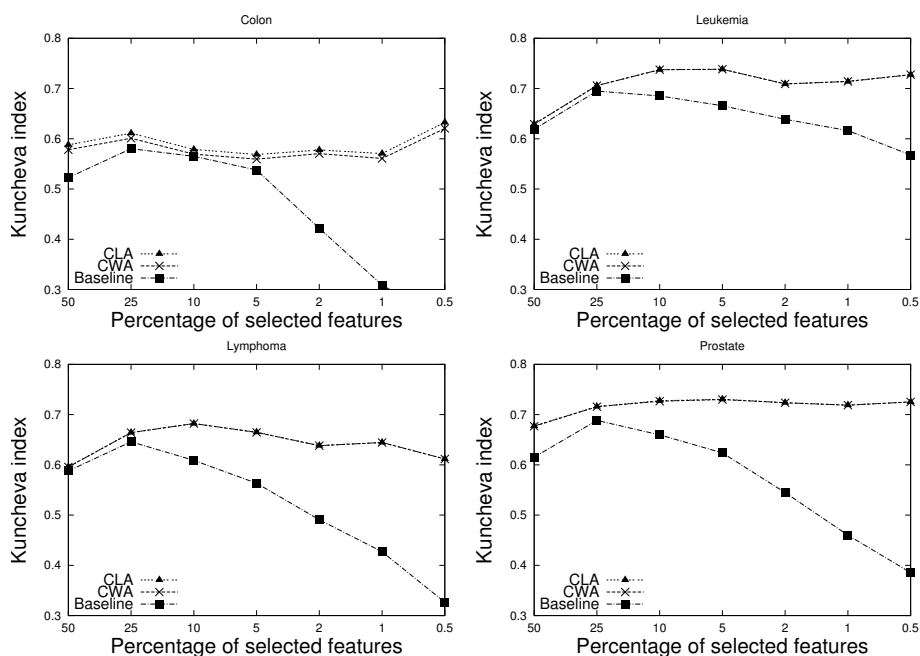


Figure F.1 Stability of the baseline method (RFE) and the ensemble methods for COL (up-left), LEU (up-right), LYM (bottom-left) and PRO (bottom-right) datasets. We used 40 bootstraps and eliminated $E = 20\%$ features at each iteration of RFE.

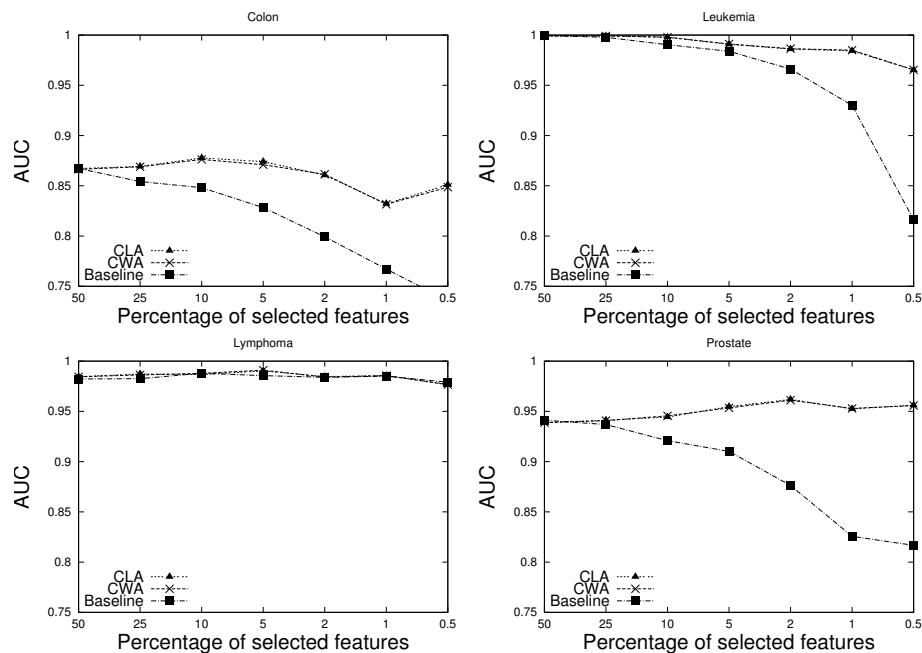


Figure F.2 Classification performance of the baseline method (RFE) and the ensemble methods for COL (up-left), LEU (up-right), LYM (bottom-left) and PRO (bottom-right) datasets. We used 40 bootstraps and eliminated $E = 20\%$ features at each iteration of RFE.

F.2 Effect of the Number of Bootstrap Rounds

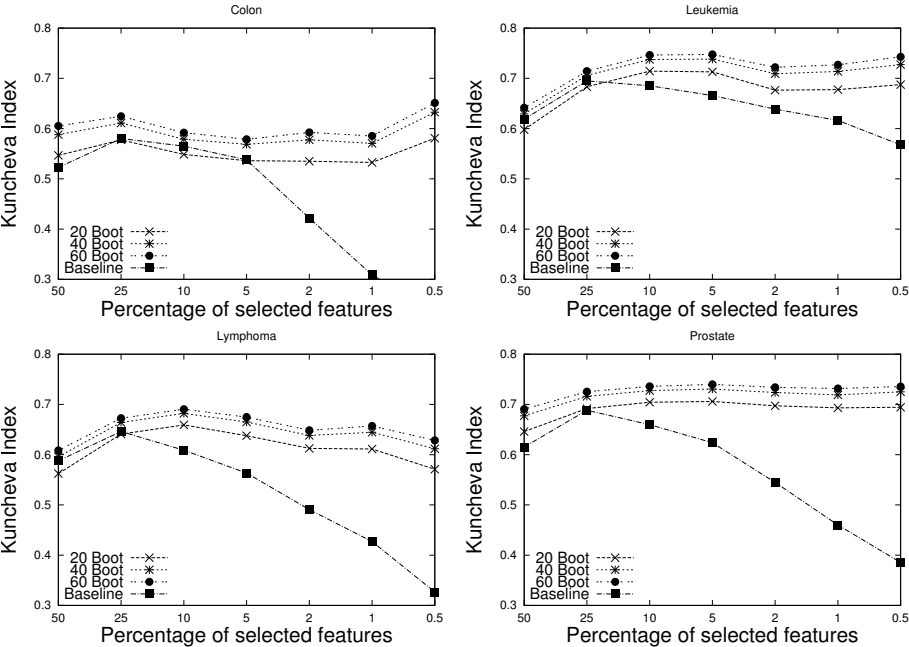


Figure F.3 Stability of the baseline method (RFE) and the ensemble methods for COL (up-left), LEU (up-right), LYM (bottom-left) and PRO (bottom-right) datasets. Effect of the number of bootstraps, when eliminating $E = 20\%$ features at each iteration of RFE.

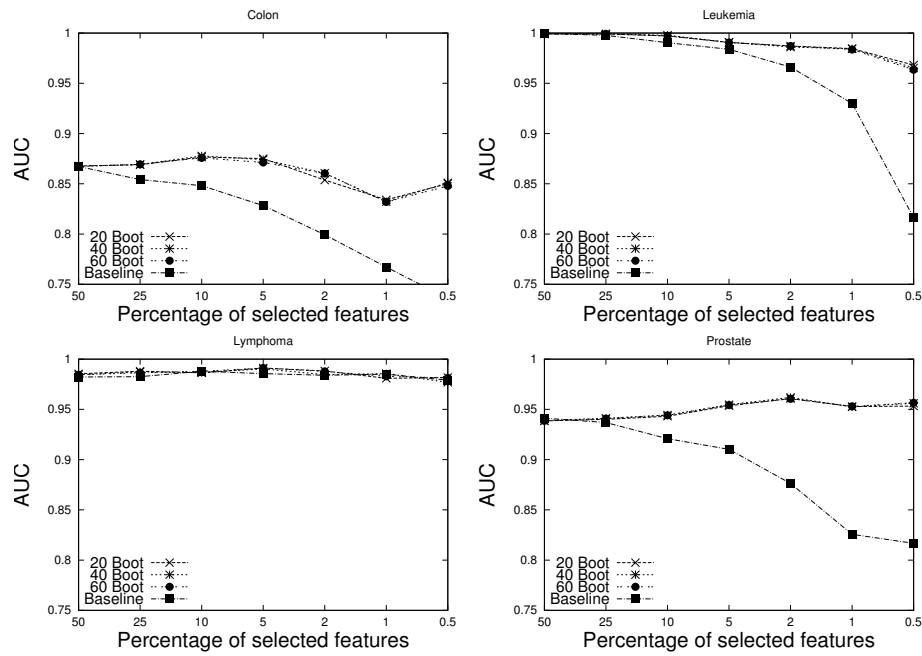


Figure F.4 Classification performance of the baseline method (RFE) and the ensemble methods for COL (up-left), LEU (up-right), LYM (bottom-left) and PRO (bottom-right) datasets. Effect of the number of bootstraps, when eliminating $E = 20\%$ features at each iteration of RFE.

F.3 Effect of the Number of Features to Eliminate

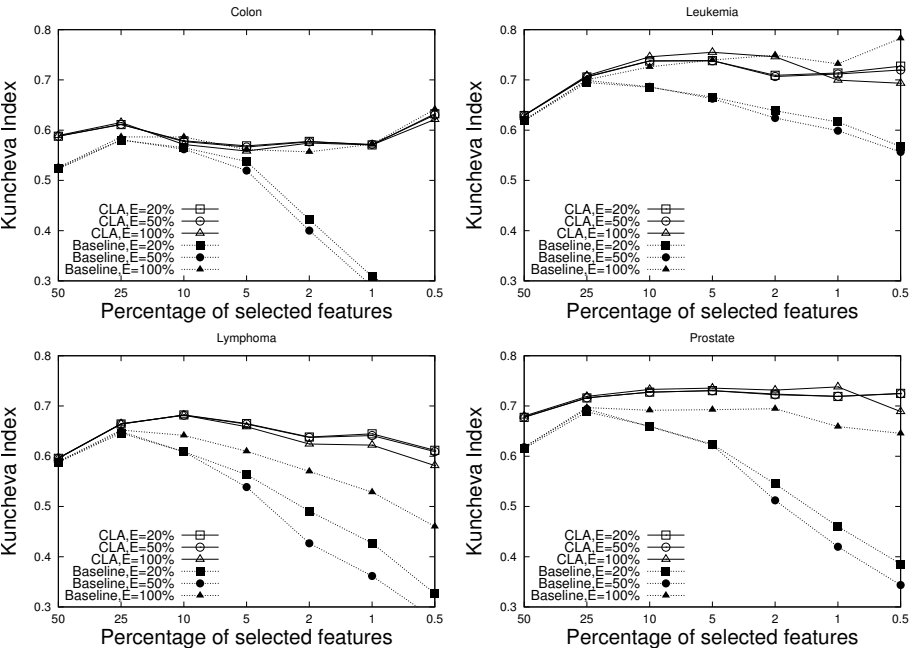


Figure F.5 Stability of the baseline method (RFE) and the ensemble methods for COL (up-left), LEU (up-right), LYM (bottom-left) and PRO (bottom-right) datasets. Effect of the number of features to eliminate at each iteration of RFE. Ensemble methods were created using 40 bootstraps.

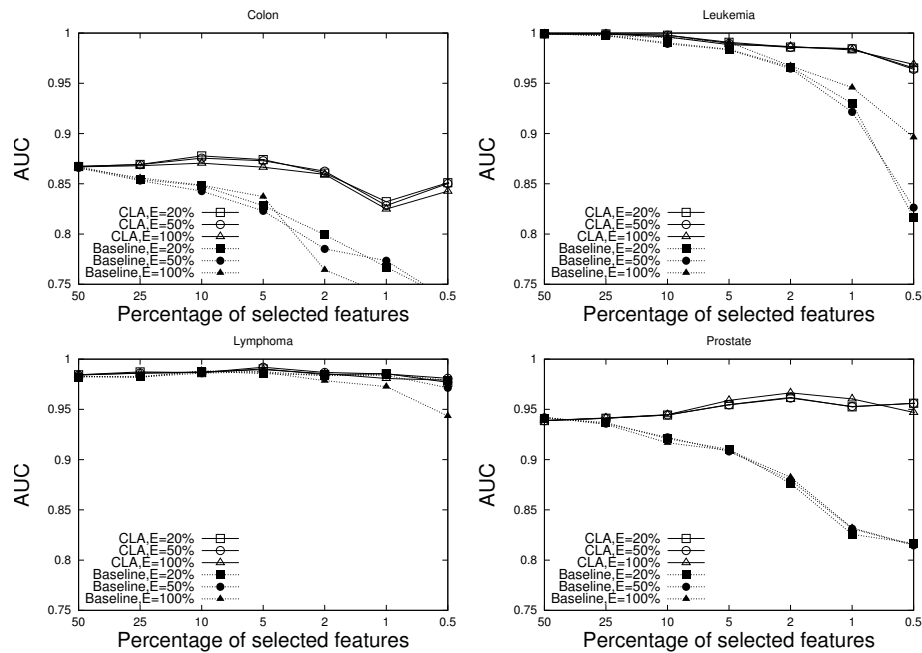


Figure F.6 Classification performance of the baseline method (RFE) and the ensemble methods for COL (up-left), LEU (up-right), LYM (bottom-left) and PRO (bottom-right) datasets. Effect of the number of features to eliminate at each iteration of RFE. Ensemble methods were created using 40 bootstraps.

F.4 Using 60 Bootstraps and a Single Run of an SVM (E=100%)

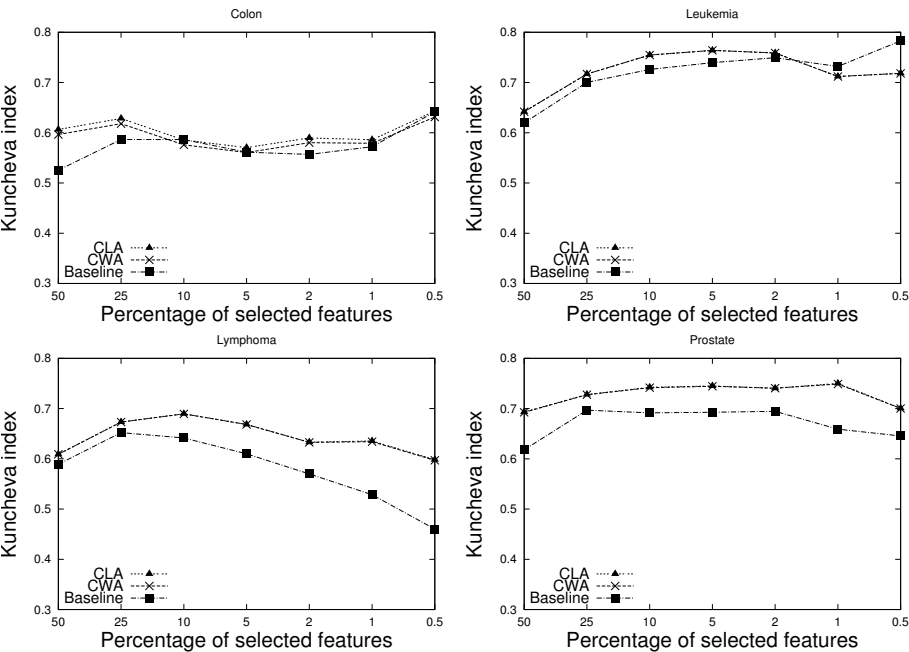


Figure F.7 Stability of the baseline method and the ensemble methods for COL (up-left), LEU (up-right), LYM (bottom-left) and PRO (bottom-right) datasets.

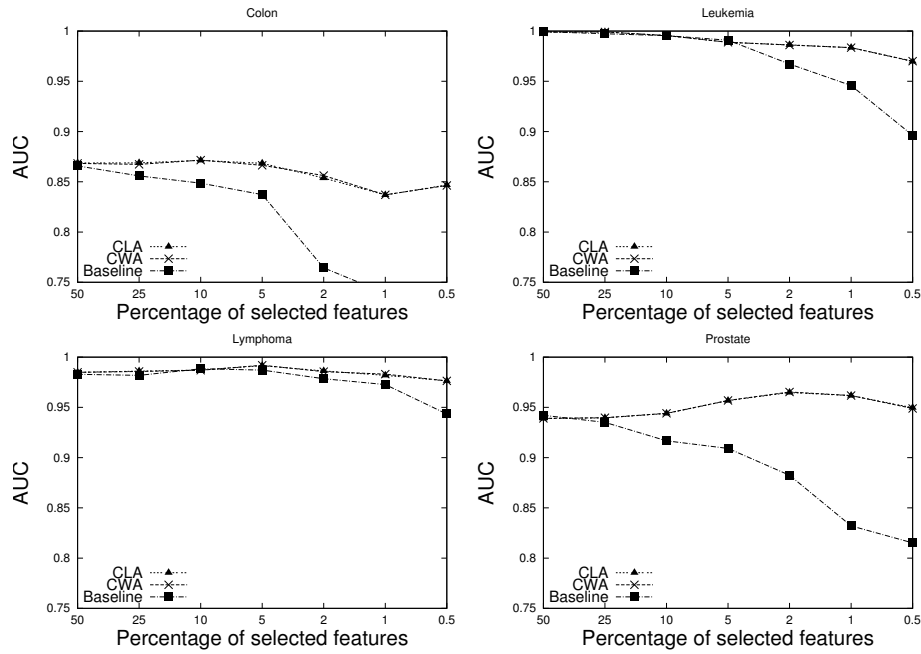
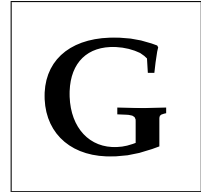


Figure F.8 Classification performance of the baseline method and the ensemble methods for COL (up-left), LEU (up-right), LYM (bottom-left) and PRO (bottom-right) datasets.

The Birthday Paradox



Chapter 5 explains the interest of random experiments to get confidence in a model or in a method. In Chapter 8, we led random experiments to assess the soundness of PS- L_2 -AROM: k genes are selected at random and favored via PS- L_2 -AROM. The rationale for such experiment is to validate the hypothesis along which this method discards irrelevantly favored features. In those experiments, we favor only $k = \{2, 3\}$ features (genes) and observe that the BCR curves fit quite well the baseline (where no features are favored) curves. The stability curves for random experiments were a bit higher than the baseline, but this is due to PS- L_2 -AROM itself, more likely to select the favored features when possible (i.e. when it does not degrade the performances). The random experiments are proven useful in this case. The work in (Louahed *et al.*, 2008; Dupont *et al.*, 2010) also included some permutation experiments.

The aim of this appendix is to outline a particular case where such random experiments produce counter-intuitive results. This misleading behavior is identified in Section G.1 and explained in Sections G.2 and G.3.

G.1 Is Randomness a Competitor?

In Chapter 9, we favor $k = 50$ genes corresponding to features selected on other datasets in a resampling protocol similar to the used in Chapter 8. However, no random experiments are reported with randomly favored features. This is no lapse of memory. This appendix begins with such an experiment. The results are displayed in Figures G.1 to G.3. On those figures, four out of five curves are just the same as in Chapter 9. The fifth one is the result of randomly favoring 50 genes. The first reasonable interpretation

of such a result is that the PS- L_2 -AROM method with randomly favored features generally provides an improvement in stability and in classification comparable to the favoring of a relevant transfer. Our selling argument for transfer learning is having a bad time.

One hypothesis to explain that phenomenon is that random selection of $k = 50$ features among $p = 12600$ features in the dataset could result in selecting *some* important genes also selected for relevant transfer with a not so low probability. This is counter intuitive because one would expect that the chance of selecting at random one of the 50 relevant features among 12600 is very low. We prove the opposite in the next section.

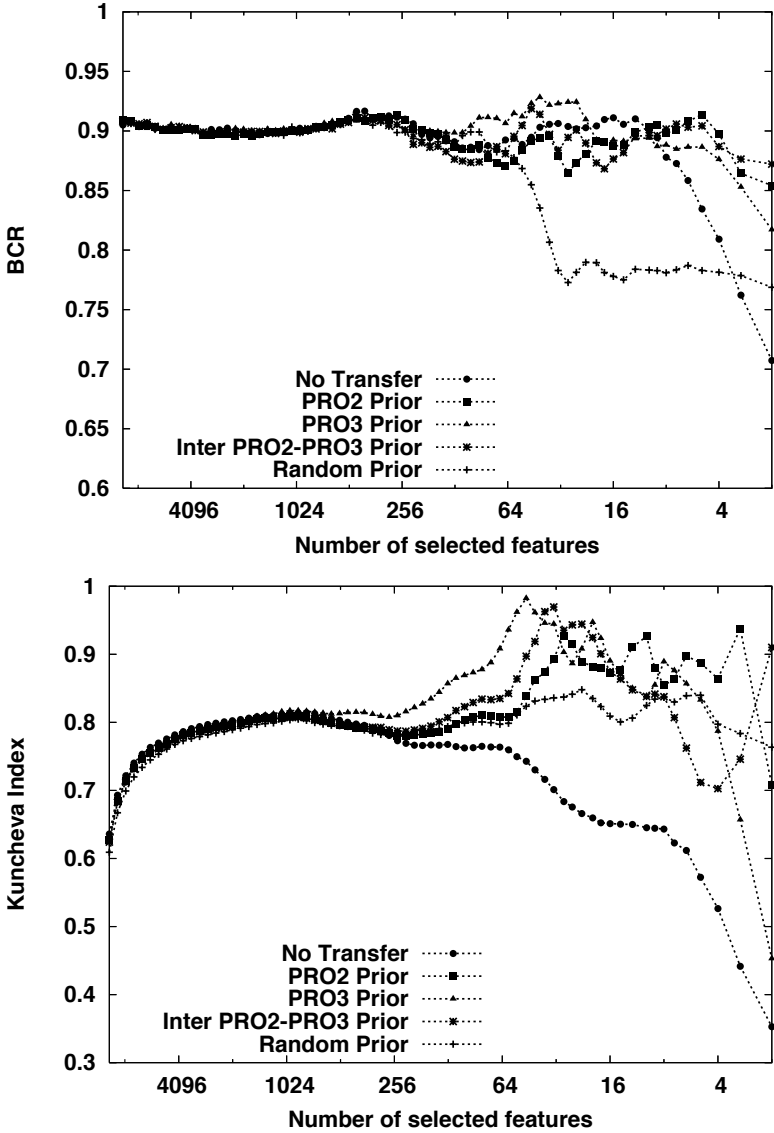


Figure G.1 Transfer learning results on on PRO1. Top: Classification performances (BCR). Bottom: Signature stability (Kuncheva index). No Transfer is the baseline: features are selected on the target dataset without prior preference. The next two curves specify which dataset was used in a single transfer setting. The fourth curve refers to the multiple transfer setting. The fifth curve correspond to the transfer of 50 randomly selected features.

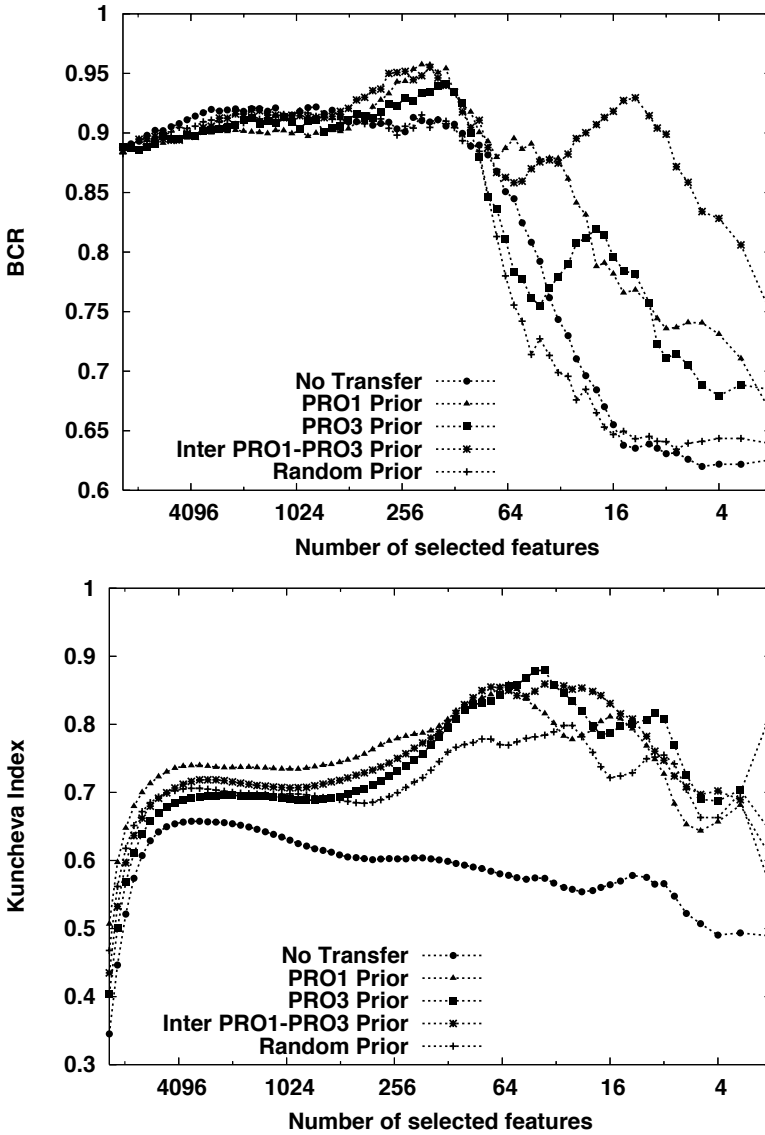


Figure G.2 Transfer learning results on on PRO2. Top: Classification performances (BCR). Bottom: Signature stability (Kuncheva index). No Transfer is the baseline: features are selected on the target dataset without prior preference. The next two curves specify which dataset was used in a single transfer setting. The fourth curve refers to the multiple transfer setting. The fifth curve correspond to the transfer of 50 randomly selected features.

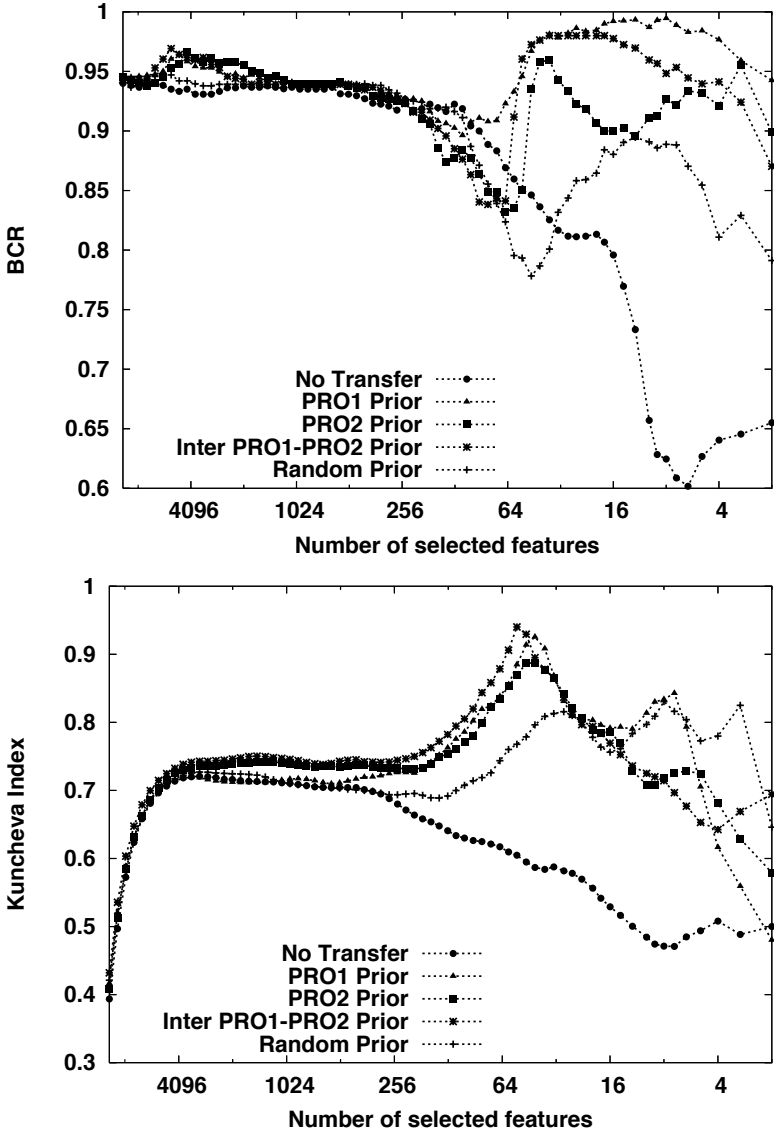


Figure G.3 Transfer learning results on on PRO3. Top: Classification performances (BCR). Bottom: Signature stability (Kuncheva index). No Transfer is the baseline: features are selected on the target dataset without prior preference. The next two curves specify which dataset was used in a single transfer setting. The fourth curve refers to the multiple transfer setting. The fifth curve correspond to the transfer of 50 randomly selected features.

G.2 Probability of Non-Selection

We want to compute the probability of **not** drawing any of the 50 features relevantly transferred via PS- L_2 -AROM during one random experiment. Let's call the set of 50 transferred features the *Signature on the Source Domain*, or $\mathbf{S}_s$. The chance that the first randomly selected feature f_1 is not contained in $\mathbf{S}_s$ is:

$$P(f_1 \cap \mathbf{S}_s = \emptyset) = \frac{12550}{12600}. \quad (\text{G.1})$$

Indeed, the random process can draw 12550 features not included in $\mathbf{S}_s$ among the 12600 available features. The chance that the second randomly selected feature f_2 is not contained in $\mathbf{S}_s$ is:

$$P(f_2 \cap \mathbf{S}_s = \emptyset) = \frac{12549}{12599}. \quad (\text{G.2})$$

We consider here drawings without repetition. Once a feature has been selected, it can't be selected again. That is why the random process can here only draw among 12549 features not included in $\mathbf{S}_s$ among the 12599 features remaining, supposing that the previously selected f_1 wasn't included also in $\mathbf{S}_s$. Continuing like that leads to draw the complete random signature $\mathbf{S}_{rand}$. The probability that $\mathbf{S}_{rand} \cap \mathbf{S}_s = \emptyset$ can thus be expressed like that:

$$P(\mathbf{S}_{rand} \cap \mathbf{S}_s = \emptyset) = \frac{12549}{12599} \times \frac{12548}{12598} \times \frac{12547}{12597} \times \dots \times \frac{12501}{12551}. \quad (\text{G.3})$$

More generally, it can be rewritten in terms of k and p as A_{p-k}^k , the number of drawings of k features among $p - k$ features (those $p - k$ features not included in the k features of $\mathbf{S}_s$):

$$A_{p-k}^k = \frac{(p-k)!}{(p-2k)!}, \quad (\text{G.4})$$

corrected by the fact that the order of selection is not important, thus leading to:

$$\begin{aligned} P(\mathbf{S}_{rand} \cap \mathbf{S}_s = \emptyset) &= \frac{(p-k)!}{(p-2k)!} \times \frac{(p-k)!}{p!} \\ &= \frac{(p-k) \times (p-k-1) \times (p-k-2) \times \dots \times (p-2k+1)}{(p) \times (p-1) \times (p-2) \times \dots \times (p-k+1)}. \end{aligned} \quad (\text{G.5})$$

G.3 Effect in Practice

Theoretically, if $p = 7129$ and $k = 2$, as it is the case for the DLBCL dataset in Chapter 8, the probability of **not** selecting at random the 2 relevant features

is: 0.999439. This means that the probability of selecting relevant features by chance is really low ($P = 0.000561$).

Still according to our hypothesis, if $p = 7129$ and $k = 3$, as it is the case for the Leukemia dataset in Chapter 8, the probability of **not** selecting at random the 3 relevant features is: 0.998738. This means that the probability of selecting relevant features by chance is also really low ($P = 0.001272$).

Again, if $p = 12600$ and $k = 50$, as it is the case for all the datasets considered in Chapter 9, the probability of **not** selecting at random the 50 relevant features is: 0.8193895. This means that the probability of selecting relevant features by chance is not so low ($P = 0.1806105$).

Considering the fact that those experiments were each repeated 10 times independently, those probabilities become respectively: 0.005595859 (0.56%), 0.01264744 (1.26%) and 0.8635719 (86.36%). Those are the probabilities of selecting at least once, in 10 independent random experiment, one or more features relevantly favored via PS- L_2 -AROM. Otherwise said, if PS- L_2 -AROM really drops irrelevant favored features, but is positively impacted by relevant features, it is not surprising that the BCR performances of the random curves shown in Figures G.1 to G.3 are much higher than the baseline.

These theoretical numbers should be compared against what is obtained in practice on the three prostate datasets. Tables G.1, G.2 and G.3 respectively show for PRO1, PRO2 and PRO3 the intersection between the 10 random prior signatures $\mathbf{R}_0$ to $\mathbf{R}_9$ and several prior signatures assumed to be relevant, extracted via a t -test from other source datasets. By taking the average of the *Empty* line from those tables for the signatures of size 50 (three first columns), we obtain 8.333 which is thus the average number of time over 10 runs where there is no intersection between the random signature and a relevant signature. This figure, 83.333% is really close to $1 - 0.1806105 \approx 81.94\%$, verifying our theoretical reasoning. The phenomenon described in this chapter is related to the *Birthday Paradox* (Mckinney, 1966) that can be expressed as follows:

The birthday paradox pertains to the probability that in a set of randomly chosen people some pair of them will have the same birthday. In a group of at least 23 randomly chosen people, there is more than 50% probability that some pair of them

Table G.1 PRO1 – Size (and indexes of the corresponding features) of the intersection between the random signatures and relevant signatures. The *Ave* line is the average size of the intersection. The *Empty* line is the number of empty intersections. PRO3_k, PRO2_k and PRO3PRO2_k respectively correspond to a signature of size *k* extracted via a *t*-test ranking on PRO3, PRO2, and the intersection of both.

	PRO3 ₅₀	PRO2 ₅₀	PRO3PRO2 ₅₀	PRO3 ₁₀₀	PRO2 ₁₀₀
Ave	0	0.1	0.1	0.2	0.2
Empty	10	9	9	8	8
R ₀					
R ₁	1 (40)				
R ₂					
R ₃					
R ₄	1 (56)				
R ₅	1 (79)				
R ₆	6 (84)				
R ₇					
R ₈					
R ₉	1 (43)				1 (43)

will have the same birthday. For 57 or more people, the probability is more than 99%, and it reaches 100% when, ignoring leap-years, the number of people reaches 366.

The problem considered here is however not strictly equivalent to the birthday paradox, because in that paradox several people can be born on the same day. However, here, features can be selected only once. Neither the birthday paradox, nor our result are in fact real paradoxes in the sense of leading to a logical contradiction, but they are called a paradox because the mathematical truth contradicts naive intuition: most people estimate that the chance is much lower than 50%.

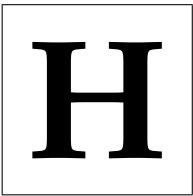
Table G.2 PRO2 – Size (and indexes of the corresponding features) of the intersection between the random signatures and relevant signatures. The *Ave* line is the average size of the intersection. The *Empty* line is the number of empty intersections. $PRO1_k$, $PRO3_k$ and $PRO1PRO3_k$ respectively correspond to a signature of size k extracted via a t -test ranking on PRO1, PRO3, and the intersection of both.

	PRO1 ₅₀	PRO3 ₅₀	PRO1PRO3 ₅₀	PRO1 ₁₀₀	PRO3 ₁₀₀
Ave	0.5	0.3	0.1	0.6	0.4
Empty	6	8	9	6	7
R ₀	1 (49)				
R ₁					
R ₂	1 (29)	2 (41,26)	1 (6)	2 (71,29)	2 (41,26)
R ₃					
R ₄	1 (29)				2 (29,80)
R ₅	1 (83)				
R ₆	1 (19)			1 (19)	
R ₇					
R ₈	2 (4,32)			2 (4,32)	
R ₉					

Table G.3 PRO3 – Size (and indexes of the corresponding features) of the intersection between the random signatures and relevant signatures. The *Ave* line is the average size of the intersection. The *Empty* line is the number of empty intersections. $PRO1_k$, $PRO2_k$ and $PRO1PRO2_k$ respectively correspond to a signature of size k extracted via a t -test ranking on PRO1, PRO2, and the intersection of both.

	PRO1 ₅₀	PRO2 ₅₀	PRO1PRO2 ₅₀	PRO1 ₁₀₀	PRO2 ₁₀₀
Ave	0.1	0.2	0.3	0.4	0.4
Empty	9	8	7	7	6
R ₀	1 (20)		1 (20)		
R ₁	1 (47)				1 (53)
R ₂	1 (26)		1 (55)		1 (26)
R ₃					1 (97)
R ₄					
R ₅	1 (34)				
R ₆					
R ₇			1 (9)	2 (56,76)	
R ₈	1 (15)		1 (15)		
R ₉					

Cristall: Supplementary Material



H.1 List of a Priori Relevant Features

IL1B

	Probeset ID	Symbol	GenBank	Names
1	205067_at	IL1B	NM_000576	interleukin 1, beta
2	39402_at	IL1B	M15330	interleukin 1, beta

IL3

	Probeset ID	Symbol	GenBank	Names
1	207906_at	IL3	NM_000588	interleukin 3 (colony-stimulating factor, multiple)

IL4

	Probeset ID	Symbol	GenBank	Names
1	207538_at	IL4	NM_000589	interleukin 4
2	207539_s.at	IL4	NM_000589	interleukin 4

IL5

	Probeset ID	Symbol	GenBank	Names
1	207952_at	IL5	NM_000879	interleukin 5 (colony-stimulating factor, eosinophil)

IL8

	Probeset ID	Symbol	GenBank	Names
1	202859_x_at	IL8	NM_000584	interleukin 8
2	211506_s_at	IL8	AF043337	interleukin 8

IL9

	Probeset ID	Symbol	GenBank	Names
1	208193_at	IL9	NM_000590	interleukin 9

IL10

	Probeset ID	Symbol	GenBank	Names
1	207433_at	IL10	NM_000572	interleukin 10

IL13

	Probeset ID	Symbol	GenBank	Names
1	207844_at	IL13	NM_002188	interleukin 13

IL17

	Probeset ID	Symbol	GenBank	Names
1	208402_at	IL17A	NM_002190	interleukin 17A
2	216876_s_at	IL17A	Z58820	interleukin 17A

STAT3

	Probeset ID	Symbol	GenBank	Names
1	208991_at	STAT3	AA634272	signal transducer and activator of transcription 3 (acute-phase response factor)
2	208992_s_at	STAT3	BC000627	signal transducer and activator of transcription 3 (acute-phase response factor)
3	225289_at	STAT3	AI139252	signal transducer and activator of transcription 3 (acute-phase response factor)
4	243213_at	STAT3	BF508977	signal transducer and activator of transcription 3 (acute-phase response factor)

RORC

	Probeset ID	Symbol	GenBank	Names
1	206419_at	RORC	NM_005060	RAR-related orphan receptor C
2	228806_at	RORC	AI218580	RAR-related orphan receptor C

IFNG

	Probeset ID	Symbol	GenBank	Names
1	210354_at	IFNG	M29383	interferon, gamma

GMCSF

	Probeset ID	Symbol	GenBank	Names
1	210228_at	CSF2	M11734	colony stimulating factor 2 (granulocyte-macrophage)
2	210229_s_at	CSF2	M11734	colony stimulating factor 2 (granulocyte-macrophage)

TGFB

	Probeset ID	Symbol	GenBank	Names
1	203084_at	TGFB1	NM_000660	transforming growth factor, beta 1
2	203085_s_at	TGFB1	BC000125	transforming growth factor, beta 1

TNFA

	Probeset ID	Symbol	GenBank	Names
1	207113_s_at	TNF	NM_000594	tumor necrosis factor (TNF superfamily, member 2)

NR3C1

	Probeset ID	Symbol	GenBank	Names
1	201865_x.at	NR3C1	AI432196	nuclear receptor subfamily 3, group C, member 1 (glucocorticoid receptor)
2	201866_s.at	NR3C1	NM_000176	nuclear receptor subfamily 3, group C, member 1 (glucocorticoid receptor)
3	211671_s.at	NR3C1	U01351	nuclear receptor subfamily 3, group C, member 1 (glucocorticoid receptor)
4	216321_s.at	NR3C1	X03348	nuclear receptor subfamily 3, group C, member 1 (glucocorticoid receptor)

TGM2

	Probeset ID	Symbol	GenBank	Names
1	201042.at	TGM2	AL031651	transglutaminase 2 (C polypeptide, protein-glutamine-gamma-glutamyltransferase)
2	211003_x.at	TGM2	BC003551	transglutaminase 2 (C polypeptide, protein-glutamine-gamma-glutamyltransferase)
3	211573_x.at	TGM2	M98478	transglutaminase 2 (C polypeptide, protein-glutamine-gamma-glutamyltransferase)
4	216183.at	TGM2	AL512703	transglutaminase 2 (C polypeptide, protein-glutamine-gamma-glutamyltransferase)

STAT6

	Probeset ID	Symbol	GenBank	Names
1	201331_s.at	STAT6	BC004973	signal transducer and activator of transcription 6, interleukin-4 induced
2	201332_s.at	STAT6	NM.003153	signal transducer and activator of transcription 6, interleukin-4 induced

GATA3

	Probeset ID	Symbol	GenBank	Names
1	209602_s.at	GATA3	AI796169	GATA binding protein 3
2	209603_at	GATA3	AI796169	GATA binding protein 3
3	209604_s.at	GATA3	BC003070	GATA binding protein 3

HSP60

	Probeset ID	Symbol	GenBank	Names
1	200806_s.at	HSPD1	BE256479	heat shock 60kDa protein 1 (chaperonin)
2	200807_s.at	HSPD1	NM.002156	heat shock 60kDa protein 1 (chaperonin)
3	241716_at	HSPD1	BF965447	heat shock 60kDa protein 1 (chaperonin)
4	243372_at	HSPD1	AW674195	heat shock 60kDa protein 1 (chaperonin)
5	243845_at	HSPD1	AI762164	heat shock 60kDa protein 1 (chaperonin)

CCL11

	Probeset ID	Symbol	GenBank	Names
1	210133_at	CCL11	D49372	chemokine (C-C motif) ligand 11

FOXP3

	Probeset ID	Symbol	GenBank	Names
1	221333_at	FOXP3	NM_014009	forkhead box P3
2	221334_s_at	FOXP3	NM_014009	forkhead box P3
3	224211_at	FOXP3	AF277993	forkhead box P3

TLR2

	Probeset ID	Symbol	GenBank	Names
1	204924_at	TLR2	NM_003264	toll-like receptor 2

TLR4

	Probeset ID	Symbol	GenBank	Names
1	1552798_a.at	TLR4	NM_138557	toll-like receptor 4
2	221060_s.at	TLR4	NM_003266	toll-like receptor 4
3	224341_x.at	TLR4	U93091	toll-like receptor 4
4	232068_s.at	TLR4	AF177765	toll-like receptor 4

TLR9

	Probeset ID	Symbol	GenBank	Names
1	223903_at	TLR9	AB045180	toll-like receptor 9

FCER1A

	Probeset ID	Symbol	GenBank	Names
1	1562537_at	FCER1A	BC015185	Fc fragment of IgE, high affinity I, receptor for; alpha polypeptide
2	211734_s_at	FCER1A	BC005912	Fc fragment of IgE, high affinity I, receptor for; alpha polypeptide

IDO

	Probeset ID	Symbol	GenBank	Names
1	210029_at	IDO1	M34455	indoleamine 2,3-dioxygenase 1

CXCL2

	Probeset ID	Symbol	GenBank	Names
1	1569203_at	CXCL2	BC005276	chemokine (C-X-C motif) ligand 2
2	209774_x_at	CXCL2	M57731	chemokine (C-X-C motif) ligand 2
3	230101_at	CXCL2	AV648479	chemokine (C-X-C motif) ligand 2

CXCL3

	Probeset ID	Symbol	GenBank	Names
1	207850_at	CXCL3	NM_002090	chemokine (C-X-C motif) ligand 3

PTGS2

	Probeset ID	Symbol	GenBank	Names
1	1554997_a.at	PTGS2	AY151286	prostaglandin-endoperoxide synthase 2 (prostaglandin G/H synthase and cyclooxygenase)
2	204748_at	PTGS2	NM_000963	prostaglandin-endoperoxide synthase 2 (prostaglandin G/H synthase and cyclooxygenase)

FLG

	Probeset ID	Symbol	GenBank	Names
1	207822_at	FGFR1	NM_023107	fibroblast growth factor receptor 1
2	207937_x.at	FGFR1	NM_023110	fibroblast growth factor receptor 1
3	210973_s.at	FGFR1	M63889	fibroblast growth factor receptor 1
4	211535_s.at	FGFR1	M60485	fibroblast growth factor receptor 1
5	215404_x.at	FGFR1	AK024388	fibroblast growth factor receptor 1
6	215704_at	FLG	AL356504	filaggrin
7	222164_at	FGFR1	AU145411	fibroblast growth factor receptor 1
8	226705_at	FGFR1	BE467261	fibroblast growth factor receptor 1

NFKB1

	Probeset ID	Symbol	GenBank	Names
1	209239_at	NFKB1	M55643	nuclear factor of kappa light polypeptide gene enhancer in B-cells 1

NFKB2

	Probeset ID	Symbol	GenBank	Names
1	207535_s.at	NFKB2	NM_002502	nuclear factor of kappa light polypeptide gene enhancer in B-cells 2 (p49/p100)
2	209636_at	NFKB2	BC002844	nuclear factor of kappa light polypeptide gene enhancer in B-cells 2 (p49/p100)
3	211524_at	NFKB2	U09609	nuclear factor of kappa light polypeptide gene enhancer in B-cells 2 (p49/p100)

HLAA

	Probeset ID	Symbol	GenBank	Names
1	213932_x.at	HLA-A	AI923492	major histocompatibility complex, class I, A
2	215313_x.at	HLA-A	AA573862	major histocompatibility complex, class I, A
3	217436_x.at	HLA-A	M80469	major histocompatibility complex, class I, A

HLAB

	Probeset ID	Symbol	GenBank	Names
1	208729_x.at	HLA-B	D83043	major histocompatibility complex, class I, B
2	209140_x.at	HLA-B	L42024	major histocompatibility complex, class I, B
3	211911_x.at	HLA-B	L07950	major histocompatibility complex, class I, B

HLAC

	Probeset ID	Symbol	GenBank	Names
1	208729_x.at	HLA-B	D83043	major histocompatibility complex, class I, B
2	209140_x.at	HLA-B	L42024	major histocompatibility complex, class I, B
3	211911_x.at	HLA-B	L07950	major histocompatibility complex, class I, B

HLA-DPA1

	Probeset ID	Symbol	GenBank	Names
1	211990_at	HLA-DPA1	M27487	major histocompatibility complex, class II, DP alpha 1
2	211991_s.at	HLA-DPA1	M27487	major histocompatibility complex, class II, DP alpha 1
3	213537_at	HLA-DPA1	AI128225	major histocompatibility complex, class II, DP alpha 1

HLA-DPB1

	Probeset ID	Symbol	GenBank	Names
1	201137_s.at	HLA-DPB1	NM_002121	major histocompatibility complex, class II, DP beta 1

HLA-DRA

	Probeset ID	Symbol	GenBank	Names
1	208894_at	HLA-DRA	M60334	major histocompatibility complex, class II, DR alpha
2	210982_s.at	HLA-DRA	M60333	major histocompatibility complex, class II, DR alpha

HLA-DQA1

	Probeset ID	Symbol	GenBank	Names
1	203290_at	HLA-DQA1	NM_002122	major histocompatibility complex, class II, DQ alpha 1
2	212671_s.at	HLA-DQA1	BG397856	major histocompatibility complex, class II, DQ alpha 1
3	213831_at	HLA-DQA1	X00452	major histocompatibility complex, class II, DQ alpha 1
4	236203_at	HLA-DQA1	AI377755	major histocompatibility complex, class II, DQ alpha 1

HLA-DRB1

	Probeset ID	Symbol	GenBank	Names
1	209312_x.at	HLA-DRB1	U65585	major histocompatibility complex, class II, DR beta 1
2	215193_x.at	HLA-DRB1	AJ297586	major histocompatibility complex, class II, DR beta 1

HLA-DQB1

	Probeset ID	Symbol	GenBank	Names
1	209480_at	HLA-DQB1	M16276	major histocompatibility complex, class II, DQ beta 1
2	209823_x_at	HLA-DQB1	M17955	major histocompatibility complex, class II, DQ beta 1
3	210747_at	HLA-DQB1	M24364	major histocompatibility complex, class II, DQ beta 1
4	211654_x_at	HLA-DQB1	M17565	major histocompatibility complex, class II, DQ beta 1
5	211656_x_at	HLA-DQB1	M32577	major histocompatibility complex, class II, DQ beta 1
6	212998_x_at	HLA-DQB1	AI583173	major histocompatibility complex, class II, DQ beta 1

HLA-G

	Probeset ID	Symbol	GenBank	Names
1	210514_x_at	HLA-G	AF226990	major histocompatibility complex, class I, G
2	211528_x_at	HLA-G	M90685	major histocompatibility complex, class I, G
3	211529_x_at	HLA-G	M90684	major histocompatibility complex, class I, G
4	211530_x_at	HLA-G	M90686	major histocompatibility complex, class I, G

H.2 Expression Profiles for the Preliminary Cohort

Figure H.1 33 samples profiles (feature-normalized) for probesets in GCRMA signatures. Non-allergic samples are in red, allergic samples are in green

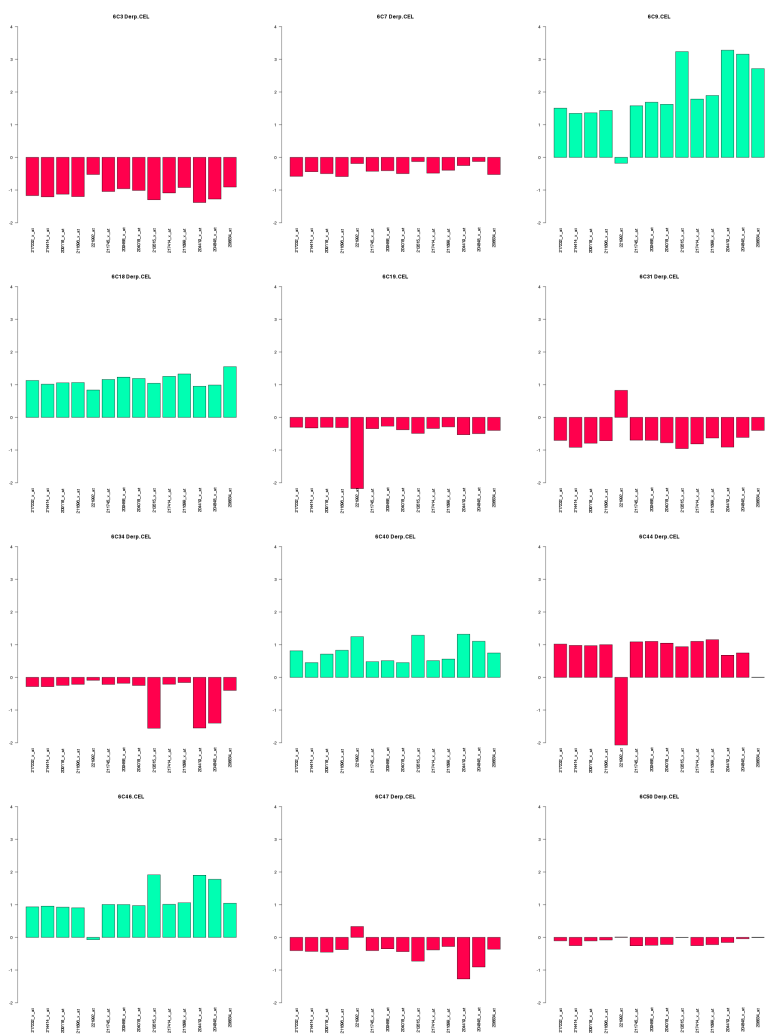
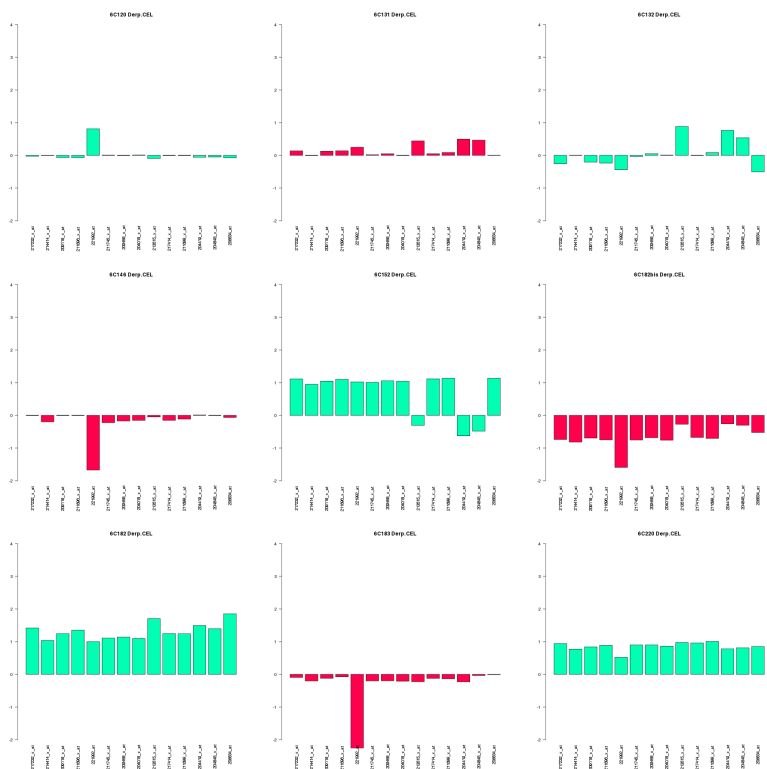


Figure H.2 (Figure H.1 Continued) 33 samples profiles (feature-normalized) for probesets in GCRMA signatures. Non-allergic samples are in red, allergic samples are in green

Figure H.3 (Figure H.2 Continued) 33 samples profiles (feature-normalized) for probesets in GCRMA signatures. Non-allergic samples are in red, allergic samples are in green



H.3 *L*₂-AROM and PS-*L*₂-AROM Comparison

Table H.1 Probesets selected by both *L*₂-AROM and PS-*L*₂-AROM methods on a single run on the 73 Cristall samples.

	Probeset ID	Symbol	Names
1	214539_at	SERPINB10	serpin peptidase inhibitor, clade B (ovalbumin), member 10
2	243610_at	C9orf135	chromosome 9 open reading frame 135
3	220784_s_at	UTS2	urotensin 2
4	241703_at	RUNDC3B	RUN domain containing 3B
5	213979_s_at	CTBP1	C-terminal binding protein 1
6	1555600_s_at	APOL4	apolipoprotein L, 4
7	222197_s_at	LOC100128008	hypothetical protein LOC100128008
8	206575_at	CDKL5	cyclin-dependent kinase-like 5
9	214591_at	KLHL4	kelch-like 4 (Drosophila)
10	233002_at	PPP4R4	protein phosphatase 4, regulatory subunit 4
11	241284_at	NA	NA
12	1552848_a_at	PTCHD1	patched domain containing 1
13	206371_at	FOLR3	folate receptor 3 (gamma)
14	243277_x_at	EVI1	ecotropic viral integration site 1
15	1569182_at	NA	NA
16	220785_at	UTS2	urotensin 2
17	229880_at	LOC643517	hypothetical LOC643517
18	230378_at	SCGB3A1	secretoglobin, family 3A, member 1
19	235214_at	C1orf190	chromosome 1 open reading frame 190
20	231458_at	NA	NA
21	210062_s_at	ZNF589	zinc finger protein 589

Table H.2 Probesets selected by PS- L_2 -AROM among the 102 favored ones.

	Probeset ID	Symbol	Names
1	205067_at	IL1B	interleukin 1, beta
2	207538_at	IL4	interleukin 4
3	208193_at	IL9	interleukin 9
4	207433_at	IL10	interleukin 10
5	208402_at	IL17A	interleukin 17A
6	216876_s.at	IL17A	interleukin 17A
7	208991_at	STAT3	signal transducer and activator of transcription 3 (acute-phase response factor)
8	208992_s.at	STAT3	signal transducer and activator of transcription 3 (acute-phase response factor)
9	243213_at	STAT3	signal transducer and activator of transcription 3 (acute-phase response factor)
10	206419_at	RORC	RAR-related orphan receptor C
11	228806_at	RORC	RAR-related orphan receptor C
12	210354_at	IFNG	interferon, gamma
13	203084_at	TGFB1	transforming growth factor, beta 1
14	201331_s.at	STAT6	signal transducer and activator of transcription 6, interleukin-4 induced
15	201332_s.at	STAT6	signal transducer and activator of transcription 6, interleukin-4 induced
16	201042_at	TGM2	transglutaminase 2 (C polypeptide, protein-glutamine-gamma-glutamyltransferase)
17	211671_s.at	NR3C1	nuclear receptor subfamily 3, group C, member 1 (glucocorticoid receptor)
18	209604_s.at	GATA3	GATA binding protein 3
19	241716_at	HSPD1	heat shock 60kDa protein 1 (chaperonin)
20	243372_at	HSPD1	heat shock 60kDa protein 1 (chaperonin)
21	243845_at	HSPD1	heat shock 60kDa protein 1 (chaperonin)

Table H.3 Table H.2 continued.

	Probeset ID	Symbol	Names
22	210133.at	CCL11	chemokine (C-C motif) ligand 11
23	221334.s.at	FOXP3	forkhead box P3
24	224211.at	FOXP3	forkhead box P3
25	204924.at	TLR2	toll-like receptor 2
26	1552798.a.at	TLR4	toll-like receptor 4
27	221060.s.at	TLR4	toll-like receptor 4
28	224341.x.at	TLR4	toll-like receptor 4
29	232068.s.at	TLR4	toll-like receptor 4
30	1562537.at	FCER1A	Fc fragment of IgE, high affinity I, receptor for; alpha polypeptide
31	211734.s.at	FCER1A	Fc fragment of IgE, high affinity I, receptor for; alpha polypeptide
32	210029.at	IDO1	indoleamine 2,3-dioxygenase 1
33	209774.x.at	CXCL2	chemokine (C-X-C motif) ligand 2
34	230101.at	CXCL2	chemokine (C-X-C motif) ligand 2
35	207850.at	CXCL3	chemokine (C-X-C motif) ligand 3
36	207535.s.at	NFKB2	nuclear factor of kappa light polypeptide gene enhancer in B-cells 2 (p49/p100)
37	209636.at	NFKB2	nuclear factor of kappa light polypeptide gene enhancer in B-cells 2 (p49/p100)
38	211524.at	NFKB2	nuclear factor of kappa light polypeptide gene enhancer in B-cells 2 (p49/p100)
39	207937.x.at	FGFR1	fibroblast growth factor receptor 1
40	210973.s.at	FGFR1	fibroblast growth factor receptor 1
41	211535.s.at	FGFR1	fibroblast growth factor receptor 1
42	215404.x.at	FGFR1	fibroblast growth factor receptor 1
43	215704.at	FLG	filaggrin
44	217436.x.at	HLA-A	major histocompatibility complex, class I, A

Table H.4 Table H.3 continued.

	Probeset ID	Symbol	Names
45	208729_x.at	HLA-B	major histocompatibility complex, class I, B
46	209140_x.at	HLA-B	major histocompatibility complex, class I, B
47	211911_x.at	HLA-B	major histocompatibility complex, class I, B
48	201137_s.at	HLA-DPB1	major histocompatibility complex, class II, DP beta 1
49	203290.at	HLA-DQA1	major histocompatibility complex, class II, DQ alpha 1
50	213831.at	HLA-DQA1	major histocompatibility complex, class II, DQ alpha 1
51	209480.at	HLA-DQB1	major histocompatibility complex, class II, DQ beta 1
52	209823_x.at	HLA-DQB1	major histocompatibility complex, class II, DQ beta 1
53	211654_x.at	HLA-DQB1	major histocompatibility complex, class II, DQ beta 1
54	211656_x.at	HLA-DQB1	major histocompatibility complex, class II, DQ beta 1
55	212998_x.at	HLA-DQB1	major histocompatibility complex, class II, DQ beta 1
56	208894.at	HLA-DRA	major histocompatibility complex, class II, DR alpha
57	209312_x.at	HLA-DRB1	major histocompatibility complex, class II, DR beta 1
58	215193_x.at	HLA-DRB1	major histocompatibility complex, class II, DR beta 1
59	210514_x.at	HLA-G	major histocompatibility complex, class I, G
60	211529_x.at	HLA-G	major histocompatibility complex, class I, G
61	211530_x.at	HLA-G	major histocompatibility complex, class I, G

References

- Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., and Saeys, Y. (2009). Robust biomarker identification for cancer diagnosis using ensemble feature selection methods. In *Machine Learning in Systems Biology*.
- Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., and Saeys, Y. (2010). Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics (Oxford, England)*, **26**(3), 392–398.
- Ahmed, A. and Brenton, J. (2005). Microarrays and breast cancer clinical studies: forgetting what we have not yet learn. *Breast Cancer Research*, **7**(3), 96–99.
- Alizadeh, A., Eisen, M., Davis, R., Ma, C., Lossos, I., Rosenwald, A., Boldrick, J., Sabet, H., Tran, T., Yu, X., Powell, J., Yang, L., Marti, G., Moore, T., Hudson, J., Lu, L., Lewis, D., Tibshirani, R., Sherlock, G., Chan, W., Greiner, T., Weisenburger, D., Armitage, J., Warnke, R., Levy, R., Wilson, W., Grever, M., Byrd, J., Botstein, D., Brown, P., and Staudt, L. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, **403**(3), 503–511.
- Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., and Levine, A. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS*, **96**, 6745–6750.
- Ambroise, C. and McLachlan, G. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *PNAS*, **99**(10), 6562–6566.
- Argyriou, A., Evgeniou, T., and Pontil, M. (2006). Multi-task feature learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 41–48.
- Argyriou, A., Evgeniou, T., and Pontil, M. (2008). Convex multi-task feature learning. *Machine Learning*, **73**(3), 243–272.

- Bach, F. (2008). Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, **9**, 1179–1225.
- Bach, F. (2009). NIPS Tutorial, Sparse methods for machine learning: Theory and algorithms.
- Baxter, J. (2000). A model of inductive bias learning. *Journal of Artificial Intelligence Research*, **12**, 149–198.
- Bekkerman, R., El-Yaniv, R., Tishby, N., and Winter, Y. (2003). Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research*, **3**, 1183–1208.
- Bellman, R. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton University Press.
- Ben-David, S. and Lindenbaum, M. (1998). Localization versus identification of semi-algebraic sets. *Machine Learning*, **32**(3), 207–224.
- Ben-David, S. and Schuller Borbely, R. (2008). A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine Learning*, **73**(3), 273–287.
- Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. (2007). Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems (NIPS)*.
- Ben-Hur, A., Ong, C., Sonnenburg, S., Schölkopf, B., and Raetsch, G. (2008). Support vector machines and kernels for computational biology. *PLoS Comput Biol*, **4**(10), e1000173.
- Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B*, **57**(1), 289–330.
- Berkson, J. (1944). Application to the logistic function to bio-assay. *Journal of the American Statistical Association*, **39**(227), 357–365.
- Bishop, C. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer.
- Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, Sydney, Australia. Association for Computational Linguistics.

- Bolstad, B., Irizarry, R., Astrand, M., and Speed, T. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, **19**(2), 185–193.
- Boser, B., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In *Fifth Annual Workshop on Computational Learning Theory*, ACM, pages 144–152.
- Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.
- Braga-Neto, U. and Dougherty, E. (2004). Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, **20**(3), 374–380.
- Branders, S. and Paul, J. (2010). *Etude comparative de techniques de sélection de variables en catégorisation de documents et en analyse de données génomiques*. Master’s thesis, University of Louvain, Louvain School of Engineering.
- Breiman, L. (1996a). Arcing classifiers. Technical report, University of California, Department of Statistics.
- Breiman, L. (1996b). Bagging predictors. *Machine Learning*, pages 123–140.
- Breiman, L. (2001). Random forests. *Machine Learning*, **45**(1), 5–32.
- Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Chapman & Hall, New York, NY.
- Candès, E. (2006). Compressive sampling. In *International Congress of Mathematics*, volume 3, pages 1433–1452.
- Candès, E. and Tao, T. (2005). Decoding by linear programming. *IEEE Transactions on Information Theory*, **51**(12), 4203–4215.
- Candès, E., Wakin, M., and Boyd, S. (2007). Enhancing sparsity by reweighted ℓ_1 minimization. *Journal Fourier Analysis and Applications*, **14**, 877–905.
- Caruana, R. (1997). Multitask learning. *Machine Learning*, **28**(1), 41–75.
- Chandran, U., Ma, C., Dhir, R., Bisceglia, M., Lyons-Weiler, M., Liang, W., Michalopoulos, G., Becich, M., and Monzon, F. (2007). Gene expression profiles of prostate cancer reveal involvement of multiple molecular pathways in the metastatic process. *BMC Cancer*, **7**(1), 64.

- Chapelle, O., Vapnik, V., Bousquet, O., and Mukherjee, S. (2002). Choosing multiple parameters for support vector machines. *Machine Learning*, **46**, 131–159.
- Chapelle, O., Schölkopf, B., and Zien, A., editors (2006). *Semi-Supervised Learning*. MIT Press, Cambridge, MA.
- Cheng, Y., Cai, Y., Sun, Y., and Li, J. (2008). Semi-supervised feature selection under logistic I-RELIEF framework. In *19th International Conference on Pattern Recognition*.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, **20**(3), 273–297.
- Cristianini, N. and Hahn, M. (2006). *Introduction to Computational Genomics: A Case Studies Approach*. Cambridge University Press.
- Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to support vector machines : and other kernel-based learning methods*. Cambridge University Press, 1 edition.
- Dai, W., Yang, Q., Xue, G., and Yu, Y. (2008). Selft-thought clustering. In *International Conference of Machine Learning (ICML)*, pages 200–207.
- Daumé III, H. and Marcu, D. (2007). Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, **26**, 101–126.
- De Mol, C., Mosci, S., Traskine, M., and Verri, A. (2009). A regularized method for selecting nested groups of relevant genes from microarray data. *Journal of Computational Biology*, **16**(5), 677–690.
- Dettling, M. (2004). Bagboosting for tumor classification with gene expression data. *Bioinformatics*, **20**(18), 3583–3593.
- DGTRE, R., editor (2006). *Rapport d'Activité*.
- Diaz-Uriarte, R. and de Andres, S. A. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, **7**(1), 3.
- Diettrich, T. (2000). Ensemble methods in machine learning. In *Proceedings of the 1st International Workshop on Multiple Classifier Systems*, pages 1–15.
- Do, H., Kalousis, A., and Hilario, M. (2009). Feature weighting using margin and radius based error bound optimization in svms. In *ECML PKDD*

- '09: *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 315–329, Berlin, Heidelberg. Springer-Verlag.
- Dupont, P., Gaulis, S., Helleputte, T. (Inventors), GlaxoSmithKline Biologicals S.A., Dupont, P., Gaulis, S., and Helleputte, T. (Applicants) (2010). Method for classifying a cancer patient as responder or non-responder to immunotherapy. Patent, WO/2010/029174.
- Edgar, R. and Barrett, T. (2006). Ncbi geo standards and services for microarray data. *Nature Biotechnology*, **24**, 1471–1472. <http://www.ncbi.nlm.nih.gov/geo/>.
- Efron, B. (1983). Estimating the error rate of a prediction rule: Some improvements on cross-validation. *Journal of the American Statistical Association*, **78**, 316–331.
- Efron, B. (1986). How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, **81**, 461–470.
- Efron, B. and Tibshirani, R. (1997). Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, **92**(438), 548–560.
- Ein-Dor, L., Kela, I., Getz, G., Givol, D., and Domany, E. (2005). Outcome signature genes in breast cancer: is there a unique set? *Bioinformatics*, **21**(2), 171–178.
- Ein-Dor, L., Zuk, O., and Domany, E. (2006). Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proc Natl Acad Sci U S A*, **103**(15), 5923–5928.
- Evgeniou, T. and Pontil, M. (2004). Regularized multi-task learning. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 109–117.
- Felsenstein, J. (2004). *Inferring Phylogenies*. Sinauer Associates.
- Francois, D. (2007). *High-dimensional data analysis: optimal metrics and feature selection*. Ph.D. thesis, Faculté des Sciences Appliquées, Université catholique de Louvain.
- Franke, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, **3**, 95–110.

- Gentleman, R., Carey, V., Bates, D., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A., Sawitzki, G., Smith, C., Smyth, G., Tierney, L., Yang, J., and Zhang, J. (2004). Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, **5**, R80.
- George, E. and McCulloch, R. (1997). Approaches for bayesian variable selection. *Statistica Sinica*, **7**(2), 339–373.
- Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, **36**(1), 3–42.
- Giard, J. (2010). *Protein Surface Properties using Signal Processing and Statistical Tools*. Ph.D. thesis, Université catholique de Louvain.
- Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., and Lander, E. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.
- Gulgezen, G., Cataltepe, Z., and Yu, L. (2009). Stable and accurate feature selection. In W. Buntine, M. Grobelnik, D. Mladenić, and J. Shawe-Taylor, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 5781, chapter 47, pages 455–468. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Gunkel, M., Erdel, F., Rippe, K., Lemmer, P., Kaufmann, R., Hörmann, C., Amberger, R., and Cremer, C. (2009). Dual color localization microscopy of cellular nanostructures. *Biotechnology Journal*, **4**, 927–938.
- Guyon, I. (2002). Rfe erratum. <http://www.clopinet.com/isabelle/Papers/RFE-erratum.html>.
- Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, **3**, 1157–1182.
- Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, **46**(1-3), 389–422.
- Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L., editors (2006). *Feature Extraction, Foundations and Applications*. Springer.

- Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class machine learning. In *International Conference on Machine Learning (ICML)*, pages 359–366.
- Helleputte, T. and Dupont, P. (2009a). Biomarker selection by transfer learning with linear regularized models. In *Machine Learning in Systems Biology*.
- Helleputte, T. and Dupont, P. (2009b). Feature selection by transfer learning with linear regularized models. In *European Conference on Machine Learning (ECML)*, number 5781 in Lecture Notes in Artificial Intelligence, pages 533–547.
- Helleputte, T. and Dupont, P. (2009c). Partially supervised feature selection with regularized linear models. In *International Conference on Machine Learning (ICML)*.
- Hernández-Lobato, D., Hernández-Lobato, J., Helleputte, T., and Dupont, P. (2010). Expectation propagation for bayesian multi-task feature selection.
- Hinton, G. and Salakhutdinov, R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science*, **313**(5786), 504–507.
- Ho, T. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**, 832–844.
- Hoos, H. and Stützle, T. (2004). *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann / Elsevier.
- Hsieh, C.-J., Chang, K.-W., Lin, C.-J., Keerthi, S., and Sundararajan, S. (2008). A dual coordinate descent method for large-scale linear svm.
- Hsu, C., Chang, C., and Lin, C. (2009). A practical guide to support vector classification. Technical report, National Taiwan University, Department of Computer Science, Taipei 106, Taiwan.
- Huang, J., Smola, A., Gretton, A., Borgwardt, K., and Schölkopf, B. (2007). Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 601–608. MIT Press.
- Huang, J., Ma, S., and Zhang, C. (2008). Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, **18**, 1603–1618.
- Hubbell, E., Liu, W. M., and Mei, R. (2002). Robust estimators for expression analysis. *Bioinformatics*, **18**(12), 1585–1592.

- Huber, P. (1981). *Robust Statistics*. John Wiley & Sons.
- Hunter, L. (2009). *The Processes of Life: An Introduction to Molecular Biology*. MIT Press.
- Hyvärinen, A., Karhunen, E., and Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons.
- Irizarry, R., Bolstad, B. M., Collin, F., Cope, L. M., Hobbs, B., and Speed, T. P. (2003). Summaries of affymetrix genechip probe level data. *Nucleic Acids Res*, **31**(4).
- Joachims, T. (2002). SVM light, <http://svmlight.joachims.org>.
- Jones, N. and Pevzner, P. (2004). *An Introduction to Bioinformatics Algorithms*. The MIT Press.
- Kalousis, A., Prados, J., Sanchez, J.-C., Allard, L., and Hilario, M. (2004). Distilling classification models from cross validation runs: An application to mass spectrometry. *Conference on Tools with Artificial Intelligence, IEEE International*, **0**, 113–119.
- Kalousis, A., Prados, J., and Hilario, M. (2007). Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowl. Inf. Syst.*, **12**(1), 95–116.
- Kessler, A. (2006). *The End of Medicine: How Silicon Valley (And Naked Mice) Will Reboot Your Doctor*. Harper Collins.
- Kifer, D., Ben-David, S., and Gehrke, J. (2004). Detecting change in data streams. In *VLDB*, pages 180–191.
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence (IJCAI)*.
- Kohavi, R. and John, G. (1997). Wrappers for feature selection. *Artificial Intelligence*, **97**(1–2), 273–324.
- Krishnapuram, B., Carin, L., and Hartemink, A. (2004). *Kernel Methods in Computational Biology*, chapter 14: Gene Expression Analysis: Joint Feature Selection and Classifier Design, pages 299–317. MIT Press, Cambridge, MA.

- Kuncheva, L. (2007). A stability index for feature selection. In *Proceedings of the 25th International Multi-Conference: Artificial Intelligence and Applications*, pages 390–395, Anaheim, CA, USA. ACTA Press.
- Lane, N. (2005). *Power, sex, suicide : mitochondria and the meaning of life*. Oxford University Press.
- Lawrence, N. and Platt, C. (2004). Learning to learn with the informative vector machine. In *International Conference on Machine Learning (ICML)*, page 65, New York, NY, USA. ACM.
- Lee, J. and Verleysen, M. (2007). *Information Science and Statistics Series*. Springer.
- Li, C. and Wong, W. (2001). Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *PNAS*, **98**(1), 31–36.
- Liao, X., Xue, Y., and Carin, L. (2005). Logistic regression with an auxiliary data source. In *International Conference on Machine Learning (ICML)*, pages 505–512.
- Liu, H. and Setiono, R. (1995). Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of the Seventh International Conference on Tools with Artificial Intelligence*, pages 388–391.
- Liu, H., Palatucci, M., and Zhang, J. (2009). Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In *International Conference on Machine Learning (ICML)*.
- Long, P. and Vega, V. (2003). Boosting and microarray data. *Machine Learning*, **52**(1-2), 31–44.
- Loscalzo, S., Yu, L., and Ding, C. (2009). Consensus group stable feature selection. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 567–576, New York, NY, USA. ACM.
- Louahed, J., Gaulis, S., Helleputte, T., Dupont, P., Gruselle, O., Spatz, A., Kruit, W., Dreno, B., Lehmann, F., and Brichard, V. (2008). Clinical response to the mage-3 immunotherapeutic in metastatic melanoma patients is associated with a specific gene profile present prior to treatment. In *33th European Society for Medical Oncology (ESMO) Congress*.

- MacKay, D. (1994). Bayesian methods for backpropagation networks. In E. Dornan, J. van Hemmen, and K. Schulten, editors, *Models of Neural Networks III*, chapter 6, pages 211–254. Springer.
- McKinney, E. (1966). Generalized birthday problem. *The American Mathematical Monthly*, **73**(4), 385–387.
- Metzker, M. (2010). Sequencing technologies – the next generation. *Nature Reviews Genetics*, **11**, 31–46.
- Mierswa, I. and Wurst, M. (2005). Efficient case based feature construction. In *European Conference on Machine Learning (ECML)*, volume 3720 of *Lecture Notes in Artificial Intelligence*, pages 641–648. Springer-Verlag.
- Minka, T. (2003 (revised in 2007)). A comparison of numerical optimizers for logistic regression. Technical report, Microsoft Research.
- Mukherjee, S. (2003). *A Practical Approach to Microarray Data Analysis*, chapter 9: Classifying Microarray Data Using Support Vector Machines, pages 166–185. Springer.
- Nadeau, C. and Bengio, Y. (2003). Inference for the generalization error. *Machine Learning*, **52**, 239–281.
- Ng, A. (2004). Feature selection, l_1 vs. l_2 regularization, and rotational invariance. In *International Conference on Machine Learning (ICML)*.
- Niccol, A. (1997). Gattaca. <http://www.imdb.com/title/tt0119177/>.
- Obozinski, G., Taskar, B., and Jordan, M. (2009). Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, pages 1–22.
- Pan, S. and Yang, Q. (2008). A survey on transfer learning. Technical Report HKUST-CS08-08, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China.
- Parkinson, H., Kapushesky, M., Kolesnikov, N., Rustici, G., Shojatalab, M., Abeygunawardena, N., Berube, H., Dylag, M., Emam, I., Farne, A., Holloway, E., Lukk, M., Malone, J., Mani, R., Pilicheva, E., Rayner, T., Rezwan, F., Sharma, A., Williams, E., Bradley, X., Adamusiak, T., Brandizi, M., Burdett, T., Coulson, R., Krestyaninova, M., Kurnosov, P., Maguire, E., Neogi, S., Rocca-Serra, P., Sansone, S., Sklyar, N., Zhao, M., Sarkans, U., and Brazma, A. (2009). ArrayExpress update—from an archive of functional genomics

- experiments to the atlas of gene expression. *Nucl. Acids Res.*, **37**(suppl-1), D868–872.
- Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, **2**(6), 559–572.
- Perneger, T. (1998). What's wrong with bonferroni adjustments. *British Medical Journal*, **316**, 1236–1238.
- Peyré, G., Bogleux, S., and Cohen, L. (2008). Non-local regularization of inverse problems. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 57–68. Springer-Verlag.
- Provost, F. and Fawcett, T. (1997). Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *Proc. of the Third Intl. Conf. on Knowledge Discovery and Data Mining*, pages 43–48.
- Qiu, J. (2006). Epigenetics: Unfinished symphony. *Nature*, **441**, 143–145.
- Quinlan, J. (1990). Induction of decision trees. In J. W. Shavlik and T. G. Dietterich, editors, *Readings in Machine Learning*. Morgan Kaufmann. Originally published in *Machine Learning* 1:81–106, 1986.
- Quinlan, J. (1993). *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann.
- R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
- Roth, V. (2002). The generalized lasso: a wrapper approach to gene selection for microarray data. In *Proceedings of the 14th International Conference on Automated Deduction*, pages 252–255.
- Rudin, L., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal. *Physica D*, **60**, 259–268.
- Saeys, Y., Inza, I., and Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, **23**(19), 2507–2517.
- Saeys, Y., Abeel, T., and Van de Peer, Y. (2008). Robust feature selection using ensemble feature selection techniques. In *European Conference on Machine Learning (ECML)*, pages 313–325.

- Sarle, W. (1997). Neural network faq. <ftp://ftp.sas.com/pub/neural/FAQ.html>. Periodic posting to the Usenet newsgroup comp.ai.neural-nets.
- Schölkopf, B. and Smola, A. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA.
- Seeger, M. (2008). Bayesian inference and optimal design for the sparse linear model. *Journal of Machine Learning Research*, **9**, 759–813.
- Shannon, C. (1949). Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers*, **37**(1), 10–21.
- Sheikh, M., Sarvotham, S., Milenkovic, O., and Baraniuk, R. (2007). Dna array decoding from nonlinear measurements by belief propagation. In *IEEE Statistical Signal Processing Workshop*.
- Shevade, S. and Keerthi, S. (2003). A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, **12**(17), 2246–2253.
- Shipp, M. A., Ross, K. N., Tamayo, P., Weng, A. P., Kutok, J. L., Aguiar, R. C., Gaasenbeek, M., Angelo, M., Reich, M., Pinkus, G. S., Ray, T. S., Koval, M. A., Last, K. W., Norton, A., Lister, T. A., Mesirov, J., Neuberg, D. S., Lander, E. S., Aster, J. C., and Golub, T. R. (2002). Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, **8**(1), 68–74.
- Silver, D. and Bennett, K. (2008). Guest editor’s introduction: special issue on inductive transfer learning. *Machine Learning*, **73**, 215–220.
- Singh, D., Febbo, P., Ross, K., Jackson, D., Manola, J., Ladd, C., Tamayo, P., Renshaw, A., D’Amico, A., Richie, J., Lander, E., Loda, M., Kantoff, P., Golub, T., and Sellers, W. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, **1**(2), 203–209.
- Smets, F. (2006). The cristall project. <http://www.cristall.be/>.
- Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Ser. B*, **36**, 111–147.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, **58**(1), 267–288.

- Tipping, M. (2001). Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, **1**, 211–244.
- Torkkola, K. (2003). Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research*, **3**, 1415–1438.
- Tukey, J. (1977). *Exploratory Data Analysis*. Addison-Wesley.
- van de Vijver, M. J., He, Y. D., van't Veer, L. J., Dai, H., Hart, A. A., Voskuil, D. W., Schreiber, G. J., Peterse, J. L., Roberts, C., Marton, M. J., Parrish, M., Atsma, D., Witteveen, A., Glas, A., Delahaye, L., van der Velde, T., Bartelink, H., Rodenhuis, S., Rutgers, E. T., Friend, S. H., and Bernards, R. (2002). A gene-expression signature as a predictor of survival in breast cancer. *N Engl J Med*, **347**(25), 1999–2009.
- van Rijsbergen, C. (1979). *Information Retrieval*. Butterworths, London, 2 edition.
- van 't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., and Friend, S. H. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**(6871), 530–536.
- Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer, New York.
- Wahba, G. (1980). Spline bases, regularization, and generalized cross-validation for solving approximation problems with large quantities of noisy data. In *Proceedings of the International Conference on Approximation Theory in Honour of George Lorenz*. Austin, TX: Academic Press.
- Wang, L., Zhu, J., and Zou, H. (2007). Hybrid huberized support vector machines for microarray classification. In *International Conference on Machine Learning (ICML)*, pages 983–990.
- Wang, Z., Song, Y., and Zhang, C. (2008). Transferred dimensionality reduction. In *European Conference on Machine Learning (ECML)*, volume 5212 of *Lecture Notes in Artificial Intelligence*, pages 550–565. Springer.
- Watson, J. and Crick, F. (1953). Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. *Nature*, **171**(4356), 737–738.

- Wehenkel, L. (1997). Discretization of continuous attributes for supervised learning: variance evaluation and variance reduction. In *Proceedings of the International Fuzzy Systems Association World Congress*, pages 381–388.
- Weigelt, B., Peterse, J., and van't Veer, L. (2005). Breast cancer metastasis: markers and models. *Nature Reviews Cancer*, **5**(8), 591–602.
- Welsh, J., Sapinoso, L., Su, A., Kern, S., Wang-Rodriguez, J., Moskaluk, C., Frierson, H., and Hampton, G. (2001). Analysis of Gene Expression Identifies Candidate Markers and Pharmacological Targets in Prostate Cancer. *Cancer Research*, **61**(16), 5974–5978.
- Weston, J., Elisseeff, A., Schölkopf, B., and Tipping, M. (2003). Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research*, **3**, 1439–1461.
- Wolpert, D. (1996). The lack of a priori distinctions between learning algorithms. *Neural Computation*, **8**(7), 1341–1390.
- Wu, Z., Irizarry, R., Gentleman, R., Martinez-Murillo, F., and Spencer, F. (2004). A model-based background adjustment for oligonucleotide expression arrays. *Journal of the American Statistical Association*, **99**(468), 909–917.
- Yu, L., Ding, C., and Loscalzo, S. (2008). Stable feature selection via dense feature groups. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 803–811, New York, NY, USA. ACM.
- Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, **68**(1), 49–67.
- Zhao, P. and Yu, B. (2006). On model selection consistency of lasso. *Journal of Machine Learning Research*, **7**, 2541–2563.
- Zhao, Z. and Liu, H. (2007). Semi-supervised feature selection via spectral analysis. In *7th SIAM International Conference on Data Mining*, pages 641–652.
- Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B*, **67**, 301–320.
- Zou, H. and Yuan, M. (2008). The f_∞ -norm support vector machine. *Statistica Sinica*, **18**(1), 379–398.

Index

Symbols

0/1-loss	27, 82
F_∞ -norm Support Vector Machine	168
L_0	33, 62, 163, 166
$L_{1,1}$	163, 165
$L_{1,2}$	163, 165
$L_{1,\infty}$	163, 166
L_1	33, 40, 59, 63, 169
L_1 -AROM	63, 126, 129
L_2	33, 40, 60
L_2 -AROM	64, 129
L_∞	33
χ^2	51
d_{VC}	<i>see</i> VC dimension
AIDS	251
allergen	207
allergy	5, 202
alternative splicing	250
amino acid	247
annotations	5
apoptosis	246
ARD	<i>see</i> automatic relevance determination
area under the curve	75
AROM methods	62, 125, 141
ArrayExpress	143
atopic march	202
ATP	<i>see</i> adenosine triphosphate
ATP synthase	252
AUC	<i>see</i> area under the curve
auto-encoding MLP	49
automatic relevance determination	68

A

absolute value	170
accuracy	7, 10, 13, 29, 72, 81, 193
activity	22
adaptive lasso	63
adenine	247, 278
adenosine diphosphate	245
adenosine triphosphate	245
ADP	<i>see</i> adenosine diphosphate
affinity	278
Affymetrix	4, 97, 207, 271
aggregation	113, 191

B

background adjustment	274
backward elimination	7, 57, 112
bacteria	249
bagging	41, 44
balanced classification rate	13, 74, 86, 176, 193
balanced error rate	74
base pairs	247

Bayesian framework 60
bayesian framework 141, 194
BCR . . . *see* balanced classification rate
Benjamini-Hochberg correction . . . 54
BER *see* balanced error rate
between-tasks stability 162, 176
binary classification 183
binary decision trees 41
Bioconductor 17, 281
bioinformatics 243
biomarker 6, 48, 203, 208
birthday paradox 301
bleed & read 253
Bonferroni correction 54
bootstrap 13, 78
bootstrap .632+ 13
bootstrap sample 45, 66, 80
bound 31, 45
breakdown point 99
breast cancer 9

C

C 17
C++ 17
C4.5 41
cancer 246
CART *see* classification and regression
trees
categorical 42, 55
causality 48
cDNA *see* complementary DNA
cell 244
cell imaging 4
cell line 5
cell wall 244
cellular respiration 246

central dogma 244
centrifugation 253
chemotherapy 10, 81
chromatin 247
chromosomes 244
CLA . *see* complete linear aggregation
class 25
classification 21
classification and regression trees . 41
classification performances 191
classifier 25
clinical annotations 22
clinical efficacy 74
clinical markers 130
clinical outcome 9
clinical setting 8
clinical status 204
clustering 227
CNP *see* copy number variation
codon 247
cohort 204
COL 283
complementary DNA 251
complete linear aggregation 114
complete weighted linear
aggregation 114
complexity 6, 32, 174
compressed sensing 62, 64, 196
computational biology 243
computational cost 48
computational intelligence 5
concave 23
confounding factor . . . 9, 201, 202, 228,
235
confusion matrix 13, 72
consensus 11, 113, 191
consistency 69
continuous variables 51

convergence 64, 173
 convex 23, 28, 37, 62, 164, 174
 convex optimization 36
 convexity 22
 coordinate descent 173
 copy number variation 4, 161, 258
 corrected confidence interval 94
 corrected resampled *t*-test 94, 152
 correlation 60, 236
 correlation-based feature selection 51
 Cristall project 5, 201
 cross-validation 9, 10, 78, 82, 164
 curse of dimensionality, 7, 31
 CWA ... *see* complete weighted linear
 aggregation
 cytoplasm 245
 cytosine 247, 278
 cytoskeleton 244
 cytosol 244

D

dataset 17, 283
 decision trees 41
 decision value 25, 38
 decisions 25
 density estimation 54
 deoxyribonucleic acid 245, 247
 dependence 48, 54
 derivatives 35, 170
 descriptive statistics 212
 diagnosis 5, 6, 8, 202
 diagnosis kit 48, 162
 dimensionality reduction 47
 dimensions 5
 discrete variables 51
 distribution 26

divergence 145
 DLB 283
 DNA *see* deoxyribonucleic acid
 DNA methylation 263
 double-stranded RNA 250
 down-regulated 234
 DRAGS 61
 dropouts 204
 drug 5
 dye-swap 261

E

EBI 143
 elastic net 194
 electrophoresis 3, 253
 embedded methods 51
 empirical error 82
 empirical risk 29, 30, 49
 empty spaces 7, 78
 ensemble feature selection 11, 110
 ensemble methods ... 14, 41, 109, 191,
 202
 ensemble ranking 114
 enzyme 249
 epigenetics 263
 ERM principle 30
 ethics 265
 eukaryotes 244
 evaluation protocol 114
 exons 247
 explanatory concerns 6
 expression 4, 247
 expression profiles 234
 extra trees 46
 extremely randomized trees ... 41, 44

F

false positives 54
FDA *see* Food and Drug
Administration
feature extraction 49, 165
feature selection 5, 7, 47
filters 51
financial cost 48
flow cytometry 203
fluorescent marker 259
fluorescent protein tagging 254
Food and Drug Administration ... 17,
269, 270
forest 45
forward selection 7, 57
forward-backward selection 57
functional risk 29, 30, 49, 82

G

GC-RMA 10, 207, 278
gene 9, 22, 48, 247
gene expression .. 4, 6, 8, 22, 161, 203,
259, 271
Gene Expression Omnibus ... 11, 143,
280
gene profiling 6
Genechip 4, 207, 271
generalization 7, 9, 30
generalized bootstrap .632+ 13, 81, 85
generalized lasso 40, 60, 192
generative process 26, 69
genetic algorithms 57
genome 258
genomic business 258
genomic data 70

GEO .. *see* Gene Expression Omnibus
GFP *see* green fluorescent protein
Gini Index 42
global optimum 62
Golub's ratio 51, 197
Google 259
gradient 60
gradient descent 6, 63, 127
greedy strategy 42
green fluorescent protein 255
group lasso 61
group-lasso 167, 195, 279
Group-Lasso-Based Summarization
279
group-regularizer 12, 165, 192
guanine 247, 278

H

hard-margin SVM 33
heatmaps 226, 234
hemoglobin 203, 231, 233
Hessian 6
heuristic 38
heuristics 57
HGU133A 280
HGU133Plus2.0 207, 271
HGU95A 149, 271, 286
HGU95Av2 149, 286
high dimensional 5, 6
high-density microarrays 4
high-throughput technology 4
hinge loss 27, 37, 41, 64, 164, 166
histones 245
HIV 251
Hu6000 286
Hu6800 286

human genome 253, 258
 hybrid huberized SVM..... 194
 hybridization..... 247, 273
 hydrogen bond 247
 hyperplane 25, 33
 hypothesis 25
 hypothesis testing..... 53

I

i.i.d. 26
 ICA *see* independent component analysis
 ID3 41
 immune system..... 202
 impurity 42
 independent component analysis.. 49
 indicator function 30
 inductive bias..... 30
 inductive transfer 143
 information theory 54
 informed selection 101
 instability 67
 inter quartile range 100, 101
 interpretability 6, 26, 48
 interpretation 9, 69
 introns 247
 IQR normalization 99
 Iteratively Reweighted Least Square 6

J

Jaccard index 72, 176
 Joachims heuristic..... 38
 joint methods 57

K

k-fold cross-validation 79
 k-NN *see* nearest-neighbors
 kernel density estimation 61
 kernel learning..... 168
 kernel methods 33
 Kuncheva index..... 71, 116

L

labels 26
 Lagrange multipliers..... 35
 Laplace distribution..... 60
 Laplace prior 141
 Laplacian prior 60
 lasso 60, 163
 LASSUM *see* group-lasso-based summarization
 learning process..... 26, 30, 42
 leave-one-out 80
 LEU 284
 linear model..... 8, 25, 31, 49, 191
 linear SVM..... 33, 58, 64, 111, 129
 local search 57
 logistic function..... 39
 logistic loss... 27, 39, 41, 164, 168, 192
 logistic regression . 12, 38, 40, 163, 167
 logistic ridge regression . 40, 177, 179, 184
 logit function 38
 loss & regularizer pattern . 32, 35, 37, 40, 58, 126, 164
 loss function..... 27, 32
 low-density microarrays 4
 LYM..... 284
 LYM1..... 284

M

machine learning 5, 64
MammaPrint 270
margin 34, 59
markers 70
MAS *see* Microarray Analysis
 Software
mass spectrometry 253
matrix inversions 6
matrix-norm 192
maximum margin 33, 59
MDS ... *see* multidimensional scaling
mean 99
median 99
medicine 71
melanoma 5
membrane 244
messenger RNA 247, 250
metabolism 22, 70
micro-RNA 215, 250
microarray ... 4, 6, 8, 22, 203, 259, 271
Microarray Analysis Software 10, 275
minimax problem 34
miRNA *see* micro-RNA
mismatch probe 273
mitochondria 245
mitochondrial DNA 245, 249
mixed-norm 162, 165
MLP *see* multi-layer-perceptron
molecular biology 3, 71, 243
mRNA *see* messenger RNA
mtDNA *see* mitochondrial DNA
multi-class 12, 183, 192
multi-layer-perceptron 36, 49
multi-task feature selection 162
multi-task learning 12, 161, 192
multidimensional scaling 49
multiple kernel learning 168

multiple transfer 144, 148
multiple-test 54
multivariate 42, 50
mutual information 54

N

N-Resamplings 93, 115, 149
national insurances 5
NCBI 143
ncRNA *see* non-coding RNA
Nearest Neighbors 86
nearest neighbors 54
negative accuracy 13
negative predicted value 74
negative transfer 144
Newton method 6, 169
next-generation sequencing . 3, 4, 264
NGS . *see* next-generation sequencing
no free lunch theorem 269
non-coding RNA 250
non-specific binding 273
non-specific filtering 52, 101
non-specific filtering evaluation ... 13
normal distribution 100
normalization 97, 275
normalization bias 9, 99, 116
NP-Hard 63
NPV ... *see* negative predicted value
nuclear pore 234, 245
nucleolus 245
nucleotides 3, 247
nucleus 244

O

objective function 32
 odds 38
 offset 25
 one-against-one 183
 one-against-the-rest 183
 optimality conditions 35, 169
 optimization 6, 163
 organelles 245
 out-of-bag sample 67, 80
 outlier 100, 202, 207, 208
 overfitting 8, 13, 30, 46, 80, 195
 oxygen 233

P

p-value 53
 partial supervision 125
 Partially Supervised AROM 126
 partially supervised feature selection
 11, 125, 202
 Parzen window 61
 patents 14
 pathology 9
 pathways 50, 61, 70
 patients 9, 201
 PCA*see* principal component analysis
 Pearson correlation 236
 penalized logistic regression 60
 perfect match probe 273
 performance 78
 permutation 84
 personalized medicine 5
 pharmaceutical companies 5
 phylogeny 246
 polynomial model 32
 pooling 143, 161, 192
 positive accuracy 13

positive predicted value 13, 74, 86
 PPV *see* positive predicted value
 pre-filtering technique 193
 pre-mRNA 250
 precision 74
 prediction 5
 preliminaries 21
 principal component analysis 49, 208
 prior knowledge 11, 71, 125, 141, 191
 prior relevance 126
 PRO 285
 PRO1 285
 PRO2 286
 PRO3 286
 probabilistic outputs 38
 probability density function 100
 probability distribution 54
 probe 10, 259, 273
 probeset 10, 22, 207, 273
 prognosis 5, 6, 8, 203, 206
 prokaryotes 244
 prostate cancer 148
 protein 3, 251
 protein docking 255
 protein knockdown 254
 protein production 251
 protein structure 252
 proteomics 253
 pruning 43
 PS- L_1 -AROM 126
 PS- L_2 -AROM 130, 146, 192, 215

Q

quantile normalization 277
 quartile 100

R

R language 17, 281
 radius-margin bound 65
 random forests 41, 44, 66
 random permutation 94
 random subspace method 45
 ranking 51
 recall 73
 recovery 59, 69
 recursive feature elimination .. 58, 65,
 112, 129
 regression 26, 168
 regularization 32, 49, 162
 regularization constant 32, 37
 regularized learning 49
 regularized risk 30
 regularizer 32
 regulated 70
 relative overfitting 84
 relevance vector machine 68
 relevance vectors 68
 resampling 78, 93
 response prediction 5
 retro-viruses 251
 reverse transcriptase 251
 reweighted- L_1 63
 RFE . *see* recursive feature elimination
 ribonucleases 280
 ribonucleic acid 207, 245, 249
 ribosomal RNA 250
 ribosome 245
 RMA *see* Robust Multi-array Average
 RNA *see* ribonucleic acid
 RNA interference 250, 254
 RNA Polymerases 245
 RNAi *see* RNA interference
 Robust Multi-array Average .. 10, 207,
 277

robust normalization 101
 ROC curve 74, 116
 rRNA *see* ribosomal RNA
 RVM ... *see* relevance vector machine

S

screening 260
 selection bias .. 9, 50, 99, 116, 131, 226
 semi-supervised learning 125, 143
 sensitivity 73, 86
 separation 8
 sequencing 253, 258
 Shannon Sampling 62
 signal processing 64
 signal to noise 52
 signature 6, 7, 9, 48, 70
 simulated annealing 57
 single nucleotide polymorphism ... 4,
 161, 258
 single transfer 148
 single-stranded RNA 250
 single-task learning 163, 192
 siRNA *see* small interfering RNA
 slack variables 37
 small interfering RNA 250
 small- n -big- p 7, 51, 194
 snip *see* single nucleotide
 polymorphism
 SNP *see* single nucleotide
 polymorphism
 SNPedia 258, 259
 soft-margin SVM 37
 solver 12, 61, 163, 169
 source domain 143
 sparse 48
 sparse learning 62

sparse regularization 60
sparsity 36, 62, 70, 75, 176, 192
sparsity-enforcing regularizer 59
sparsity-enforcing regularizers ... 162
specific filters 51
specific selection 101
specificity 73, 86
spectrometry 4, 161
spike-and-slab 141, 194
square loss 27, 60, 164, 168
stability ... 9, 69, 70, 113, 176, 191, 202
standard deviation 99
standardization 98
statistical inference 5
statistical learning theory ... 26, 33, 49
stimulation 207
strictly convex 23, 28, 37, 62
strongly convex 24
sub-gradient 169, 170
subsamplings 9, 93
summarization .. 10, 97, 194, 207, 274,
275
summarization bias 281
supervised learning 26
support 59, 162
support vector machine 68
support vector machines 33, 40
support vectors 34, 36, 68
SVM *see* support vector machine
SVM dual 35
SVM primal 35

T

t-test 51, 197
target domain 143
target value 22

thymine 247, 250, 278
TNR *see* true negative rate
total variational norm 60, 196
TPR *see* true positive rate
train-test setting 78
training samples 8
training set 9, 26
transcription 4
transcription factors 251
transcriptome 4, 264, 280
transductive transfer 144
transfer learning 11, 143, 192
transfer RNA 250
treatment 5
tree 41, 66
tRNA *see* transfer RNA
true negative rate 73
true positive rate 73
tumor 10
TV-norm ... *see* total variational norm
two-stages procedure 67

U

ultra-high sequencing 264
unbalanced 10
univariate 42, 50
unsupervised learning 49, 227
unsupervised transfer 144
up-regulated 234
uracil 250

V

validation loop 6, 78

Vapnik-Chervonenkis dimension . *see*
 VC dimension

VC dimension 31, 49, 145

viruses 251

visualization 47, 202, 208

voting system 43

W

warm start 166

Welch statistic 53

within-task stability 176

wrappers 51

X

x-ray crystallography 253

Z

Z-Score 98

zero-pseudo-norm 126, 166