Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi

Statistical learning methods for high dimensional genomic data

by

Salvatore Masecchia

Theses Series **DIBRIS-TH-2013-04**

DIBRIS, Università di Genova
Via Opera Pia, 13 16145 Genova, Italy http://www.dibris.unige.it/

Università degli Studi di Genova

Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi

Dottorato di Ricerca in Informatica

Ph.D. Thesis in Computer Science

Statistical learning methods for high dimensional genomic data

by

Salvatore Masecchia

July, 2013

Dottorato di Ricerca in Informatica
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università degli Studi di Genova

DIBRIS, Università di Genova
Via Opera Pia, 13
I-16145 Genova, Italy
http://www.dibris.unige.it/

Ph.D. Thesis in Computer Science (S.S.D. INF/01)

Submitted by Salvatore Masecchia
DIBRIS, Università di Genova
salvatore.masecchia@unige.it

Date of submission: February 2013

Title: Statistical learning methods for high dimensional genomic data

Advisor: Alessandro Verri
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università di Genova
alessandro.verri@unige.it

Ext. Reviewers:
Barbara Di Camillo
Dipartimento di Ingegneria dell'Informazione
Università di Padova
barbara.dicamillo@dei.unipd.it

Nicola Ancona
Istituto di Studi sui Sistemi Intelligenti per l'Automazione
Consiglio Nazionale delle Ricerche
ancona@ba.issia.cnr.it

Abstract

Due to their high-dimensionality, -omics technologies require the development of computational methods that are able to work with large number of variables. Each data type is characterized by its method of measurement and by the biological aspect under study. Understanding the data properties allows the design of sophisticated and effective computational models that are able to uncover and explain complex biological phenomena.

This thesis aims at exploring the use of statistical learning methods for dealing with different high-throughput molecular data, in order to answer heterogeneous biological questions related to various diseases. We address problems at different biological levels (e.g. gene expression or genomic alteration) but exploiting different peculiarities of the data under analysis.

We propose a computational framework in which biological questions can be modeled as solution of a minimization problem of a functional where data properties are described via a composition of penalties and constraints. This framework includes a wide range of regularized least squares and regularized matrix factorization methods.

We focus on two main questions. First, we apply the $\ell_1\ell_2$-norms regularization to extract gene signatures from gene expression data related to neurodegenerative diseases like Alzheimer and Parkinson. Such feature selection method is nested in a pipeline where functionally related pathways are extracted from the list of relevant genes. The last step of the pipeline, moreover, plans to infer interaction network related to each pathways from the data in order to evaluate differences between different phenotypes (*e.g.* patients *vs* controls).

Second, dealing with aCGH data, in the context of Dictionary Learning, we combine a set of penalties (e.g. ℓ_1-norm and Total Variation) and hard constraints in order to automatically detect common genomic alterations from a set of high risk Neuroblastoma patients. Genomic alterations identified by the regularized method are used as input of an algorithm for oncogenesis tree estimation.

Finally, we present a set of well structured software modules, tools and libraries that implement the above methods and models.

Contents

Chapter 1

Introduction

1.1 Motivations

This thesis is set in the context of Statistical Learning applied to Computational Biology. The *learning-from-examples* paradigm was originally employed to tackle prediction problems (Vapnik, 1998). A machine is trained, in order to perform a given task, on a set of *input-output* pairs. In this context, *training* means to infer the function which best describe the map from *inputs* to *outputs*. When this *black-box* was been applied to genomic data, it showed its inadequacy. Here, the final goal is to understand what is inside the *black-box* and understand what are the mechanisms that are behind the prediction (Golub et al., 1999).

Regularization Theory (Girosi et al., 1995; Evgeniou et al., 2002), gives the tools to answer this question. In a context where *inputs* live in a high-dimensional space (*e.g.* expression of thousands of genes), proper penalization functions used into the learning model may enforce regularization by inducing sparsity, that is to select what variables of the *input* samples (*e.g.* what genes) are responsible of the mapping to the *outputs*.

More explicitly, in computational biology we often would like to find a small subset of predictors or *biomarkers* that exhibit the strongest correlation with the biological phenomenon under study. Appropriate feature selection techniques can produce a model that is interpretable and has possibly lower prediction error than the full model.

It is clear that the emphasis is on finding the best way to understand the biological phenomena behind them. Dictionary Learning (Elad and Aharon, 2006) methods are an evolution of the classical *prediction-oriented* approaches which look for the best data representation. The assumption is that data may be conveniently represented by a linear combination of *few* elements from a given dictionary of basic *atoms*. The goal is to learn

such dictionary directly by the given data, also using appropriate regularizing penalties (for handling sparsity and complexity).

In this thesis we studied, designed and implemented statistical learning methods for high dimensional genomic data, with a major interest in regularized linear models. The interest is on methods that incorporate (possibly in the penalty term) prior biological knowledge (*e.g.* public databases) and that are able to deal with different data types.

1.2 Contributions

In this thesis we present different approaches for the analysis of different genomic data. Our aim is to propose complete approaches centered around a given biological question, analyzing properties, drawbacks and lacks of our methods.

The main contribution of this thesis is a novel statistical method for the analysis of array-based Comparative Genomic Hybridization (aCGH) data, namely CGHDL (Masecchia et al., 2013b,a). CGHDL is a dictionary learning method, based on the minimization of a functional combining several penalties that explicitly exploit the structured nature of aCGH signals. Each atom contains a meaningful *common pattern* of genomic alterations and the resulting model provides a biologically sound representation of aCGH data. Our main goal is to obtain atoms that possibly capture co-occurrences of Copy Number Variations (CNVs) across samples, leading to results that are more easily interpretable by the domain experts. We demonstrate the effectiveness of our method also designing standard machine learning experiments, like classification and clustering.

Secondly, we present two pipelines for study cancer development (oncogenesis) (Masecchia et al., 2012a,b). The first pipeline is a combination of *off-the-shelf* and custom methods and tools which lead to a hierarchical organization of genetic events, that are gains or losses of specific chromosomes segments, extracted from aCGH data. In this context we were not satisfied by the interpretability of the results and started to study different approaches to that particular problem. Exploiting CGHDL properties we were able to improve the approach, obtaining more reliable models of cancer progression.

Finally, we present a pipeline for gene profiling which works on gene expression data and aims at return a set of relevant genes for the disease under study. The core statistical learning method of this pipeline had already been extensively studied in our research group (De Mol et al., 2009b; Barla et al., 2008). Our main contributions, here, were the implementation and development of well-structured software libraries which allow us to easily execute the experiments, get the results and evaluate their statistical significance. Our aim is to make our methodologies reproducible distributing our tools as Open Source software libraries.

The method is nested into a complete pipeline that, from data preprocessing to model assessment and evaluation, tries to give reliable answers, in particular for neurodegenerative diseases like Parkinson's and Alzheimer's (Barla et al., 2010; Squillario et al., 2010, 2011, 2012; Barla et al., 2011a, 2012).

1.3 Structure of the thesis

This thesis is organized as follows:

- In **Chapter 2** we introduce the problem of gene profiling describing how feature selection can accomplish the task. We describe the context of the analysis of molecular data presenting preprocessing steps of our pipeline and underlining major complications and drawbacks. Then we concentrate our attention on the statistical learning method and on the model assessment framework which have an important role for the reliability of the results. In this chapter we also present all the adopted tools and methodologies which complete the analysis pipeline and present a variation where prior knowledge is injected before the statistical learning phase. Finally, we also analyze sources of variability when different methods are used in different steps of the process.

- **Chapter 3** is dedicated to our main theoretic and algorithmic contribution. We present a novel Dictionary Learning method for the identification of genomic aberrations. We first present the biological context highlighting the importance to have reliable and interpretable results. The proposed method is described with the aim to explain how we incorporate prior knowledge defining a regularized minimization problem. In this chapter we also present a novel synthetic data model which help us to evaluate our results.

- In **Chapter 4** we present our pipelines for oncogenesis. We describe the problem and the biological context trying to explain the intrinsic difficulties of the task. A standard and well-known tree model of inference is briefly described highlighting its properties and limitations. Than we present the two pipelines: the first follows a standard methodology adopted in literature to solve such problem, while the novel second approach exploits the model we presented in the Chapter 3.

- **Chapter 5** contains an introduction of the tools we developed that are behind all the results presented in the central three chapters. For each software library we presents the context of applicability also given some coding examples.

- **Chapter 6** is dedicated to conclusions and future developments. We summarized the problems approached in this thesis and highlighted future directions of the work.

Chapter 2

Feature selection for gene profiling

In the context of feature selection we focus on regularized methods based on the combination of ℓ_1 and ℓ_2 norms applied to the problem of the gene expression profiling.

In this chapter we first introduce, in Section 2.1, the general view, summarizing the biological questions we aim to answer and the data under analysis, highlighting their properties and issues (Section 2.2).

In Section 2.3 we introduce our approach and in Section 2.4 we describe the methods of our pipeline which we extend to the case of functional groups identification.

In the last Section 2.5 we report some experimental results and discuss our findings. We also integrate our work discussing the variability of this kind of biological analysis pipeline.

2.1 Biological context: molecular data

Biological data have undergone a radical transformation since the 1980s, being finally able to measure the activity of biological systems at their molecular level.

Genomics (Dubitzky et al., 2007) can be broadly defined as the systematic study of genes, their functions and their interactions. Analogously, proteomics is the study of proteins, protein complexes, their localisation, their interactions, and post-translational modifications. Some years ago, genomics and proteomics studies focused on one gene or one protein at a time. With the advent of high-throughput technologies in biology and biotechnology, this has changed dramatically. We are currently witnessing a paradigm shift from a traditionally hypothesis-driven to a data-driven research. The activity and interaction of thousands of genes and proteins can now be measured si-

multaneously. Technologies for genome and proteome-wide investigations have led to new insights into mechanisms of living systems. There is a broad consensus that these technologies will revolutionize the study of complex human diseases such as Alzheimer's disease, HIV, and particularly cancer. With its ability to describe the clinical and histopathological phenotypes of cancer at the molecular level, gene expression profiling based on microarrays holds the promise of a patient-tailored therapy. Recent advances in high-throughput mass spectrometry allow the profiling of proteome patterns in bio-fluids such as blood and urine, and complement the genomic portray of diseases. Despite the undoubted impact that these technologies have made on bio-medical research, there is still a long way to go from bench to bedside.

In order to extract the information from the data and properly transform it in usable knowledge, it is necessary to develop adequately complex methods. Indeed, a large plethora of studies using high-throughput technologies has been published until now, providing the research community with a wealth of potentially valuable gene expression data in biology and medicine. These studies essentially consist in the application of several biomolecular and statistical techniques aiming at examining the expression of many genes at the same time, allowing the identification of the most significant ones and, ultimately, of the altered pathways underlying the biological question of interest. In the last decade (Dupuy and Simon, 2007), microarrays have been adopted to enlighten the complex biology of cancer and they have been applied to several areas such as genetic screening, safety assessment and diagnostics. In this field, the use of microarrays has expanded exponentially during the past few years but repeatability of published microarray studies is apparently limited (Ioannidis, 2005; Ioannidis et al., 2009). As all the high-throughput techniques, the open problem in microarray studies is how to find robust, reproducible and reliable results. Usually, the identification of the relevant genes and pathways (i.e. a *gene signature*) is achieved by analyzing the gene expression profiles from a set of samples, homogenous with respect to a criterion that is strictly related to the biological question of interest. From the statistical viewpoint, selecting the most suitable method can be an issue because of the large number of available approaches. To date, there is no consensus method for statistical analysis, and thus array data are processed in a variety of different ways.

2.2 Processing molecular data

Every microarray experiment produces images. Image analysis software reduces these images to raw intensity data. To be useful for a data analyst this raw intensity data need to be converted into measures. Pre-processing is used to describe these procedures. Unfortunately, many users of microarrays treat low-level analysis as a "black box", using whatever software is supplied by their system vendor, without much idea of what is

really being done with their data. Microarray experiments are usually conducted to answer one or more questions of biological interest, for instance topics such as determining gene function, discriminating between cases and controls or tumor sub-classes studying the cell cycle and pathway analysis. Typically, low-level analysis methodologies do not attempt to answer these questions. Instead, the primary goal of a low-level analysis of a microarray data experiment is to provide better measures which can be used in higher-level analysis. Ideally, such values should be both precise (low variance) and accurate (low bias) (Bolstad, 2004).

The experiments described in the remainder of this chapter were all performed starting from raw DNA microarray data. We performed data normalization on the raw data with the robust multi-array average *rma* algorithm of the R Bioconductor *affy* package. The *rma* algorithm (Irizarry et al., 2003) is used to create an expression matrix from Affymetrix data. The raw intensity values are background corrected, log2 transformed and then quantile normalized. Next a linear model is fit to the normalized data to obtain an expression measure for each probe set on each array.

2.3 Feature selection for relevant gene sets identification

In a context where typically one has to deal with few samples in a high dimensional space, the definition of statistical models based on a sparse subset of the input measurements (*biomarkers*) is one of the most research interest in the recent past (Hilario and Kalousis, 2008; Saeys et al., 2007). Appropriate feature selection techniques can produce a model that is interpretable and has possibly lower prediction error than the full model (Figure 2.1).

The abundance of techniques has led to the publication of some interesting reviews about feature selection with the intent of classifying the different state-of-the-art techniques following some specific criteria. The most referred one (Guyon and Elisseeff, 2003) organizes feature selection techniques in three main categories: filter methods,

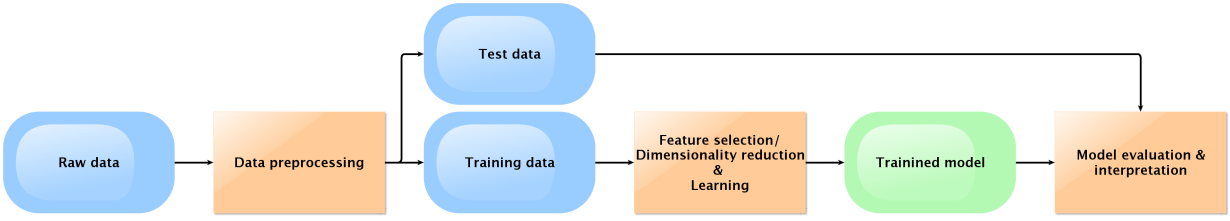
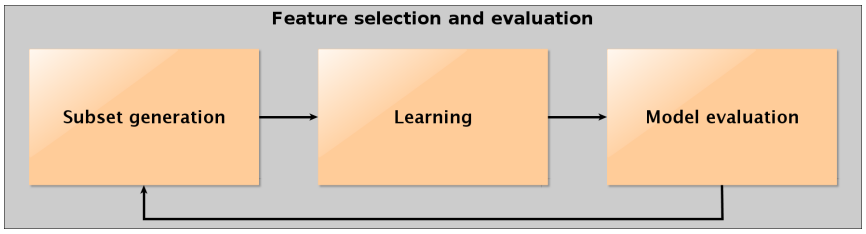
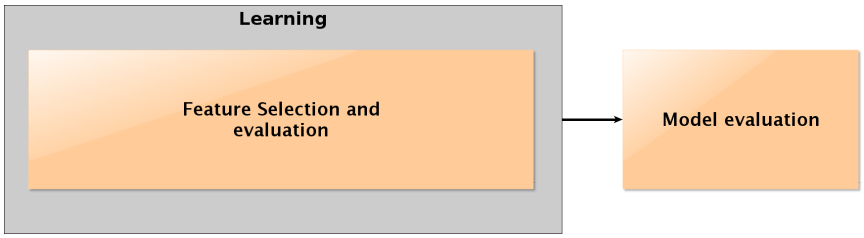


Figure 2.1: A standard learning pipeline. The output is the trained model and all the information one can extract from its interpretation in a biological context.

(a) Filter methods



(b) Wrapper methods



(c) Embedded methods

Figure 2.2: Feature selection techniques schemes.

wrapper methods and embedded methods.

Filter methods assess the relevance of features by looking only at the intrinsic properties of the data, usually with an iterative procedure that calculates a relevance score and removes the last ranked over. Standard feature selection techniques that are widely used in bioinformatics are statistical univariate tests. Those approaches are usually very fast and scale easily to very high dimensional data but they usually ignore the feature dependencies, even if some multivariate filter methods were also introduced in order to overcome this main disadvantage (see Saeys et al., 2007, for a complete review).

In *wrapper methods* the evaluation of a specific subset of features is obtained by training and testing a specific classification model by wrapping a chosen features ranking algorithm around the classification model. Wrapper methods take into account feature dependency with despite the great risk of overfitting (Guyon and Elisseeff, 2003).

Finally the *embedded methods*, include techniques where an optimal subset of features is built into the classifier construction such as *regularized methods* which have been applied in computational biology on different data type such as genomics, epigenetics and

proteomics data (see Ma and Huang, 2008, and reference therein).

When dealing with high-throughput data, the choice of a consistent selection algorithm is not sufficient to guarantee good results. It is therefore essential to introduce a robust methodology (Ancona et al., 2006) to select the significant variables not susceptible of selection bias (Ambroise and McLachlan, 2002).

The following results are based on a feature selection framework (Barla et al., 2008), namely $\ell_1\ell_2 fs$, implementing an unbiased framework for gene expression analysis. The gene selection relies on a regularized least square model based on the $\ell_1\ell_2$ regularization (De Mol et al., 2009a,b) and is presented in Section 2.3.1. The assessment of the statistical significance of the model is performed via cross validation and is presented in Section 2.3.2.

2.3.1 Model selection

In the field of computational biology, one consolidated idea is that a good algorithm for gene signature[1] extraction should take into account at least the linear interactions of multiple genes. Standard statistical univariate methods take into consideration one gene at the time, and then rank them according to their fold-change or to their prediction power. Since most diseases are multi-factorial, a multivariate model is usually preferable.

Another drawback of many statistically-based gene selection algorithms is the rejection of part of the relevant genes due to redundancy. In many biological studies, some of the input variables may be highly correlated with each other. As a consequence, when one specific variable is considered relevant to the problem, its correlated variables should be considered relevant as well.

Given the above premises, we focus on the $\ell_1\ell_2$ (or *elastic net*) feature selection method solved by an iterative soft-thresholding algorithm (De Mol et al., 2009a), introduced by De Mol et al. (2009b) and further studied by Mosci et al. (2010) as proximal algorithm. We are given a data matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$, a column vector of labels $\mathbf{Y} \in \mathbb{R}^{n\times 1}$ and an unknown model vector $\boldsymbol{\beta} \in \mathbb{R}^{p\times 1}$. Our algorithm consists of two stages. In the **first stage** we obtain a model $\hat{\boldsymbol{\beta}}$ with minimal cardinality and small bias by coupling two optimization regularized procedures based, respectively, on $\ell_1\ell_2$ and ℓ_2 penalties. In the

[1] In this section, for convenience and clarity we use the term "gene". We are considering "virtual" data matrices when each variable is associated to one gene. In a real context, on a standard microarray platform we have to consider probes and/or probesets as variables.

first optimization, we perform a minimal gene set selection by minimizing

$$\tilde{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \frac{1}{n} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\| + \mu \|\boldsymbol{\beta}\|_2^2 + \tau \|\boldsymbol{\beta}\|_1 \right\}, \tag{2.1}$$

where τ and μ are two positive regularization parameters. In the second step we rely on regularized least squares by minimizing

$$\overline{\boldsymbol{\beta}} = \underset{\tilde{\boldsymbol{\beta}}}{\operatorname{argmin}} \left\{ \frac{1}{n} \|\tilde{\mathbf{Y}} - \tilde{\mathbf{X}}\tilde{\boldsymbol{\beta}}\| + \lambda \|\tilde{\boldsymbol{\beta}}\|_2^2 \right\}, \tag{2.2}$$

where $\tilde{\mathbf{X}}$ and $\tilde{\boldsymbol{\beta}}$ represent the data matrix and the vector model restricted to the genes selected by the first procedure (2.1), and λ is a positive regularization parameter. The resulting model $\hat{\boldsymbol{\beta}}$ is reconstructed extending $\overline{\boldsymbol{\beta}}$ to the data dimensionality p and filling the positions related to non-selected genes with 0. The parameter selection was performed using a cross validation schema with fixed and small value of μ (close to 0, only with the aim of increase the stability), large values of τ (minimum genes selection) and small values of λ (minimal regularization to overcome the $\ell_1\ell_2$ over-shrinkage drawback). In this cross validated step we search the best pair (τ^*, λ^*).

In the second stage we gradually increase the model cardinality by exploiting the grouping effect of the $\ell_1\ell_2$ regularizer with running the two optimization procedures for increasing values of μ and fixed (τ^*, λ^*). The resulting families of models $\boldsymbol{\beta}$ yield almost nested lists of relevant genes of gradually increasing cardinality.

In Algorithm 2.1 we report the scheme of the algorithm.

2.3.2 Model assessment

In order to obtain an unbiased estimate (Ambroise and McLachlan, 2002) of the classification performances this step must be carefully designed by holding out a blind test set. Since the available samples are usually few compared to the number of variables, this step has to be performed on different sub-samplings and its results averaged to guarantee statistical robustness (see Algorithm 2.2).

The gene selection step is nested in an external cross validation loop, needed to verify the goodness of the estimated model both in terms of performance stability and significance. The training and testing sets required by the **model selection** step (see Algorithm 2.1) come from an external *B-fold* cross validation loop. Each internal model selection loop returns a complete family of almost nested lists of genes, each one associated with a cross validation error. For each increasing correlation value (μ_1 to μ_{m-1}) we have a list of gene sets and we can compute an average cross-validation error. A final

Algorithm 2.1 NESTED-LISTS: Extraction of a family of almost nested gene signatures

Require: $(\mathbf{X}, \mathbf{Y})$ training set, $(\mathbf{X}^{test}, \mathbf{Y}^{test})$ test set
$\{(\mathbf{X}_1, \mathbf{Y}_1), \ldots, (\mathbf{X}_k, \mathbf{Y}_k)\}$ partition of $(\mathbf{X}, \mathbf{Y})$ $\mu_0 < \mu_1 < \cdots < \mu_{m-1}$

Stage I
$\mu \leftarrow \mu_0$, $(\tau_t, \lambda_l)_{t \in \mathcal{T}, l \in \mathcal{L}}$ a grid in parameter space
for $t \in \mathcal{T}, l \in \mathcal{L}$ **do**
  for $i = 1 \rightarrow k$ **do**
    $\mathbf{X}_i^{tr} \leftarrow \mathbf{X}_1, \ldots, \mathbf{X}_{i-1}, \mathbf{X}_{i+1}, \ldots, \mathbf{X}_k$
    $\mathbf{Y}_i^{tr} \leftarrow \mathbf{Y}_1, \ldots, \mathbf{Y}_{i-1}, \mathbf{Y}_{i+1}, \ldots, \mathbf{Y}_k$
    $\boldsymbol{\beta}(t, l, i) \leftarrow$ DOUBLE-OPTIMIZATION$(\mathbf{X}_i^{tr}, \mathbf{Y}_i^{tr}, \tau_t, \lambda_l, \mu_0)$
    $Err(t, l, i) \leftarrow$ error made by $\boldsymbol{\beta}(t, l, i)$ on $(\mathbf{X}_i, \mathbf{Y}_i)$
  end for
  $\overline{Err}(t, l) \leftarrow \frac{1}{k} \sum_{i=1}^{k} Err(t, l, i)$
end for

Stage II
$(t^*, l^*) \leftarrow \text{argmin}_{t \in \mathcal{T}, l \in \mathcal{L}} \{\overline{Err}(t, l)\}$
$(\tau^*, \lambda^*) \leftarrow (\tau_{t^*}, \lambda_{l^*})$
for $i = 0 \rightarrow m - 1$ **do**
  $\boldsymbol{\beta}^*(i) \leftarrow$ DOUBLE-OPTIMIZATION$(\mathbf{X}, \mathbf{Y}, \tau^*, \lambda^*, \mu_i)$
  $Err^{test}(i) \leftarrow$ error made by $\boldsymbol{\beta}^*(i)$ on $(\mathbf{X}^{test}, \mathbf{Y}^{test})$
end for

list of genes for each value of μ is calculated by merging all the B lists corresponding to the external cross validation folds. Such merging algorithm counts the occurrences of each selected gene in the lists and keeps only the most frequent ones (up to a user defined threshold), discarding outliers and providing a more robust and reliable gene signature.

Algorithm 2.2 Gene signature model assessment (cross validation)

Require: $(\mathbf{X}, \mathbf{Y})$ data set, $\{(\mathbf{X}_1, \mathbf{Y}_1), \ldots, (\mathbf{X}_b, \mathbf{Y}_b)\}$ partition of $(\mathbf{X}, \mathbf{Y})$
$\mu_0 < \mu_1 < \cdots < \mu_{m-1}$, $(\tau_t, \lambda_l)_{t \in \mathcal{T}, l \in \mathcal{L}}$

Cross validated nested lists
for $i = 1 \to b$ **do**
$\mathbf{X}_i^{tr}, \mathbf{X}_i^{ts} \leftarrow (\mathbf{X}_1, \ldots, \mathbf{X}_{i-1}, \mathbf{X}_{i+1}, \ldots, \mathbf{X}_b), \mathbf{X}_i$
$\mathbf{Y}_i^{tr}, \mathbf{Y}_i^{ts} \leftarrow (\mathbf{Y}_1, \ldots, \mathbf{Y}_{i-1}, \mathbf{Y}_{i+1}, \ldots, \mathbf{Y}_b), \mathbf{Y}_i$
$\boldsymbol{\beta}^*(i, \mu_0^{m-1}), Err(i, \mu_0^{m-1}) \leftarrow$ NESTED-LISTS$(\mathbf{X}_i^{tr}, \mathbf{Y}_i^{tr}, \mathbf{X}_i^{ts}, \mathbf{Y}_i^{ts}, \mu_0^{m-1}, (\tau_t, \lambda_l)_{t \in \mathcal{T}, l \in \mathcal{L}})$
end for

Resulting stable lists
for $i = 0 \to m - 1$ **do**
$\boldsymbol{\beta}^*(i) \leftarrow$ MERGE-LISTS$(\boldsymbol{\beta}^*(b, \mu_i), \ldots, \boldsymbol{\beta}^*(b, \mu_i))$
$\overline{Err}(i) \leftarrow \frac{1}{b} \sum_{j=1}^{b} Err(j, \mu_i)$
end for

2.4 Functional characterization of gene sets

Functional genomics attempts to make use of the vast wealth of data produced by genomic projects to describe gene (and protein) functions and interactions. Functional characterization of gene sets usually derives a list of relevant pathways from sets of discriminant features, moving the focus of the analysis from single genes to functionally related pathways. The promise of functional genomics is to expand and synthesize genomic and proteomic knowledge into an understanding of the dynamic properties of an organism at cellular and/or organismal levels. This would provide a more complete picture of how biological function arises from the information encoded in an organism's genome.

2.4.1 A posteriori approach: enrichment methods

Pathway enrichment methods are widely used in bioinformatic analysis, for example to assess the relevance of biomarker lists (like gene sets), or as a first step in network analysis. The idea is to identify functional modules of genes that cooperate for one common functional role. Starting with the list of discriminating genes, we would like to generate a list of functional gene groups.

Retrieving all the useful information from the literature, as well as to manipulate the interrelated data, is a complex and time consuming task. As outlined in the review

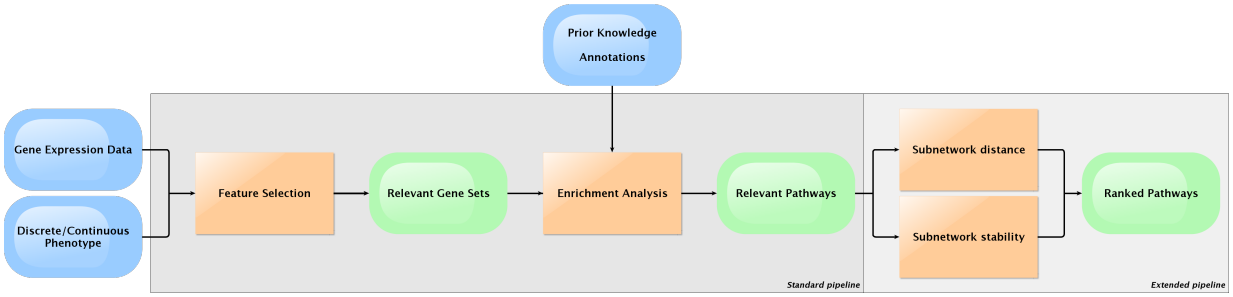


Figure 2.3: Standard analysis pipeline (Section 2.4.1) with gene sets selection and a posteriori functional enrichment. See Section 2.5.2 for experimental results. Such pipeline could be extended with a further network inference analysis (Section 2.4.3). See Section 2.5.4 for experimental results and sources of variability evaluation.

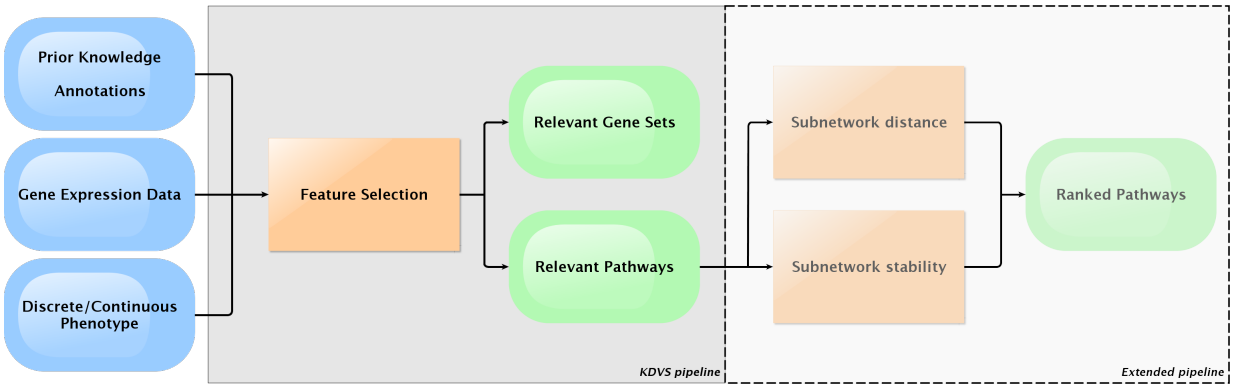


Figure 2.4: Knowledge driven analysis pipeline (KDVS, Section 2.4.2) with gene sets selection and a priori functional enrichment. See Section 2.5.2 for experimental results. Such a pipeline could be extended with a further network inference analysis (sections 2.4.3) but in this thesis we do not treat this particular extension.

by Huang et al. (2009), in the last 10 years the gene-annotation enrichment analysis field has been growing rapidly and several bioinformatics tools have been designed for this task. Huang et al. (2009) provide a unique categorization of these enrichment tools in three major categories based on the underlying algorithm: singular enrichment analysis (SEA), gene set enrichment analysis (GSEA), and modular enrichment analysis (MEA). The goal is to perform a functional characterization of the generated gene set that produces a list of functional gene groups, using biological domain knowledge, such as the Gene Ontology (GO, Ashburner et al. (2000)) or the Kyoto Encyclopedia of Genes and Genomes (KEGG, Kanehisa and Goto (2000)).

GO is a database of controlled vocabularies (ontologies) that describes gene products in terms of their associated domains, that are *biological process* (BP), *cellular component* (CC) and *molecular function* (MF), in a species independent manner. GO is structured as a

directed acyclic graph where each term has a defined relationship to one or more terms in the same domain and sometimes to other domains. The most common visual representation of GO is a graph where the relations among the terms (*nodes*) are represented by connecting lines (*arcs*).

KEGG is a repository that stores the higher-order systemic behaviors of the cell and the organism from genomic and molecular information. It is an integrated database resource consisting of 16 main databases, broadly categorized into *systems information*, *genomic information*, and *chemical information*. All the available KEGG pathways have been biologically validated before publishing.

One of the on-line and free-to-use tools for functional analysis of gene sets is `WebGestalt` (Zhang et al., 2005)[2]. `WebGestalt` belongs to the first category proposed by Huang et al. (2009) (SEA) and takes as input a list of relevant genes/probesets. The enrichment analysis is performed in KEGG and GO identifying the most relevant pathways and ontologies that can be associated with the gene in the given gene set. `WebGestalt` adopts the hypergeometric test to evaluate functional category enrichment and performs a multiple test adjustment, the default method being the one from Benjamini and Hochberg (1995). The user may choose different significance levels and the minimum number of genes belonging to the selected functional groups.

In Section 2.5, for evaluating all sources of variability in gene expression analysis, we also compared `WebGestalt` with two methods belonging to the other two categories. `GSEA` (Subramanian et al., 2005) is the representative of the second class. It first performs a correlation analysis between the features and the phenotype by obtaining a ranked list of features. Second it determines whether the members of given gene sets are randomly distributed in the ranked list of features obtained above, or primarily found at the top or bottom. Finally, the tool in the MEA class is the Pathways and Literature Strainer (`PaLS`) (Alibés et al., 2008), which takes a list or a set of lists of genes (or protein identifiers), and shows which ones share the same GO terms or KEGG pathways, following a criterion based on a threshold t set by the user. The tool provides as output those functional groups that are shared at least by the $t\%$ of the selected genes. `PaLS` is aimed at easing the biological interpretation of results from studies of differential expression and gene selection, without assigning any statistical significance to the final output.

[2] `http://bioinfo.vanderbilt.edu/webgestalt/`

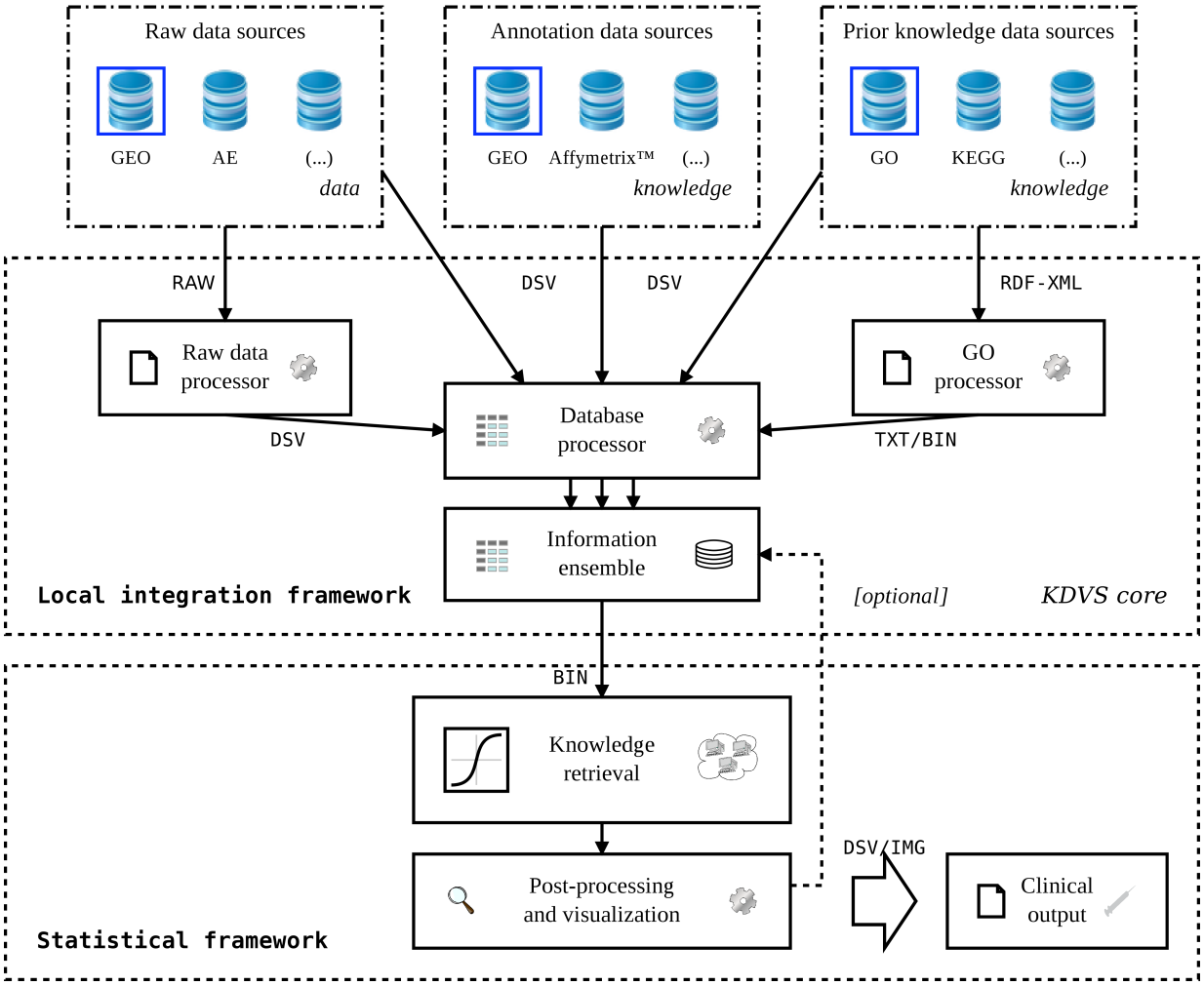


Figure 2.5: Schema of the structure of the KDVS pipeline (Zycinski, 2012). It is composed by two parts named Local Integration Framework and Statistical Framework. The inputs to the first part are represented by the integration of data with the information about the data and with the prior knowledge source. The second part receive as input data matrices built by the first one. Successively the results are post-processed and eventually visualized. DSV and TXT/BIN indicate the files formats: DSV is Delimiter separated values and TXT/BIN is text/binary. GEO and AE are the two most used sources of microarray data: Gene Expression Omnibus and ArrayExpress. GO and KEGG are the most used sources of prior knowledge: Gene Ontology and the Kyoto Encyclopedia of Genes and Genomes.

2.4.2 A priori approach: knowledge driven selection (KDVS)

In this section we introduce a different approach for identifying relevant pathways from Gene expression data, namely KDVS (Knowledge Driven Variable Selection, Zycinski et al. (2011, 2013)).

The general schema of KDVS is presented in Figure 2.5. It consists of a local integration framework that performs an integration of microarray platform data and annotations (Edgar et al., 2002) with prior biological knowledge. Outside of this framework, the raw data are pre-processed for normalization and summary, with state of the art algorithms for microarray technologies (Gentleman et al., 2005). The result of the local integration framework is a dynamically created information ensemble, where for every GO term, the corresponding set of probesets is collected, the expression values are extracted across all samples and considered for the classification/feature selection task performed by $\ell_1\ell_2 fs$. Therefore, the original $p \times n$ gene expression data matrix serves as a template for generation of submatrices $ps \times n$, where $ps < p$. In turn, each submatrix is analyzed by $\ell_1\ell_2 fs$, within the knowledge retrieval phase. We used Gene Ontology (GO) as the source of prior knowledge, and we focused especially on molecular function domain. The final output of KDVS consists of the list of discriminant GO terms, identified with classification, as well as the list of selected genes counted across discriminant terms, identified with feature selection[3].

2.4.3 Functional gene sets selection with interaction network inference

After the identification of relevant pathways from gene sets, the ultimate goal of the analysis is to identify and rank such pathways reflecting major changes between two conditions (*e.g.* healthy and non-healthy), or during a disease evolution. The reconstruction of molecular pathways from high-throughput data is then based on the theory of complex networks (*e.g.* Strogatz (2001); Newman (2003); Boccaletti et al. (2006); Newman (2010); Buchanan et al. (2010)) and, in particular, in the reconstruction algorithms for inferring networks topology and wiring from data (He et al., 2009).

For each pathway, networks are reconstructed (inferred) separately on data from the different classes (phenotypes). The subnetwork inference phase requires to reconstruct a network $N_{p_i,y}$ on the pathway p_i by using the steady state expression data of the samples of each class y. The network inference procedure is limited to the sole genes belonging to the pathway p_i in order to avoid the problem of intrinsic underdeterminacy of the task.

[3]For more details see (Zycinski, 2012).

Relying on a properly defined distance measure, we evaluate networks corresponding to the same pathway for different classes, *i.e.* all the pairs $(N_{p_i,+1}, N_{p_i,-1})$ and rank the pathways themselves from the most to the least changing across classes. We attached to each network a quantitative measure of stability with respect to data subsampling, in order to evaluate the reliability of inferred topologies.

We adopt four different subnetwork reconstruction algorithms: the Weighted Gene Co-Expression Networks algorithm (WGCNA) (Horvath, 2011), the Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE) (Margolin et al., 2006), the Context Likelihood of Relatedness (`CLR`) approach (Faith et al., 2007), and the Reverse Engineering Gene Networks using Artificial Neural Networks (RegnANN) (Grimaldi et al., 2011).

WGCNA is based on the idea of using (a function of) the absolute correlation between the expression of a couple of genes across the samples to define a link between them. ARACNE is a method for inferring networks from the transcription level (Margolin et al., 2006) to the metabolic level (Nemenman et al., 2007). Beside it was originally designed for handling the complexity of regulatory networks in mammalian cells, it is able to address a wider range of network deconvolution problems. This information-theoretic algorithm removes the vast majority of indirect candidate interactions inferred by co-expression methods, by using the data processing inequality property (Cover and Thomas, 1991). CLR belongs to the relevance networks class of algorithms, and it is employed for the identification of transcriptional regulatory interactions (Faith et al., 2007). In particular, interactions between transcription factors and gene targets are scored by using the mutual information between the corresponding gene expression levels, coupled with an adaptive background correction step. The most probable regulator-target interactions are chosen by comparing the mutual information score versus the "background" distribution of mutual information scores for all possible pairs within the corresponding network context (*i.e.* all the pairs including either the regulator or the target). RegnANN is a newly defined method for inferring gene regulatory networks based on an ensemble of feed-forward multilayer perceptrons. Correlation is used to define gene interactions. For each gene a one-to-many regressor is trained using the transcription data to learn the relationship between the gene and all the other genes of the network. The interaction among genes are estimated independently and the overall network is obtained by joining all the neighborhoods.

Subnetwork distance. Despite its common use even in biological contexts (Sharan and Ideker, 2006), the problem of quantitatively comparing networks (*e.g.*, using a metric instead of evaluating network properties) is a still an open issue in many scientific disciplines. The central problem is of course which network metrics should be used to evaluate stability, whether focusing on local changes or global structural changes.

As discussed in Jurman et al. (2011), the classic distances in the edit family focus only on the portions of the network interested by the differences in the presence/absence of matching links and quantitatively evaluate the differences between two networks (with the same number of nodes) in terms of minimum number of edit operations (with possibly different costs) transforming one network into the other, *i.e.* deletion and insertion of links. Spectral distances - based on the list of eigenvalues of the Laplacian matrix of the underlying graph - are instead particularly effective for studying global structures. Within them, we considered the Ipsen-Mikhailov ϵ distance: originally introduced by Ipsen and Mikhailov (2002) as a tool for network reconstruction from its Laplacian spectrum, it has been proven to be the most robust in a wide range of situations by Jurman et al. (2011). We are also aware that spectral measures are not flawless: they cannot distinguish isomorphic or isospectral graphs, which can correspond to quite different conditions within the biological context. In (Jurman et al., 2012), both approaches are improved by proposing a *glocal* distance ϕ as a possible solution against both issues: ϕ is defined as the product metric of the Hamming distance H (as representative of the edit-family) and the ϵ distance. Full mathematical details are available in (Jurman et al., 2012).

Subnetwork stability. For each $N_{p_i,y}$, we extracted a random subsampling (of a fraction r of $\mathbf{X}$ labelled as y) on which the corresponding $N_{p_i,y}$ will be reconstructed. Repeating m times the subsampling/inferring procedure, a set of m nets is generated for each $N_{p_i,y}$. Then all mutual $\binom{m}{2} = \frac{m(m-1)}{2}$ distances are computed, and for each set of m graphs we build the corresponding distance histogram. Usually, our choices are $m = 20$ and $r = \frac{2}{3}$. Mean and variance of the constructed histograms quantitatively assess the stability of the subnetwork inferred from the whole dataset: the lower the values, the higher the stability in terms of robustness to data perturbation (subsampling).

2.5 Experiments and results

The identification of the genes involved in a particular disease is one of the most challenging questions in the field of the computational biology. In the context of neurodegenerative disorders, like Parkinson's and Alzheimer's, the issue is even more crucial, given their unknown etiology at molecular level. Shedding light on the molecular mechanisms of such complex diseases could be the first step towards the development of reliable early diagnostic tools.

In this section we show two different approaches for gene signature and pathways profiling: one with the standard *a posteriori* enrichment analysis (Section 2.5.2) and one with *a priori* injection of biological validated knowledge (Section 2.5.3). In the last subsection

(2.5.4) we evaluate with an *ad hoc* designed experiments the sources of variability across a standard pipeline for gene profiling completed with network inference methods.

2.5.1 Datasets description

In this section different public datasets are used with the approaches described in this chapter. We are particularly interested in the analysis of neurodegenerative diseases like Parkinson's (PD) and Alzheimer's (AD), for their high impact in the current society.

Datasets come from different laboratories and are measured on different Affymetrix microarray platforms. Data were downloaded from the Gene Expression Omnibus (GEO[4]) as raw files and normalized before the downstream analysis with our methods.

GSE6613 refers to early-stage PD and contains whole blood expression data from 50 patients with PD, 33 with neurodegenerative diseases other than PD, and 22 healthy controls.

GSE20295 refers to multiple brain regions in PD. The dataset is a super-series of data referring to other 3 GEO datasets and contains 93 samples of tissues extracted from prefrontal area 9 (**GSE20168**, 29 samples), putamen (**GSE20291**, 35 samples) and whole substantia nigra (**GSE20292**, 29 samples).

GSE9770 refers to non-demented individuals with intermediate Alzheimer's neuropathologies. The dataset contains 34 samples collected on 6 different neuronal regions: entorhinal cortex, hippocampus, middle temporal gyrus, posterior cingulate, superior frontal gyrus and primary visual cortex.

GSE5281 refers to 87 AD samples and 74 controls. Individuals were stratified with respect to diagnostic groups, age groups, and Apolipoprotein E genotype. The dataset contains 150 individual brain tissues from 6 different regions: entorhinal cortex, hippocampus, middle temporal gyrus, posterior cingulate, superior frontal gyrus and primary visual cortex.

In Table 2.1 we summarize the main information about the data. We refer to the following section for details about each particular use of the datasets.

2.5.2 Gene selection and functional enrichment

In (Squillario et al., 2010) we studied **PD** on the **GSE6613** dataset considering 50 patients with PD and 55 controls or patients with neurodegenerative diseases other than PD. The

[4] `http://www.ncbi.nlm.nih.gov/geo/`

GEO ID	Reference	Disease	Platform
GSE6613	Scherzer et al. (2007)	PD (early)	Affymetrix U133A
GSE20295	Zhang et al. (2005)	PD (late)	Affymetrix U133A
GSE9770	BioProject PRJNA103705[5]	AD (early)	Affymetrix U133 Plus 2.0
GSE5281	Liang et al. (2007, 2008)	AD (late)	Affymetrix U133 Plus 2.0

Table 2.1: Gene expression datasets adopted in the experiments.

last two groups of people were considered as belonging to one class that harbors those individuals not affected by PD.

The feature selection analysis resulted in a 64% prediction accuracy, associated to a signature composed by 378 selected probesets for the maximum correlation allowed. The functional analysis shows that many of the genes are involved in pathways related to the immune system, to some particular cell basic processes (apoptosis, cell cycle, motility, communication) but also to the nervous system, to the metabolism and to some diseases/infections. The reliability of the statistical method is confirmed by the signature, that includes genes known to be involved in PD (like SNCA, PRDX2, RNF11) and genes that are known to be implicated in other neurodegenerative diseases (like BCL2, CASP1). The signature is also intersecting with the one by Scherzer et al. (2007) (10 overlapping genes on 22). The functional characterization of the signature confirmed that the majority of the genes are included in categories already known to be affected by PD (metabolism, cell related pathways as signaling and apoptosis, the nervous and the immune system related pathways).

Table 2.2 shows that most of the genes in the signature are involved in pathways related to systems of the human organism, specially involving the immune (*i.e.* hematopoietic cell lineage, leukocyte transendothelial migration), the endocrine (insulin signaling pathway, GnRH signaling pathway) and the nervous systems (*i.e.* long-term potentiation and long-term depression). The other enriched pathways are comprehended in two KEGG categories that are the environmental information processing and the human diseases. In particular the genes involved in the pathways of the first category encode for signaling molecules that are responsible for the transduction of the signals (*i.e.* MAPK or Wnt signaling pathways), also through the interactions with other proteins (*i.e.* cytokine-cytokine receptor interaction, cell adhesion molecules), while the genes included in the second category participate in the development of diseases or infections (*i.e.* colorectal cancer, type I diabetes mellitus, neurodegenerative disorders, epithelial cell signaling in Helicobacter pylori infection). The last enriched categories of pathways are cellular processes, genetic information processing and metabolism: the first category includes those pathway related to the cell communication system (*i.e.* gap junction and focal adhesion) and to the cell growth and dead (*i.e.* apoptosis, cell cycle) while the second and the third categories harbor one pathway each. These pathways

are named respectively ribosome and arachidonic acid metabolism.

We also analyzed our signature with respect to GO and we found that the *cellular components* tree shows that most part of the proteins encoded by the genes selected with $\ell_1\ell_2 fs$ framework, are located in the cytoplasm: some proteins constitute the proteasome complex (*i.e.* PSMD14, PSMF1 and PSMD10), other functions are vacuoles or lysosomes (*i.e.* USP4, LAMP2, IGF2R) others are located in the outer membrane of the mitochondrion (*i.e.* TOMM20, RAF1, BCL2) and others constitute the large subunit of the ribosome (*i.e.* RPL18A, RPL4, RPL7). The *molecular function* tree shows that most part of our proteins bind to nucleic acids, both DNA and RNA, or to nucleotides, specially purines that are needed to make ATP and GTP molecules. Other proteins bind to cytokines (*i.e.* IL6ST,IL8RA, CCR3), to growth factors (*i.e.* IL1R2, IL6R, IL7R) and to unfolded proteins (HSPD1, HSP90AB1, CCT2). The *biological process* tree shows that, among the enriched nodes, those that contain the higher number of genes are:

- *cellular processes* that is subsequently divided in nodes that contains genes involved in cell proliferation or cell death, a process the latter that is positively and negatively regulated and can be apoptosis or other processes to lead to cell death;
- *physiological processes* that is divided in metabolism that includes several type of it; there are the cellular, the primary and the macromolecule metabolism;
- *response to stimulus* that includes response to defense, to chemical stimulus, to stress and to biotic stimulus; the last two nodes subsequently are combined in one node that is named response to unfolded protein (that includes all the most important heat shock proteins, that are the 60, 70 and 90kDa). This node recalls the most known and important cause of PD.

KEGG Pathway	E	R	p-value
MAPK signaling pathway	2.37	4.65	$2.97 \cdot 10^{-5}$
Hematopoletic cell lineage	0.75	13.38	$4.12 \cdot 10^{-9}$
Cytokine-cytokine receptor interaction	2.15	4.18	$3.43 \cdot 10^{-4}$
Insulin signaling pathway	1.16	5.19	$1.12 \cdot 10^{-3}$
Leukocyte transendothelial migration	1	6.02	$5.12 \cdot 10^{-4}$
Ribosome	0.85	7.02	$2.23 \cdot 10^{-4}$
Antigen processing and presentation	0.64	9.37	$4.50 \cdot 10^{-5}$
Cell adhesion molecules (CAMs)	1.1	5.44	$8.75 \cdot 10^{-4}$
Gap junction	0.76	6.61	$9.89 \cdot 10^{-4}$
Colorectal cancer	0.71	7.03	$7.51 \cdot 10^{-4}$
Type I diabetes mellitus	0.36	14.05	$2.75 \cdot 10^{-5}$
GnRH signaling pathway	0.84	5.98	$1.55 \cdot 10^{-3}$
Long-term potentiation	0.59	8.52	$3.09 \cdot 10^{-4}$
Focal adhesion	1.69	2.96	$2.79 \cdot 10^{-2}$
Wnt signaling pathway	1.27	3.93	$9.24 \cdot 10^{-3}$
Apoptosis	0.72	6.94	$7.95 \cdot 10^{-4}$
Jak-STAT signaling pathway	1.27	3.14	$3.93 \cdot 10^{-2}$
Cell cycle	0.99	4.05	$1.75 \cdot 10^{-2}$
Regulation of actin cytoskeleton	1.75	2.28	$9.98 \cdot 10^{-2}$
Adipocytokine signaling pathway	0.61	6.52	$3.37 \cdot 10^{-3}$
Fc epsilonRI signaling pathway	0.66	4.56	$2.84 \cdot 10^{-2}$
Arachidonic acid metabilism	0.49	6.13	$1.30 \cdot 10^{-2}$
Purine metabolism	1.3	2.31	$1.42 \cdot 10^{-1}$
Long-term depression	0.68	4.44	$3.04 \cdot 10^{-2}$
Chronic myeloid leukemia	0.64	4.68	$2.65 \cdot 10^{-2}$
PPAR signaling pathway	0.6	5.03	$2.20 \cdot 10^{-2}$
Calcium signaling pathway	1.48	2.03	$1.84 \cdot 10^{-1}$
Epithelial cell signaling in Helicobacter pyiori infection	0.59	5.11	$2.11 \cdot 10^{-2}$
T cell receptor signaling pathway	0.8	3.75	$4.65 \cdot 10^{-2}$
Neurodegenerative disorders	0.3	9.92	$3.40 \cdot 10^{-3}$
Neuroactive ligand-receptor interaction	2.55	1.18	$4.71 \cdot 10^{-1}$

Table 2.2: Most of the genes of our signature are involved in pathways related to systems of the human organism, specially involving the immune, the endocrine and the nervous systems. The other enriched pathways are comprehended in two KEGG categories that are the environmental information processing and the human diseases (in *italics*). (E) expected number in the category; (R) ratio of enrichment, when R > 0 the pathway is enriched.

In (Barla et al., 2010), we studied **AD** on the **GSE5281** dataset measured on a set of diseased and control samples from 6 different regions of the brain (Table 2.3). This gives us the opportunity of performing the analysis on the entire dataset and to focus also on different regions, to verify their involvement in the disease.

After the selection phase, we characterized the list of relevant probesets with a functional analysis, leveraging on `WebGestalt`. The final assessment on the selected variables is done by a thorough search on the available literature, to confirm their biological soundness. The biological characterization was based on: a) Gene Ontology using the hypergeometric test with level of significance $p \leq 0.05$ and minimum number of genes equal to 2; b) Entrez Gene where for each selected gene we query the engine and look for prior knowledge (Entrez[6] & PubMed[7]).

The analysis on the complete dataset (Table 2.4 (a)) was perfomed with our $\ell_1\ell_2 fs$ framework with an external 5-fold cross-validation obtaining a 5% classification error, selecting 47 probesets. The majority of the identified 47 probesets corresponds to genes whose functions are still unknown. Nevertheless, the prior knowledge on the remaining well characterized genes show that they are expressed in the brain and/or they are already known to be involved in AD. Exploiting the prior knowledge retrieved from PubMed we found that SST is the somatostatin hormone that is known to affect the rates of neurotransmission in the central nervous system (CNS) and of proliferation in both normal and tumorigenic cells; MINK1 is known to be up-regulated during postnatal mouse cerebral development and because of the great genomic and functional similarity between human and mouse, we can suggest that it could preserve the same function also in the human organism; CXCR4, like all the other chemokines, could be implicated in AD, beside being implicated in other brain diseases; GFAP encodes one of the major intermediate filament proteins of mature astrocytes and it is known that there is a marked increase of this protein in AD patients in comparison with healthy controls; CTSB is involved in the proteolytic processing of amyloid precursor protein (APP) and the in-

[6] `http://www.ncbi.nlm.nih.gov/sites/gquery`
[7] `http://www.ncbi.nlm.nih.gov/pubmed`

Brain Region	Patients	Controls
Hippocampus	10	13
Entorhinal Cortex	10	13
Medial Temporal Gyrus	16	12
Posterior Cingulate	9	13
Superior Frontal Gyrus	23	11
Primary Visual Cortex	19	12

Table 2.3: Patients and controls samples in GSE5281 dataset for different brain areas

complete processing of APP is known to be a causative factor of AD; PENK is a neuropeptide hormone; UBE3A is known to take part to the ubiquitin protein degradation system and since the poor clearance specially of amyloid-beta is a certified characteristic of AD, we suggest that this protein could be involved in AD.

The analysis restricted to the hippocampal region (Table 2.4 (b)) was performed with our $\ell_1\ell_2 fs$ framework with an external LOO-cross-validation obtaining a classification error $\leq 1\%$, selecting 10 probesets. From the GO tree, we note that three proteins, LMO4, SPG7, YWHAH are involved in the nervous system development and, exploiting prior knowledge retrieved from PubMed, six proteins (SPG7, NAV1, SCAMP1, LMO4, WASL, YWHAH) are known to be expressed in the brain or to be involved in other brain diseases: SPG7 codes for a mitochondrial metalloprotease protein and, as member of the AAA protein family, it has an ATP domain that has several roles including protein folding and proteolysis. Since AD is know to be characterized by deposits of amyloid protein, it is feasible that this SPG7 could have a role in the folding or the degradation of this protein. NAV1 belongs to the neuron navigator family and it is predominantly expressed in the nervous system, it is similar to a C. elegans gene that is involved in the axon guidance. It is supposed to play a role in neuronal development and regeneration. SCAMP1, that is a secretory carrier membrane proteins, is included in a list of novel susceptibility genes for AD identified in the hippocampus region (Potkin et al., 2009). The function of LMO4 is still unknown, but it could behave like a transcriptional regulator or as an oncogene; furthermore it protects neurons from ischemic brain injury, its overexpression interferes with neuritic outgrowth. WASL belongs to the Wiskott-Aldrich syndrome (WAS) family. It shows highest expression in neural tissues and it is known to be an important molecular signal for regulating spines and synapses. In particular it interacts with HSP90, known to act as a regulator of pathogenic changes that lead to the neurodegenerative phenotype in AD. YWHAH is involved in the neurotrophin signaling pathway.

2.5.3 Knowledge driven selection and functional characterization

In (Squillario et al., 2011) we aimed at identifying gene signatures specific for the early and late stages of AD and PD. The underlying idea is that the concerted analysis of AD and PD, that share common characteristics at least in nervous and immune systems, might improve the identification of all those genes whose deregulation is common to both the diseases or is specific to only one. Both the *a priori* and *a posteriori* functional analyses in GO aim at revealing the most significant functions and process associated to the selected genes.

We analyzed all 4 collected datasets: 2 for AD early and late stages (**GSE9770**, **GSE5281**) and 2 PD early and late stages (**GSE6613**, **GSE20295**). The analysis of the datasets with

(a) Complete Dataset

Affymetrix probeset ID	Gene Symbol
213921_at	SST
...	...
214246_x_at	MINK1
217028_at	CXCR4
...	...
203540_at	GFAP
213274_s_at	CTSB
213791_at	PENK
214980_at	UBE3A

(b) Hippocampus only

Affymetrix probeset ID	Gene Symbol
201020_at	YWHAH
205809_s_at	WASL
209204_at	LMO4
212417_at	SCAMP1
...	...
224771_at	NAV1
...	...
20629_s_at	SPG7

Table 2.4: Complete dataset (left): the majority of the identified 47 probesets corresponds to genes whose functions are still unknown. Nevertheless, the prior knowledge on the remaining well characterized genes, show that they are expressed in the brain and/or they are already known to be involved in AD. Hippocampus only (right): From the GO tree, we note that the proteins are involved in the nervous system and/or are known to be expressed in the brain or to be involved in other brain diseases.

Dataset	# of Genes	Classification accuracy of $\ell_1\ell_2 fs$
PD early stage	73	62%
PD late stage	90	80%
AD early stage	132	90%
AD late stage	106	95%

Table 2.5: Gene signature found analyzing PD and AD datasets for early and late stages

the standard *a posteriori* approach leads to 4 gene signatures specific for the early and late stages of AD and PD detailed reported in Table 2.5.

The intersection between the signatures resulted in four common genes among the two stages of each disease: XIST, RPS4Y1, DEFA1/DEFA3, HLA-DQB1 for PD while for AD HBB, PMS2L1/PMS2L2, SCAMP1, XIST. The only common gene between these two lists is XIST that was recently found, together with RPS4Y1, to be expressed in a subset of neurons as part of a group of gender-specific genes differentially expressed in dorsolateral prefrontal cortex, anterior cingulate cortex and cerebellum. The intersection between the gene signatures of the early stages did not give any result while that one among the gene signatures of late stages resulted in 5 probesets corresponding to 4 known genes that are TAC1, HBB, SST, CD44. The gene set enrichment analysis performed *a posteriori* by WebGestalt in GO, identified relevant enriched nodes for each

signature both *specific* and partially *overlapping* among the two stages within the same disease.

For PD the *overlapping* nodes are few and with a very general meaning (*e.g.* intracellular, cytoplasm, negative regulation of biological process). The specific ones for the early stage concern the immune system, the response to stimulus (*i.e.* stress, chemicals or other organism like virus), the regulation of metabolic processes, the biological quality and the cell death. The *specific* nodes for the late stage are related to the nervous system (*e.g.* neurotransmitter transport, transmission of nerve impulse, learning or memory) and to the response to stimuli (*e.g.* behavior, temperature, organic substances, drugs or endogenous stimuli).

For AD the *overlapping* nodes are few but some of these are more specific and concern the circulatory system, in particular the blood circulation and the neurological system processes that is bounded to the transmission of nerve impulse node. The *specific* nodes for the late stage are mostly related to negative regulation of several processes (*e.g.* of cell proliferation, of cell communication, of macromolecule biosynthetic process) but also to neurological system process (connected to the last enriched node named visual perception) the behavior and the response to stimuli (behavior, drug, hormone).

We were able to identify functionally significant and statistically robust signatures for the early and late stages of both AD and PD also with the *a priori* analysis where results for PD are reported in Figure 2.6 and Table 2.6. Both the *a posteriori* and *a priori* functional analysis allowed us to characterize the signatures in GO in order to identifies the process and functions shared by the diseases or specific of only one disease. The *a priori* functional analysis showed an advantage in the rapid visualization and therefore comprehension of the results by the use of the specific structure of GO. The identification in all the signatures of genes already known to be involved in their respective diseases, confirmed the reliability of the method in identifying relevant genes and increased the likelihood that the remaining genes could be involved in specific stages of the diseases.

(a) Both

GO ID	Description
GO:0003824	Catalytic activity
GO:0016787	Hydrolase activity
GO:0003676	Nucleic acid binding
GO:0016740	Transferase activity
GO:0030528	Transcription regulator activity

(b) Early

GO ID	Description
GO:0016209	Antioxidant activity
GO:0004672	Protein kinase activity
GO:0016301	Kinase activity

(c) Late

GO ID	Description
GO:0008289	Lipid binding
GO:0005198	Structural molecule activity
GO:0004518	Nuclease activity
GO:0003682	Chromatin binding
GO:0005509	Calcium ion binding

Table 2.6: Relevant GO nodes from *a priori* functional analysis

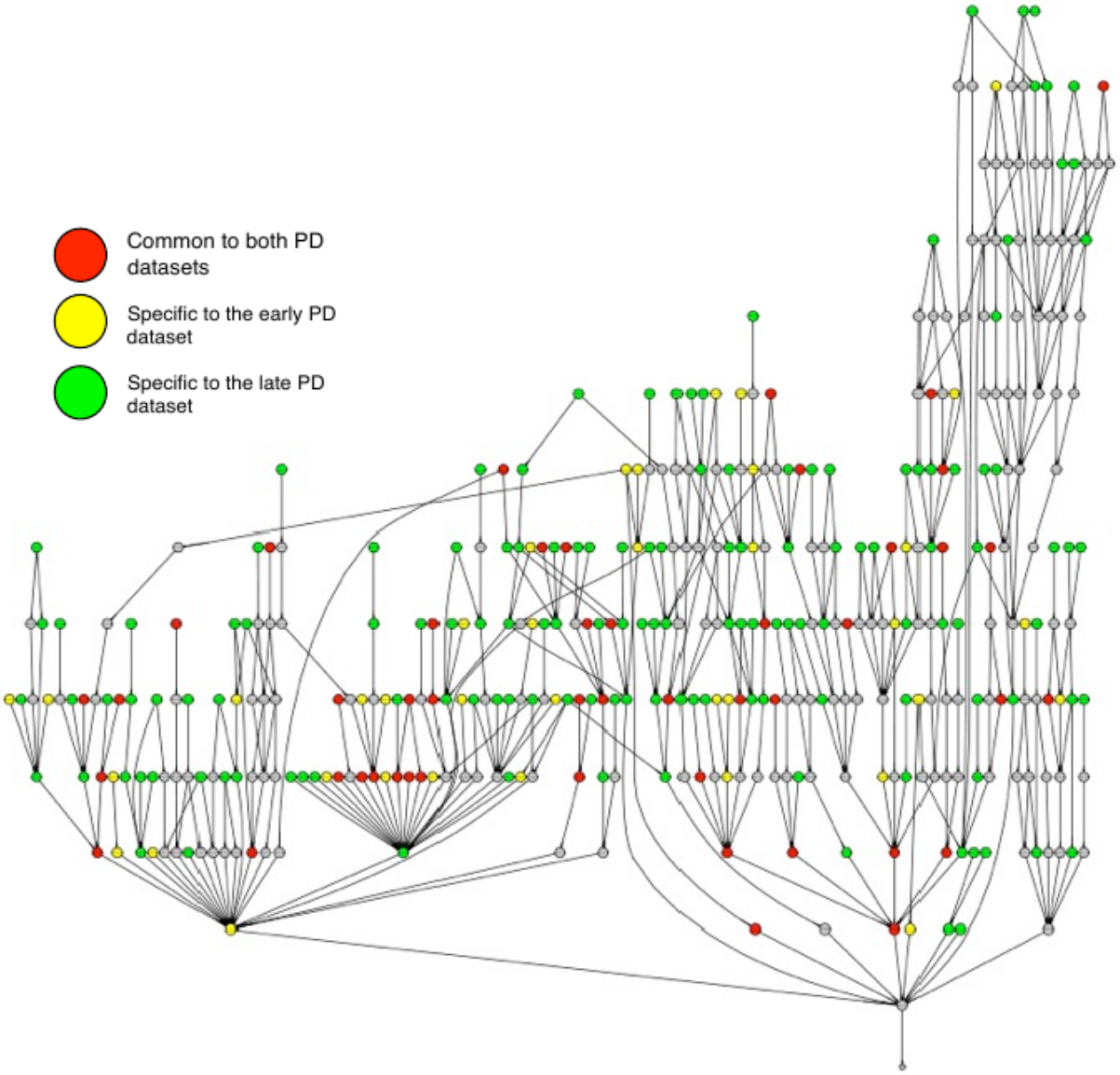


Figure 2.6: Relevant GO nodes identified with *a priori* approach

In (Squillario et al., 2012) we compared the results of the two different pipelines in the analysis of an early PD dataset (Barrett et al. (2011), GSE6613) whose samples derived from the whole blood of 50 individuals affect by early PD, of 33 individuals affect by other neurodegenerative diseases and of 22 healthy controls. In the analysis, gene selection is performed by $\ell_1\ell_2 fs$ with Gene Ontology (GO) as source of prior knowledge for both. Both pipelines identify a list of discriminant GO terms and a list of selected genes. It is not possible to compare these results directly, because they are derived from two different procedures. Hence, we resorted to compare the information produced by each pipeline with the currently available domain knowledge regarding PD. To this aim, we performed two analyses, namely *literature characterization* and *benchmark analysis*. In the first one, we used the following procedure, implemented with BioPython[8]. For each gene, we performed a PubMed search using gene symbol as a query, and noted the resulting papers. For each paper, we obtained the title, the abstract, the authors, the journal and the PubMed ID. We searched the title and abstract for a set of arbitrarily chosen keywords: Alzheimer, Parkinson, Amyotrophic Lateral Sclerosis, Huntington, brain, neuro. Besides "Parkinson", other neurodegenerative diseases were considered as well, because it is known that they share a relevant biological background (von Bernhardi et al., 2010; Vicente Miranda and Outeiro, 2010). Also, more general terms, like "neuro" and "brain", were included, in order to consider genes that are normally, if not specifically, expressed in the brain.

In the second analysis, to obtain necessary benchmark lists, we started with the union of three gene lists: the first one from the "Parkinson's disease - Homo sapiens" pathway of KEGG PATHWAY database, the second one from the "Parkinson's disease (PD)" entry of KEGG DISEASE database, and the third one from the result of Gene Prospector tool (Yu et al., 2008) when queried for "Parkinson's disease". While the first two lists contain genes that have been experimentally verified to be involved in the disease, the third list could contain also genes connected to the disease, because they derive from high-throughput experiments and need further experimental validation. Next we merged the three lists and eliminated duplicates. The merged list contained a total of 482 genes. To proceed, we used Gene Ontology Annotations (GOA), compiled for *homo sapiens*. Here, each gene is associated with some GO term(s), based on specific evidences. The evidences describe the work or analysis upon which the association between a specific GO term and gene product is based; there are 22 possible evidence codes in GO[9] Each single association of a gene to an evidence is tagged with the annotation date as well. Based on that information, we constructed a filtering schema to derive *benchmark gene list* and *benchmark GO terms list*, that are strongly associated with PD. Two filters were applied: based on the annotation date (for GO terms), and based on the evidence strength (for both genes and GO terms). While examining the annotations, we noticed that each

[8] `http://www.biopython.org/`

[9] `http://www.geneontology.org/GO.annotation.SOP.shtml`

	Standard	KDVS
Discriminant/Enriched GO terms	65	150
Discriminant Genes	66	4286
Benchmark coverage	7%(g) & 3% (GO)	44%(g) & 12%(GO)

Table 2.7

gene could be associated with the same GO term but with different evidences, due to the internal history of GO curation process. During the construction of benchmark lists, we kept those associations whose evidences displayed the most recent annotation date. Some of the evidences can be more reliable than others. Based on this consideration, we arbitrarily defined the trustability of the evidences, as follows. The evidences recognized as more trustable include all those belonging to the Experimental Evidence Codes (i.e. EXP, IDA, IPI, IMP, IGI, IEP), the Traceable Author Statement (i.e. TAS), and the Inferred by Curator (i.e. IC). During the construction of benchmark lists, we kept the genes and GO terms associated with strong evidences.

For the *benchmark analysis* the results are reported in Table 2.7. For the *literature characterization*, we evaluated that the percentage of selected genes linked to at least one of a set of keywords related with PD are the 69% and 80% respectively for KDVS and the standard *a posteriori* pipeline. Moreover, the percentage of selected genes linked to the keyword "Parkinson", are 12% and 20% respectively for KDVS and the standard *a posteriori* pipeline.

The *literature characterization* analysis showed that the list of selected genes produced by *a posteriori* pipeline contained more genes related to PD than the respective list produced by KDVS pipeline. In contrast, the *benchmark analysis* showed the opposite, namely both lists of selected genes and discriminant GO nodes were more PD-related in case of KDVS. Since the *literature characterization*, as performed, is more prone to identify false positives than the other analysis, we concluded that both pipelines were able to identify reliable lists of genes and GO terms, however the results obtained from KDVS pipeline in particular covered more certified knowledge related to PD, than the results from the *a posteriori* pipeline.

Therefore, the application of prior knowledge before the statistical analysis of gene expression data, proved to be more valuable in obtaining trustable knowledge for the discovery of potential biomarkers and molecular mechanisms related to PD. Moreover the comparison highlighted two genes and few GO terms common to both the pipelines: the genes were SNCA and ATXN1, the latter known to be associated with another neurodegenerative disease named ADCA, while the GO terms concerned the iron and the heme binding, in addition to the bond of same proteins to form homodimers.

2.5.4 From genes to networks: evaluating sources of variability

In this section we present a computational framework (Barla et al., 2011a, 2012) for the study of reproducibility in network medicine studies (Barabási et al., 2011). Networks, molecular pathways in particular, are increasingly looked at as a better organized and more rich version of gene signatures. However, high variability can be injected by the different methods that are typically used in system biology to define a cellular wiring diagram at diverse levels of organization, from transcriptomics to signalling, of the functional design. For example, to identify the link between changes in graph structures and disease, we choose and combine in a workflow a classifier, the feature ranking/selection algorithm, the enrichment procedure, the inference method and the networks comparison function. Each of these components is a potential source of variability, as shown in the case of biomarkers from microarrays (MAQC Consortium, 2010). Considerable efforts have been directed to tackle the problem of poor reproducibility of biomarker signatures derived from high-throughput -omics data (MAQC Consortium, 2010), addressing the issues of selection bias (Ambroise and McLachlan, 2002; Furlanello et al., 2003) and more recently of pervasive batch effects (Leek et al., 2010). We argue that it is now urgent to adopt a similar approach for network medicine studies. Stability (and thus reproducibility) in this class of studies is still an open problem (Baralla et al., 2009). Underdeterminacy is a major issue (De Smet and Marchal, 2010), as the ratio between network dimension (number of nodes) and the number of available measurements to infer interactions plays a key role for the stability of the reconstructed structure. Furthermore, the most interesting applications are based on inferring networks topology and wiring from high-throughput noisy measurements (He et al., 2009). The stability analysis is applied to networks identifying common and specific network substructures that could be basically associated to disease. The overall goal of the pipeline is to identify and rank the pathways reflecting major changes between two conditions, or during a disease evolution. We start from a profiling phase based on classifiers and feature selection modules organized in terms of experimental procedure by Data Analysis Protocol (MAQC Consortium, 2010), obtaining a ranked list of genes with the highest discriminative power. Classification tasks as well as quantitative phenotype targets can be considered by using different machine learning methods in this first phase. The problem of underdeterminacy in the inference procedure is avoided by focusing only on subnetworks, and the relevance of the studied pathways for the disease is judged in terms of discriminative relevance for the underlying classification problem.

As a testbed for the *glocal* stability analysis, we compare network structures on a collection of microarray data for patients affected by Parkinson's disease (PD), together with a cohort of controls (Zhang et al., 2005). The goal of this task is to identify the most relevant disrupted pathways and genes in late stage PD. On this dataset, we show that choosing different profiling approaches we get low overlapping in terms of common

enriched pathways found. Despite this variability, if we consider the most disrupted pathways, their *glocal* distances (between case and control networks) share a common distribution assessing equal meaningfulness to pathways found starting from different approaches.

2.5.4.1 Description of the pipeline

The machine learning pipeline described in this section has been originally introduced in (Barla et al., 2011b). As shown in Figure 2.3 (extended pipeline), it handles case/control transcription data through four main steps, from a profiling task to the identification of discriminant pathways. The pipeline is independent from the algorithms used: here we describe each step and the implementation adopted for the following experiments evaluating the impact of different enrichment methods in pathway profiling.

Feature selection step. The prediction model $\mathcal{M}$ is built by using two different algorithms for classification and feature ranking. Aside from $\ell_1\ell_2 fs$ framework, we consider `LibLinear`, a linear Support Vector Machine (SVM) classifier specifically designed for large datasets (millions of instances and features) (Fan et al., 2008). In particular, the classical dual optimization problem with L2-SVM loss function is solved with a coordinate descent method. For our experiment we adopt the ℓ_2-regularized penalty term and the module of the weights for ranking purposes within a 100×3-fold cross validation schema. We build a model for increasing feature sublists where the feature ranking is defined according to the importance for the classifier. We choose the model, and thus the top ranked features, providing a balance between the accuracy of the classifier and the stability of the signature (MAQC Consortium, 2010). The output of this first step is a gene signature $g_1, ..., g_k$ (one for each model $\mathcal{M}$) containing the k most discriminant genes, ranked according their frequency score.

Pathway enrichment. The successive enrichment phase derives a list of relevant pathways from the discriminant genes, moving the focus of the analysis from single genes to functionally related pathways. As indicated in Section 2.4 we choose one representative $\mathcal{E}$ for each class of enrichment tools indicated by Huang et al. (2009): `WebGestalt` (`WG`), `GSEA` and `PaLS`. We refer as sources of information $\mathcal{D}$ both to the KEGG, to explore known information on molecular interaction networks, and GO, to explore functional characterization and biological annotation. For `GSEA` we use the *preranked* analysis tool, feeding the ranked lists of genes produced by the profiling phase directly to the enrichment step of GSEA. To avoid a miscalculation of the enrichment score ES, we provide as input the complete list of variables (not just the selected ones), assigning to the not-selected a zero score. Note that GSEA calculates enrichment scores that reflect the de-

$\mathcal{M}$	$\mathcal{D}$	$\mathcal{E}$ WG	GSEA	PaLS
$\ell_1\ell_2 fs$	GO	114 (92)	7 (7)	381 (331)
	KEGG	43 (43)	2 (2)	71 (71)
LibLinear	GO	83 (45)	0 (0)	404 (356)
	KEGG	56 (55)	1 (1)	77 (77)

Table 2.8: Summary of pathways retrieved in the pathway enrichment step. The numbers in brackets refer to the pathways considered for the network inference step.

gree to which a pathway is overrepresented at the top or the bottom of the ranked list. In our analysis we considered only pathways enriched with the top of the list. Applying the above mentioned pathway enrichment techniques, we retrieve for each gene g_i the corresponding whole pathway $p_i = \{h_1, ..., h_t\}$, where the genes $h_j \neq g_i$ not necessarily belong to the original signature $g_1, ..., g_k$. Extending the analysis to all the h_j genes of the pathway allows us to explore functional interactions that would otherwise get lost.

Data description and preprocessing. The presented approach is applied to PD data originally introduced in Zhang et al. (2005) and publicly available at Gene Expression Omnibus (GEO), with accession number GSE20292. The biological samples consist of whole substantia nigra tissue in 11 PD patients and 18 healthy controls. Expressions were measured on Affymetrix HG-U133A platform. We perform the data normalization on the raw data with the *rma* algorithm of the R Bioconductor *affy* package with a custom CDF (downloaded from BrainArray: `http://brainarray.mbni.med.umich.edu`http://brainarray.mbni.med.umich.edu) adopting Entrez identifiers.

2.5.4.2 Results and Discussion

The feature selection results varied accordingly to the chosen method: $\ell_1\ell_2 fs$ identified 458 discriminant genes associated to an average prediction performance of 80.8%, while with LibLinear we selected the top-500 genes associated to an accuracy of 80% (95% boostrap Confidence Interval: (0.78;0.83)) coupled with a stability of 0.70. The lists have 119 common genes.

The number of enriched pathways greatly varied depending on the selection and enrichment tools. With $\ell_1\ell_2 fs$, we found globally for GO and KEGG, 157, 452 and 9 pathways as significantly enriched, for WG, PaLS and GSEA respectively. Similarly, for LibLinear, the identified pathways were: 139, 481 and 1. Table 2.8 reports the detailed results for model $\mathcal{M}$, enrichment $\mathcal{E}$ and database $\mathcal{D}$.

$\mathcal{M}$	$\mathcal{D}$	$\lvert\cap(\text{All}_{\mathcal{E}})\rvert$	$\lvert\cap(\mathcal{E}_{\text{WG, PaLS}})\rvert$
$\ell_1\ell_2 fs$	GO	0	17
	KEGG	1	22
LibLinear	GO	0	5
	KEGG	0	21

Table 2.9: Summary of common most disrupted pathways ($\phi \geq 0.05$).

If we consider the $\ell_1\ell_2 fs$ selection method and the enrichment performed within the GO, we may note that no common GO terms were selected across enrichment methods. A significant overlap of results was found only between WG and PaLS, with 30 GO common terms. Similar considerations may be drawn with the results from the LibLinear feature selection method. Within the GO enrichment we did not identify any common GO term among the three enrichment tools. Considering only WG and PaLS, we were able to select 12 common GO terms.

If we consider the $\ell_1\ell_2 fs$ selection method and the enrichment performed within KEGG, two common pathways are identified across enrichment methods. A significant overlap of results was found between WG and PaLS, with 43 common pathways. For LibLinear, only one common pathway was selected among the three enrichment tools. A significant overlap of results was found between WG and PaLS, with 55 common pathways.

Following the pipeline, we also performed a comparison of the three network reconstruction methods. As an additional caution against this problem, in the following experiments we limit the analysis to pathways having more than 4 nodes and less than 1000 nodes. We considered the most disrupted networks, keeping for the analysis those pathways that had a *glocal* distance greater or equal to the chosen threshold $\tau = 0.05$. The choice of such threshold was made considering the distribution of *glocal* distances ϕ for the methods $\mathcal{M}$. For instance, if we consider the LibLinear selection method and the KEGG database, we have a cumulative distribution as depicted in Figure 2.7(a). The threshold τ is set to 0.05 and allows retaining at least 50% of pathways. The plot in Figure 2.7(b) represents the *glocal* distances distribution for all enrichment methods $\mathcal{E}$ with respect to the two components of the *glocal* distance: the Ipsen distance ϵ and the Hamming distance H. The red curved line represents the threshold τ in this space. The plot in Figure 2.7(c) is detailed for subnetwork inference method $\mathcal{N}$.

After retaining the most distant pathways, we performed a comparison of common terms for fixed selection method $\mathcal{M}$ and database $\mathcal{D}$. The results are reported in Table 2.9.

In Tables 2.10 and 2.11 we report the most disrupted GO terms and KEGG pathways that have a *glocal* distance ϕ greater or equal to the chosen threshold τ.

As an example of a selected pathway within KEGG, the networks (thresholded at edge weight 0.1 for graphic purposes) inferred by `WGCNA` (together with the corresponding stability) on the *Amyotrophic Lateral Sclerosis* KEGG pathway (ALS - 05014) are displayed in Figure 2.8. We also plot the inferred network by the `RegnANN` algorithm. Similarly, in Figure 2.9 we plot the *Pathogenic E. coli infection* KEGG pathway, reconstructed by `WGCNA`, its stability plot, and the corresponding inferred networks by the `RegnANN` algorithm.

The variability in the results, as expected, strongly depends on the method of choice. For feature selection, the nature of the method is the key. In the proposed pipeline we limited the impact of this step by choosing two approaches within the regularization methods family. Both classifiers adopt a ℓ_2-regularization penalty term, combined with different loss functions and, for $\ell_1\ell_2 fs$ with another regularization term. We used similar but not equal model selection protocols. Both guarantee that the results are not affected by selection-bias. In this work, the main source of variability was the choice of the gene enrichment module. Therefore, the experimenter must be careful in choosing one method or another and in using it compliantly with the experimental design. For instance, GSEA was designed for estimating the significance levels by considering separately the positively and negatively scoring gene sets within a list of genes selected with *filter* methods based on classical statistical tests. It is worth noting that, if one uses the preranked option, as we did, negative regulated groups might not be significant at all (we indeed discarded them). WG uses the Hypergeometrical test to assess the functional groups but, differently from GSEA, does not use any significance assessment based on permutation of phenotype labels. PaLS is the simplest method, being just a measure of occurrences of a given descriptor in the list of selected genes. However, enrichment methods from different categories are complementary and can identify different but equally meaningful biological aspects of the same process. Thus, the integration of information across different methods is the best strategy.

Moreover, the assessment of the reconstruction distance between case and control version of the same pathways help in providing a quantitative focus on the key pathway involved in the process. The use of a distance mixing the effects of structural changes with those due to the differences in rewiring, moreover, warrants a more informative view on the difference assessment itself. The limited effect of different feature selection methods is confirmed by the plots in Figure 2.11.

For $\ell_1\ell_2 fs$, the only most disrupted pathway shared by the three enrichment tools $\mathcal{E}$ and the three reconstruction methods $\mathcal{N}$ is ALS. This pathway is relevant in this context because, like PD, ALS is another neurodegenerative disease, therefore they share significant biological features, in particular at the mithocondrial level. Moreover, at the phenotypic level the skeletal muscles of the patients are severely affects influencing the movements. In Figure 2.8 it is evident that a high number of interactions is established

among the genes going from the control (below) to the affected (above) pathways. It is also interesting to underline that CYCS (Entrez ID: 54205), one of the hub genes (represented by a red dot in the graph) within the pathway, was identified by $\ell_1\ell_2 fs$ as discriminant. This gene is highly involved in several neurodegenerative diseases (*e.g.,* PD, Alzheimer's, Huntington's) and in pathways related to cancer. Furthermore its protein is known to functions as a central component of the electron transport chain in mitochondria, and to be involved in initiation of apoptosis, known cause of the neurons loss in PD. Across variable selection algorithms $\mathcal{M}$, five highly disrupted pathways were found as shared between two of the three enrichment methods (see Table 2.11, bold items). In particular, we represented in Figure 2.9 the corresponding inferred networks. To further highlight the different outcomes occurring from the same dataset when diverse inference methods are employed, we reconstructed the *ALS* and *Pathogenic E. coli infection* by the `RegnANN` algorithm, which tends to spot also second order correlation among the network nodes, see Figures 2.8 and 2.9.

Two genes in the *E. coli infection* pathway were selected both by $\ell_1\ell_2 fs$ and LibLinear, namely ABL1 (Entrez ID:71) and TUBB6 (Entrez ID: 84617). ABL1 seems to play a relevant role as hub both in the `WGCNA` and in the `RegnANN` networks. ABL1 is a protooncogene that encodes protein tyrosine kinase that has been implicated in processes of cell differentiation, cell division, cell adhesion, and stress response. It was also found to be responsible of apoposis in human brain microvascular endothelial cells.

In Figure 2.12 we note that pathways with high number of genes are similar in term of local distance, instead a wider range of variability is found looking at the spectral distance. The red line in 2.12(b) divides the 2 cluster. Pathway targets beyond and within the red line are represented in the cumulative histogram in 2.12(a). Pathways beyond the threshold are equally distributed and they represent a wider range of targets, instead pathways within the threshold show a smaller number of targets 2.12(a) on the right.

2.5.4.3 Conclusion

Moving from gene profiling towards pathway profiling can be an effective solution to overcome the problem of the poor overlapping in *-omics* signatures. Nonetheless, the path from translating a discriminant gene panel into a coherent set of functionally related gene sets includes a number of steps each contributing in injecting variability in the process. To reduce the overall impact of such variability, it is thus critical that, whenever possible, the correct tool for each single step is adopted, accurately focusing on the desired target to be investigated. This mainly holds for the choice of the most suitable enrichment tool and biological knowledge database, and, to a lower extent, to the inference method for the network reconstruction: all these ingredients are planned for

different objectives, and their use on other situations may result in misleading. As a final observation and a possible future development to explore, the emerging instability can be tackled by obtaining the functional groups identification as the result of a prior knowledge injection in the learning phase, rather than a procedure a posteriori (Zycinski et al., 2011, 2013).

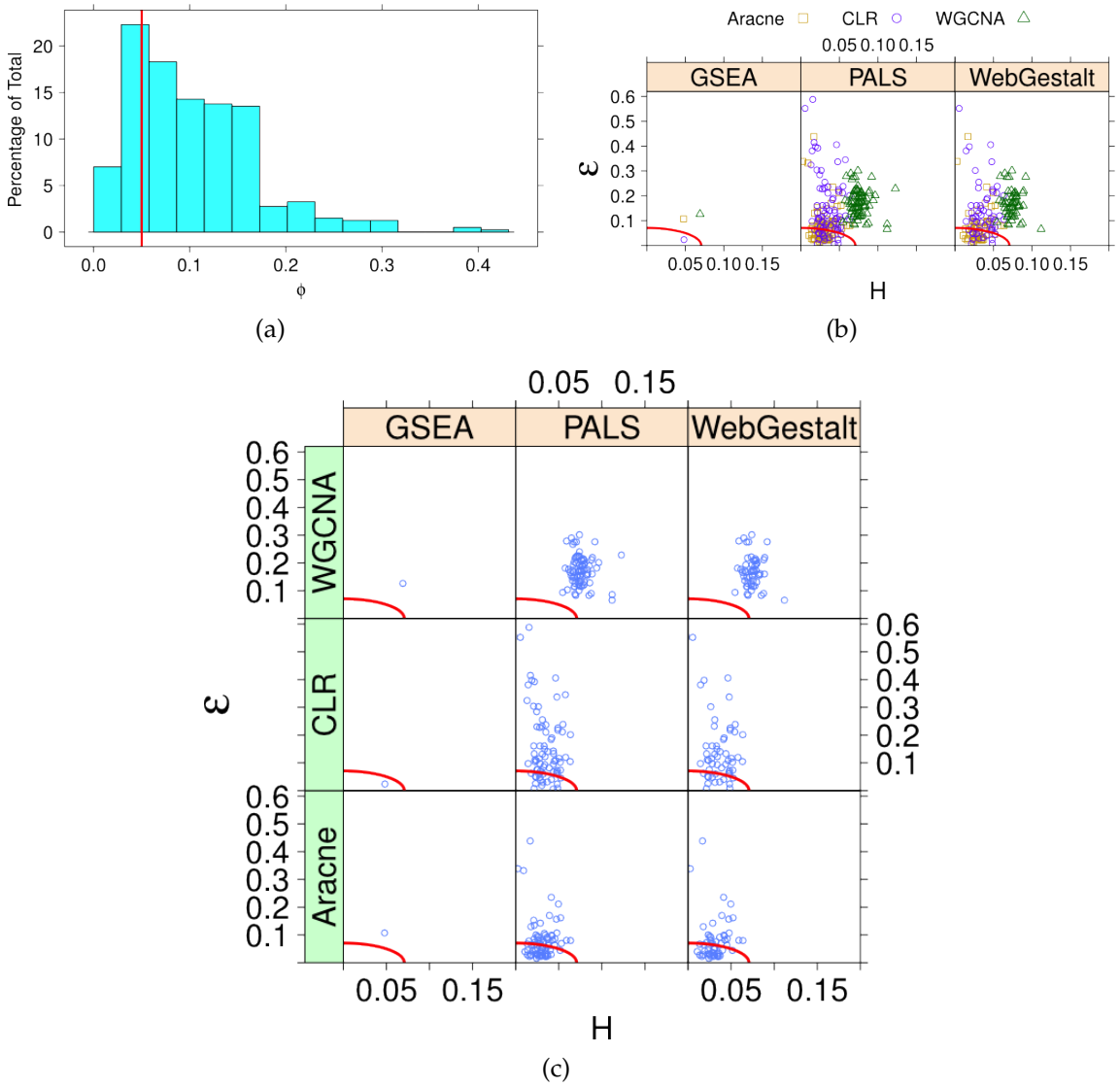


Figure 2.7: Detailed distance plot for LibLinear and KEGG (see Figure 2.11b). The histogram plot in (a) represents the cumulative histogram for all distances across enrichment methods $\mathcal{E}$ and subnetwork inference algorithms $\mathcal{N}$. The threshold τ is set to retain at least 50% of pathways. In (b) a plot of Hamming vs. Ipsen distances (H vs. ϵ) for all possible combinations of $\mathcal{E}$ and $\mathcal{N}$, which is detailed in (c).

$\ell_1\ell_2 fs$		LibLinear	
ID	Term name	ID	Term name
GO:0005739	Mitochondrion	GO:0042127	Regulation of cell proliferation
GO:0031966	Mitochondrial membrane	GO:0005783	Endoplasmic reticulum
GO:0005743	Mitochondrial inner membrane	GO:0015629	Actin cytoskeleton
GO:0042802	Identical protein binding	GO:0006469	Negative regulation of protein kinase activity
GO:0007018	Microtubule-based movement	**GO:0005747**	**Mitochondrial respiratory chain complex I**
GO:0046961	Proton-transp. ATPase activity, rotat. mechanism		
GO:0005753	Mitoch. proton-transp.ATP synthase complex		
GO:0000502	Proteasome complex		
GO:0015986	ATP synthesis coupled proton transport		
GO:0045202	Synapse		
GO:0048487	Beta-tubulin binding		
GO:0042734	Presynaptic membrane		
GO:0005747	**Mitochondrial respiratory chain complex I**		
GO:0006120	Mitoch. electron transport, NADH to ubiquinone		
GO:0015078	Hydrogen ion transmembrane transp. activity		
GO:0015992	Proton transport		
GO:0005874	Microtubule		

Table 2.10: Summary of most disrupted GO terms common between WG and PaLS, for different models $\mathcal{M}$. Each GO term is associated to a *glocal* distance $\phi \geq 0.05$ for all subnetwork reconstruction algorithms $\mathcal{N}$. GO terms are sorted according decreasing average ϕ. Bold fonts represent the GO terms shared by model $\mathcal{M}$.

$\ell_1\ell_2 fs$		LibLinear	
ID	Pathway name	ID	Pathway name
01100	**Metabolic pathway**	04630	Jak-STAT signaling pathway
05130	**Pathogenic Escherichia coli infection**	**01100**	**Metabolic pathway**
04910	**Insulin signaling pathway**	**05130**	**Pathogenic Escherichia coli infection**
00310	Lysine degradation	04623	Cytosolic DNA-sensing pathway
04140	Regulation of autophagy	00330	Arginine and proline metabolism
03050	Proteasome	**04910**	**Insulin signaling pathway**
00230	Purine metabolism	05212	Pancreatic cancer
05014	Amyotrophic lateral sclerosis*	03030	DNA replication
00980	Metabolism of xenobiotics by cytochrome P450	**05213**	**Endometrial cancer**
00620	Pyruvate metabolism	04660	T cell receptor signaling pathway
05213	**Endometrial cancer**	04310	Wnt signaling pathway
00270	Cysteine and methionine metabolism	05210	Colorectal cancer
00240	Pyrimidine metabolism	04912	GnRH signaling pathway
05120	Epithelial cell sign. in Helicobacter pylori infec.	05332	Graft-versus-host disease
05110	Vibrio cholerae infection	04520	Adherens junction
00020	Citrate cycle (TCA cycle)	04621	NOD-like receptor signaling pathway
00562	Inositol phosphate metabolism	04370	VEGF signaling pathway
00600	Sphingolipid metabolism	04662	B cell receptor signaling pathway
05218	Melanoma	**04722**	**Neurotrophin signaling pathway**
00010	Glycolysis / Gluconeogenesis	05214	Glioma
00051	Fructose and mannose metabolism	04330	Notch signaling pathway
04722	**Neurotrophin signaling pathway**		

**Note: This is the only selected pathway shared across all enrichment methods $\mathcal{E}$.*

Table 2.11: Summary of most disrupted KEGG pathways common between WG and PaLS, for different models $\mathcal{M}$. Each pathway is associated to a *glocal* distance $\phi \geq 0.05$ for all subnetwork reconstruction algorithms $\mathcal{N}$. KEGG pathways are sorted according decreasing average ϕ. Bold fonts represent the KEGG pathways shared by model $\mathcal{M}$.

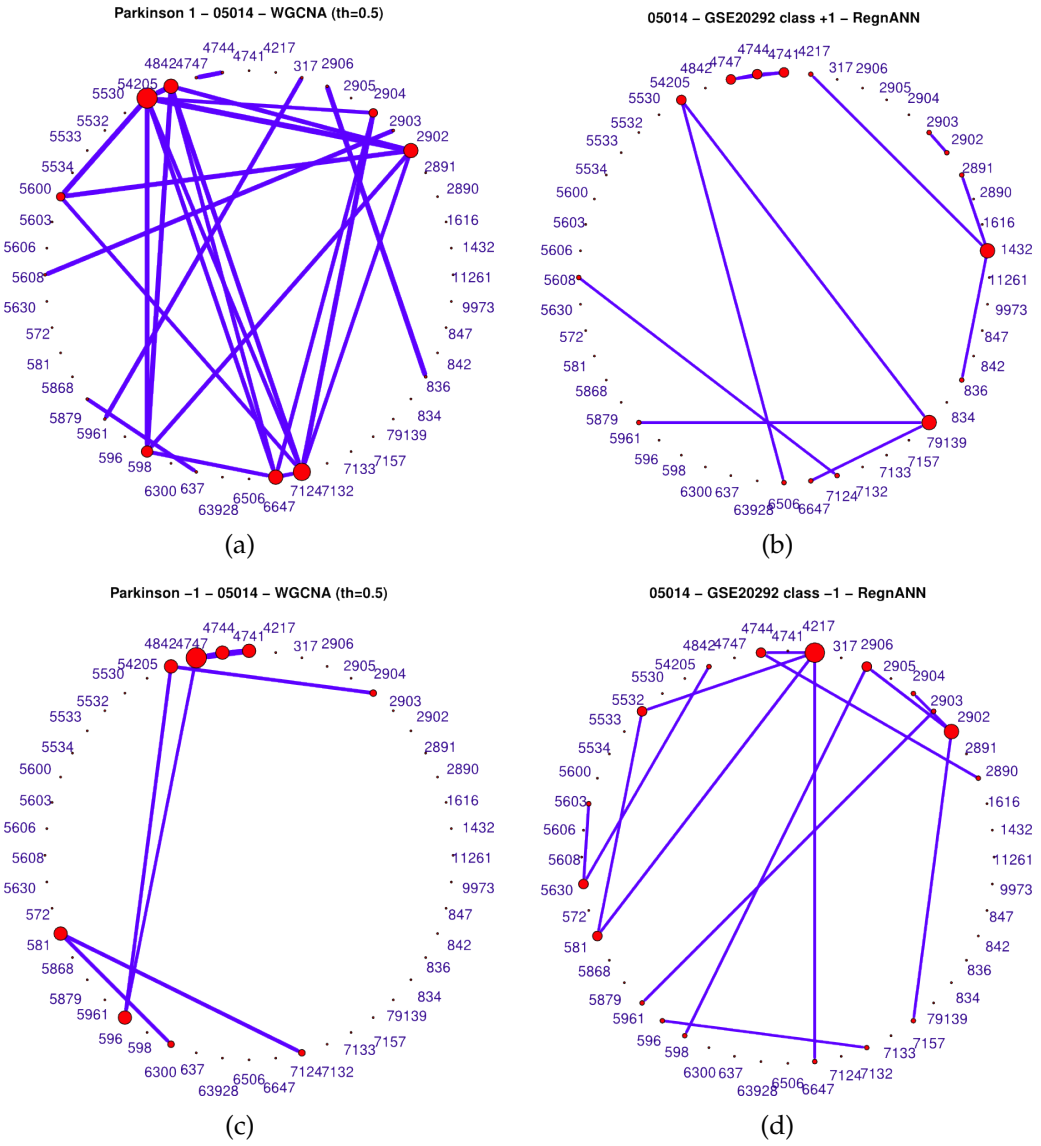


Figure 2.8: Networks inferred by `WGCNA` (a, c) and RegnANN (b, d) algorithm for the *ALS* KEGG pathway for PD patients (a, b) and controls (c, d).

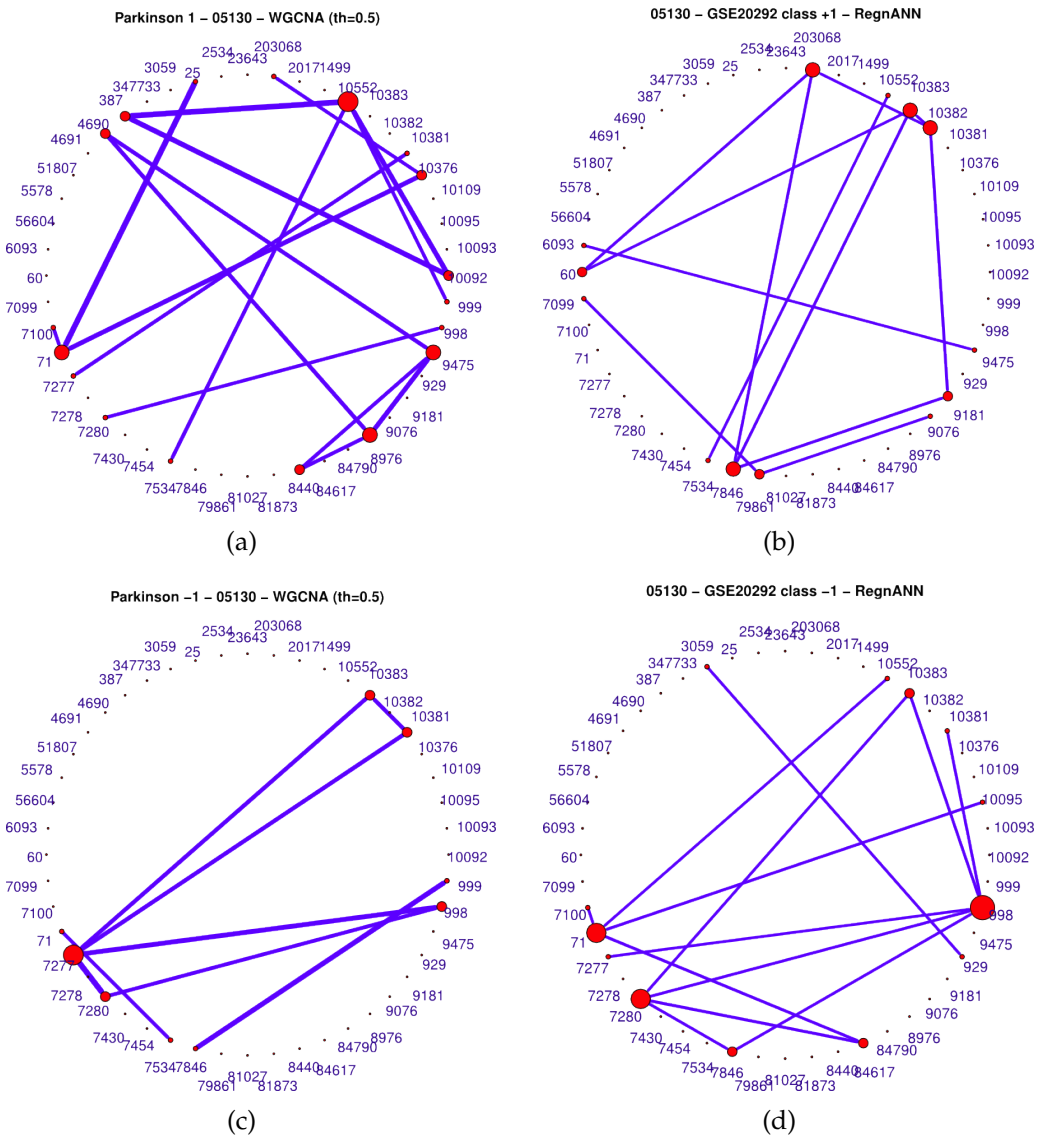


Figure 2.9: Networks inferred by `WGCNA` (a, c) and `RegnANN` (b, d) algorithm for the *Pathogenic E. coli infection* KEGG pathway for PD patients (a, b) and controls (c, d).

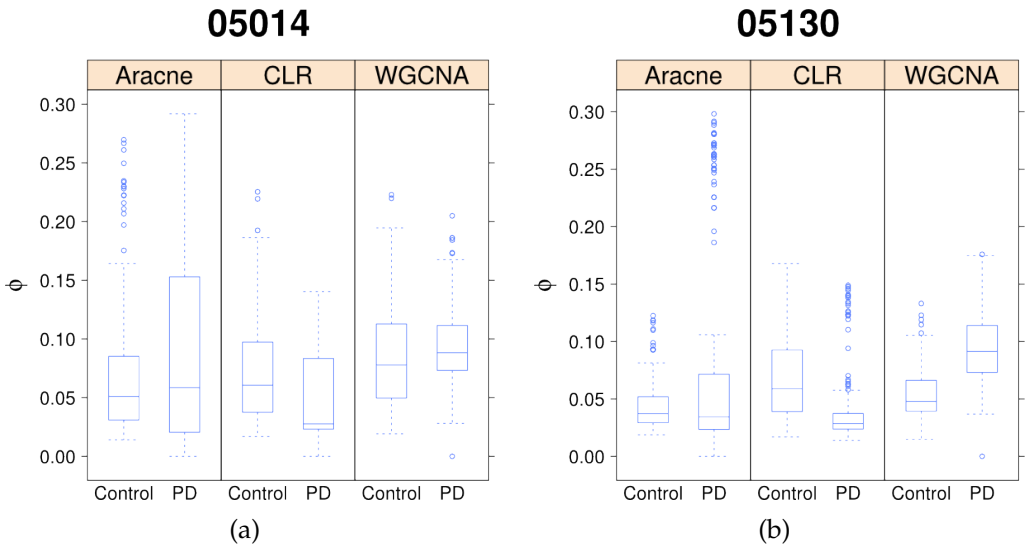
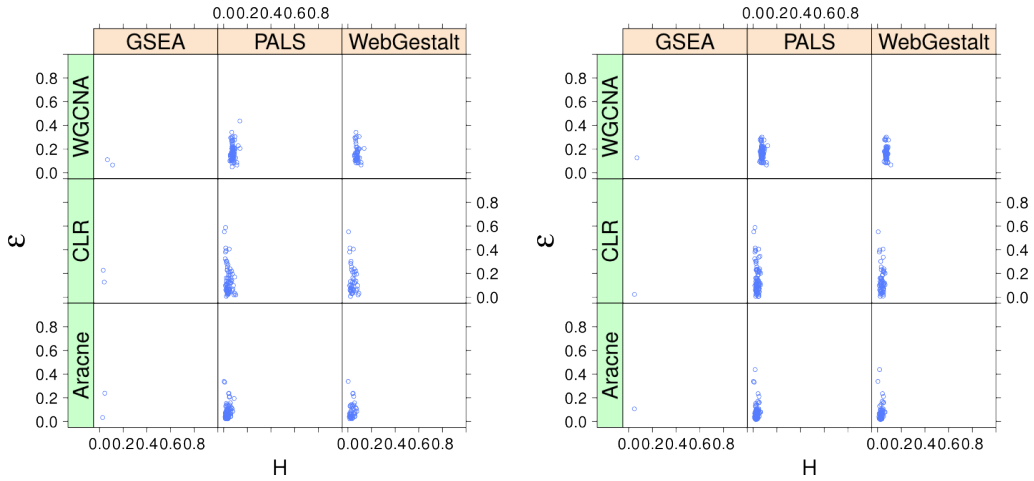


Figure 2.10: (a) For networks inferred for the *ALS* (a) and for the *Pathogenic E. coli infection* (b) KEGG pathway `WGCNA` is the method showing the highest stability on the two classes.



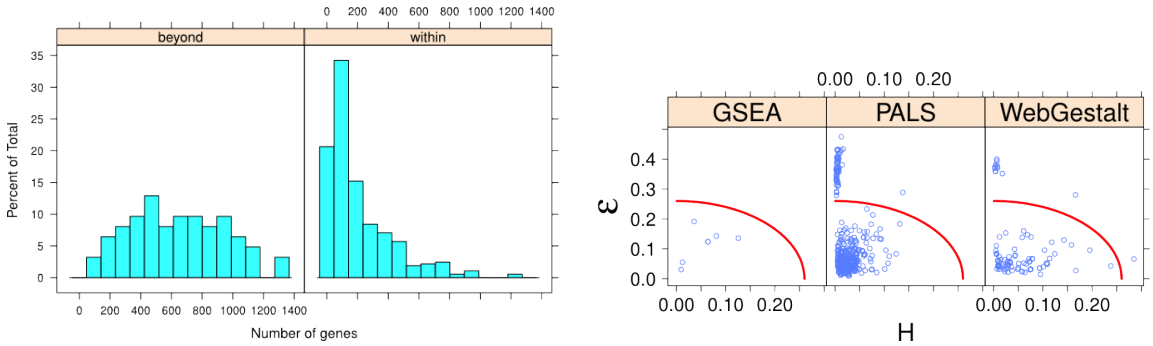
(a) $\ell_1\ell_2$ and KEGG

(b) LibLinear and KEGG

(c) $\ell_1\ell_2$ and GO

(d) LibLinear and GO

Figure 2.11: Plots of Hamming vs. Ipsen distances (H vs. ϵ) for all possible combinations of $\mathcal{M}, \mathcal{D}, \mathcal{E}$ and $\mathcal{N}$. In our analysis we considered the *glocal* distance ϕ, defined as the normalized product of H and ϵ.



(a) (b)

Figure 2.12: (a) Pathway target cumulative histogram. (b) Hamming versus Ipsen (H vs. ϵ) distance, and thresholding of high populated pathways.

Chapter 3

Dictionary learning for genomic aberrations identification

In the context of statistical learning, dictionary learning is a class of methods that aims to reconstruct a given set of signals as combination of *atoms* belonging to a dictionary. The key aspect in dictionary learning is that the dictionary is also learned by the given data.

In this chapter we present a novel dictionary learning method specifically developed to analyze array-based Comparative Genomic Hybridization (aCGH) data, whose goal is to extract recurrent patterns (the *atoms*) of genomic alterations.

We first introduce, in Section 3.1, the biological context describing the problem we want to solve and in Section 3.2 we introduce the data and the state-of-the-art methods to handle them.

In Section 3.3 we introduce the proposed model, describing its peculiarities, and a novel minimization algorithm that we use to solve it and that is also general enough to solve a wider class of dictionary learning problems.

Section 3.4 describes a model for simulated aCGH data that exploit several state-of-the-art synthetic models and combines them in order to generate "realistic" aCGH signals.

Section 3.5 illustrates results of several experiments on toy, simulated and real data.

3.1 Biological context: copy number variation from aCGH

Multifactorial pathological conditions, such as tumors, are often associated with structural and numerical chromosomal aberrations. Copy number variations (CNVs) are alterations of the DNA that result in the cell having an abnormal number of copies of one or more sections of the DNA. Recurrent aberrations across samples may indicate an oncogene or a tumor suppressor gene, but the functional mechanisms that link altered copy numbers to pathogenesis are still to be explained.

3.2 Processing aCGH data

Array-based Comparative Genomic Hybridization (aCGH) (Davies et al., 2005a) is a modern whole-genome measuring technique that evaluates the occurrence of copy variants across the genome of samples (patients) versus references (controls) on the entire genome, extending the original CGH technology (Kallioniemi et al., 1992).

Modern array-based CGH technologies aim to improve the CGH in terms of resolution passing from tens of Mb (Megabases) to kb (kilobases). The technology improvement is obtained substituting the hybridization target, moving from metaphase chromosome spreads to genomic segment spotted in an array format (Figure 3.1). Even if aCGH allows a deeper analysis of chromosomal imbalances/alterations, some intrisic problems related to this technology are not (and cannot be) solved. For example, a limitation regards the tracking of balanced translocations that cannot be detected (Davies et al., 2005a).

aCGH is used to detect the aberrations in segmental copy numbers at chromosomal *loci* represented by DNA clones with known genomic locations. The technique visualizes CNVs by hybridizing patient and control DNAs with two different fluorescent dyes, usually Cy3 (green) and Cy5 (red), as shown in Figure 3.1.

After preprocessing, for each aCGH we obtain a signal as the one depicted in Figure 3.2, where for each chromosomal location (x-axis) we can estimate a log-ratio of the CNV for the hybridized control and patient (y-axis). Arranging the x-axis ordering the chromosomal locations, clinicians can actually see numerical (entire chromosome) and segmental (part of chromosome) genomic alterations, in term of deletions or amplifications.

Data Normalization. A crucial step in the analysis of aCGH data is the normalization. There are several technological and biological sources of noise that need to be managed in order to produce a signal ready for the downstream analysis.

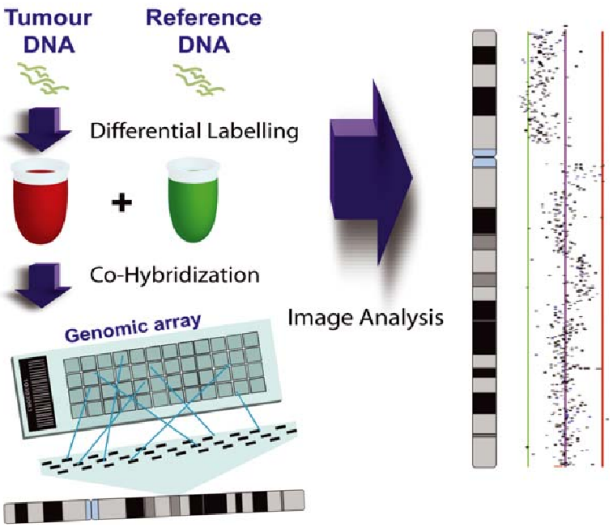


Figure 3.1: Array-based Comparative Genomic Hybridization process(Davies et al., 2005a).

In literature many methods were proposed (Curry et al., 2008). Khojasteh et al. (2005) tested existing normalization methods commonly used for gene expression data in order to deduce a stepwise normalization framework tailored to handling high density aCGH data. Recently, novel techniques non-inherited from gene expression microarray have been proposed (Neuvial et al., 2006; Chen et al., 2008; Miecznikowski et al., 2011; Fitzgerald et al., 2011).

In this thesis, when we deal with real raw data, we adopt a method belonging to the class of population-based normalization methods, namely CGHNormaliter proposed by van Houte et al. (2009) (van Houte et al., 2010, for details on the R package). This method, an extension of the one proposed by Staaf et al. (2007), offers a more sophisticated normalization of aCGH data. First it performs segmentation of the signal in order to separate accurately normals, gains and losses population of clones. Next they fit a LOWESS (LOcally WEighted Scatterplot Smoothing) (Cleveland et al., 1988) regression curve through the normals only and use that to normalize the entire population of clones. Iteratively the method performs the two steps of segmentation and normalization until convergence. The separation of the normal clones from the others allows to reduce the impact of imbalanced copy numbers that lead to improper normalization using conventional methods.

Segmentation and Calling. A signal measured with an aCGH is made of a piecewise linear (and constant) component plus some noise. The typical mode of analysis of such data is *segmentation*, that is the automatic detection of *loci* where copy numbers alterations (amplifications or deletions) occur. Differently from other molecular data, for

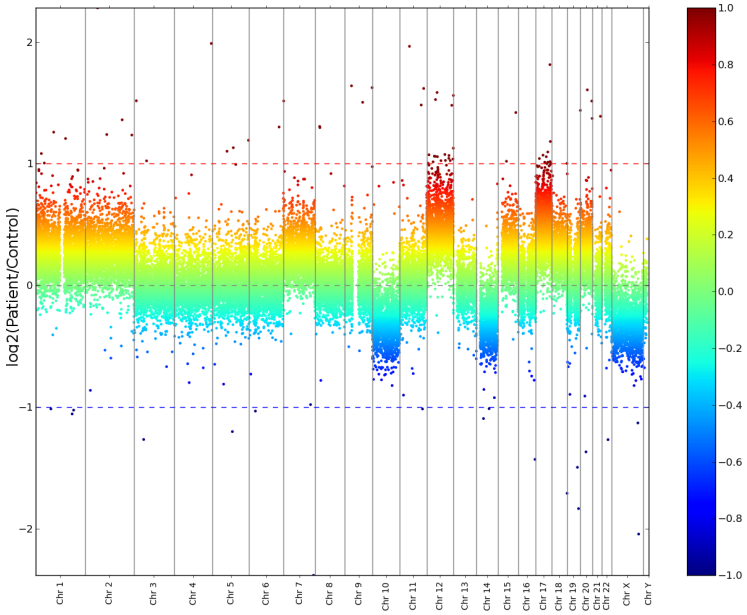


Figure 3.2: An example of aCGH signal profile throughout whole genome.

example gene expression, with aCGH it is possible to exploit the intrinsic structure of the data in order to improve the downstream analysis.

Many methods have been proposed for the extraction of CNVs based on different principles, like filtering (or smoothing), segmentation, breakpoint-detection and calling (Hupé et al., 2004; Olshen et al., 2004; Fridlyand et al., 2004; Willenbrock and Fridlyand, 2005; Wang et al., 2005; Price et al., 2005; Pique-Regi et al., 2008; Tibshirani and Wang, 2008), taking into account one sample at a time (see Lai et al., 2005, for a review)[1].

Especially in cancer diseases, where random mutations happen very frequently, joint-analysis of aCGH samples could be helpful to filter out unshared mutations between (at least a subset of) samples. One of the first works in that sense is the one by Pique-Regi et al. (2009) where they extend their previous model (Pique-Regi et al., 2008) to the multi-sample analysis. As for the single-sample segmentation, different approaches are adopted, usually extending the one by one algorithm to a multi-sample application (Mei et al., 2010; Zhang et al., 2010; de Ronde et al., 2010), and in some cases extending the approach also to the data joint-normalization (Picard et al., 2011).

Some interesting recent results exploit the possibility to adopt regularized methods for a joint segmentation of many aCGH profiles. The works proposed by Tian et al. (2012); Nowak et al. (2011); Vert and Bleakley (2010); Wang and Hu (2010) follow this

[1]Some algorithms for aCGH segmentation may be easily tested through the CGHWeb application by Lai et al. (2008) available at `http://compbio.med.harvard.edu/CGHweb`

stream, and are based on total variation (TV) or fused lasso signal approximation (see Section 3.3.1).

3.3 CGHDL: a new model for aCGH data analysis

In this section, we present a novel model for aCGH segmentation, namely CGHDL (Masecchia et al., 2013b), a dictionary learning method based on the minimization of a functional combining several penalties (see Section 3.3.1). In dictionary learning (Elad and Aharon, 2006), the original signal (e.g. an aCGH sample) is approximated by a linear weighted combination of the atoms, that are the elements of a learned dictionary. In our model, we assume that each sample uses just some atoms enforcing sparsity on the coefficient matrix.

Our method is an extension of the model proposed by Nowak et al. (2011), namely FLLat, addressing the following improvements:

- the segmentation is performed on a signal possibly composed of multiple chromosomes, still preserving independency among chromosomes;
- the coefficients are constrained to be positive, hence simplifying the interpretability of the coefficients matrix in favor of selecting more representative atoms, especially when co-occurent alterations take place.

The direct result of CGHDL is a denoised version of the input data as well as a representative dictionary of atoms. Each atom contains a meaningful *common pattern* of genomic alterations. Our model provides a more biologically sound representation of aCGH data, thanks to the combination of more complex penalties that explicitly exploit the structured nature of aCGH signals. Despite having a less simple model, we obtain atoms that possibly capture co-occurrences of CNVs across samples, leading to results that are more easily interpretable by domain experts. Moreover, our proposed model is able to deal with signals spanning the entire genome, whereas FLLat by Nowak et al. (2011) takes into account one chromosome at a time.

The proposed approach aims to extract latent features (atoms) and perform segmentation on aCGH data. The model is based on the minimization of a functional combining several penalties, properly designed to treat the whole genomic signal and to select more representative atoms. We propose to solve the optimization task by an inexact version of the alternating proximal algorithm originally studied by Attouch et al. (2010). A general analysis of the problem of computing *reliable* approximations of proximity operators for sums of composite functions is provided, which turns to be critical for the

effectiveness of the alternating proximal algorithm. In this respect, we extend some results given by Villa et al. (2012).

3.3.1 The proposed model

We are given $S \in \mathbb{N}$ samples $(\mathbf{y}_s)_{1\leq s\leq S}$, with $\mathbf{y}_s \in \mathbb{R}^L$. The objective is to find J *simple* atoms $(\boldsymbol{\beta}_j)_{1\leq j\leq J}$ which possibly give complete representation of all samples, in the sense that:

$$\mathbf{y}_s \approxeq \sum_{j=1}^{J} \theta_{js}\boldsymbol{\beta}_j \quad \forall s = 1, \ldots, S$$

for some vectors of coefficients $\boldsymbol{\theta}_s = (\theta_{js})_{i\leq j\leq J} \in \mathbb{R}^J$. The word *simple* in the problem statement, means that we want them to resemble *step functions*; more precisely we require each $\boldsymbol{\beta}_j$ to be of *small* total variation.

To achieve this, Nowak et al. (2011) proposed the following model:

$$\begin{aligned} \min_{\boldsymbol{\theta}_s,\boldsymbol{\beta}_j} & \sum_{s=1}^{S} \left\| \mathbf{y}_s - \sum_{i=1}^{J} \theta_{js}\boldsymbol{\beta}_j \right\|^2 + \lambda \sum_{j=1}^{J} \|\boldsymbol{\beta}_j\|_1 + \mu \sum_{j=1}^{J} TV(\boldsymbol{\beta}_j) \\ s.t. \quad & \|\theta_{j\cdot}\|_2^2 \leq 1 \quad \forall j = 1, \ldots, J, \end{aligned} \tag{3.1}$$

where the ℓ_1 term forces each atom $\boldsymbol{\beta}_j$ to be sparse and the total variation term

$$TV(\boldsymbol{\beta}_j) = \sum_{s=2}^{S} |\beta_{js} - \beta_{j,s-1}|,$$

induces small variations in the atoms[2]. The hard constraints on the coefficients $\theta_{\cdot j}$ are imposed for consistency and identifiability of the model. Indeed, multiplying a particular feature $\boldsymbol{\beta}_j$ by a constant, and dividing the corresponding coefficients by the same constant leaves the fit unchanged, but reduces the penalty.

Our model is an extension of (3.1) which aims to improve the interpretability of the atoms by the adoption of penalty that better capture the intrinsic properties of the data under analysis. First, we use a *weighted total variation*:

$$TV_{\mathbf{w}}(\boldsymbol{\beta}_j) = \sum_{l=1}^{L-1} w_l |\beta_{l+1,j} - \beta_{l,j}|\,,$$

where $\mathbf{w} = (w_l)_{1\leq l\leq L-1} \in \mathbb{R}^{L-1}$ is a given vector of weights. $TV_{\mathbf{w}}$ is a generalized total variation due the the presence of the weights $\mathbf{w}$. A simple observation is that there is

[2]The combination of an ℓ_1 (or *lasso*) and a TV penalty is also called *fused lasso*.

no biological meaning to force *continuity* between the last probe of a chromosome and the first probe on the next one. For this reason, in Nowak et al. (2011) the algorithm is run chromosome-by-chromosome. However, this solution has an evident drawback: it does not allow to directly identify recurrent alterations, occurring on two different chromosomes (*e.g.,* due to an unbalanced translocation).

Then, the aCGH data analysis is driven by the following optimization problem depending on the three regularization parameters $\lambda, \mu, \tau > 0$,

$$\begin{aligned} \min_{\boldsymbol{\theta}_s, \boldsymbol{\beta}_j} \sum_{s=1}^{S} \left\| \mathbf{y}_s - \sum_{i=1}^{J} \theta_{js} \boldsymbol{\beta}_j \right\|^2 + \lambda \sum_{j=1}^{J} \|\boldsymbol{\beta}_j\|_1^2 + \mu \sum_{j=1}^{J} TV_{\mathbf{w}}(\boldsymbol{\beta}_j) + \tau \sum_{s=1}^{S} \|\boldsymbol{\theta}_s\|_1^2 \\ \text{s.t. } 0 \leq \theta_{js} \leq \theta_{\max}, \quad \forall j = 1, \ldots, J \quad \forall s = 1, \ldots, L. \end{aligned} \tag{3.2}$$

This problem can be put in matrix form. In fact, if we define the matrices

$$\begin{aligned} \mathbf{B} &= \begin{bmatrix} \boldsymbol{\beta}_1 & \boldsymbol{\beta}_2 & \cdots & \boldsymbol{\beta}_J \end{bmatrix} \in \mathbb{R}^{L \times J}, \\ \mathbf{Y} &= \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_S \end{bmatrix} \in \mathbb{R}^{L \times S}, \\ \boldsymbol{\Theta} &= \begin{bmatrix} \boldsymbol{\theta}_1 & \boldsymbol{\theta}_2 & \cdots & \boldsymbol{\theta}_S \end{bmatrix} \in \mathbb{R}^{J \times S}, \end{aligned}$$

then (3.2) turns into

$$\min_{\mathbf{B}, \boldsymbol{\Theta}} \frac{1}{2} \|\mathbf{Y} - \mathbf{B}\boldsymbol{\Theta}\|_F^2 + h(\boldsymbol{\Theta}) + g(\mathbf{B}), \tag{3.3}$$

where

$$g(\mathbf{B}) = \lambda \sum_{j=1}^{J} \|\mathbf{B}(:, j)\|_1^2 + \mu \sum_{j=1}^{J} TV_{\mathbf{w}}(\mathbf{B}(:, j)), \tag{3.4}$$

$$h(\boldsymbol{\Theta}) = \delta_{\Delta^{J \times S}}(\boldsymbol{\Theta}) + \tau \sum_{s=1}^{S} \|\boldsymbol{\Theta}(:, s)\|_1^2 \tag{3.5}$$

and $\delta_{\Delta^{J \times S}}$ denoting the indicator function of the box set $\Delta^{J \times S} = [0, \theta_{max}]^{J \times S}$.

This model improves the one used in Nowak et al. (2011) in several aspects. First, it employs the penalization terms

$$\lambda \sum_{i=1}^{J} \|\mathbf{B}(:, j)\|_1^2, \quad \tau \sum_{s=1}^{S} \|\boldsymbol{\Theta}(:, s)\|_1^2,$$

which are sums of squares of ℓ_1 norms (mixed norms). This possibly forces a structured sparsity only along the columns of the matrix of atoms $\mathbf{B}$ and of coefficients $\boldsymbol{\Theta}$. Such choice is more faithful to the actual purposes of dictionary learning, and contrasts with

the majority of similar models, including the one proposed by Nowak et al. (2011), that instead uses a simple ℓ_1 norm that induces just a global sparsity on the matrices.

Secondly, in problem (3.3), the total variation term is indeed a generalized total variation due the the presence of the weights $\mathbf{w}$. This modification is introduced in order to relax at some points the constraint of *small jumps* on the atoms. In the aCGH data, this permits to treat the signal of the genome as a whole, still guaranteeing independency among chromosomes.

Finally, we constrain the coefficients to be positive. This reduces the complexity of the matrix of coefficients $\mathbf{\Theta}$ and forces the matrix of atoms $\mathbf{B}$ to be more informative: *e.g.*, for deletions and amplifications occurring in different samples but on the same locus on the chromosome different atoms may be selected. Ultimately the interpretability of the results is improved.

In the implementation it is customary to normalize the problem as follows

$$\min_{\mathbf{B},\mathbf{\Theta}} \frac{1}{2SL} \|\mathbf{Y} - \mathbf{B}\mathbf{\Theta}\|_F^2 + \frac{\lambda}{JL} \sum_{j=1}^{J} \|\mathbf{B}(:,j)\|_1^2 + \frac{\mu}{JL} \sum_{j=1}^{J} TV_{\mathbf{w}}(\mathbf{B}(:,j)) + \frac{\tau}{SJ} \sum_{s=1}^{S} \|\mathbf{\Theta}(:,s)\|_1^2, \tag{3.6}$$

and multiplying by SL

$$\min_{\mathbf{B},\mathbf{\Theta}} \frac{1}{2} \|\mathbf{Y} - \mathbf{B}\mathbf{\Theta}\|_F^2 + \frac{\lambda S}{J} \sum_{j=1}^{J} \|\mathbf{B}(:,j)\|_1^2 + \frac{\mu S}{J} \sum_{j=1}^{J} TV_{\mathbf{w}}(\mathbf{B}(:,j)) + \frac{\tau L}{J} \sum_{s=1}^{S} \|\mathbf{\Theta}(:,s)\|_1^2, \tag{3.7}$$

which has again the form (3.2) as soon as we make the substitution $\lambda \leftrightarrow \lambda S/J$, $\mu \leftrightarrow \mu S/J$ and $\tau \leftrightarrow \tau L/J$. Form (3.7) is the one that we actually consider in the implementation.

3.3.2 An alternating proximal algorithm

In this section we describe our proposed algorithm for solving the aCGH model (3.3). The standard algorithmic scheme widely adopted to solve dictionary learning problems is the so called *alternating minimization*. Considering the objective function

$$\varphi(\mathbf{\Theta}, \mathbf{B}) = \frac{1}{2} \|\mathbf{Y} - \mathbf{B}\mathbf{\Theta}\|_F^2 + h(\mathbf{\Theta}) + g(\mathbf{B}), \tag{3.8}$$

and the partial functions $\varphi(\cdot, \mathbf{B})$ and $\varphi(\mathbf{\Theta}, \cdot)$, the minimization scheme (in Algorithm 3.1) alternatively minimize with respect to $\mathbf{\Theta}$ and $\mathbf{B}$ until convergence is reached. This algorithm may provide satisfactory results in practice. However, it is well known that such an alternating minimization procedure requires quite restrictive conditions to guarantee convergence to a local minimizer (Chan and Wong, 2000).

Algorithm 3.1 Alternating minimization algorithm

Require: $\boldsymbol{\Theta}_0$, $\mathbf{B}_0$
  $k \leftarrow 0$
  repeat
    $\boldsymbol{\Theta}_{k+1} = \operatorname{argmin}_{\boldsymbol{\Theta}} \varphi(\cdot, \mathbf{B}_k)$
    $\mathbf{B}_{k+1} = \operatorname{argmin}_{\mathbf{B}} \varphi(\boldsymbol{\Theta}_{k+1}, \cdot)$
    $k \leftarrow k + 1$
  until convergence on $\boldsymbol{\Theta}_k$ and $\mathbf{B}_k$

Algorithm 3.2 Inexact alternating proximal algorithm

Require: $\boldsymbol{\Theta}_0$, $\mathbf{B}_0$, $\eta_k, \zeta_k \in [\rho_1, \rho_2]$, $\varepsilon_k \downarrow 0$, and $0 < \rho_1 \leq \rho_2$
  $k \leftarrow 0$
  repeat
    $\boldsymbol{\Theta}_{k+1} \approx_{\varepsilon_k} \operatorname{prox}_{\eta_k \varphi(\cdot, \mathbf{B}_k)}(\boldsymbol{\Theta}_k)$
    $\mathbf{B}_{k+1} \approx_{\varepsilon_k} \operatorname{prox}_{\zeta_k \varphi(\boldsymbol{\Theta}_{k+1}, \cdot)}(\mathbf{B}_k)$
    $k \leftarrow k + 1$
  until convergence on $\boldsymbol{\Theta}_k$ and $\mathbf{B}_k$

We propose an *inexact alternating proximal algorithm* described by Algorithm 3.2. In (Attouch et al., 2010) a deep analysis of this algorithm is presented for general functions, but with exact computation of the proximity operators ($\varepsilon_k = 0$). In that case the following result can be worked out (Attouch et al., 2010, see Lemma 3.1 and Theorem 3.2)),

Theorem 3.1. *If $(\boldsymbol{\Theta}_0, \mathbf{B}_0) \in \mathbb{R}^{J \times S} \times \mathbb{R}^{L \times J}$, and η_k, ζ_k and $(\boldsymbol{\Theta}_k, \mathbf{B}_k)_{k \in \mathbb{N}}$ are defined according to the Algorithm 3.2 with $\varepsilon_k = 0$, then $(\varphi(\boldsymbol{\Theta}_k, \mathbf{B}_k))_{k \in \mathbb{N}}$ is decreasing and $(\boldsymbol{\Theta}_k, \mathbf{B}_k)_{k \in \mathbb{N}}$ converges to a critical point of φ.*

Until now, in case of inexactness ($\varepsilon_k > 0$) no convergence result is available. Thus, computing the proximity operators in Algorithm 3.2, with high and verifiable precision becomes critical.

We propose an algorithm for the computation of the proximity operators of the partial functions $\varphi(\cdot, \mathbf{B}_k)$ and $\varphi(\boldsymbol{\Theta}_{k+1}, \cdot)$ that fulfills this requirement. The first step towards that purpose is to exploit the structure of those functions. We define

$$\begin{aligned} \omega &: \mathbb{R}^{L \times S} \to \mathbb{R}\,, \quad \omega(\mathbf{Z}) = 1/2 \left\| \mathbf{Y} - \mathbf{Z} \right\|_F^2 \\ \omega_J &:= \tau \left\| \cdot \right\|_1^2 : \mathbb{R}^J \to \mathbb{R} \\ \omega_L &:= \lambda \left\| \cdot \right\|_1^2 : \mathbb{R}^L \to \mathbb{R} \\ \chi_l &:= \mu w_l |\cdot| : \mathbb{R} \to \mathbb{R} \quad \text{for } l = 1, \ldots, L-1\,, \end{aligned}$$

and the linear maps

$$Q_{\mathbf{B}} : \mathbb{R}^{J\times S} \to \mathbb{R}^{L\times S}, \quad Q_{\mathbf{B}}(\mathbf{\Theta}) = \mathbf{B}\mathbf{\Theta},$$
$$T_{\mathbf{\Theta}} : \mathbb{R}^{L\times J} \to \mathbb{R}^{L\times S}, \quad T_{\mathbf{\Theta}}(\mathbf{B}) = \mathbf{B}\mathbf{\Theta},$$

together with the discrete derivative $D : \mathbb{R}^L \to \mathbb{R}^{L-1}$, such that

$$(D\boldsymbol{\beta})_l = \beta_{l+1} - \beta_l \text{ for } l \leq L-1\,.$$

Moreover, we need to consider the following (canonical) projections

$$\begin{aligned}
&\mathrm{pr}_j^{L\times J} : \mathbb{R}^{L\times J} \to \mathbb{R}^L, \quad \mathrm{pr}_j^{L\times J}\mathbf{B} = \mathbf{B}(:,j) \quad \text{(along columns)},\\
&\mathrm{pr}_s^{J\times S} : \mathbb{R}^{J\times S} \to \mathbb{R}^J, \quad \mathrm{pr}_s^{J\times S}\mathbf{\Theta} = \mathbf{\Theta}(:,s) \quad \text{(along columns)},\\
&\hat{\mathrm{pr}}_j^{J\times S} : \mathbb{R}^{J\times S} \to \mathbb{R}^S, \quad \hat{\mathrm{pr}}_j^{J\times S}\mathbf{\Theta} = \mathbf{\Theta}(j,:) \quad \text{(along rows)},\\
&\mathrm{pr}_l^{L-1} : \mathbb{R}^{L-1} \to \mathbb{R}, \qquad \mathrm{pr}_l^{L-1}(D\boldsymbol{\beta}) = (D\boldsymbol{\beta})_l \quad \text{(extraction)},
\end{aligned}$$

Then, the partial functions $\varphi(\cdot, \mathbf{B})$ and $\varphi(\mathbf{\Theta}, \cdot)$ can be respectively written as

$$\varphi(\cdot, \mathbf{B}) = \omega(Q_{\mathbf{B}}(\mathbf{\Theta})) + \delta_{\Delta^{J\times S}}(\mathbf{\Theta}) + \sum_{s=1}^{S} \omega_J(\mathrm{pr}_s^{J\times S}\mathbf{\Theta}) \tag{3.9}$$

$$\varphi(\mathbf{\Theta}, \cdot) = \omega(T_{\mathbf{\Theta}}(\mathbf{B})) + \sum_{j=1}^{J} \omega_L(\mathrm{pr}_j^{L\times J}\mathbf{B}) + \sum_{j=1}^{J}\sum_{l=1}^{L-1} \chi_l(\mathrm{pr}_l^{L-1} \circ D \circ \mathrm{pr}_j^{L\times J}\mathbf{B})\,. \tag{3.10}$$

3.3.3 Proximity operator of composite penalties

In view of the proximal alternating algorithm presented in this section, it is desirable to compute *reliable* approximations of proximity operators for penalties which are sums of composite functions. In this section we answer that issue in a general setting. Then, we also present algorithms tailored to the Algorithm 3.2.

3.3.3.1 The general theory

Let $g : \mathcal{H} \to \overline{\mathbb{R}}$ be a function of the following form

$$g(x) = \sum_{i=1}^{m} \omega_i(A_i x)\,,$$

where $A_i : \mathcal{H} \to \mathcal{G}_i$ are bounded linear operators between Hilbert spaces and $\omega_i : \mathcal{G}_i \to \overline{\mathbb{R}}$ are proper convex and lower semi-continuous functions. Our purpose is to show how to compute the proximity operator $\operatorname{prox}_{\lambda g} : \mathcal{H} \to \mathcal{H}$ for $\lambda > 0$, in terms of the mappings A_i and the proximity operators of ω_i.

First of all, we note that g is actually of the form

$$g(x) = \omega(Ax)\,, \tag{3.11}$$

for a suitable operator $A : \mathcal{H} \to \mathcal{G}$ and $\omega : \mathcal{G} \to \overline{\mathbb{R}}$. Indeed, it is sufficient to consider the direct sum of the Hilbert spaces $(\mathcal{G}_i)_{1\leq i\leq m}$

$$\mathcal{G} := \bigoplus_{i=1}^{m} \mathcal{G}_i \quad \langle \mathbf{u}, \mathbf{v}\rangle_{\mathcal{G}} := \sum_{i=1}^{m} \langle u_i, v_i\rangle_{\mathcal{G}_i}\,,$$

define the operator $A : \mathcal{H} \to \mathcal{G}$, $Ax = (A_i x)_{1\leq i\leq m}$ and the function

$$\omega : \mathcal{G} \to \overline{\mathbb{R}}\,, \quad \omega(\mathbf{v}) = \sum_{i=1}^{m} \omega_i(v_i).$$

Then, computing $\operatorname{prox}_{\lambda g}(y)$ aims to solve the following minimization problem

$$\min_{x\in\mathcal{H}} \omega(Ax) + \frac{1}{2\lambda}\,\|x-y\|_{\mathcal{H}}^2 := \Phi_\lambda(x)\,. \tag{3.12}$$

Its dual problem, in the sense of Fenchel-Rockafellar duality (Zălinescu, 2002), is

$$\min_{\mathbf{v}\in\mathcal{G}} \frac{1}{2\lambda}\,\|y-\lambda A^*\mathbf{v}\|_{\mathcal{H}}^2 + \omega^*(\mathbf{v}) - \frac{1}{2\lambda}\,\|y\|_{\mathcal{H}}^2 := \Psi_\lambda(\mathbf{v})\,. \tag{3.13}$$

The adjoint operator $A^* : \mathcal{G} \to \mathcal{H}$ and the Fenchel conjugate $\omega^* : \mathcal{G} \to \overline{\mathbb{R}}$ are both decomposable. Indeed it is easy to see that for every $\mathbf{v} = (v_i)_{1\leq m} \in \mathcal{G}$, it holds

$$A^*\mathbf{v} = \sum_{i=1}^{m} A_i^* v_i\,, \quad \omega^*(\mathbf{v}) = \sum_{i=1}^{m} \omega_i^*(v_i)\,. \tag{3.14}$$

From the separability of ω^*, it follows that for $\gamma > 0$

$$\begin{aligned} \operatorname*{prox}_{\gamma\omega^*}(\mathbf{v}) &= \operatorname*{argmin}_{u\in\mathcal{G}} \left\{ \omega^*(\mathbf{u}) + \frac{1}{2\gamma}\,\|\mathbf{u}-\mathbf{v}\|_{\mathcal{G}}^2 \right\} \\ &= \left(\operatorname*{prox}_{\gamma\omega_i^*}(v_i) \right)_{1\leq i\leq m}\,, \end{aligned} \tag{3.15}$$

meaning that the proximity operator of ω^* can be computed component-wise.

In Villa et al. (2012), the following inexact notion of proximal point is studied

$$z \approx_\varepsilon \operatorname*{prox}_{\lambda g}(y) \iff \frac{y-z}{\lambda} \in \partial_{\frac{\varepsilon^2}{2\lambda}} g(z) \tag{3.16}$$

where $\partial_\varepsilon g(z)$ is the ε-subdifferential of g (Zălinescu, 2002). There, it is also proved (Proposition 2.2) that approximations in the sense (3.16), can be obtained by controlling the duality gap. More precisely if one set $z_{\mathbf{v}} = y - \lambda A^* \mathbf{v} = y - \lambda \sum_{i=1}^m A_i^* v_i \in \mathcal{H}$, (the *primal variable*), it holds

$$G(z_{\mathbf{v}}, v_1, \dots, v_m) \leq \frac{\varepsilon^2}{2\lambda} \implies z_{\mathbf{v}} \approx_\varepsilon \operatorname*{prox}_{\lambda g}(y)\,, \tag{3.17}$$

where the *duality gap* G is defined for every $(z, \mathbf{v}) \in \mathcal{H} \times \mathcal{G}$ as

$$G(z, \mathbf{v}) := \Phi_\lambda(z) + \Psi_\lambda(\mathbf{v}) = \sum_{i=1}^m \omega_i(A_i z) + \omega_i^*(v_i) + \frac{1}{\lambda}\langle z - y, z\rangle + \frac{1}{2\lambda}\big(\,\|z_{\mathbf{v}}\|^2 - \|z\|^2\,\big)\,. \tag{3.18}$$

Theorem 6.1 by Villa et al. (2012), states that to minimize the duality gap (3.18), one can minimize the dual problem (3.13), if the following condition is satisfied

$$\operatorname{dom}\omega = \mathcal{G}\,. \tag{3.19}$$

In that case, *any* algorithm that produces a minimizing sequence for the dual problem (3.13) yields an ε-approximation of $\operatorname{prox}_{\lambda g}(y)$. More precisely, if $(\mathbf{v}_n)_{n\in\mathbb{N}}$ is such that $\Psi_\lambda(\mathbf{v}_n) \to \inf \Psi_\lambda$, then $z_{\mathbf{v}_n} \in \operatorname{dom}\Phi_\lambda$, $z_{\mathbf{v}_n} \to \hat{z}$ and $G(z_n, \mathbf{v}_n) \to 0$. Therefore for every $\varepsilon > 0$, we can stop when $G(z_n, \mathbf{v}_n) \leq \varepsilon^2/(2\lambda)$ and, because of (3.17), be sure to get close to $\operatorname{prox}_{\lambda g}(y)$ with precision ε.

However, in general, if condition (3.19) is not satisfied, by minimizing the dual problem (3.13), it might end up with primal variables $z_n = y - \lambda A^* \mathbf{v}_n$ that do not belong to $\operatorname{dom}\Phi_\lambda$ and possibly $G(z_n, \mathbf{v}_n) = +\infty$.

Note that condition (3.19) keeps out hard constraints, as the one considered in problem (3.3). Here we give a generalization of Theorem 6.1 cited above (by Villa et al., 2012), that allows to get proper approximation of the proximal point, under weaker condition then (3.19), and ultimately to treat all the constraints of problem (3.3).

Proposition 3.1. *Assume* $\operatorname{dom}\omega$ *closed and* $\omega_{|\operatorname{dom}\omega}$ *continuous. Moreover suppose* ω *is continuous in* Ax_0 *for some* x_0[3]*. Let* $(\mathbf{v}_n)_{n\in\mathbb{N}}$ *be such that* $\Psi_\lambda(\mathbf{v}_n) \to \inf \Psi_\lambda$*. Then defining*

$$\hat{z}_n = P_{\operatorname{dom}\Phi_\lambda}(y - \lambda A^* \mathbf{v}_n)\,, \tag{3.20}$$

[3]This hypothesis is needed just to ensure $\inf \Phi_\lambda + \inf \Psi_\lambda = 0$. However weaker conditions can be employed, as $0 \in \operatorname{sri}(R(A) - \operatorname{dom}\omega)$ (Zălinescu, 2002).

where $P_{\mathrm{dom}\,\Phi_\lambda}$ is the projection operator onto the closed convex set $\mathrm{dom}\,\Phi_\lambda = A^{-1}(\mathrm{dom}\,\omega)$, one has

$$\Phi_\lambda(\hat{z}_n) - \inf \Phi_\lambda \leq G(\hat{z}_n, \mathbf{v}_n) \to 0 \tag{3.21}$$

Proof. Set $\hat{z} = \mathrm{prox}_{\lambda g}(y)$, which is the solution of the primal problem (3.12), let $\hat{\mathbf{v}}$ be a solution of the dual problem (3.13) and $z_n = y - \lambda A^* \mathbf{v}_n$. One can prove (Villa et al., 2012, see Theorem 6.1) that

$$\frac{1}{2\lambda} \|z_n - \hat{z}\|^2 \leq \Phi_\lambda(\mathbf{v}_n) - \Phi_\lambda(\hat{\mathbf{v}})\,.$$

Since $P_{\mathrm{dom}\ \Phi_\lambda}$ is continuous, we have

$$\hat{z}_n = P_{\mathrm{dom}\ \Phi_\lambda}(z_n) \to P_{\mathrm{dom}\ \Phi_\lambda}(\hat{z}) = \hat{z}$$

and hence $A\hat{z}_n \to A\hat{z}$, $A\hat{z}_n, A\hat{z} \in \mathrm{dom}\ \omega$. Now from the continuity of $\omega_{|\mathrm{dom}\,\omega}$ it follows $\Phi_\lambda(\hat{z}_n) \to \Phi_\lambda(\hat{z}) = \inf \Phi_\lambda$. Since $\Phi(\hat{z}) = -\Psi(\hat{\mathbf{v}})$, we have

$$\begin{aligned} G(\hat{z}_n, \mathbf{v}_n) &= \Phi_\lambda(\hat{z}_n) + \Psi_\lambda(\mathbf{v}_n) \\ &= \underbrace{\Phi_\lambda(\hat{z}_n) - \Phi_\lambda(\hat{z})}_{\geq 0} + \underbrace{\Psi_\lambda(\mathbf{v}_n) - \Psi_\lambda(\hat{\mathbf{v}})}_{\geq 0} \to 0\,. \end{aligned} \tag{3.22}$$

and the statements follows. □

Proposition 3.1 shows that, for every $\varepsilon > 0$, if one stops the algorithm when $G(\hat{z}_n, \mathbf{v}_n) \leq \varepsilon^2/(2\lambda)$, then

$$\Phi_\lambda(\hat{z}_n) \leq \inf \Phi_\lambda + \frac{\varepsilon^2}{2\lambda}, \tag{3.23}$$

obtaining another kind of approximation of the prox, i.e. in the sense of equation (2.13) in (Villa et al., 2012). We remark that both criteria (3.16) and (3.23) give $\left\|z - \mathrm{prox}_{\lambda g}(y)\right\| \leq \varepsilon$.

We are thus justified in solving the dual problem (3.13) by any algorithm that provides just a minimizing sequence. We underline that for the dual problem no convergence on the minimizers is required, but convergence in value is sufficient. Taking advantage of the splitting properties (3.14)-(3.15), in Algorithm 3.3 we present a generalization, to sum of composite functions, of the algorithm given by Villa et al. (2012, see Section 6.1)

If ω_i is positively homogeneous, it holds $\omega_i^* = \delta_{S_i}$, the indicator function $S_i = \delta\omega_i(0)$ and $\mathrm{dom}\,\omega_i = \mathcal{G}_i$. Hence $\mathrm{prox}_{\gamma_n \omega_i^*} = P_{S_i}$ and $v_{n+1,i}$ is computed by

$$v_{n+1,i} = P_{S_i}(u_{n,i} + \gamma_n A_i z_n)\,. \tag{3.24}$$

Algorithm 3.3 Prox of composite functions algorithm

Require: $\mathbf{u}_0 \leftarrow \mathbf{v}_0 \leftarrow 0, t_0 \leftarrow 1, n \leftarrow 0$
  repeat
    $z_{tmp} \leftarrow y - \lambda \sum_{i=1}^{m} A_i^* u_{n,i}$
    $0 < \gamma_n \leq (\lambda \|A\|^2)^{-1}$
    for $i = 1 \rightarrow m$ **do**
      $v_{n+1,i} \leftarrow \operatorname{prox}_{\gamma_n \omega_i^*}\left(u_{n,i} + \gamma_n A_i z_{tmp}\right)$
    end for
    $z_{n+1} \leftarrow y - \lambda \sum_{i=1}^{m} A_i^* v_{n+1,i}$
    $t_{n+1} \leftarrow \frac{1+\sqrt{1+4t_n^2}}{2}$
    $u_{n+1,i} \leftarrow v_{n+1,i} + \frac{t_n - 1}{t_{n+1}}(v_{n+1,i} - v_{n.i})$
    $n \leftarrow n + 1$
  until convergence

If $\omega_i = \delta_{S_i}$ for a closed convex set $S_i \subseteq \mathcal{G}_i$, then $\omega_i^* = \sigma_{S_i}$ the support function of S_i and using the Moreau decomposition formula

$$\operatorname*{prox}_{\gamma_n \omega_i^*}(y) = y - P_{\gamma_n S_i} y,$$

the vectors $v_{n+1,i}$ in Algorithm 3.3 can be computed by the formula

$$v_{n+1,i} = (I - P_{\gamma_n S_i})\big(u_{n,i} + \gamma_n A_i z_n\big)\,. \tag{3.25}$$

Note that in this case $\omega_{i|\operatorname{dom}\omega_i}$ is continuous.

3.3.3.2 The computation of $\operatorname{prox}_{\eta_k \varphi(\cdot, \mathbf{B}_k)}(\mathbf{\Theta}_k)$

We refer to notation established in Section 3.3.2. It is simple to show that for $\gamma_n > 0$

$$\operatorname*{prox}_{\gamma_n \omega^*}(\mathbf{Z}) = \frac{1}{1+\gamma_n}\Big(\mathbf{Z} - \gamma_n \mathbf{Y}\Big)\,. \tag{3.26}$$

The dual variables are

$$(\mathbf{V}^{(i)})_{1 \leq i \leq 3}, (\mathbf{U}^{(i)})_{1 \leq i \leq 3}, \in \mathbb{R}^{L \times S} \times \mathbb{R}^{J \times S} \times \underbrace{\mathbb{R}^{J \times S}}_{(\mathbb{R}^J)^S}\,.$$

and the update for the corresponding primal variable is

$$\mathbf{\Gamma} = \mathbf{\Theta}_k - \eta_k(\mathbf{B}_k^* \mathbf{V}^{(1)} + \mathbf{V}^{(2)} + \mathbf{V}^{(3)})\,. \tag{3.27}$$

Algorithm 3.4 $\operatorname{prox}_{\eta_k \varphi(\cdot, \mathbf{B}_k)}(\boldsymbol{\Theta}_k)$ algorithm

Require: $\mathbf{V}^{(i)} \leftarrow \mathbf{U}^{(i)} \leftarrow 0, n \leftarrow 0$
  repeat
    $\boldsymbol{\Gamma}_{tmp} \leftarrow \boldsymbol{\Theta}_k - \eta_k(\mathbf{B}_k^* \mathbf{U}_n^{(1)} + \mathbf{U}_n^{(2)} + \mathbf{U}_n^{(3)})$
    $\mathbf{V}_{n+1}^{(1)} \leftarrow (1+\gamma_n)^{-1}\big(\mathbf{U}_n^{(1)} + \gamma_n(\mathbf{B}_k \boldsymbol{\Gamma}_{tmp} - \mathbf{Y})\big)$
    $\mathbf{V}_{n+1}^{(2)} \leftarrow (I - P_{\gamma_n \Delta^{J\times S}})\big(\mathbf{U}_n^{(2)} + \gamma_n \boldsymbol{\Gamma}_{tmp}\big)$

    for $s = 1 \to S$ **do**
      $\mathbf{V}_{n+1}^{(3)}(:, s) \leftarrow \operatorname{prox}_{\gamma_n \omega_J^*}\big(\mathbf{U}_n^{(3)}(:, s) + \gamma_n \boldsymbol{\Gamma}_{tmp}(:, s)\big)$
    end for
    $\boldsymbol{\Gamma}_{n+1} \leftarrow \boldsymbol{\Theta}_k - \eta_k(\mathbf{B}_k^* \mathbf{V}_{n+1}^{(1)} + \mathbf{V}_{n+1}^{(2)} + \mathbf{V}_{n+1}^{(3)})$
    $t_{n+1} \leftarrow \frac{1+\sqrt{1+4t_n^2}}{2}$
    for $i = 1 \to 3$ **do**
      $\mathbf{U}_{n+1}^{(i)} \leftarrow \mathbf{V}_{n+1}^{(i)} + \frac{t_n - 1}{t_{n+1}}(\mathbf{V}_{n+1}^{(i)} - \mathbf{V}_n^{(i)})$
    end for
    $n \leftarrow n + 1$
  until convergence

Algorithm 3.3, becomes Algorithm 3.4, where

$$\operatorname*{prox}_{\gamma_n \omega_J^*}(\boldsymbol{\beta}) = \boldsymbol{\beta} - \gamma_n \operatorname*{prox}_{\gamma_n^{-1}\omega_J}(\gamma_n^{-1}\boldsymbol{\beta})$$

and $\operatorname{prox}_{\gamma_n^{-1}\omega_J}$ can be computed with the following finite procedure by Kowalski and Torrésani (2009):

1. order the components of the vector $\boldsymbol{\beta}$ such that $|\beta_1| \geq \cdots \geq |\beta_J|$;
2. determine the number j such that

$$(\gamma_n + 2\lambda j)|\beta_{l+1}| \leq 2\lambda \left\|\boldsymbol{\beta}(1:j)\right\|_1,$$
$$(\gamma_n + 2\lambda j)|\beta_j| > 2\lambda \left\|\boldsymbol{\beta}(1:j)\right\|_1$$

3. set $\rho = 2\lambda/(\gamma_n + 2\lambda j)\left\|\boldsymbol{\beta}(1:j)\right\|_1$, then compute $\operatorname{prox}_{\gamma_n^{-1}\omega_J}(\gamma_n^{-1}\boldsymbol{\beta})$ component-wise by $(\gamma_n^{-1} S_\rho(\beta_j))_{1\leq j\leq J}$, where $S_\rho(\beta) = \operatorname{sgn}(\beta)(|\beta| - \rho)_+$ is the soft-thresholding operator.

In Section 3.3.3, we saw that a reasonable stopping criterion for the above algorithm is obtained by controlling the duality gap. We also remark that in this case $\operatorname{dom} \Phi_\lambda = \Delta^{J\times S}$.

Fenchel conjugate of $\lambda \|\cdot\|_1^2$

Let us consider the function $\omega_L = \lambda \|\cdot\|_1^2 : \mathbb{R}^L \to \mathbb{R}$. We shall compute the conjugate function

$$\omega_L^*(u) = \sup_{y \in \mathbb{R}^L} (\langle u, y \rangle - \omega_L(y)) = - \inf_{y \in \mathbb{R}} (\omega_L(y) - \langle u, y \rangle) \tag{3.28}$$

Clearly $\omega_L^*(u) = \langle u, z \rangle - \omega_L(z)$, where z is a minimizer of $\omega_L - \langle u, \cdot \rangle)$, and $z \in \operatorname{argmin}(\omega_L - \langle u, \cdot \rangle) \iff u \in \partial \omega_L(z)$. The subdifferential of ω_L is

$$\partial \omega_L(z) = 2\lambda \|z\|_1 \, \partial \|\cdot\|_1 (z) = 2\lambda \|z\|_1 \prod_{l=1}^{L} \partial |\cdot|(z_l) \, . \tag{3.29}$$

Therefore

$$u \in \partial \omega_L(z) \iff \begin{cases} u_l = 2\lambda \|z\|_1 \operatorname{sgn}(z_l) & \text{if } z_l \neq 0 \\ |u_l| \leq 2\lambda \|z\|_1 & \text{if } z_l = 0 \, . \end{cases} \tag{3.30}$$

If we take the index $k \in [1, L]$ such that $|u_k| = \max_{1 \leq l \leq L} |u_l|$ and define

$$z_l = \begin{cases} u_k/(2\lambda) & \text{if } l = k \\ 0 & \text{if } l \neq k \, , \end{cases}$$

then it holds $2\lambda \|z\|_1 = |u_k| \geq |u_l|$ and the right hand side of (3.30) is satisfied. In the end we find that

$$\omega_L^*(u) = \langle u, z \rangle - \omega_L(z) = \frac{u_k^2}{2\lambda} - \frac{u_k^2}{4\lambda} = \frac{1}{4\lambda} \max_{1 \leq l \leq L} u_l^2 \, .$$

Taking into account (3.18) and (3.9), for $\mathbf{\Gamma} \in \Delta^{J\times S}$, it is

$$
\begin{aligned}
&G(\mathbf{\Gamma}, \mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \mathbf{V}^{(3)}) \\
&= \frac{1}{2}\left\|\mathbf{Y} - \mathbf{B}_k\mathbf{\Gamma}\right\|_F^2 + \sum_{s=1}^{S} \tau \left\|\mathbf{\Gamma}(:,s)\right\|_1^2 \\
&+ \frac{1}{2}\left\|\mathbf{V}^{(1)}\right\|_F^2 + \langle \mathbf{V}^{(1)}, \mathbf{Y}\rangle + \theta_{max}\sum_{j=1}^{J}\sum_{s=1}^{S}\left(v^{(2)}(j,s)\right)_+ \\
&+ \sum_{s=1}^{S}\frac{1}{4\tau}\max_{1\le j\le J}(v^{(3)}(j,s))^2 + \frac{1}{\eta_k}\langle \mathbf{\Gamma} - \mathbf{\Theta}_k, \mathbf{\Gamma}\rangle \\
&+ \frac{1}{2\eta_k}\Big(\left\|\mathbf{\Theta}_k - \eta_k(\mathbf{B}_k^*\mathbf{V}^{(1)} + \mathbf{V}^{(2)} + \mathbf{V}^{(3)})\right\|^2 - \left\|\mathbf{\Gamma}\right\|^2\Big)\,.
\end{aligned}
$$

The stopping criterion is

$$
G(P_{\Delta^{J\times S}}(\mathbf{\Gamma}_n), \mathbf{V}_n^{(1)}, \mathbf{V}_n^{(2)}, \mathbf{V}_n^{(3)}) \le \frac{\varepsilon^2}{2\eta_k}\,, \tag{3.31}
$$

and in that case, setting $\mathbf{\Theta}_{k+1} = P_{\Delta^{J\times S}}(\mathbf{\Gamma}_n)$, it holds $\mathbf{\Theta}_{k+1} \approxeq_\varepsilon \operatorname{prox}_{\eta_k\varphi_{\mathbf{B}_k}}(\mathbf{\Theta}_k)$ in the sense of (3.23).

3.3.3.3 The computation of $\operatorname{prox}_{\zeta_k\varphi(\mathbf{\Theta}_{k+1},\cdot)}(\mathbf{B}_k)$

The dual variables are

$$
(\mathbf{V}^{(i)})_{1\le i\le 3}, (\mathbf{U}^{(i)})_{1\le i\le 3} \in \mathbb{R}^{L\times S} \times \underbrace{\mathbb{R}^{L\times J}}_{(\mathbb{R}^L)^J} \times \mathbb{R}^{(L-1)\times J}\,,
$$

and the rule for the update of the primal variable is

$$
\mathbf{Z} = \mathbf{B}_k - \zeta_k\Big(\mathbf{V}^{(1)}\mathbf{\Theta}_{k+1}^* + \mathbf{V}^{(2)} + \sum_{j=1}^{J}\operatorname{pr}_j^* D^*\mathbf{V}^{(3)}(:,j)\Big)\,.
$$

Moreover the proximity operator $\operatorname{prox}_{\gamma_n\chi_l^*}$ can be explicitly computed since (being $\gamma_n\chi_l^* = \delta_{\partial\chi_l(0)}$)

$$
\operatorname*{prox}_{\gamma_n\chi_l^*}(t) = \begin{cases} \mu w_l & \text{if } t > \mu w_l \\ t & \text{if } |t| \le \mu w_l \\ -\mu w_l & \text{if } t < -\mu w_l\,. \end{cases} \tag{3.32}
$$

For parameters $\gamma_n > 0$ in a suitable range, Algorithm 3.3, becomes Algorithm 3.5.

Algorithm 3.5 $\operatorname{prox}_{\zeta_k\varphi(\boldsymbol{\Theta}_{k+1},\cdot)}(\mathbf{B}_k)$ algorithm

Require: $\mathbf{V}^{(i)} \leftarrow \mathbf{U}^{(i)} \leftarrow 0, n \leftarrow 0$
  repeat
    $\mathbf{Z}_{tmp} \leftarrow \mathbf{B}_k - \zeta_k\Big(\mathbf{U}_n^{(1)}\boldsymbol{\Theta}_{k+1}^* + \mathbf{U}_n^{(2)} + \sum_{j=1}^J \operatorname{pr}_j^* D^* \mathbf{U}_n^{(3)}(:,j)\Big)$
    $\mathbf{V}_{n+1}^{(1)} \leftarrow (1+\gamma_n)^{-1}(\mathbf{U}_n^{(1)} + \gamma_n(\mathbf{Z}_{tmp}\boldsymbol{\Theta}_{k+1} - \mathbf{Y}))$
    for $j = 1 \to J$ **do**
      $\mathbf{V}_{n+1}^{(2)}(:,j) \leftarrow \operatorname{prox}_{\gamma_n\omega_L^*}(\mathbf{U}_n^{(2)} + \gamma_n\mathbf{Z}_{tmp}(:,j))$
    end for
    for $l = 1 \to L, j = 1 \to J$ **do**
      $v_{n+1}^{(3)}(l,j) \leftarrow P_{\mu[-w_l,w_l]}(u_n^{(3)}(l,j) + \gamma_n D_l \mathbf{Z}_{tmp}(:,j))$
    end for
    $\mathbf{Z}_{n+1} \leftarrow \mathbf{B}_k - \zeta_k\Big(\mathbf{V}_{n+1}^{(1)}\boldsymbol{\Theta}_{k+1}^* + \mathbf{V}_{n+1}^{(2)} + \sum_{j=1}^J \operatorname{pr}_j^* D^* \mathbf{V}_{n+1}^{(3)}(:,j)\Big)$
    $t_{n+1} \leftarrow \frac{1+\sqrt{1+4t_n^2}}{2}$
    for $i = 1 \to 3$ **do**
      $\mathbf{U}_{n+1}^{(i)} \leftarrow \mathbf{V}_{n+1}^{(i)} + \frac{t_n-1}{t_{n+1}}(\mathbf{V}_{n+1}^{(i)} - \mathbf{V}_n^{(i)})$
    end for
    $n \leftarrow n+1$
  until convergence

In this case the duality gap, taking into account (3.18) and (3.10), is as follows

$$\begin{aligned} & G(\mathbf{Z}, \mathbf{V}^{(1)}, \mathbf{V}^{(2)}, \mathbf{V}^{(3)}) \\ &= \frac{1}{2}\left\|\mathbf{Y} - \mathbf{Z}\boldsymbol{\Theta}_{k+1}\right\|_F^2 + \lambda \sum_{j=1}^J \left\|\mathbf{Z}(:,j)\right\|_1^2 \\ &\quad + \mu \sum_{j=1}^J \sum_{l=1}^{L-1} w_l |D_l \mathbf{Z}(:,j)| \\ &+ \frac{1}{2}\left\|\mathbf{V}^{(1)}\right\|_F^2 + \langle \mathbf{V}^{(1)}, \mathbf{Y}\rangle + \sum_{j=1}^J \frac{1}{4\lambda} \max_{1\le l\le L} (v^{(2)}(l,j))^2 \\ &\quad + \frac{1}{\zeta_k}\langle \mathbf{Z} - \mathbf{B}_k, \mathbf{Z}\rangle\,. \end{aligned}$$

Here condition (3.19) is satisfied, the stopping criterion is

$$G(\mathbf{Z}_n, \mathbf{V}_n^{(1)}, \mathbf{V}_n^{(2)}, \mathbf{V}_n^{(3)}) \le \frac{\varepsilon^2}{2\zeta_k} \tag{3.33}$$

and if $\mathbf{B}_{k+1} = \mathbf{Z}_n$, it is $\mathbf{B}_{k+1} \approxeq_\varepsilon \operatorname{prox}_{\zeta_k\psi_{\boldsymbol{\Theta}_{k+1}}}(\mathbf{B}_k)$ in the sense of (3.16).

3.3.3.4 The range of γ_n

According to Algorithm 3.3, we just need to determine an estimate of $\|A\|$ for the two cases: Algorithm 3.4 and Algorithm 3.5.

For Algorithm 3.4, it is easy to recognize that

$$A : \mathbb{R}^{J\times S} \to \mathbb{R}^{L\times S} \times \mathbb{R}^{J\times S} \times \mathbb{R}^{J\times S}, \quad A(\mathbf{\Theta}) = (\mathbf{B\Theta}, \mathbf{\Theta}, \mathbf{\Theta})\,.$$

and hence $\|A(\mathbf{\Theta})\|_F^2 = \|\mathbf{B\Theta}\|_F^2 + 2\,\|\mathbf{\Theta}\|_F^2$. Now recall that the Frobenius norm (also known as Hilbert–Schmidt norm and denoted by $\|\cdot\|_2$) and it holds

$$\|\mathbf{\Theta}\|_F^2 = \|\mathbf{\Theta}\|_2^2 = \mathrm{tr}(\mathbf{\Theta}^*\mathbf{\Theta}) = \|\mathbf{\Theta}^*\mathbf{\Theta}\|_1\ .$$

where $\|\cdot\|_1$ is the trace class norm. Moreover $\|\mathbf{\Theta}\| \le \|\mathbf{\Theta}\|_2 \le \|\mathbf{\Theta}\|_1$. We have

$$\|\mathbf{B\Theta}\|_2^2 = \mathrm{tr}(\mathbf{B}^*\mathbf{B\Theta\Theta}^*) = \langle \mathbf{B}^*\mathbf{B}, \mathbf{\Theta\Theta}^*\rangle_2\,. \tag{3.34}$$

Thus, for $\|\mathbf{\Theta}\|_2^2 = \|\mathbf{\Theta}^*\mathbf{\Theta}\|_1 \le 1$ it holds $\|\mathbf{B\Theta}\|_2^2 \le \|\mathbf{B}^*\mathbf{B}\|_2\,\|\mathbf{\Theta\Theta}^*\|_2 \le \|\mathbf{B}^*\mathbf{B}\|_2\,\|\mathbf{\Theta\Theta}^*\|_1 \le \|\mathbf{B}^*\mathbf{B}\|_2$. This proves that

$$\|A\|^2 = \sup_{\|\mathbf{\Theta}\|_2^2\le 1} \|A(\mathbf{\Theta})\|_2^2 \le \|\mathbf{B}^*\mathbf{B}\|_2 + 2\,. \tag{3.35}$$

Moreover, if we suppose $S \ge J$, there exists $\mathbf{\Gamma} \in \mathbb{R}^{J\times S}$ such that $\mathbf{\Gamma\Gamma}^* = \mathbf{B}^*\mathbf{B}$ (take $\mathbf{\Gamma} = [\sqrt{\mathbf{B}^*\mathbf{B}}\,|\,0]$) and consider the matrix $\mathbf{\Theta} = \mathbf{\Gamma}/\,\|\mathbf{\Gamma}\|_2$. Then $\|\mathbf{B\Theta}\|_2^2 = \langle \mathbf{B}^*\mathbf{B}, \mathbf{\Theta\Theta}^*\rangle_2 = \langle \mathbf{B}^*\mathbf{B}, (\mathbf{\Gamma\Gamma}^*)/\,\|\mathbf{\Gamma}\|_2^2\rangle_2 = \|\mathbf{B}^*\mathbf{B}\|_2^2\,/\,\|\mathbf{B}^*\mathbf{B}\|_1 \ge \|\mathbf{B}^*\mathbf{B}\|_2$.

$$\|\mathbf{B\Theta}\|_2^2 = \langle \mathbf{B}^*\mathbf{B}, \mathbf{\Theta\Theta}^*\rangle_2 = \langle \mathbf{B}^*\mathbf{B}, \frac{\mathbf{\Gamma\Gamma}^*}{\|\mathbf{\Gamma}\|_2^2}\rangle_2 = \frac{\|\mathbf{B}^*\mathbf{B}\|_2^2}{\|\mathbf{B}^*\mathbf{B}\|_1} \ge \|\mathbf{B}^*\mathbf{B}\|_2 \tag{3.36}$$

Therefore

$$\|A\|^2 = \sup_{\|\mathbf{\Theta}\|_2^2\le 1} \|A(\mathbf{\Theta})\|_2^2 \ge \|\mathbf{B}^*\mathbf{B}\|_2 + 2 \tag{3.37}$$

In the end we found that, under the hypothesis that $S \ge J$, it holds

$$\|A\| = (\|\mathbf{B}^*\mathbf{B}\|_F + 2)^{1/2}\,. \tag{3.38}$$

In any case, a valid range for γ_n in Algorithm 3.4 is as follows[4]

$$0 < \gamma_n \le \frac{\eta_k^{-1}}{\|\mathbf{B}^*\mathbf{B}\|_F + 2}\,. \tag{3.39}$$

[4]Note that $\|\mathbf{B}^*\mathbf{B}\|_2 \le \|\mathbf{B}^*\mathbf{B}\| = \|\mathbf{B}\|^2$, where $\|\mathbf{B}\|$ is the operator norm. Thus the bound $(\|\mathbf{B}\|^2 + 2)^{-1}$ for the range of γ_n is worse then the one provided in (3.39).

For Algorithm 3.5, we have

$$A(\mathbf{Z}) = (\mathbf{Z}\boldsymbol{\Theta}, \mathbf{Z}, D\mathbf{Z}(:,1), \ldots, D\mathbf{Z}(:,J))$$

and hence

$$\|A(\mathbf{Z})\|_F^2 = \|\mathbf{Z}\boldsymbol{\Theta}\|_F^2 + \|\mathbf{Z}\|_F^2 + \sum_{j=1}^{J} \|D\mathbf{Z}(:,j)\|_2^2 \,.$$

This calls for computing an upper bound for the norm of the derivative operator $D : \mathbb{R}^L \to \mathbb{R}^{L-1}$. To that purpose it is easy to show that $\|D\| \leq 2$. Indeed

$$\|D\boldsymbol{\beta}\|_2^2 \leq 2\left(|\beta_1|^2 + 2\sum_{l=2}^{L-1}|\beta_l|^2 + |\beta_L|^2\right) \leq 4\,\|\boldsymbol{\beta}\|_2^2 \,.$$

Then, as done before, for $\|\mathbf{Z}\|_F \leq 1$, it holds

$$\|A(\mathbf{Z})\|_F^2 \leq \|\boldsymbol{\Theta}\boldsymbol{\Theta}^*\|_F + 1 + 4\sum_{j=1}^{J} \|\mathbf{Z}(:,j)\|_2^2 \leq \|\boldsymbol{\Theta}\boldsymbol{\Theta}^*\|_F + 5\,.$$

In the end $\|A\|^2 \leq \|\boldsymbol{\Theta}\boldsymbol{\Theta}^*\|_F + 5$ and the corresponding range for γ_n is as follows

$$0 < \gamma_n \leq \frac{\zeta_k^{-1}}{\|\boldsymbol{\Theta}\boldsymbol{\Theta}^*\|_F + 5}\,.$$

3.4 aCGH signal model for synthetic data generation

Many synthetic CGH/aCGH models have been proposed in literature (Daruwala et al., 2004; Hupé et al., 2004; Hsu et al., 2005; Olshen et al., 2004; Fridlyand et al., 2004), but usually with the goal of demonstrating feasibility of a normalization, segmentation or calling method. Log-ratio signal is unrealistically simple and does not usually model all possible artifacts and bias due to technology and measurement limitations.

Willenbrock and Fridlyand (2005) tried to define a more realistic simulation schema which generates genomic profiles of comparable complexity with real life data. We aim, as Wang et al. (2007), to extend such a schema adding some other bias and noisy effects usually found in aCGH data.

We developed this synthetic data model in order to generate "realistic" (Di Camillo et al., 2009) data to test our and third-party algorithms and tools related to different aspect of the aCGH data analysis (*e.g* normalization, segmentation, calling *etc.*). Data generation is composed of a pipeline (Figure 3.3) of simple steps that will be detailed in the next sections after a short introduction on the notation.

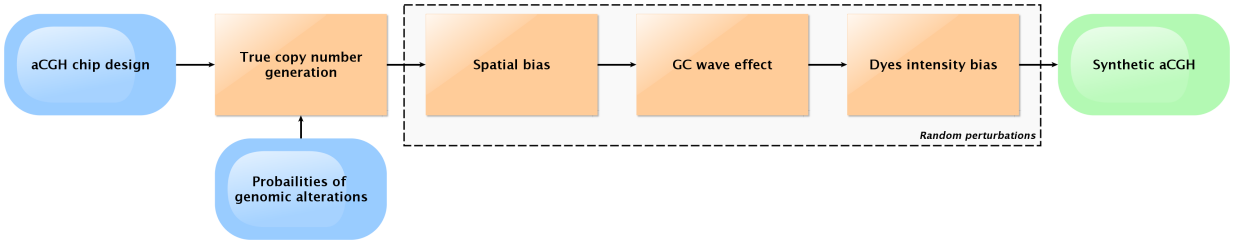


Figure 3.3: Synthetic aCGH data generation pipeline.

3.4.1 Notation

In the remainder of section, the following notation is used: $N \in \mathbb{N}$ and $P \in \mathbb{N}$ are, respectively, the number of samples and the number of aCGH clones to be generated. In this section, we will denote with x_j the j-th clone of a generic aCGH-related signal.

In our simulation procedure, we affect the signal with a spatial chip bias (as in the real data), therefore we require information about chip geometry, in terms of the number of columns $R \in \mathbb{N}$ and rows $C \in \mathbb{N}$, where the relation $P \leq R \cdot C$ must hold. All the other $(R + C - P)$ are considered as control/unused chip probes. All probes, clones and controls, are randomly distributed across the $R \times C$ grid, and we denote with x_j and y_j, respectively, the row and column coordinates of the j-th clone.

Note that, usually, in a real aCGH there are some clone replicas introduced in order to estimate noise and hybridization errors, by analyzing the intensities differences of replicated clones. In our generation procedure we do not simulate such clones, because we already have information about different kind of noises synthetically introduced.

Each clone is denoted by a quadruple $c_j = (id_j, chr_j, sb_j, eb_j)$ composed of an identifier (id_j) and a coordinate triple with the chromosome number (chr_j), the starting base (sb_j) and the ending base (eb_j). We will denote as "chip design" the sequence of $(c_1, \ldots, c_j, \ldots, c_P)$. Reference (**r**) and Test (**t**) signals are described as

$$\begin{aligned} r_j &= \{[cn_j^r + g^r(x_j, y_j)] \cdot 2^{w_j}\} \cdot b_j^r \\ t_j &= \{[(cn_j^t \cdot T) + (cn_j^r \cdot (1 - T)) + g^t(x_j, y_j)] \cdot 4^{w_j}\} \cdot b_j^t \end{aligned} \tag{3.40}$$

where cn_j is an estimated copy-number for the j-clone, T is a proportion of tumor cells in the test tissue, g is a spatial bias function, b_j is dye hybridization intensity reference, and w encodes a GC-wave bias effect. All the components will be detailed in next sections.

3.4.2 Copy numbers generation

We aim to simulate a full aCGH data generation producing both signal channels (test and reference), instead of directly generate the log-ratio signal as usually done (see, for example the Section 3.5 where we generate this kind of "toy" datasets).

The main assumption is that each sample is diploid and each clone c_j is associated with a **probability mass function** (*pmf* or *normalized histogram*) $H_j : \mathbb{N}^0 \to \mathbb{R}$ such that $H_j(\cdot) > 0$, $\sum_{l \in \mathbb{N}^0} H_j(l) = 1.0$ where $H_j(l)$ is the probability having a copy number l associated to clone c_j. Next, two different strategies are adopted for generating reference and test copy numbers.

Reference signal r: it is assumed that $cn_j^r \sim H_j^r(2) = 1.0$ for each c_j belonging to an autosome. That is, if we are generating a male sample, we assume that $H_j^r(1) = 1.0$ for each c_j belonging to both allosomes (sexual chromosomes X and Y). Otherwise, if we are generating a female sample, we assume that $H_j^r(2) = 1.0$ for each c_j belonging to X and that $H_j^r(0) = 1.0$ for each c_j belonging to Y.

Test signal t: for each clone c_j a copy number value $cn_j^t \sim H_j^t$ was drawn from an associated *pmf* given in input. Moreover, a parameter $T \sim U(T_{min}, T_{max})$, was introduced to resemble the proportion of *non-healty* cells and to incorporate this into the model.

3.4.3 Spatial bias

A spatial bias (Khojasteh et al., 2005; Neuvial et al., 2006) is successively added to the copy number signal using the chip geometry information. Such bias is modeled as a bivariate gaussian function (randomly) added to the test or reference signal separately

$$g(p_j) = s \cdot e^{-\frac{(p_j-\mu)^T \Sigma^{-1} (p_j-\mu)}{2}}$$

where $s \sim U(-1, 1)$ determines intensity and gaussian function orientation, $p_j = [x_j, y_j]$ is a clone chip-coordinate vector,

$$\mu = [\mu_x \sim U(0, R), \mu_y \sim U(0, C)]$$

is a random position on the chip, and

$$\Sigma = U \Lambda U^T, \tag{3.41}$$

is the covariance matrix of the multivariate gaussian distribution of the noise. In (3.41),

$$diag(\Lambda) = [\lambda_x \sim U(0, 2R), \lambda_y \sim U(0, 2C)],$$

contains the two-directional variances, while

$$U = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix},$$

is a rotation matrix, where $\theta \sim U(0, 2\pi)$.

3.4.4 Wave effect

Currently available aCGH technologies show a genome-wide artifact commonly known as "GC waves" (Hsu et al., 2005; Picard et al., 2011), which may be due to the guanine/-cytosine (GC) content of the probes used in aCGH (Marioni et al., 2007).

GC-waves add large scale variability to the probe signal ratio and usually interfere with data analysis algorithms. Those artifacts can increase the potential for false positive aberration calls in specific genomic regions and can also obscure true aberration calls. According to recent studies (Marioni et al., 2007), it seems that regions with a low GC content correspond roughly to the peaks of the wave, while regions with high GC content correspond to depressions.

Pique-Regi et al. (2009) tried to model this specific wave bias into CGH log-ratios. Because in our model we are trying to separately simulate test (**t**) and reference (**r**) signals, we extend Pique-Regi et al. (2009) wave model, moving the bias from log-ratio to raw signals. It is already known that for each sample, test and reference signals will have the same (aligned) wave perturbed with independent Gaussian noise (Marioni et al., 2007; Diskin et al., 2008). The amplitude of the wave may change across different samplings of aCGH but the alignment still remain the same.

Following Pique-Regi et al. (2009), the wave effect is modeled with a sinusoidal function applied on the resulting log-ratio. We will not introduce the wave bias based on real GC content, because for many purposes a roughly sinusoidal relation with chromosome coordinates is appropriate. The wave effect on log_2-ratios associated with the the j-th clone $c_j = (id_j, chr_j, sb_j, eb_j)$ may be defined as

$$w_j = a \cdot sin(f \pi sb_j) \tag{3.42}$$

where $a \sim U(A_{min}, A_{max})$ is a sample wave amplitude and $f = \frac{8}{max_j\{eb_j\}}$ is a wave frequency fixed for all sample and depending on the maximum chromosome length (in number of bases).

Such wave effect leads to a log-ratios model defined as

$$
\begin{aligned}
& log_2(t_j/r_j) + w_j = && (3.43)\\
&= log_2(t_j/r_j) + log_2(2^{w_j}) = && (3.44)\\
&= log_2(t_j/r_j \cdot 2^{w_j}) = && (3.45)\\
&= log_2(t_j \cdot 4^{w_j}/r_j \cdot 2^{w_j}), && (3.46)
\end{aligned}
$$

from which the definition in (3.40).

3.4.5 Dyes intensity and outliers

Raw noisy signals $\mathbf{r}$ and $\mathbf{t}$ need to be contextualized into modern aCGH technology. Log2-ratio signal profile from a real aCGH sample show noisy measurements, where the two channels (Cy3 and Cy5) have different median value with a lot of outliers clones, partially imputable to a dye-bias effect (Rosenzweig et al., 2004). First we define a noisy (hybridization) response as a combination of two factors

$$b_j = (\alpha_j \cdot \epsilon_j) \tag{3.47}$$

such that $\alpha_j \sim Log\mathcal{N}(a, \sigma_r^2)$ is a base response signal intensity and $\epsilon_j \sim Log\mathcal{N}(0, \sigma_\epsilon^2)$ models the noise in such responses. Therefore, r_j is shared between reference and test signals such that

$$
\begin{aligned}
b_j^t &= (\alpha_j \cdot \epsilon_j^t),\\
b_j^r &= (\alpha_j \cdot \epsilon_j^r).
\end{aligned}
\tag{3.48}
$$

Note that the $Log\mathcal{N}$ is defined as the distribution such that if $X \sim Log\mathcal{N}$ then $log_a(X) \sim \mathcal{N}$, which means that the logarithms of the test and reference signals are normally distributed, and the noise is the sum of two normally distributed noise factors (derived from the logarithms of α_j and ϵ_j).

Figures 3.4 and 3.5 compare a real Agilent 44k aCGH sample with a synthetic generated one where we duplicated the alterations that can be easily detected by a visual inspection on the original sample, that are: a loss of the chromosome 4, a gain and a loss of part of the chromosome 11 and a gain of part of the chromosome 17.

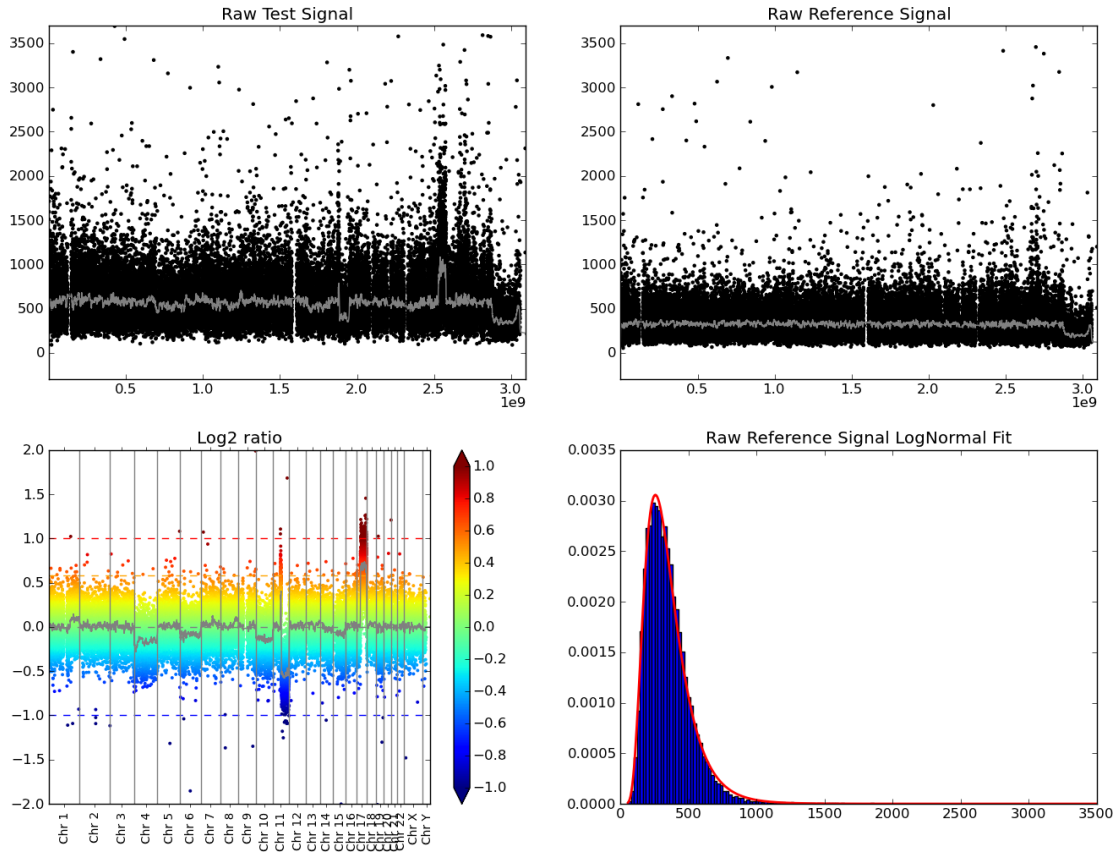


Figure 3.4: Raw test (top right) and reference (top left) signals from an Agilent 44k aCGH sample. The log-ratio profile (bottom right) shows four alterations on three chromosomes (4, 11 and 17). The reference raw signal follows a log-normal random distribution (bottom right).

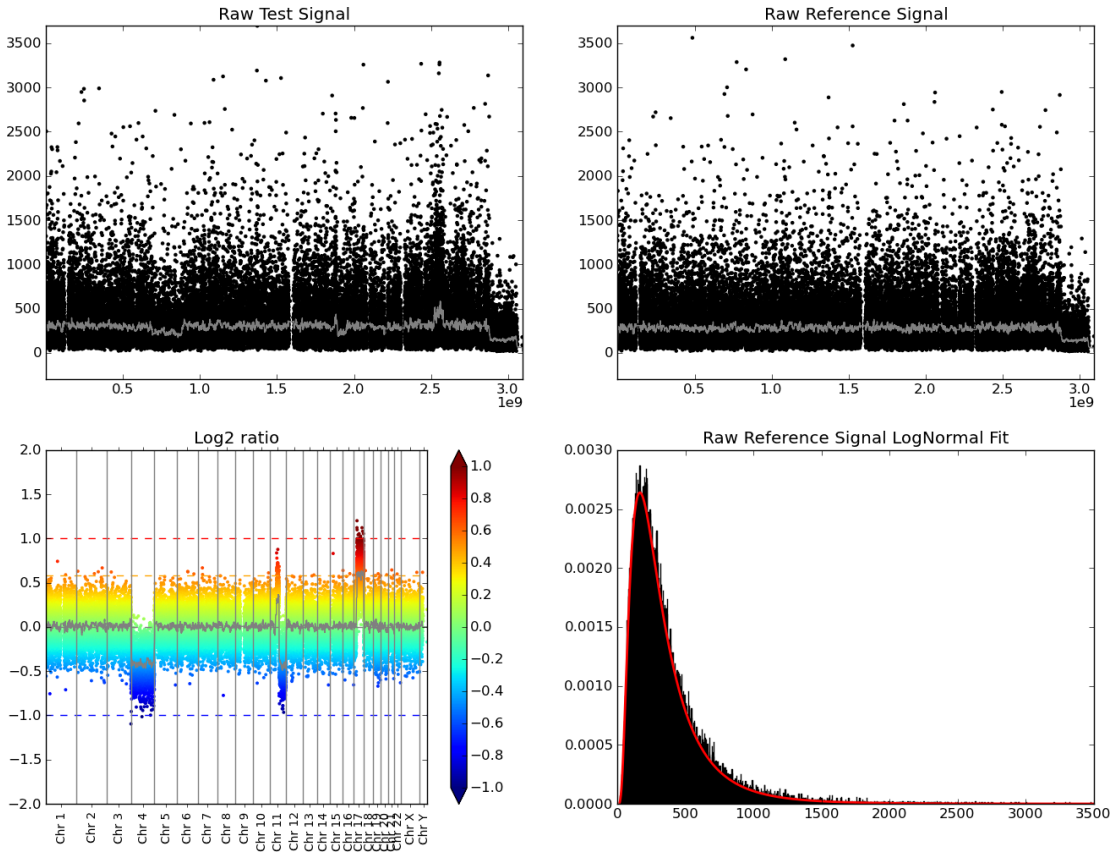


Figure 3.5: Raw test (top right) and reference (top left) signals synthetically generated from an an Agilent 44k platform design, resembling the real sample in Figure 3.4. The log-ratio profile (bottom right) shows four alterations on three chromosomes (4, 11 and 17). The reference raw signal follows a log-normal random distribution (bottom right).

3.5 Experiments and results

In this section we show the main properties of CGHDL model and compare it with FLLat. The reported experimental results are based on the analysis of different synthetic and real data, and are designed in order to show different aspect of our approach in term of main output (interpretability and reliability of the atoms and coefficients matrix, Masecchia et al. (2013b)), and in possible downstream analysis (classification or clustering tasks, Masecchia et al. (2013a)).

3.5.1 Datasets description

In this section we describe all datasets involved in our experiments. We used two different synthetic data models: one proposed by Olshen et al. (2004) and one using our simulator (Section 3.4). The model from Olshen et al. (2004) was used to compare CGHDL with FLLat because such model was used also by Nowak et al. (2011) proposing FLLat. In this model, the signal is defined as:

$$y_{ls} = \mu_{ls} + \epsilon_{ls}, \quad \mu_{ls} = \sum_{m=1}^{M_s} c_{ms} I_{\{l_{ms} \leq l \leq l_{ms}+k_{ms}\}}, \quad \epsilon_{ls} \sim N(0, \sigma^2), \tag{3.49}$$

where $l = 1, \ldots, L$, $s = 1, \ldots, S$, μ_{ls} is the mean, and σ is the standard deviation of the noise ϵ_{ls}. The mean signal $\mu_{\cdot s}$ is a step function where M_s is the number of segments (*i.e.* regions of CNVs) generated for sample s, and c_{ms}, l_{ms} and k_{ms} are the height, starting position and length, respectively, for each segment. Strictly following the model in (3.49), choosing $M_s \in \{1, 2, 3, 4, 5\}$, $c_{ms} \in \{\pm 1, \pm 2, \pm 3, \pm 4, \pm 5\}$, $l_{ms} \in \{1, \ldots, L - 100\}$ and $k_{ms} \in \{5, 10, 20, 50, 100\}$, $L = 1000$, $S = 20$, we generated tree types of datasets.

Dataset 1. The samples are generated in order to minimize the probability of sharing segments, following the same schema as in Nowak et al. (2011, Sec. 4.1, Dataset 1). Therefore separately for each sample, we chose the value of M_s, c_m l_m and k_m.

Dataset 2. Following Nowak et al. (2011, Sec. 4.1, Dataset 2), the samples are designed to have common segments of CNVs. Each shared segment appears in the samples according to a fixed proportion, randomly picked between $(0.25, 0.75)$. Starting points and lengths are shared among the selected samples, whereas the amplitudes c_{ms} may still vary within samples. The unshared segments are built as in Dataset 1, for a maximum of 5 segments per sample.

Dataset 3. The atoms $\boldsymbol{\beta}_j$ are generated according the same schema of (3.49). The coefficients θ_{js} are randomly sampled in $[0, 1]$, and the signal is built as $\mathbf{Y} = \mathbf{B}\boldsymbol{\Theta}$.

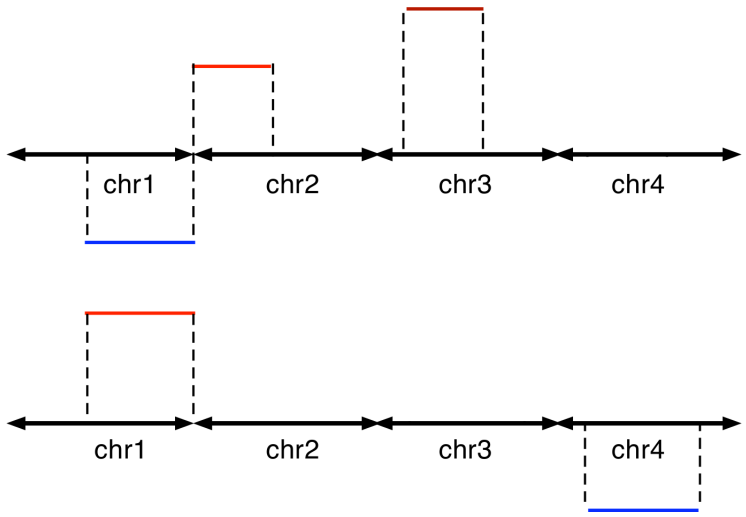


Figure 3.6: Mean signals of the two patterns used for Dataset 4 generation.

We also generate two datasets explicitly designed to mimic a real signal composed of different chromosomes. In order to have different levels of complexity, to gradually test our model before a real data application, the first dataset follows the model in (3.49) and the second follows our aCGH signal model (Section 3.4), which also needs a full preprocessing pipeline (*e.g.* normalization).

Dataset 4. We built three classes of samples. One third of the samples has mean signal as in the upper panel of Figure 3.6, one third has mean signal as in the lower panel of Figure 3.6, and the remaining third is built as Dataset 1. As for Datasets 1-3, we generate $S = 20$ samples on $L = 1000$ *loci*.

Dataset 5. As chip design, we used the Agilent 44k aCGH platform. To deal with a simpler model, we restricted the dataset to 4 out of 23 chromosomes, choosing those with a relatively small number of probes, that are chromosomes 13, 15, 18 and 21 (Figure 3.7). Overall we considered 80 samples composed of 3750 probes. We built three classes of samples. One third of the samples (G1) followed a pattern with a loss on chromosome 13, a gain on chromosome 15 and a gain on the longer arm of chromosome 18. The alterations occur randomly with a probability of 80%. Similarly, one third of the samples (G2) followed a pattern with a gain on chromosome 13 and a loss on the shorter arm of chromosome 21. Moreover, in groups G1 and G2, alterations (either gain or loss) on the chromosomes not involved in the patterns, can occur randomly with 20% probability. Finally, group G3 had random alterations (either gain or loss) on all chromosomes with low probability (10%).

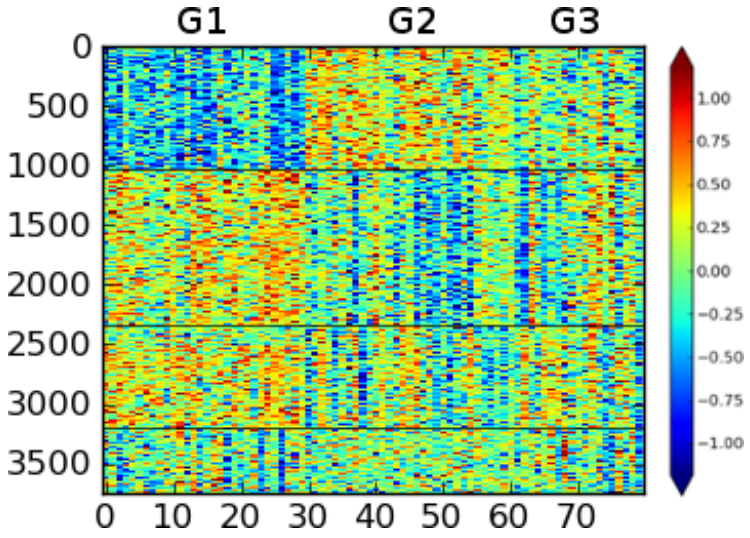


Figure 3.7: Synthetic generated Dataset 5.

3.5.1.1 Breast cancer dataset

For an experiment on real data, we considered the aCGH dataset by Pollack et al. (2002), already used by Nowak et al. (2011) to test FLLat. The dataset consists of 44 samples of advanced primary breast cancer. Each signal measures the CNV of 6691 human genes. The samples were assigned to different clinical information. In our experiments we are interested in tumor grading and tumor size. The sample belongs to 3 classes of tumor grading: 5 samples were assigned to grade 1, 21 to grade 2, 17 to grade 3 and 1 unassigned. In the dataset we have 4 classes of tumor sizes: overall, 12 samples were associated to a *small* tumor size (classes 1 and 2) and 32 samples to a *big* tumor size (classes 3 and 4).

As reported by Pollack et al. (2002), breast cancer has already been studied extensively and associated to recurrent alterations especially on chromosomes 8 and 17. Because, we also aim to compare our result with FLLat (which can only performs a chromosome-by-chromosome analysis), we concentrate our attention on these two relevant chromosomes in some experiments of the following sections.

3.5.2 Model selection

Similarly to Nowak et al. (2011), choice of the parameters (J, λ, μ, τ) is done according to the Bayesian information criterion (BIC) (Schwarz, 1978). The BIC mitigates the problem of overfitting by introducing a penalty term for the complexity of the model. In our case

the BIC is written as:

$$(SL) \cdot \log\left(\frac{\|\mathbf{Y} - \mathbf{B}\boldsymbol{\Theta}\|_F^2}{SL}\right) + k(\mathbf{B}) \log(SL), \tag{3.50}$$

where $k(\mathbf{B})$ is computed as the number of jumps in $\mathbf{B}$, and ultimately depends on the parameters (J, λ, μ, τ). Differently from Nowak et al. (2011), when required by the experiment, we also use the BIC criterion to select the number of atoms J. Note that the reconstruction accuracy increases with J, but our aim is not to achieve a perfect fit, but rather detecting the relevant alterations. In this context, the value of J may be chosen keeping in mind the compromise between model complexity (smaller J) and reconstruction accuracy (higher J).

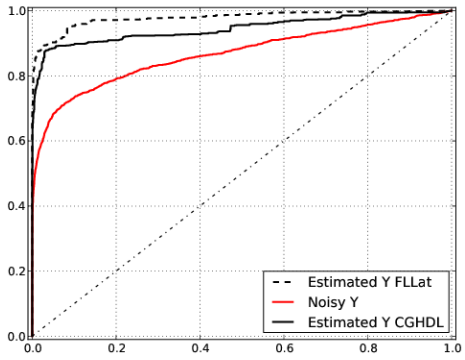
3.5.3 Representations interpretability and reliability

The first set of experiments aims at understanding how interpretable and reliable are the representations returned by CGHDL, both in terms of atoms (main alteration patterns) and coefficients (how samples use the given dictionary of atoms). For the experiments we used Python scripts, implementing *ex novo* our approach and wrapping the available R code for FLLat.

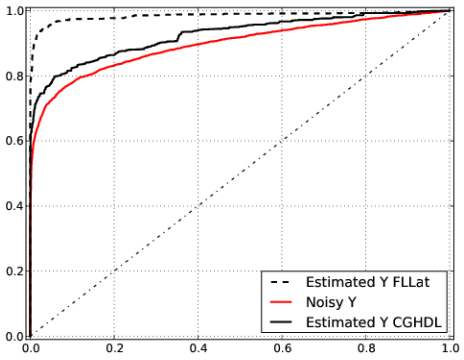
We start analyzing the reconstruction of the aCGH signals on the Datasets 1-4 where, following the model in (3.49), we applied two different levels of noise, $\sigma = 1.0$ and $\sigma = 2.0$. The number of atoms J varied in $\{5, 10, 15, 20\}$.

Figure 3.8 shows the performances of CGHDL and FLLat. Following Nowak et al. (2011), ROC curves are built by evaluating the correct detection of alterations based on the denoised signal $\hat{y}$, as varying a threshold value $t > 0$ the signal represents a deletion ($\hat{y} < -t$) or an amplification ($\hat{y} > t$).

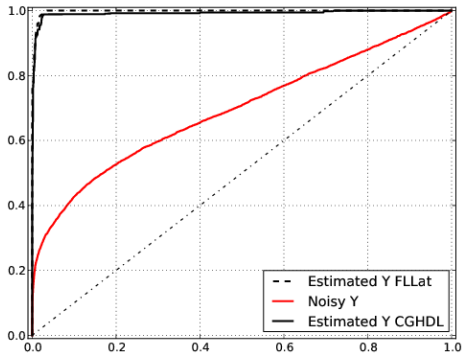
When analyzing the ROC curves, one can easily see how CGHDL and FLLat return comparable results in terms of data denoising, improving the detection of the genomic alterations with respect to the analysis performed directly on the noisy data matrix ($\mathbf{Y}$, reported in red as reference). FLLat, in terms of data fit, performs better than CGHDL on Dataset 1 and Dataset 2. Note that both datasets were built without the the assumption that subgroups of data share common subgroups of alterations. In this context, considering that CGHDL impose more constraints concerning this starting hypothesis, it is obvious to obtain better performances by FLLat, which are anyway worse than the other two cases (Dataset 3 and Dataset 4).



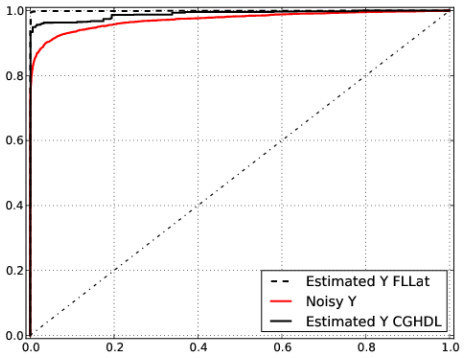
(a) Dataset 1



(b) Dataset 2

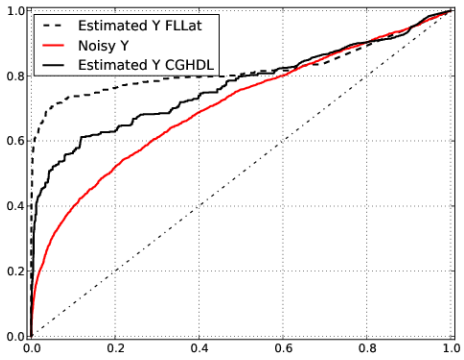


(c) Dataset 3

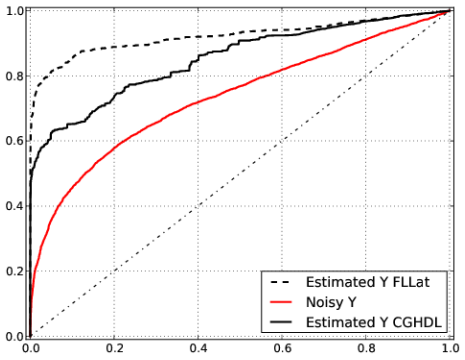


(d) Dataset 4

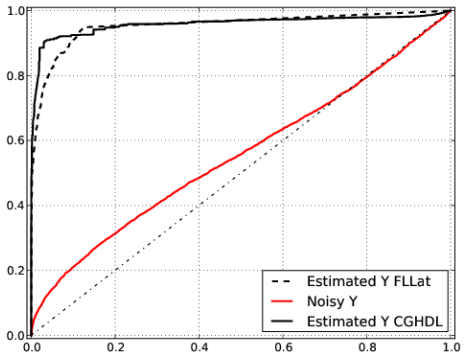
Figure 3.8: ROC curves for different dataset type and noise level $\sigma = 1.0$.



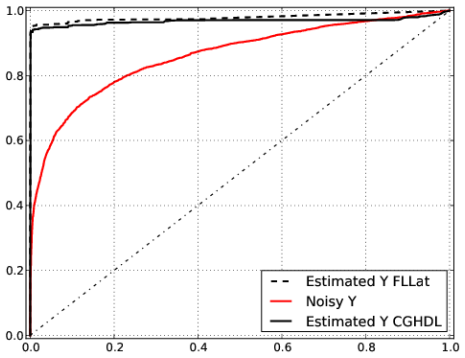
(a) Dataset 1



(b) Dataset 2



(c) Dataset 3



(d) Dataset 4

Figure 3.9: ROC curves for different dataset type and noise level $\sigma = 2.0$.

Focusing on the Dataset 4 ($\sigma = 2.0$), we can also show how the reconstruction was performed by the two algorithms. Figure 3.10 shows a plot of the solutions obtained by the two approaches. In both panels, left for FLLat and right for CGHDL, we report the input noisy matrix (up-right), the true solution without noise (bottom-right), the estimated denoised solution (bottom-left), and the set of found atoms (up-left). The algorithm implementing (3.1) (left panel) achieves good results in denoising, selecting $J = 10$ atoms (Figure 3.11), but fails in detecting the underlying patterns seen in Figure 3.6. The selected atoms represent single alterations which are scattered across the matrix $\mathbf{B}$. Conversely, CGHDL (right panel in Figure 3.10) selects $J = 5$ atoms which clearly comprise the two patterns (Figure 3.12).

We obtain a similar and promising result analyzing the Dataset 5 with CGHDL, which is the one created using the aCGH signal model proposed in Section 3.4. Keeping in mind that we had three main groups of data, we chose $J = 5$. Figures 3.13 and 3.14 report the atoms and coefficients of CGHDL on the simulated dataset. One can recognize that in Figure 3.13 the first two atoms capture the pattern of group G1, while the third corresponds to group G2. The fourth atom represents one of the deletions that have been introduced randomly as noise. The platform we used for simulations (Agilent 44k) maps only the longer arms of the chromosomes 13 and 15 (namely 13q and 15q). For chromosome 18, in Figure 3.13 the dashed vertical line indicates the boundary between the shorter and longer arm of the chromosome: 18p and 18q. Chromosome 21 has a very small p arm which is mapped on the platform by only 3 probes, so catching an alteration in this point is a very difficult task. Atoms $\#1$ and $\#5$ actually capture this small alteration.

3.5.4 Clustering for breast cancer sub-typing

The aim of this experiment is to prove that CGHDL allows for a more informative representation of the data in terms of main shared patterns of alterations. In order to demonstrate this hypothesis, we performed two different experiments. First we would like to demonstrate that CGHDL, even if more complex than FLLat, is able to extract useful information in a chromosome-by-chromosome analysis. Then we performed an experiment considering all the chromosomes at the same time, and showed how CGHDL can extract all the meaningful genomic alteration, providing an overall informative result.

In the first experiment we compared CGHDL and FLLat focusing on chromosomes 8 (241 mapped genes) and 17 (382 mapped genes), identified by Pollack et al. (2002) as chromosomes with biologically relevant CNVs. Clustering was performed on Y^c, the original raw noisy data matrix restricted to the chromosome $c \in \{8, 17\}$, on coefficients matrices $\mathbf{\Theta}^c_{cghdl}$ and $\mathbf{\Theta}^c_{fllat}$, and on the denoised samples matrices $\hat{\mathbf{Y}}^c_{cghdl}$ and $\hat{\mathbf{Y}}^c_{fllat}$.

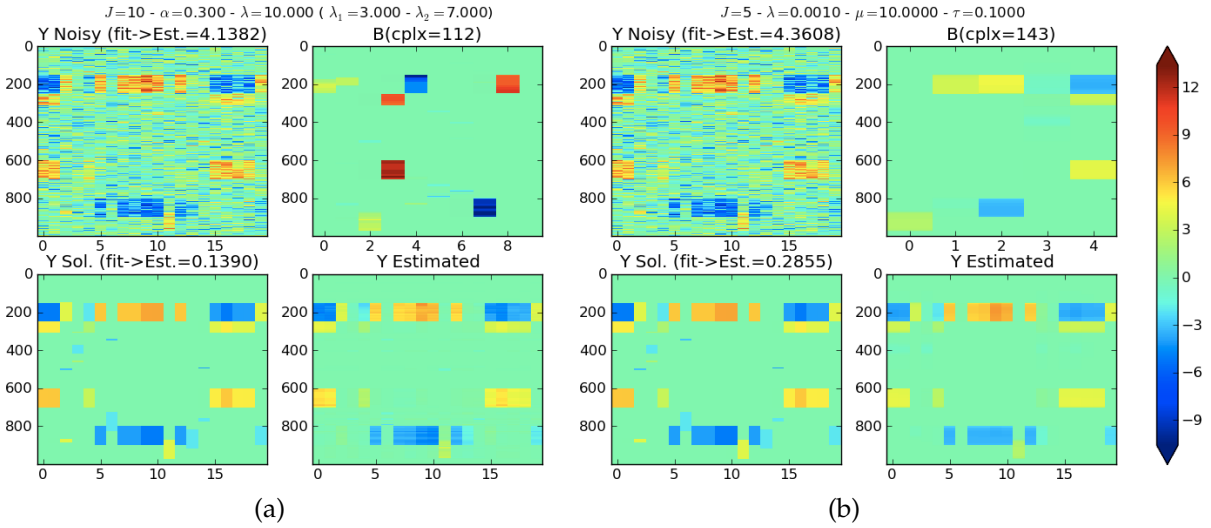


Figure 3.10: Dataset 4 analyzed by FLLat (left panel) and our method (right panel). Each panel shows 4 subplots: top left plot represents the noisy data matrix, top right plot shows the atom matrix with atoms as columns, bottom left subplot is the *true* data matrix and bottom right is the estimated signal.

As explained by Nowak et al. (2011), FLLat cannot analyze an aCGH signal along the entire genome due to the unweighted total variation included into its model, therefore, in the second experiment, we compared the results of CGHDL with a clustering procedure on the raw dataset (6691 probes). Clustering was performed on the original raw noisy data matrix $\mathbf{Y}$, on the coefficients matrix $\mathbf{\Theta}_{cghdl}$ and the denoised samples matrix $\hat{\mathbf{Y}}_{cghdl}$ calculated by CGHDL.

For clustering, we adopted a hierarchical agglomerative algorithm, using the the *city block* or *manhattan* distance between points

$$d(a,b) = \sum_i |a_i - b_i|$$

and the *single linkage* criterion (Sibson, 1973)

$$d(A,B) = min\{d(a,b) : a \in A, b \in B\}.$$

The cluster A is linked with the cluster B if the distance $d(A,B)$ is the minimum with respect to all the other clusters B'. The *manhattan* distance allows us to calculate a point-wise difference both for the coefficients vectors and the raw/denoised aCGH signals.

Moreover, to evaluate the coherence of the obtained dendrogram with respect to the groups G1, G2 and G3, we measured the *cophenetic distance* among the samples within

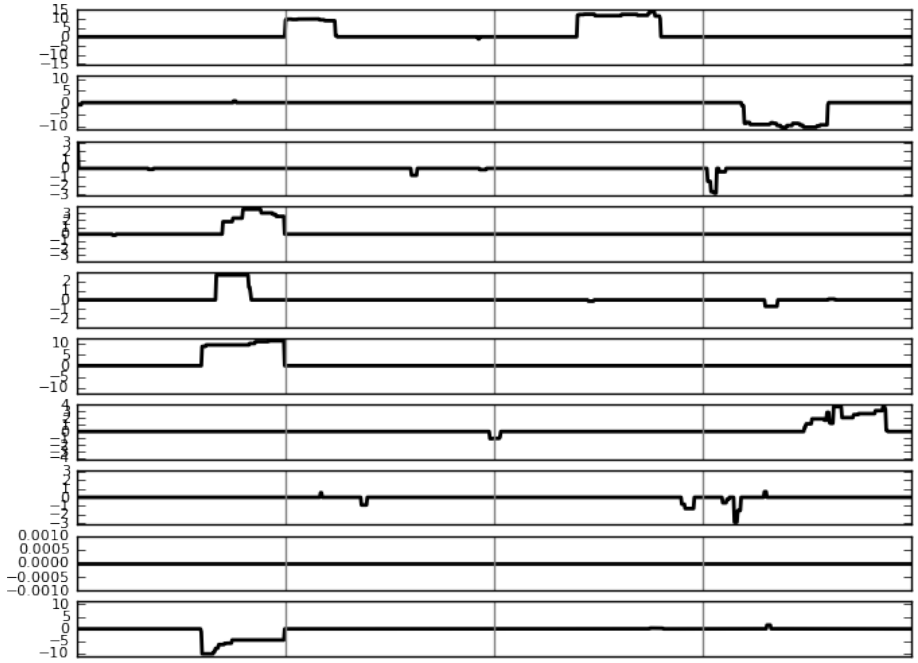


Figure 3.11: Dictionary learned by FLLat on Dataset 4 ($\sigma = 2.0$). Comparing this figure with the dictionary depicted in Figure 3.10, here the atoms are ordered with respect to the their usage for the reconstruction of the original signals. Atoms at the top are more used than atoms at the bottom.

each group (Sokal and Rohlf, 1962). For each pair of observations (a, b), the cophenetic distance is the distance between the two clusters that were merged to assign the two points in a single new cluster. The average of the cophenetic distances within each clinical group provides an objective measure of how the resulting dendrogram "describes" the differences between observations, using the clinical grades as ground truth.

Note that, by design, the values contained into the coefficients matrix produced by FLLat and CGHDL could have different range of values (in CGHDL the values are positive and bounded). In order to calculate comparable distance metrics, before clustering and cophenetic distances evaluation, each estimated coefficients matrix $\mathbf{\Theta}$ was normalized by is maximum absolute value. The same preprocessing was also applied on the original aCGH signals and on the estimated ones $\hat{\mathbf{Y}} = \mathbf{B\Theta}$.

Both FLLat and CGHDL choose the optimal parameters over a grid using a BIC-based searching algorithm. In particular, for FLLat the grid was defined by some heuristics implemented in the given R package. The parameter θ_{max} in CGHDL was set to 1.0. This choice forces the algorithm to find atoms with signal amplitude comparable with the original data.

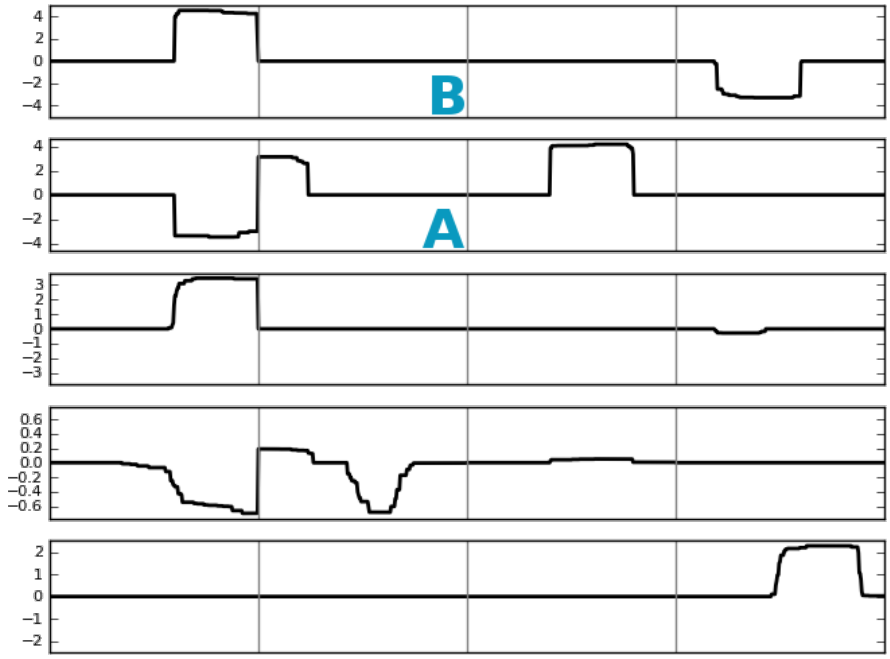


Figure 3.12: Dictionary learned by CGHDL on Dataset 4 ($\sigma = 2.0$). Comparing this figure with the dictionary depicted in Figure 3.10, here the atoms are ordered with respect to the their usage for the reconstruction of the original signals. Atoms at the top are more used than atoms at the bottom.

Analysis restricted to chromosomes 17 and 8. In Figure 3.15 (left) we show the means of the cophenetic distances calculated for each group of samples (the unannotated sample was not considered) restricted to the chromosome 17. In this experiment, following Nowak et al. (2011), we fixed $J = 5$ and initialized B with the first 5 principal components of the matrix Y. We searched, for CGHDL, the best triple of parameters in $\mu \in \{0.01, 0.1, 1.0, 10, 100\}$, $\lambda \in \{0.01, 0.1, 1.0, 10, 100\}$ and $\tau \in \{0.1, 1.0, 10\}$. It is clear that the clustering on the coefficients matrix produced by CGHDL places the samples belonging to homogeneous clinical groups (G1, G2 and G3) closer in the dendrogram. Moreover, also the denoised data matrix $\hat{\mathbf{Y}}^{17}_{cghdl}$ shows better discriminative performances with respect to $\hat{\mathbf{Y}}^{17}_{fllat}$. This may be due to the capability of our model to better detect the main altered patterns in the signals, despite a possibly higher reconstruction error Masecchia et al. (2013b). Such property ultimately induces a more effective clustering.

In Table 3.1 we report a summary of the averaged cophenetic distances, also including the clustering on raw signals.

The analysis on chromosome 8 gives similar results. Following Nowak et al. (2011), we fixed $J = 6$, initialized B with the first 6 principal components of the matrix Y.

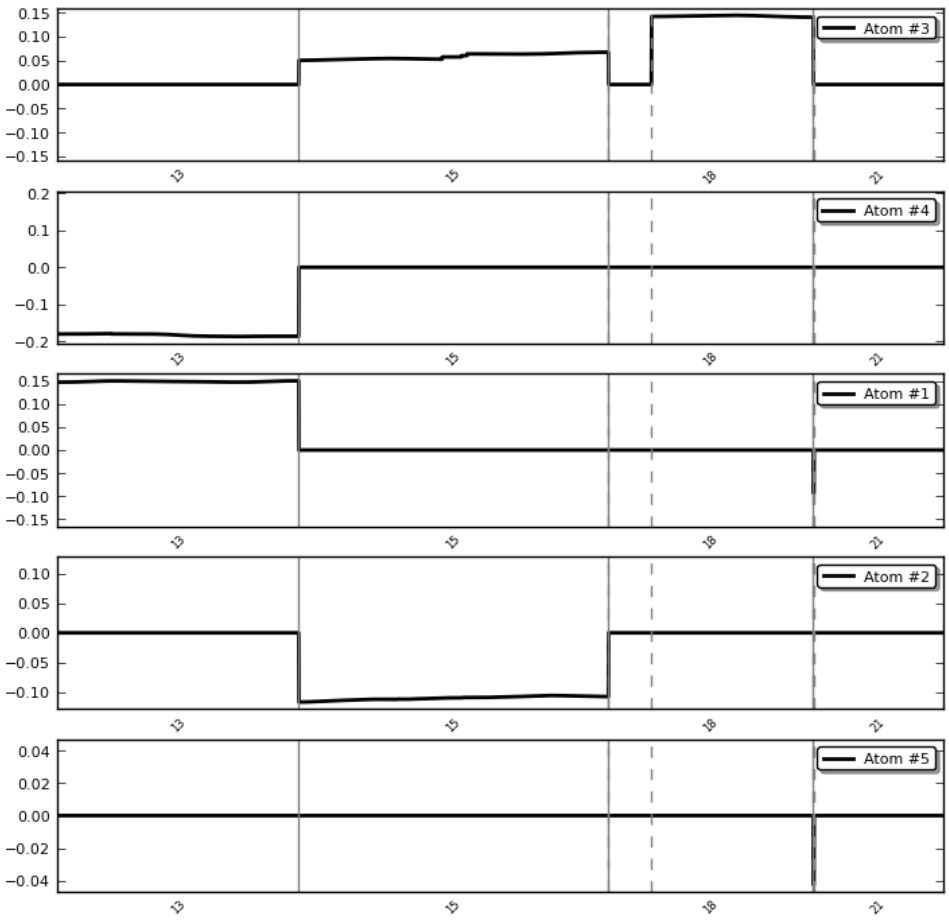


Figure 3.13: Profiles of the five atoms identified by CGHDL for the simulated Dataset 5.

Then, we searched, for CGHDL, the best parameters in $\mu \in \{0.01, 0.1, 1.0, 10, 100\}$, $\lambda \in \{0.01, 0.1, 1.0, 10, 100\}$ and $\tau \in \{0.1, 1.0, 10\}$. Figure 3.15 (right) shows the means of the cophenetic distances calculated for each group of samples, and Table 3.2 shows the corresponding averaged cophenetic distances.

Whole genome analysis. We ran the experiments with three different $J \in \{10, 18, 24\}$ which correspond to the number of principal components of Y able to explain respectively the 50%, 70% and 80% of the variance. Then we searched the best parameters $\mu \in \{0.01, 0.1\}$, $\lambda \in \{0.01, 0.1\}$ and $\tau \in \{0.01, 0.1\}$. Here, we present the results obtained with $J = 10$: the resulting atoms (see Figure 3.16(a)) describe co-occurrent alterations along different chromosomes but are still fairly simple for a visual interpretation by the domain experts. For different Js we did not note relevant differences in terms of fit and clustering.

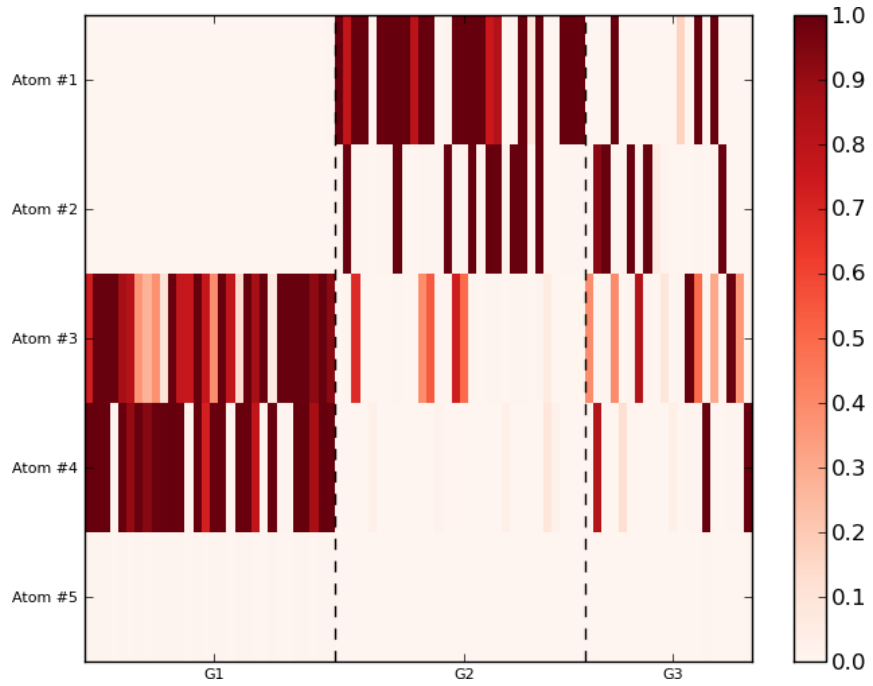


Figure 3.14: The matrix of the CGHDL coefficients for the simulated Dataset 5.

	G1	G2	G3
$\mathbf{\Theta}^{17}_{cghdl}$	$\mathbf{0.008 \pm 0.004}$	$\mathbf{0.079 \pm 0.112}$	$\mathbf{0.111 \pm 0.124}$
$\hat{\mathbf{Y}}^{17}_{cghdl}$	0.022 ± 0.019	0.476 ± 0.720	0.687 ± 0.795
$\mathbf{\Theta}^{17}_{fllat}$	0.178 ± 0.044	0.265 ± 0.173	0.517 ± 0.446
$\hat{\mathbf{Y}}^{17}_{fllat}$	1.737 ± 0.484	2.945 ± 2.074	5.212 ± 3.851
Y^{17}	19.284 ± 2.374	19.589 ± 3.961	23.941 ± 5.870

Table 3.1: Average cophenetic distances after clustering for the analysis restricted to chromosome 17

It is important to note that the four more used atoms of the dictionary extracted by CGHDL detect the main genomic alterations on chromosomes 8 and 17 as well as a co-occurrence of deletions on chromosome 3 and 5. In Pollack et al. (2002) all these alterations were already indicated as very common but the relation between chromosomes 3 and 5 was not indicated as co-occurrence and needs further biological validation.

3.5.5 Classification for tumor size prediction

In the following set of experiments, we considered the aCGH dataset from Pollack et al. (2002) designing a standard classification problem. The aim of this experiment was to

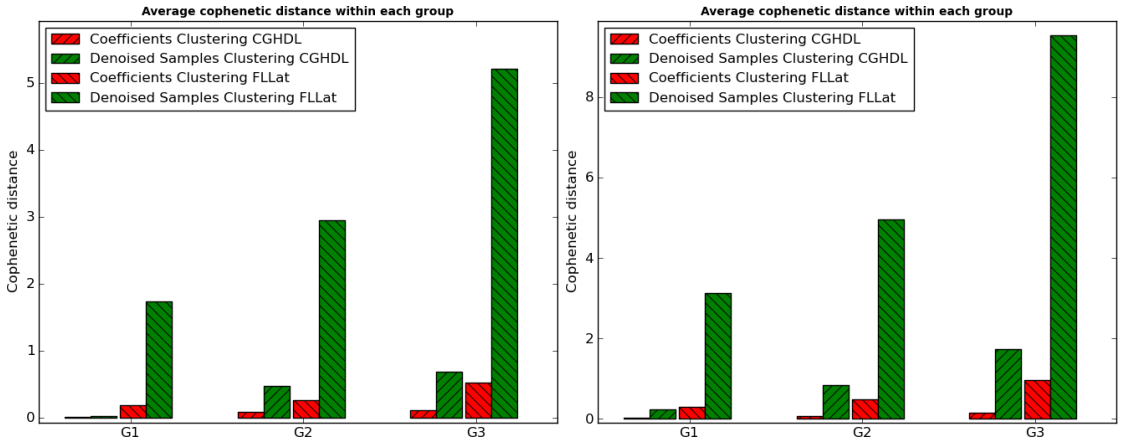


Figure 3.15: Average cophenetic distances for the groups G1, G2 and G3 on chromosome 17 (left) and chromosome 8 (right). CGHDL always has better clustering results (see also Tables 3.1 and 3.2). Moreover, is also interesting to note that clustering the denoised samples by CGHDL and FLLat, the former has better results, suggesting also an higher quality of the dictionary atoms used to reconstruct the samples.

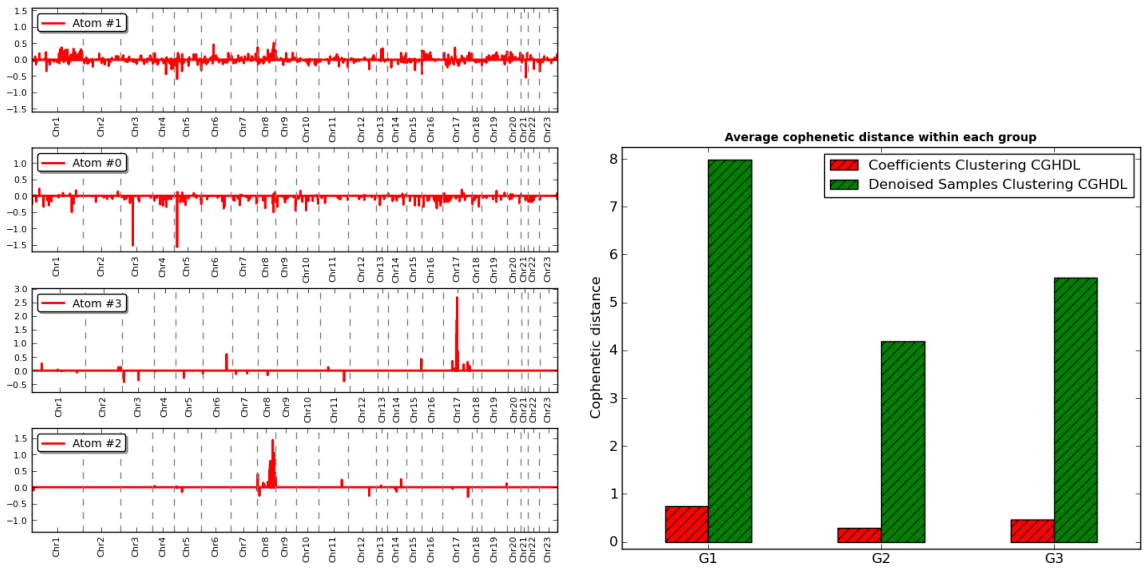
	G1	G2	G3
$\boldsymbol{\Theta}^8_{cghdl}$	$\mathbf{0.016 \pm 0.007}$	$\mathbf{0.054 \pm 0.024}$	$\mathbf{0.147 \pm 0.142}$
$\hat{\mathbf{Y}}^8_{cghdl}$	0.222 ± 0.095	0.842 ± 0.410	1.720 ± 1.135
$\boldsymbol{\Theta}^8_{fllat}$	0.301 ± 0.095	0.469 ± 0.236	0.951 ± 0.657
$\hat{\mathbf{Y}}^8_{fllat}$	3.135 ± 1.090	4.962 ± 2.605	9.547 ± 6.638
Y^8	12.363 ± 1.165	15.484 ± 4.124	20.150 ± 6.200

Table 3.2: Average cophenetic distances after clustering for the analysis restricted to chromosome 8

use CGHDL to possibly identify discriminant alterations for the tumor size. To this aim, we considered a binary classification setting (*small* vs. *big* tumor sizes) using the coefficient matrix as a representation of the dataset.

We devised two cases: first, we analyzed the signal restricted to chromosomes 8 and 17 only, searching in these two characterizing chromosomes further signal alterations correlating with the tumor size. Next, we considered the entire genome, in case the discriminant alterations were contained in other chromosomes. The experiments were also performed using the FLLat algorithm for comparison purposes.

For each run, we chose J corresponding to the number of principal components of the data matrix $\mathbf{Y}$ able to explain at least the 50% of the variance. Then, we searched for the best triple of parameters (μ, λ, τ) in $\mu \in \{0.01, 0.1, 1.0, 10, 100\}$, $\lambda \in \{0.01, 0.1, 1.0, 10, 100\}$



(a) (b)

Figure 3.16: (a) Profiles of the first 4 more used atoms for sample reconstruction (sum of the row of $\mathbf{\Theta}$) extracted by CGHDL on all chromosomes. The atom #1 maps a general pattern of alterations, and it is responsible of a high proportion of signal reconstruction. Note that CGHDL found the alterations on chromosomes 8 and 17, and also detected co-occurring alterations on chromosomes 3 and 5. (b) Average cophenetic distances for the groups G1, G2 and G3 on all chromosomes and $J = 10$. See also Table 3.3

and $\tau \in \{0.1, 1.0, 10\}$. For both scenarios, the training and validation sets were randomly sampled 100 times from the dataset according to a two-third one-third proportion. The sampling was performed taking into account the unbalance of the classes.

We trained a linear SVM classifier (Fan et al., 2008) choosing the best regularization parameter $C \in \{1, 10, 100, 1000\}$ in a 5-fold cross validation schema. We then evaluated the performance of the best classifier on the validation set. The pipeline was repeated 100 times and performances scores averaged. As score, we used the Matthews Correlation

	G1	G2	G3
$\mathbf{\Theta}_{cghdl}$	$\mathbf{0.738 \pm 0.541}$	$\mathbf{0.290 \pm 0.213}$	$\mathbf{0.463 \pm 0.406}$
$\hat{\mathbf{Y}}_{cghdl}$	7.988 ± 4.663	4.191 ± 2.795	5.512 ± 3.632
Y	305.76 ± 39.85	290.26 ± 38.04	302.86 ± 34.76

Table 3.3: Average cophenetic distances after clustering for the analysis extended to all chromosomes with $J = 10$

aCGH Model	Chr	J	MCC
FLLat	8+17	7	-0.016
CGHDL	8+17	7	0.126
CGHDL	8&17	5	0.006
FLLat	Conc.	131	0.330
FLLat	ALL	10	0.275
CGHDL	ALL	10	**0.464**
Raw data	ALL	6691	0.436

Table 3.4: Classification results for FLLat and CGHDL. For each aCGH model, we indicate the coefficient matrix Chr, the number of atoms J and the corresponding MCC score.

Coefficient (MCC) (Matthews, 1975)

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

explicitly designed for binary classification, particularly suitable for unbalanced classes and is able to describe the confusion matrix of true and false positives and negatives in a single measure. This score returns a value between -1 and $+1$: a coefficient of $+1$ represents a perfect prediction, 0 indicating no better than random prediction and -1 indicates total disagreement between prediction and observation.

In Table 3.4 we show the performances of the SVM classification based on FLLat and CGHDL analysis on different cases. In "$8+17$", the matrix of coefficients are obtained independently on each chromosomes and then concatenated. In "$8\&17$" the analysis is performed jointly on both chromosomes. In "$Conc.$" the FLLat procedure is applied to each chromosome separately and then concatenated. Finally, "ALL" indicates the procedures applied on the entire genome.

We remark that the best result is obtained by CGHDL in the "ALL" case. These results favor a global analysis that takes into account the genome as a whole. Comparable results with the FLLat analysis may be achieved at the expense of handling higher dimensional feature vectors. Averaged scores were comparable with the ones calculated on the validation sets, guaranteeing unbiased results.

Chapter 4

A Computational pipeline for oncogenesis

This chapter describes a computational pipeline for oncogenesis. Such problem is approached exploiting a well known algorithm for inferring oncogenetic tree models. This methods assumes given as input a list of genomic events (deletions or amplification) detected on a dataset of aCGH data.

Limitations related with the straightforward application of the pipeline, led us to take advantages of use CGHDL, the proposed model for aCGH segmentation described in the Chapter 3, exploiting its peculiarities.

In Section 4.1 we describe the biological context and the intrinsic difficulties of the task.

Section 4.2 describes the general pipeline for oncogenesis and the adopted tree model formalizing the concept of "oncogenetic tree". In this section we explain how CGHDL can be nested into the pipeline and improve interpretability and reliability of the inferred oncogenetic trees.

Finally, in Section 4.3, we illustrate the results of two different experiments on real data related to the Neuroblastoma disease. We test the standard approach with respect to the one based on CGHDL.

4.1 Biological context: oncogenesis

It has been long thought that cancer is due to an accumulation of (specific) genes mutations. Tumor development may starts from a single genetically altered cell and proceeds by successive clonal expansions of cells that have acquired additional advantageous

mutations. The progression of cancer is characterized by the accumulation of these genetic changes. Usually, the need of effective algorithms is due to the fact that the real tumor progression cannot be inferred by chronological ordered data but by different point-wise progression processes.

Vogelstein et al. (1988) pioneered the oncogenesis related research, working on a genetic model for colorectal tumorgenesis (Fearon and Vogelstein, 1990). They were able to associate specific genetic changes with four of the stages of cancer progression, as depicted in Figure 4.1. The genetic changes are assumed as irreversible and the presence of all four changes indicates that the cell is cancerous. They also underlined an important aspect: tumor development its not related to the progression but with the mutations accumulation. The tumor development model they found was a simple path where the cell is assumed starting from an healthy (normal) state and proceeding through a path with four step representing different genetic change. Such genetic changes will not always occur following exactly the path order, but the path defines a preferred order. Formally, the standard *multistage theory* of tumor progression states that tumor occurs at the end of a multistep pipeline between k states, where each step from one to the next is a rare event. Let us denote the cancer stages by $0, 1, 2, \ldots, k$, where stage 0 refers to the normal precancerous state, 1 to the first adenomatous stage, and k to a defined cancerous endpoint, such as the formation of metastases. The process is started at time $t = 0$ in state 0.

Unfortunately, attempts to find similar path models for other types of cancer have not been successful. An important problem in solid tumors is that when a set of crucial genetic alterations develops, the cancer starts to accumulate seemingly random alterations. CGH studies suggest that it happens because many cancers are genetically heterogeneous, in that clinically similar cancers have different genetic causes. There exist also the possibility that the model is completely independent. This means that mutations can occur in parallel and the tumor *appear* when all are present. A more realistic model allows a partial dependency between mutations: some can occur randomly but each one can promotes a further mutation, for example if the mutation compromises a cancer suppressor gene or an oncogene.

The works of Desper et al. (1999, 2000); Radmacher et al. (2001) were aimed at inferring trees or graph models for oncogenesis from CGH data (a novel technique at the time). Since then, an increasing number of tumor progression models (Bilke et al., 2005; Liu et al., 2009; Navin et al., 2010) has been proposed due to the constantly increasing availability of public datasets.

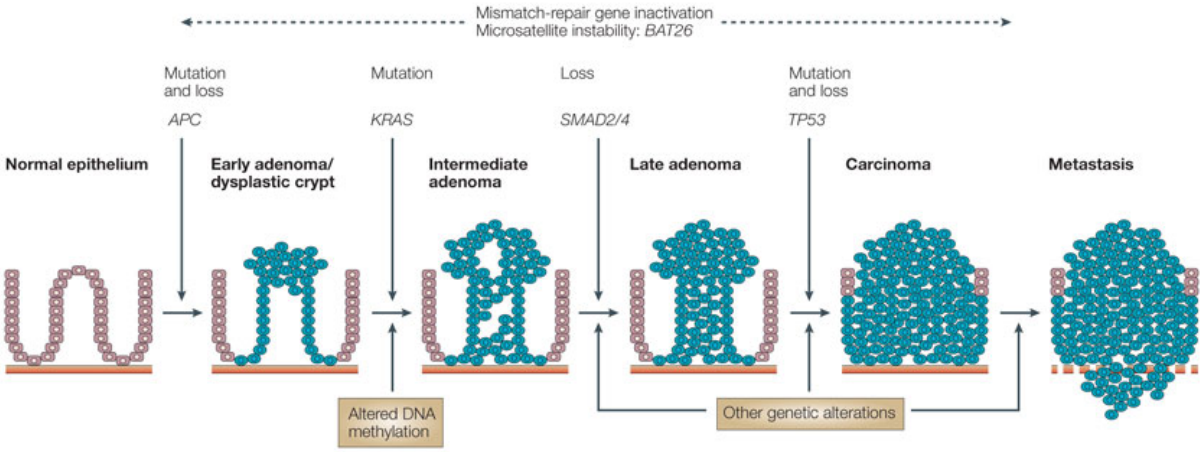


Figure 4.1: Colorectal cancer multi-step model Davies et al. (2005b).

4.2 A pipeline for oncogenensis from aCGH data

Given a set of aGCH, we aim at building an oncogenetic tree describing the progression of genomic events. A generic pipeline is described by three computational steps as depicted in Figure 4.2:

1. **Normalization**: a standard normalization of the data is always required in order to remove technical and biological noise and perform a multi sample analysis.

2. **Alterations Extraction**: after the normalization of the sample this step analyzes the aCGH data (separately or as a whole) in order to extract a list of genomic alterations or CNVs associated to it. *In this section we present two different approaches for extracting such alterations*, namely **standard** and **CGHDL-based**. These variations are presented respectively in Sections 4.2.2 and 4.2.3.

3. **Oncogenesis model inference**: tumorigenesis trees are finally produced using a well-known oncogenesis tree models described in Section 4.2.1. Depending by the input from the previous step, this phase may infer oncogenetic trees where nodes are single genomic events (chromosomes gains or losses) or pattern of genomic events.

In the next section we first describe the oncogenetic inference method in order to give an idea of the input these methods require, then we describe the two variations of step number 2 implemented.

Figure 4.2: Schema of the implemented pipeline for oncogenesis.

4.2.1 Inferring tree models for oncogenesis

In this section, we briefly describe the theory and algorithm developed by Desper et al. (1999) for inferring oncogenetic tree models. This method was the first attempt to propose a statistical well founded algorithm to solve this difficult combinatorial problem.

The inference method assumes that some mutations may happen at random, but others may be caused by previous ones. In some cases, the connection between the events is specific and directly causal, while in other cases later events occur seemingly at random due to the tumor cells instability.

The method requires, as given input, a family of sets of Copy Number Variations (CNVs). Each CNV may be associated to a genomic event (amplification or deletion) on a particular chromosome portion. The family of sets is sampled from a *probability distribution* over all sets of genetic events. In this context a model for oncogenesis defines a distribution over sets of genetic events.

Let V be a finite set of genetic events plus a root node r. A probability distribution on $\mathcal{P}(V)$ (the power set of V) is a function p such that

$$p(S) \geq 0, \forall S \in \mathcal{P}(V) \qquad \sum_{S \in \mathcal{P}(V)} p(S) = 1.$$

A *rooted tree* on V is a triple $\mathcal{T} = (V, E, r)$. $\mathcal{T}$ defines a distribution on $\mathcal{P}(V)$, where E is a set of pairs of vertices such that

- for each $v \in V$ there is at most one edge $(u, v) \in E$;
- there is no edge (u, r);
- no sequence of edges in E form a cycle $((v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k), (v_k, v_0))$.

Note that this model allows trees with disconnected components and two special kind of trees: *stars* and *paths*. A star is a tree in which all edges leave the root, a path is a tree with at most one edge leave each vertex.

An *oncogenetic tree*

$$\mathcal{T} = (V, E, r, \alpha) \tag{4.1}$$

is a rooted labeled tree where for all $e \in E$ $0 < \alpha(e) \leq 1$. The label $\alpha(e)$ may be interpreted as the probability of e to be present in the graph and edges are considered as independent events. The tree $\mathcal{T}$ then generates a distribution $P_{\mathcal{T}}$ on $\mathcal{P}(V)$ where for each $S \subseteq V$ we have:

- if $r \in S$ and there is a subset $E' \subseteq E$ such that S is the set of all vertices reachable from r in $\mathcal{T} = (V, E', r)$, then

$$P_{\mathcal{T}}(S) = \prod_{e \in E'} \alpha(e) \cdot \prod_{e=(u,v) \in E, u \in S, v \notin S} (1 - \alpha(e)),$$

- otherwise $P_{\mathcal{T}}(S) = 0$.

Moreover, given the probability distribution $P_{\mathcal{T}}$ induced by $\mathcal{T}$ a tree (rooted or oncogenetic), for events (tree vertices) $v_i \in V$ and a root $r \in V$, the following probabilities are defined:

$$\begin{aligned} p_i = p_{ri} &= \sum_{v_i \in Y, Y \subset V} P_{\mathcal{T}}(Y), \\ p_{ij} &= \sum_{\{v_i, v_j\} \subseteq Y, Y \subset V} P_{\mathcal{T}}(Y), \\ p_{i\neg j} &= \sum_{Y \mid v_i \in Y, v_j \notin Y} P_{\mathcal{T}}(Y), \\ p_{i|j} &= \frac{p_{ij}}{p_j}, \quad p_{i|\neg j} = \frac{p_{i\neg j}}{1 - p_j}. \end{aligned}$$

As noted by Desper et al. (1999), this model is rigorous but simple because it is assumed that the causality between events is tree-like and each causation is independent from each other. These assumptions are obviously questionable and we hope that there is a tree-like model that captures, accurately enough, how the dominant genetic events occur.

Given these probabilistic models and starting from the lists of CNVs extracted from CGH data, one can define a proper *weight functional* which maps probability distributions over $\mathcal{P}(V)$ to real weight for the pairs of genetic events in $V \times V$. These weights can be used to reconstruct an oncogenetic tree as the optimum branching[1] tree or equivalently the *maximum-weight rooted tree*. The weight w_{ij} should:

- reflect the likelihood ratio for i and j occurring together: $\frac{p_{ij}}{p_i p_j}$;

[1]Note that in the optimization literature to which Desper et al. (1999) refer, a *direct tree* is usually called *branching*, while the term *tree* is reserved for the undirected version

- reflect which CNV is likely to occur first: $p_i > p_j$ if and only if event i occurs more often than event j, then it is more advantageous to have an edge from i to j than from j to i.

The weighting scheme proposed by Desper et al. (1999) is the following:

$$w_{ij} = \frac{p_i}{p_i + p_j} \cdot \frac{p_{ij}}{p_i p_j},$$

which is actually used in logarithm form:

$$w_{ij} = log(p_i) - log(p_i + p_j) - log(p_j). \tag{4.2}$$

Desper et al. (1999) also proved the following theorem which guarantees the reconstruction of the tree using a standard maximum-branching algorithm:

Theorem 4.1. *Let T be an oncogenetic tree $\mathcal{T}$ defined as* (4.1)*. The maximum branching over V with respect to the weights defined by* (4.2) *from the distribution $P_{\mathcal{T}}$ correctly reconstruct $\mathcal{T}$.*

Obviously, Theorem 4.1 applies when we know the probability distribution $P_{\mathcal{T}}$. In practice, given a set of samples we can only calculate an estimation of p_i and p_j, namely $\hat{p}_i$ and $\hat{p}_j$ for $v_i, v_j \in V$. From these we can estimate $\hat{w}_{ij}$ and find the maximum branching tree. In such context, let $\epsilon > 0$ defined such that $p_i > \epsilon$ for each event i and that for each pair of events i, j either $|p_i - p_j| > \epsilon$, or $p_i - p_{ij} > \epsilon$, then the following Theorem holds

Theorem 4.2. *If $\mathcal{T}$ is a tree with n vertices (non including the root r), and p_{min} is the minimum probability of observing any event, then with $N = \frac{8lnn}{\epsilon^2 p_{min}}$ samples of $P_{\mathcal{T}}$, the probability that the algorithm returns a false edge is less than $1/n^2$.*

Theorem 4.2 lead very restricting conditions to the use of the method on a very complex cancer dataset with a possibly high number of genetic events. In order to obtain reliable trees, one have to focus the interest on a restricted list of alterations with an higher p_{min}. Anyway, because ϵ cannot be estimated from the data, the number of needed sample still remain indicative.

A software implementation of the method proposed by Desper et al. (1999) is available at `http://www.ncbi.nlm.nih.gov/CBBresearch/Schaffer/cgh.html`.

4.2.2 Standard alterations extraction

The goal of this step is to produce a list of alterations associated to each aCGH in input. These lists will be used as input for the oncogenesis tree inference model presented above.

After the normalization, we have a series of log-ratio signals from which extract the alterations. We used a standard method proposed for smoothing and segmentation of aCGH data, namely cghFLasso by Tibshirani and Wang (2008).

This method is a precursor and a single-sample version of the FLLat model presented in Section 3.2. It is a regularized minimization model based on the combination of an ℓ_1 (*Lasso*) and a *Total Variation* (TV) penalties, also called *Fused Lasso* (FL) by (Tibshirani et al., 2005),

$$\boldsymbol{y}^* = argmin_{\hat{\boldsymbol{y}}} \left\{ \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 + \lambda \|\hat{\boldsymbol{y}}\|_1 + \mu TV(\hat{\boldsymbol{y}}) \right\},$$

where the resulting $\boldsymbol{y}^*$ is a smoothed aCGH input signal $\boldsymbol{y}$ with a forced simpler shape: sparse (ℓ_1) and piecewise constant (TV).

From such a smoothed signal the non-zero probes, which indicates gains or losses, can be extract easily. In order to manage noise and false positives, a list of altered chromosomes bands is extracted, taking into account the agreement between probes related to the same band: only chromosome segments where at least 50% of probes are in agreement with respect to the alteration status are included into the list.

Those lists of alterations associated to each aCGH sample are the inputs required by the oncogenetic inference algorithm.

4.2.3 CGHDL-based alteration extraction

The Theorem 4.2 states that for a good tree reconstruction the number of nodes (genomic events) that can be managed depends by the number of samples involved into the reconstruction. When to add new sample to an oncogenetic analysis is unfeasible, a low number of genetic events should be managed. Unfortunately in cancer the number of genomic alteration cannot be always controlled. Moreover, with a small samples size can be difficult to manually distinguish between relevant and not relevant events to include into the oncogenetic analysis.

To approach this problem we propose to exploit the ability of CGHDL, the model for aCGH data analysis and segmentation proposed in Section 3.3. Our model returns two direct output: a dictionary of main atoms $\mathbf{B}$ and a coefficients matrix $\mathbf{\Theta}$ from which calculate a smoothed version $\hat{\mathbf{Y}} = \mathbf{B}\mathbf{\Theta}$ of the aCGH signals stacked as column of the input matrix $\mathbf{Y}$.

The idea, also exploited by Subramanian et al. (2012) but with a different approach, is to use the atoms produced by CGHDL as *patterns of genomic events* to arrange in the oncogenesis tree model. This choice is justified by the fact that CGHDL is able to extract from data, pattern of co-occurent genomic events shared by a subgroup of samples. Each pattern will be described by one (or more) atoms into the dictionary $\mathbf{B}$. Moreover,

the matrix Θ directly indicates the set of atoms/patterns occurred on each sample.

We can prepare a valid input for the oncogenesis inference algorithm, transforming Θ in a binary matrix (value 1 associated to the non-zero entries) and adding a virtual row with all $1s$. The added row, introduce a *root* for the tree we would like to reconstruct. Such a root describe the *null-pattern*, that is the possibly starting healthy status shared by all the samples. Each column of Θ indicates the list of patterns of genomic events (atoms plus root) associated to each sample. With such approach we are able to reduce the number of nodes, going from the number of single genomic events to the number of atoms.

4.3 Experiments and results

In this section we report the results obtained inferring tree models of oncogenesis with the pipelines presented in this chapter.

We focus on a the Neuroblastoma disease, and we show how both approaches can infer useful information from the data. We also discuss about the improved interpretability of the results as effect of the inclusion of CGHDL in the analysis.

4.3.1 Datasets description

Different public datasets are used with the approaches described in this chapter. We are particularly interested in the analysis of Neuroblastoma (NB) disease. Neuroblastoma (NB) is the most frequent pediatric extra-cranial solid tumor that develops from nervous tissue. NB presents itself as a disseminated disease with an **heterogeneous clinical behavior**. Patients were classified according to the International Neuroblastoma Staging System (INSS, see Figure 4.3) (Brodeur et al., 1993). Our analysis investigates the oncogenesis of NB using aCGH technology and focuses on the differences between metastatic high-risk stages **4** (rapid progression of disease) and **4S** (spontaneous disease regression).

Datasets come from different laboratories and are developed on different aCGH platforms. Data were downloaded from Gene Expression Omnibus (GEO[2]) as raw files and normalized before the downstream analysis with our methods. From the following datasets, we considered only the sample for which we can remap all the probes to the last human genome reference (hg19) and manually perform a normalization in order to have the same genome reference and the same preprocessing protocol.

[2] `http://www.ncbi.nlm.nih.gov/geo/`

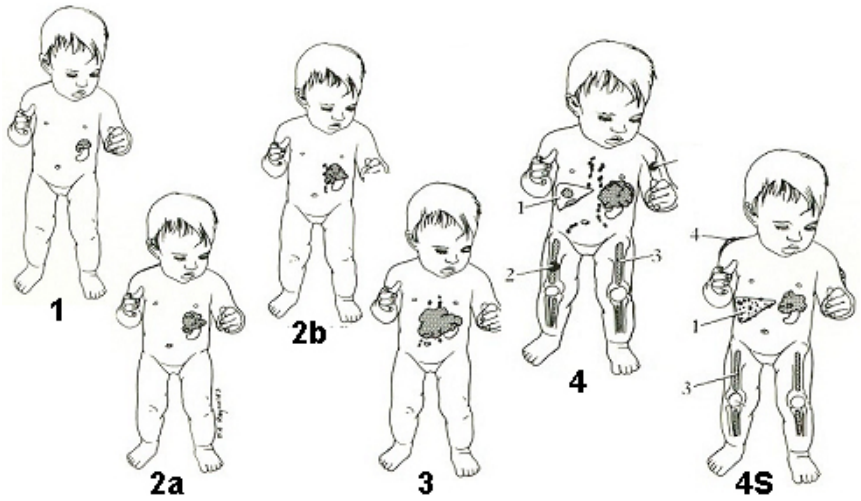


Figure 4.3: International Neuroblastoma Staging System (INSS): (1) localized tumor confined to the area of origin; (2A) unilateral tumor with incomplete gross resection, identifiable ipsilateral and contralateral lymph node negative for tumor; (2B) unilateral tumor with complete or incomplete gross resection with ipsilateral lymph node positive for tumor, identifiable contralateral lymph node negative for tumor; (3) tumor infiltrating across midline with or without regional lymph node involvement or unilateral tumor with contralateral lymph node involvement or midline tumor with bilateral lymph node involvement; (4) dissemination of tumor to distant lymph nodes, bone marrow, bone, liver, or other organs except as defined by Stage 4S; (4S) age < 1 year old with localized primary tumor as defined in Stage 1 or 2, with dissemination limited to liver, skin, or bone marrow (less than 10% of nucleated bone marrow cells are tumors).

GSE25771 consisted in 133 samples measured on 4 different Agilent platforms (GPL2873, GPL2879, GPL5477 and GPL4093) ranging from 44k to 105k probes of resolution. Three different groups of patients with NB were collected:

- **G1**: **49** patients at stage **4S**, MYCN-;
- **G2**: **37** patients at stage **4**, younger than 18 months of age, MYCN- and without disease progression with at least 3 years of follow-up;
- **G3**: **47** patients at stage **4**, older than 19 months of age, with unfavorable outcome, characterized by progression and dead for disease, within 3 years to diagnosis;

GSE14109 contains NB primary tumors collected at the onset of disease. All patients were classified as stage 4 and they were older than 1 year of age at time of diagnosis. All the tumor samples were hybridized on Agilent 44k resolution platform (GPL2873 and GPL5477). This dataset has an intersection with the GSE25771. The number of new samples is then 14.

GSE35953 contains 22 new samples with respect to the datasets GSE14109 and GSE25771 hybridized on Agilent 44k resolution platform (GPL2873 and GPL5477) and classified as stage 4.

GSE26494 is made by aCGH profiling of human NB samples obtained from infants included in the INES99.1, INES99.2 and INES99.3 trials. Each of the tumoral genomic DNAs was hybridized against non-tumoral DNA reference on BAC/PAC array or commercial supports in order to determine an overall genomic profile. The reference DNA was obtained from the blood of a single normal individual. The dataset comprise 5 different aCGH platform: a NimbleGen 72k resulution platform, an Agilent 44k resultion platform and 3 different BAC-based platforms of variable number of probes. From this datasets, 108 samples belonging to the platform GPL11633 (Agilent), GPL8971 (NimbleGen), GPL9715 (IntegraChip BAC), were considered. These platforms are characterized by 28 stages 4 and 80 stages 4S NBs respectively.

4.3.2 Neuroblastoma oncogenetic trees from genomic events

We tested the pipeline described in Sections 4.2.1 and 4.2.2 on the dataset **GSE25771**. The study of tumorigenesis of NB focuses on the differences between stages 4 and 4S, generating two different oncongenetic trees presented in Figure 4.4 and Figure 4.5.

The results of our analysis were consistent with well-known NB properties already verified in the literature. Indeed, both stages are characterized by chromosomes 7, 2, 12 alterations, and also 3, 18 and 6 aberrations are detected as very relevant (Schleiermacher et al., 2007). According to the literature (Krona et al., 2008), 17q gain is an early event, but 4p1 loss seems to be more frequent in stage 4 tumors (poor prognosis). Moreover, according to Coco et al. (2012), stage 4S tumors are characterized by numerical aberrations (alteration of a chromosome as a whole), while stage 4 by structural aberrations (alteration of a chromosome segment) . This aspect can be easily highlighted, organizing all the genomic events in decreasing order of distance with respect to the root. Given a node, the higher is such a distance, the lower is the probability to see the associated genomic event into the data. As expected (Table 4.1), the ordered events related with stage 4S are grouped by chromosomes showing that there is an high probability that these alterations occur together. In Table 4.1, the genomic events are colored consistently between the two columns and with the nodes in the trees reported in Figures 4.4 and 4.5.

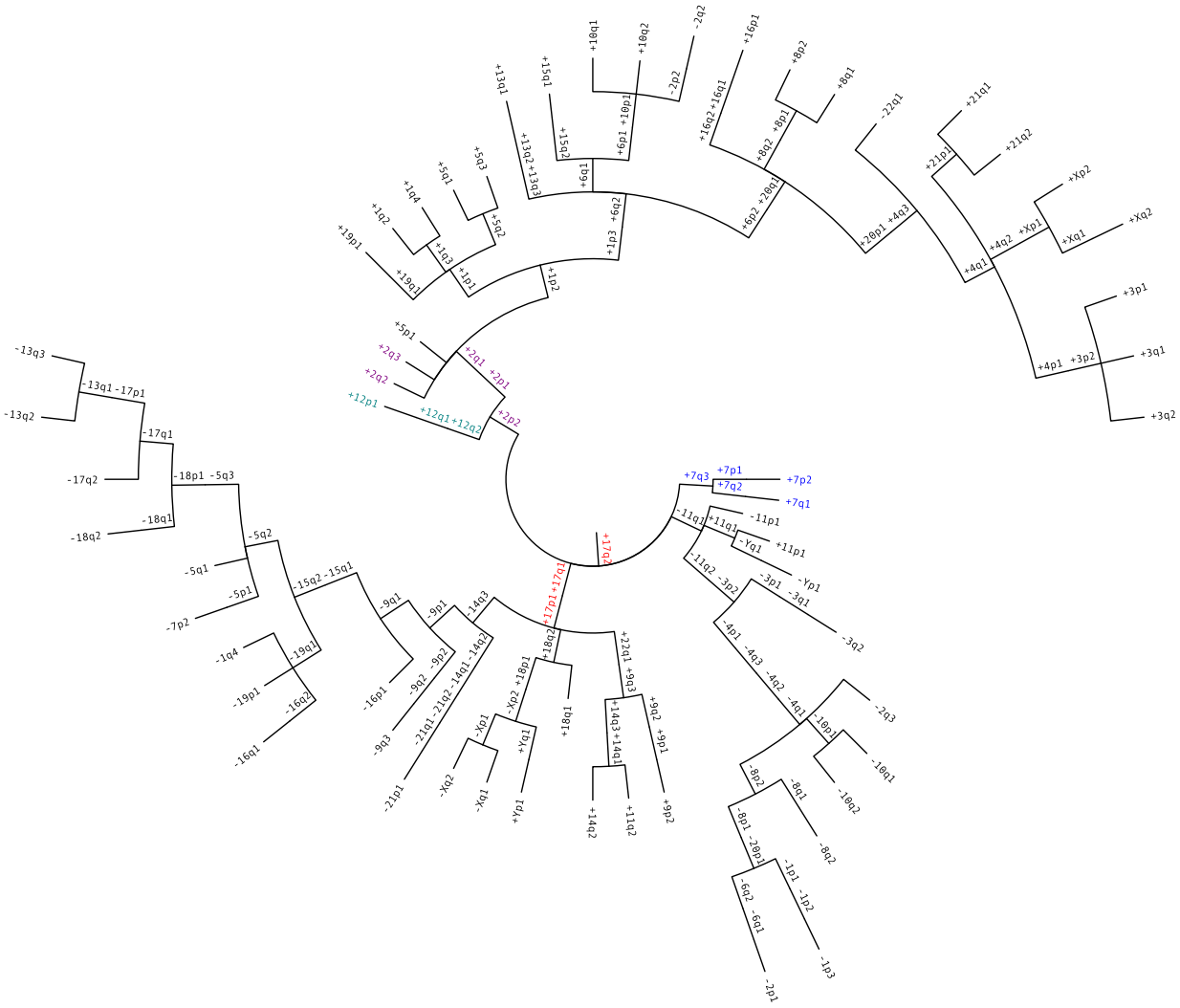


Figure 4.4: Oncogenetic branching trees for NB 4 stages.

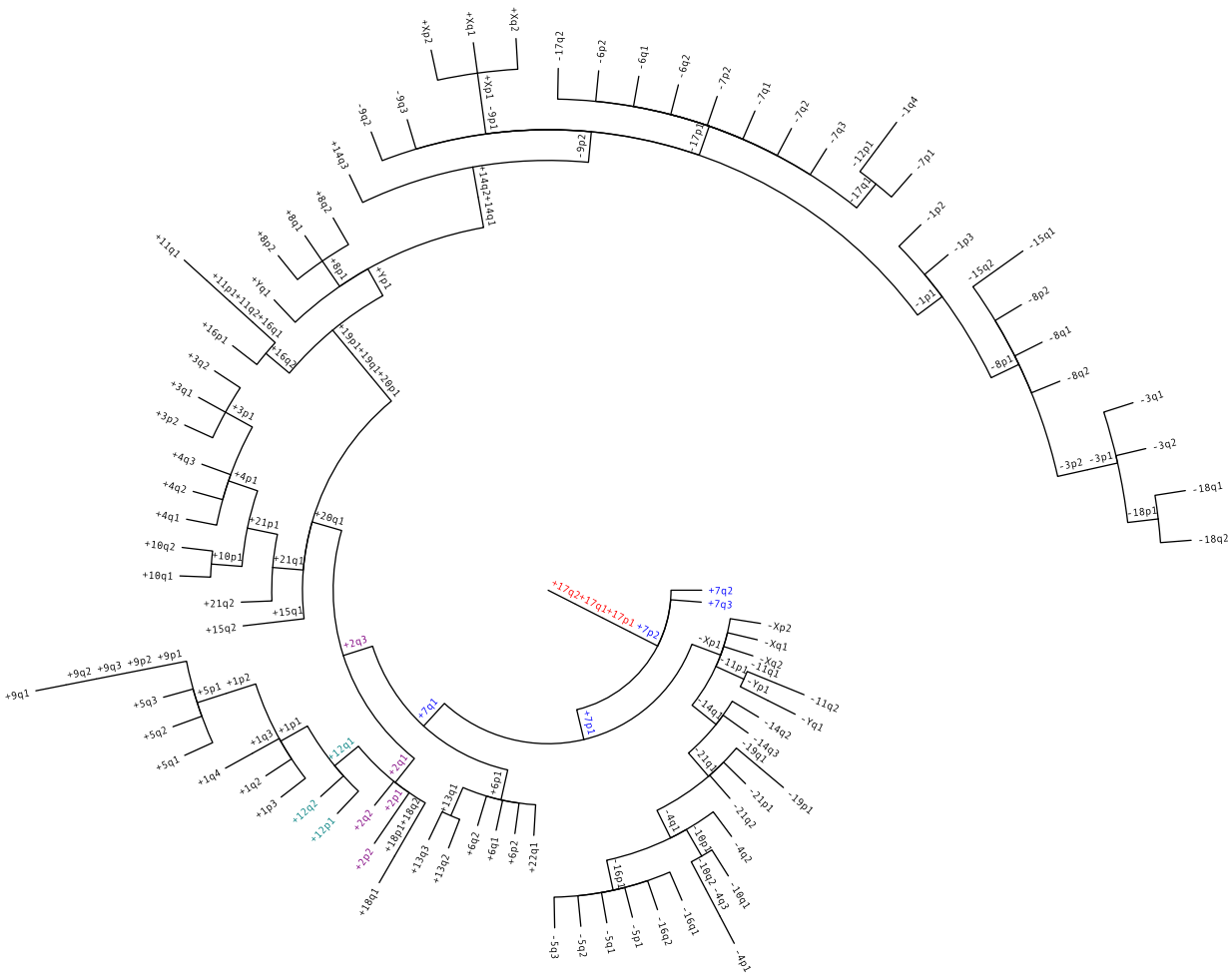


Figure 4.5: Oncogenetic branching trees for NB 4S stages.

Stage 4	Stage 4S
+17q2	+17q2
+7q3	+17q1
+7q2	+17p1
+17q1	+7q2
-11q1	+7q3
+7p1	+7p2
+2p2	+7p1
+7p2	+7q1
-11q2	+2q3
+7q1	+2p2
+12q2	+2p1
+17p1	+2q2
-3p2	+2q1
+11q1	+6p2
+2p1	+6q1
+12q1	+6q2
+18q2	+6p1
+18q1	+12q2
+18p1	+12q1
+11p1	+12p1
+22q1	+13q2
+2q1	+13q1
+12p1	+13q3
+2q2	+1p3
+2q3	+1q2
+5p1	+1q4
-3p1	+1q3
-4p1	+1p1
-14q3	+1p2

Table 4.1: First nodes in ascending order with respecting to the distance from the root.

4.3.3 Neuroblastoma oncogenetic trees from genomic patterns

In this section, we present an experiment on real Neuroblastoma data where we aim to infer oncogenesis tree of genomic events patterns with the approach described in Section 4.2.3.

Conversely to the previous experiment, here we use all the aCGH data we were able to collect from GEO. Obviously, in order to analyze our dataset with CGHDL, we need to construct a single data matrix $\mathbf{Y}$ from row data hybridized on different platform at different genomic resolution.

The alignment was performed after the normalization. The approach we adopted is a variation of the algorithm proposed by Jong et al. (2007) and is composed by three steps:

- **Sampling**: to deal with the varying positions of the different clones on the genome, N positions were sampled on each **chromosome band** at equal spacing. This approach weighs each band equally and distribute the sampled clones with a similar original proportion.

- **Interpolation**: the DNA copy number ratio for each sampled position was set to the mean value of the closer K position in the data.

- **Standardization**: The dynamic range for the aCGH ratios may vary across platforms and across hybridizations. A single-copy alteration, for example, may gives an higher or lower value compared to other platforms (Ylstra et al., 2006). Moreover, a similar effect is due to a different purity of the samples, that is the proportion of tumor and healthy cells into the hybridized tissue. For each sample we divide the interpolated log-ratios for the sample standard deviation. Originally, Jong et al. (2007), trasformed the log-ratios in z-scores, by also subtracting from the log-ratios the average over all positions. However, this approach has the effect to shift the aCGH signal with respect to 0. Because we assume that the aCGHs were previously normalized and centered we decide only to divide by the standard deviation.

In this experiment, we choose $K = 10$ number of neighborhood and $N = 10$ number of points for each band, giving us aCGH signal composed by approximately 8k probes. This is a reasonable trade-off between high and low resolution platforms that we aimed to combine together.

Actually in literature, there are other alignment approaches (Tian and Kuang, 2010). We decide to use the simplest one to not introduce a source of variability out of our control and understand if the oncogenesis approach we are proposing is feasible and reliable.

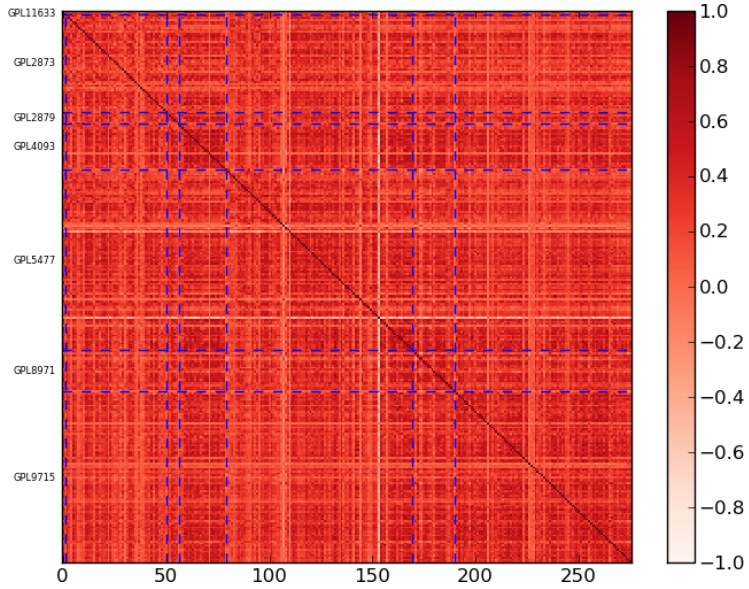


Figure 4.6: Pearson Correlation Coefficient between the 277 samples after the alignment process. Dotted blue lines separate samples belonging to different platforms. Rectangles along the diagonal represent PCC values for samples belonging to the same platform.

To evaluate the performance of pre-processing, we used the Pearson Correlation Coefficient (PCC) (Guo et al., 2011)

$$PCC(x,y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}.$$

The PCC value was calculated between each pair of samples after the alignment process. The PCC values of any two samples should not show the dependencies on the platform. In Figure 4.6 all the PCC values are reported. The only evident correlation patter is between the platform GPL4093 and all the other platforms. Note that the GPL4093 platform is the one with the higher resolution (Agilent 105k). On this platform the alignment phase has also an higher impact, reducing the size from 105k clone to 8k sampled clones.

The matrix $\mathbf{Y}$ thus obtained was used as input for CGHDL searching for a dictionary of 12 atoms with a BIC-based model selection schema. The best solution, reported in Figures 4.7 and 4.8, was obtained with $\mu = 1.0$, $\lambda = 0.001$ and $\tau = 0.01$. The learned dictionary $\mathbf{B}$ contains only the major patterns of alterations distributed across the 12 atoms (see Figure 4.7). Such atoms are used by different subgroups of samples, as explained by the sparsity patterns into the matrix $\mathbf{\Theta}$.

Note that we performed an unsupervised extraction of the main patterns from the data. The information related to the cancer status will be used only when generating the on-

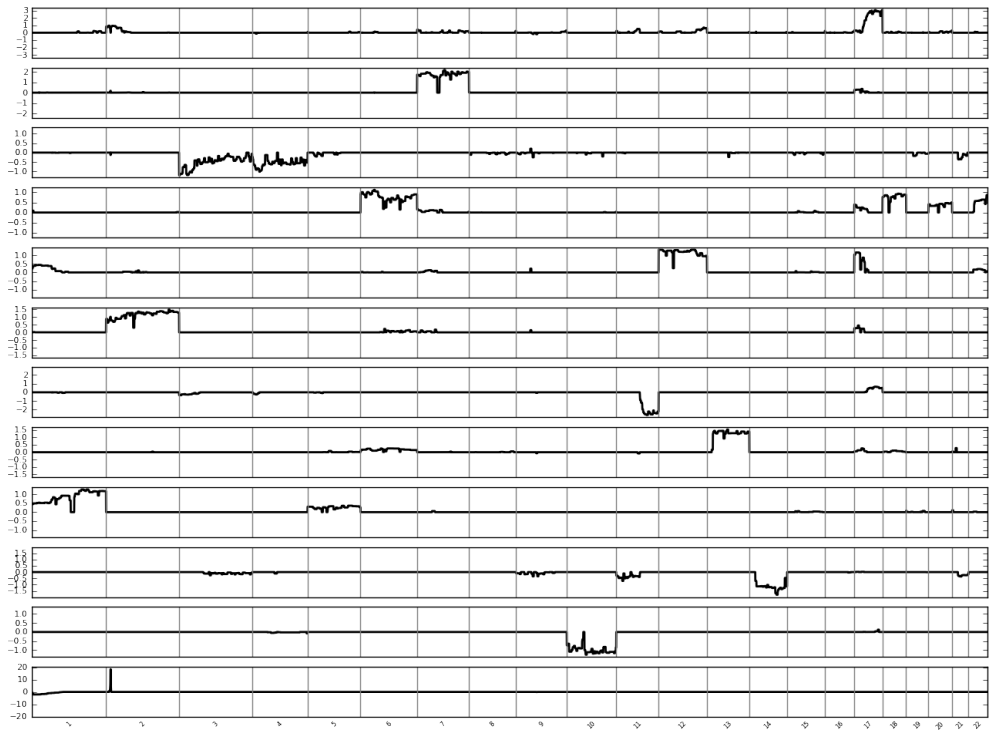


Figure 4.7: Resulting dictionary obtained by CGHDL on the aligned NBs aCGH data. Atoms are ordered with respect to the their usage for the reconstruction of the original signals. Atoms at the top are more used than atoms at the bottom.

congenetic trees. In this way we aim at describing the two *flows* of alterations starting from the same basic "ingredients". The list of alterations associated with the atoms and ordered by the sum of the rows of $\mathbf{\Theta}$, are reported in Table 4.2. The significance of the atoms is represented by the sum of the rows of $\mathbf{\Theta}$.

The analysis of the atoms reported in Table 4.2 shows similarities with the results obtained from a sample-by-sample analysis and the oncongenetic tree obtained with the experiment reported in Section 4.3.2. Also in this case we obtain that the main alterations in NB are the gains related with chromosomes 17, 7 as reported in literature (Schleiermacher et al., 2007). Very interesting is the atom 3 which includes a $4p1$ loss and is highly used to reconstruct the data. This genomic alteration was reported by Krona et al. (2008) as an early event together with $17q$ gain (atom 1).

At this point, we aim to use the matrix $\mathbf{\Theta}$ as an input for the algorithm proposed by Desper et al. (1999) in order to obtain a hierarchical organization of the atoms. The resulting trees are reported in Figures 4.9 and 4.10. Carefully observing the trees reconstructed by the algorithm, we can note an interesting aspect with respect to the results reported by Krona et al. (2008). In the tree related to the stage 4 NBs (poor outcome), we have

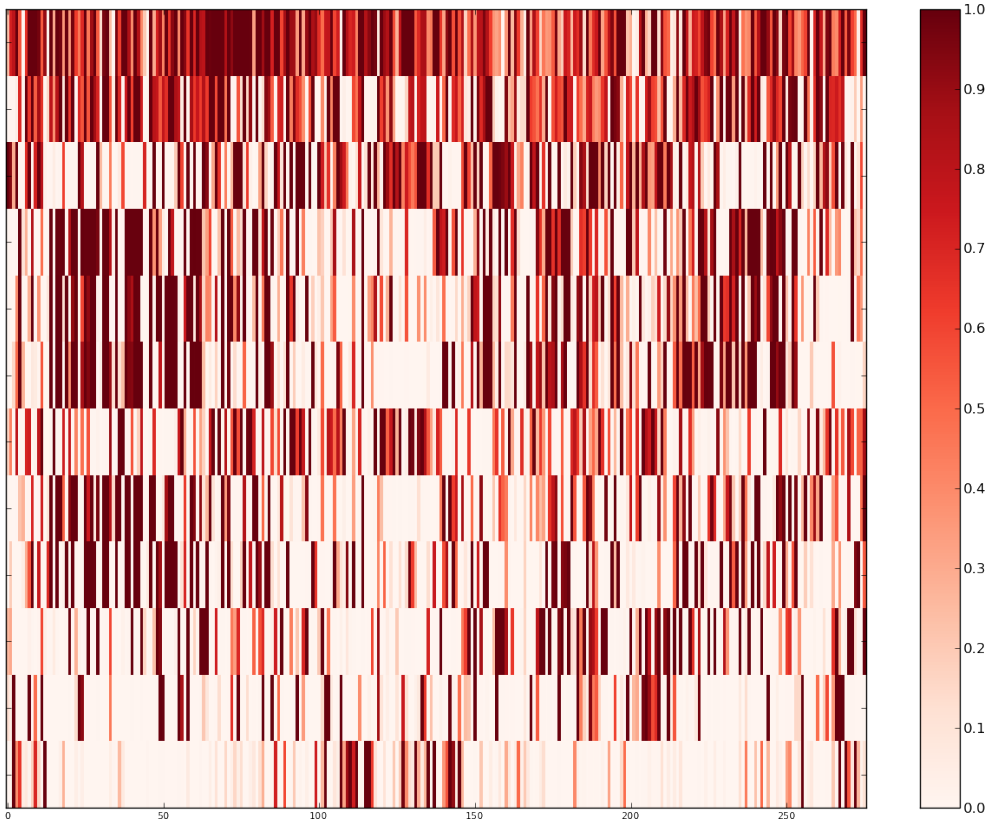


Figure 4.8: Resulting coefficients matrix obtained by CGHDL on the aligned NBs aCGH data. The order of the coefficients (by row) follows the order of the atoms in Figure 4.7. Atoms at the top are more used than atoms at the bottom.

atom 1 (with gain on $17q$) and 3 (with loss on $4p1$) in a path starting from the root of the tree. Conversely for stage 4S (spontaneous disease regression), the two atoms follows to separate paths from the root. The analysis of the relation between this two alterations could be a good starting point for further analysis.

Atom	List of alterations
Atom #1	+11q1, +12q2, **+17q1**, **+17q2**, +2p2
Atom #2	+7p1, +7p2, +7q1, +7q2, +7q3
Atom #3	-3p1, -3p2, -3q1, **-4p1**, -4q1, -4q2, -4q3
Atom #4	+18p1, +18q1, +18q2, +20q1, +22q1, +6p1, +6p2, +6q1, +6q2
Atom #5	+12p1, +12q1, +12q2, +17p1, +17q1
Atom #6	+2p1, +2p2, +2q1, +2q2, +2q3
Atom #7	+17q2, -11q1, -11q2
Atom #8	+13q1, +13q2, +13q3
Atom #9	+1p1, +1p2, +1p3, +1q2, +1q3, +1q4
Atom #10	-11p1, -14q1, -14q2, -14q3
Atom #11	-10p1, -10q1, -10q2
Atom #12	+2p2, -1p2, -1p3

Table 4.2: Lists of alterations associated with the atoms ordered by the sum of the rows of the coefficients matrix $\boldsymbol{\Theta}$.

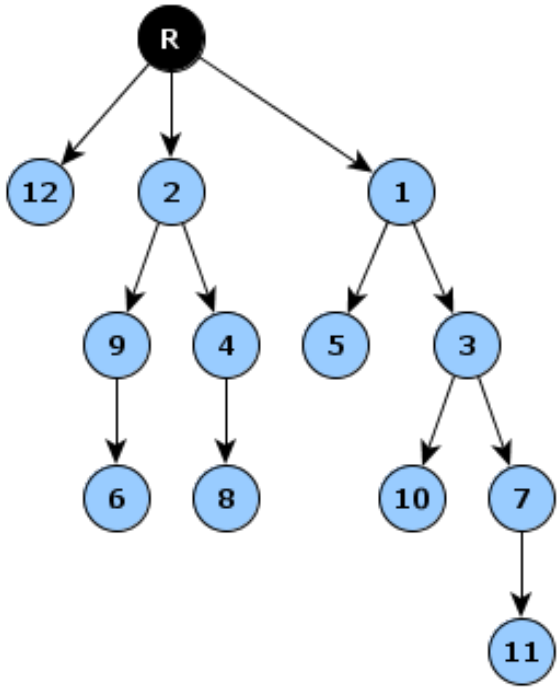


Figure 4.9: Oncogenetic branching trees for NB 4 stages from CGHDL analysis. See Table 4.2 for alterations associated to each node.

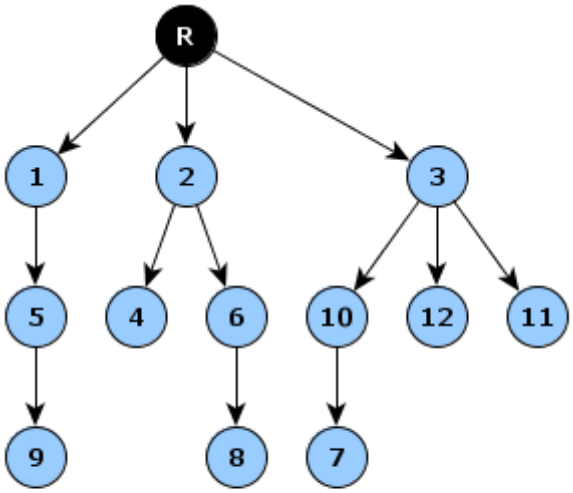


Figure 4.10: Oncogenetic branching trees for NB 4S stages from CGHDL analysis. See Table 4.2 for alterations associated to each node.

Chapter 5

Developed software libraries

In this chapter we present a set of tools and libraries implemented to performs the experiments described in this thesis. All the libraries and modules are always designed, from the very beginning, in order to be publicly released as open source projects.

Together, these libraries can be viewed as a set of modules that can be easily composed to build more complex analysis tools. A working example is KDVS (Knowledge Driven Variable Selection), created by Zycinski et al. (2013), and L1L2Signature (Section 5.3) which rely on L1L2Py (Section 5.1) and PPlus (Section 5.2).

5.1 L1L2Py: feature selection by means of $\ell_1\ell_2$ regularization with double optimization

L1L2Py[1] implements the $\ell_1\ell_2$ regularization framework proposed by De Mol et al. (2009b), described in Section 2.3. The method implements the algorithm 2.1 and is based on the optimization of the $\ell_1\ell_2$ functional presented by De Mol et al. (2009a). The estimator is consistent, provides a sparse solution, preserves variables correlation and a second optimization, a step of regularized least squares, is needed to allow for a good prediction accuracy (De Mol et al., 2009b).

We recall that the framework is based on a double optimization and is composed of two stages. The first stage aims at identifying a minimal list of relevant variables. The second stage aims at extracting the almost completely nested models for increasing values of the correlation parameter.

[1] `http://slipguru.disi.unige.it/Software/L1L2Py`

The library is divided in three parts: main functions, algorithms and tools. Upon importing L1L2Py, the functions implementing the two main stages will be placed in the `l1l2py` namespace:

- Stage I (`l1l2py.minimal_model`): this stage aims at selecting the optimal values for the regularization parameters τ_{opt} and λ_{opt} within a k-fold cross validation loop, for a fixed and small value of the correlation parameter μ.
- Stage II (`l1l2py.nested_models`): for fixed τ_{opt} and λ_{opt}, this stage identifies the set of relevant lists of variables, for increasing values of the correlation parameter μ.

This module also provides a wrapper function (`l1l2py.model_selection`) that runs the two stages sequentially.

The module `l1l2py.algorithms` implements two regularization algorithms, that are essential to the main functionals, `ridge_regression` solves a classical regularized least squares (RLS) problem and `l1l2_regularization` minimizes the Elastic Net functional using an iterative shrinkage-thresholding algorithm (De Mol et al., 2009a). The module also provides a function (`l1_bound`) that estimates the maximum value for the sparsity parameter τ and the `l1l2_path` that evaluates the regularization path for a fixed value of μ and increasing values of τ. This acceleration method based on *warm starts* has been theoretically proved by Hale et al. (2008).

The module `l1l2py.tools` is composed by utilities like linear and geometric range generators, functions for data normalization, classification/regression error functions and cross validation utilities.

Figure 5.1 depicts a typical use case: provide a training and a test set, the former is firstly divided in K splits that are used by Stage I (`minimal_model`) to estimate the optimal pair $\tau_{opt}, \lambda_{opt}$ for a fixed (and close to zero) value of μ. Stage II (`nested_models`) allows to identify a family of almost nested lists of relevant variables for different values of the parameter μ. The prediction error is evaluated on each list.

The few lines of code reported in Listing 5.1 perform a full experiment, following the pipeline sketched in Figure 5.1. In this simple example, the training set is divided in 5 cross-validation splits (`l1l2py.tools.kfold_splits`).
The `l1l2py.model_selection` function executes Stage I with $\mu = 10^{-6}$, searching for the optimal pair $\tau_{opt}, \lambda_{opt}$ over a grid of 10×10 values. Then, for 4 increasing values of μ, the nested models will be calculated and their prediction ability will be tested on the test set. A complete tutorial on how to install and use L1L2Py is available online[2].

[2] `http://slipguru.disi.unige.it/Software/L1L2Py/tutorial.html`

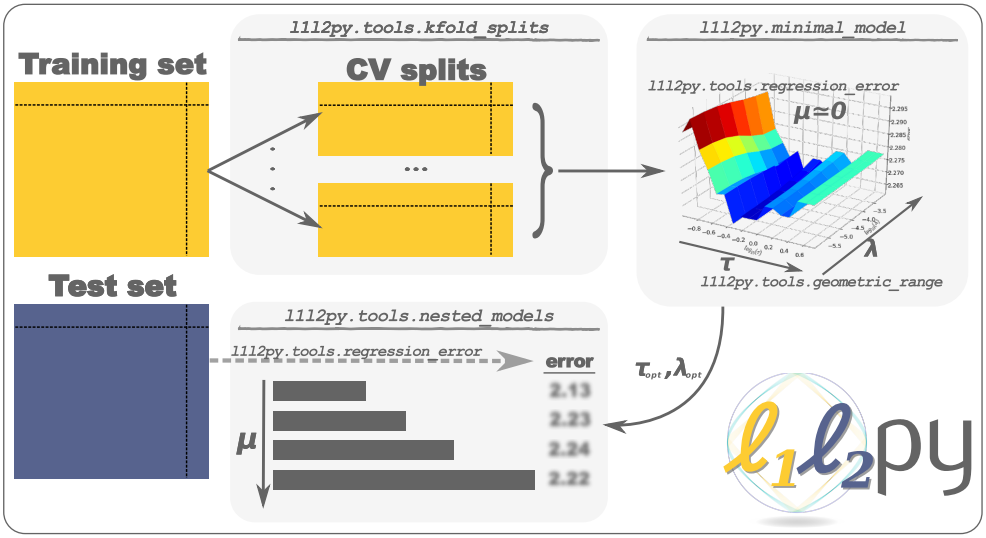


Figure 5.1: Provided a training and a test set, Stage I performs a K-fold cross validation to select the optimal values τ_{opt} and λ_{opt} for fixed μ and estimates the test error. Stage II builds the models for larger values of μ.

Listing 5.1: Python code implementing a simple example with L1L2Py

```
import l1l2py
cv_splits = l1l2py.tools.kfold_splits(train_labels, k=5)
tau_range = l1l2py.tools.geometric_range(tau_min, tau_max, 10)
lambda_range = l1l2py.tools.geometric_range(lam_min, lam_max, 10)
mu_range = l1l2py.tools.geometric_range(1e-6, mu_max, 4)

result = l1l2py.model_selection(train_data, train_labels,
                        test_data, test_labels,
                        mu_range, tau_range, lambda_range, cv_splits,
                        cv_error_function=l1l2py.tools.regression_error,
                        error_function=l1l2py.tools.regression_error,
                        data_normalizer=l1l2py.tools.center)

for mu, sel in zip(mu_range, result['selected_list']):
    print "%.3f:" % mu, sel.nonzero()[0]
```

5.2 PPlus: a parallel Python environment with easy data sharing

PPlus[3] is a simple environment to execute Python code in parallel on many machines. It is actually a fork of Parallel Python[4], another simple but powerful framework for parallel execution of python code, which lacks features needed for effective use in our daily research.

More specifically, PPlus was created to answer the following needs:

- to facilitate data transport over distributed environment of usually very big file, exposing a simple interface while handling details in the background;
- to separate file handling between different experiments, so one machine can participate in many computational experiments simultaneously.

A distributed environment controlled by PPlus is composed of a set of machines (nodes) that offer their resources to execute assigned tasks. All those nodes are running the `pplusserver.py` process in the background, that provides visibility over the local network of a prefixed number of computational workers and controls all data transfers.

The Python code to be executed by PPlus consists of the following conceptual pieces:

- the worker code is distributed over the network to the node machines to be executed there; it produces partial results saved locally and ready to be collected;
- the master code that distributes the worker code pieces, collects all partial results and produce master (final) results.

When a Python code needs to be executed in parallel, it is placed on one of the machines. That process is designated to be the master process: it distributes all parallel tasks to node workers and it receives all the results. Both worker code and master code can do any computations, import modules (with some restrictions), and produce files. Internally, PPlus uses the concept of experiments to organize code and data. The experiment consists of the code that performs a specific task, including pieces to be executed in parallel (i.e. master code and all worker code), as well as all regular files produced by that code. A single instance of the worker code, submitted for remote execution, is also called worker task or worker job (Figure 5.2).

[3] `http://slipguru.disi.unige.it/Software/PPlus`
[4] `http://www.parallelpython.com/`

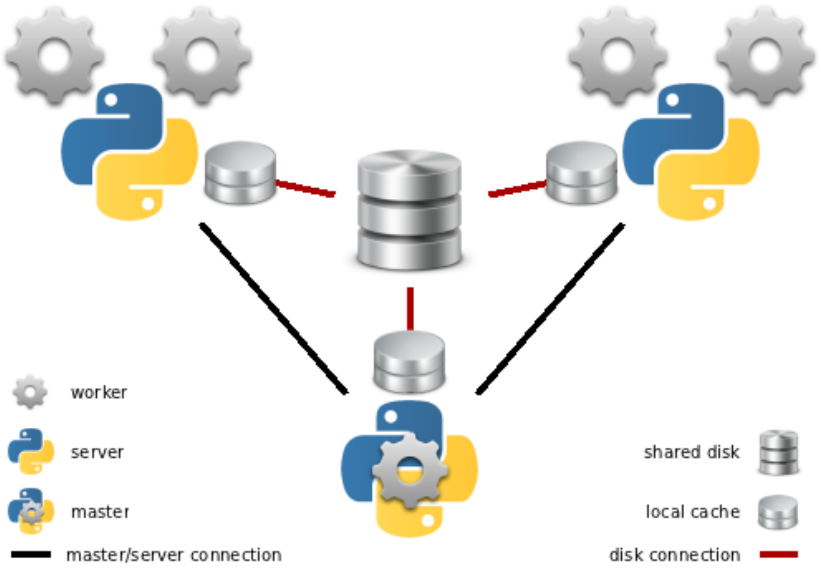


Figure 5.2: General scheme of the PPlus environment. Each node has a local cache disk space and expose one ore more workers. All the nodes share a central disk space. An experiment starts from the master node which also has the responsibility of collecting results.

To ease running of multiple subsequent experiments over the network, the experiment code can use a shared file system resource to store files produced during execution and to access them back if needed. This functionality is controlled by PPlus in a transparent way and it is exposed through a simple API. The experiment code can access and store any remote files in a dedicated experiment directory, created specifically for that purpose on the shared disk resource. Both experiment master code and all worker code have an access to the experiment directory. The files produced by different experiments are physically separated; no direct support is provided for accessing data from outside the experiment. As a result, many experiments can run simultaneously without data corruption.

Each execution of the code, of both master and worker type, on each machine, is considered a session. All the activity during a session is stored in a separated session log file. More specifically all worker tasks are considered running in separate sessions, and will produce separate session log files. The master code is also treated as running in separate session, but it will produce two logs. *Session log* documents the activity regarding accessing shared disk resource from within master code. *Experiment log* documents the activity regarding distribution of worker tasks.

Remote files are managed based on file keys. They serve as identifiers for accessing physical files without knowing their precise location, regardless of the network protocols. The following rules apply to file keys regarding experiment level:

- between different experiment directories, the same file keys may be used; that is, key "BIGFILE" used in experiment A, and key "BIGFILE" used in experiment B, are both referring to two different files within experiment directory;
- within experiment:
  - all file keys must be unique to avoid unwanted data corruption.
  - if the same file key is used for opening new remote file for writing, by default the content of the existing file will be overwritten without warning; otherwise, an error will be reported;

PPlus uses logging to record its activity during the execution of experiment code. The following logs are used:

- **Experiment log**: this log is created by master code when experiment ID is granted. It documents the activity of the master code regarding control of worker tasks and interaction with Parallel Python. Also, all errors in worker tasks will be logged here. It is considered private and is not exposed through public API.
- **Master session log**: this log is created by master code when experiment ID and session ID are granted. It primarily documents the activity of the master code regarding remote file access. It is considered public and is exposed through public API.
- **Session log**: this log is created by each single worker task, with experiment ID given and session ID granted. It documents the activity of the worker code regarding remote file access. It is considered public and is exposed through public API.

Logs produced in local caches (*session log*) are never transferred to shared disk resource after the experiment has been finished. They must be accessed manually on each machine.

5.3 L1L2Signature: unbiased framework for -omics data analysis

L1L2Signature[5] is the natural companion of the L1L2Py library. It implements the $\ell_1\ell_2 fs$ feature selection framework proposed by Barla et al. (2008) and described in Section 2.3.

[5] `http://slipguru.disi.unige.it/Software/L1L2Signature`

This library is composed by a set of Python scripts and a set of useful classes and functions that could be used to manually read and/or analyze high-throughput data extending/integrating the proposed pipeline.

L1L2Signature other than L1L2Py for the gene selection core, relies on PPlus which is used to parallelize cross validation splits in an easy and effective way.

Input data (gene expression matrix and labels) are assumed to be textual ad separated by a char (delimiter). Labels contains information about the given samples, indicating a real "score" or a class.

L1L2Signature configuration file is a standard Python script. It is imported as a module, then all the code is executed. Actually all configuration options are mandatory and it is possible to generate a default configuration file, as the one in Listing 5.2, with the `l1l2_run.py` script.

Configuration file is fully documented and it imports L1L2Py in order to use some useful tools. User is free to use personalized functions if they follow the same API.

After the user defines all the option needed to read the data and to perform the model assessment, the crucial phase is to properly define a set of ranges of parameter involved. In order to help users choosing a good relative τ range, they can use the `l1l2_tau_choice.py`.

The `l1l2_run.py` script, executes the full framework. When launched, the script reads and splits the data, then it runs L1L2Py on each external split and collects the results in a new sub-directory of the `result_path` directory. Such a directory is named as: `l1l2_result_<TIMESTAMP>` and it contains all information needed for the following analysis step.

Note that data and configuration file are hard-linked inside the result directory which, in that way, becomes completely portable and self contained.

In the last step, users can get some useful summaries and plots from an already executed experiment. The `l1l2_analysis.py` script accepts as only parameter a result directory already created. The script prints some results and produces a set of textual and graphical results:

Cross Validation Errors. The script generates a list of `kcv_err_split_*.png`, one for each external split (as averaged error across internal splits). Moreover, it generates an averaged plot: `avg_kcv_err.png`. On each plot, a blue dot indicates the minimum.

Prediction Errors. The script generates a box plot for both test and training errors, respectively `prediction_error_ts.png` and `prediction_error_tr.png`. They show the averaged prediction error over external splits in order to asses performance and stability of the signatures (for each level of considered correlation, μ values).

Listing 5.2: L1L2Signature default configuration file

```
# Configuration file example for L1L2Signature
# version: '0.2.2'

import l1l2py

#~~ Data Input/Output ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# * Data assumed csv with samples and features labels
# * All the path are w.r.t. config file path
data_matrix = 'data/gedm.csv'
labels = 'data/labels.csv'
delimiter = ','
samples_on = 'col' # or 'row': samples on cols or rows
result_path = '.'

#~~ Data filtering/normalization ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
sample_remover = None # removes samples with this label value
variable_remover = 'affx' # remove vars where name starts with
data_normalizer = l1l2py.tools.center
labels_normalizer = None

#~~ Cross validation options ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# * See l1l2py.tools.{kfold_splits, stratified_kfold_splits}
external_k = 4    # (None means Leave One Out)
internal_k = 3
cv_splitting = l1l2py.tools.stratified_kfold_splits

#~~ Errors functions ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# * See l1l2py.tools.{regression_error, classification_error,
#                     balanced_classification_error}
cv_error = l1l2py.tools.regression_error
error = l1l2py.tools.balanced_classification_error
positive_label = None # Indicates the positive class in case of 2-class task

#~~ L1l2 Parameters ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
# * Ranges will be sorted from smaller to bigger value!
# * See l1l2py.tools.{geometric_range, linear_range}
tau_range = l1l2py.tools.geometric_range(1e-3, 0.5, 20) # * MAX_TAU
mu_range = l1l2py.tools.geometric_range(1e-3, 1.0, 3) # * CORRELATION_FACTOR
lambda_range = l1l2py.tools.geometric_range(1e0, 1e4, 10)

#~~ Signature Parameters ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
frequency_threshold = 0.5

#~~ PPlus options ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
debug = True # If True, the experiment runs only on the local pc cores
```

Frequencies Threshold. In order to help the user defining a good stability threshold (see `frequency_threshold` in configuration file 5.2) the script also plots (and actually print and save as `selected_over_threshold.png`), an overall summary of the number of genes selected for different thresholds and for each correlation level.

Signatures Heatmaps. In case of classification (automatically identified when labels assume only two values), the script creates a heatmap plot for each final signature (then they also depend by `frequency_threshold` option value). Images are saved as `heatmap_mu*.png` files where samples and variables are properly clustered in order to improve the visualization. Using a very small `frequency_threshold` (e.g. 0.0), signature contains the full list of variables. In that case, variables are not clustered but only ordered by frequency across splits. In order to generate such a plot, the `l1l2signature.plots.heatmap` function can be used.

Samples in PCA space. In case of classification (automatically identified when labels assume only two values), the script plots samples in a 3D space, using Principal Component Analysis (PCA), for each final signature. Images are saved as `pca_mu*.png` files.

Performance Statistics. The analysis script also produces some textual results, saved into a stats.txt file. That file is divided into some sections, each one containing at least a short table.

- **Optimal parameters**: this table describes the best parameter pairs found in each cross validation split.
- **Prediction errors**: These tables show the averaged prediction error of the signatures, before frequency/stability thresholding, for each value of correlation. They correspond to the generated prediction errors box plots.
- **Classification performances**: This table, generated only in classification for each μ, summarizes classification performances of the signature through a standard confusion matrix. Moreover, this section contains some other performance measures, as Accuracy, Balanced Classification and Matthews correlation coefficient. At last, if the `positive_label` parameter is given, into the configuration file, the script is able to calculate some other measures that assume the presence of a positive class as in the case of patients vs. controls.

Signatures. Obviously, the script generates a set of signatures, each one written in a separated text file `signature_mu*.txt`, in order to eventually simplify the parsing. Each file contains the ordered list of probes belonging to the signature. The file is tab-delimited, the signatures are thresholded with respect to the `frequency_threshold` option, and they correspond to the signatures used to generate heatmaps.

5.4 PyCGH: a Comparative Genomic Hybridization toolkit

PyCGH is Python library which implements a set of utilities for handle and analyze aCGH data.

The library contains the following main modules: `datatypes`, `readers`, `plots`, `synth`, `analysis`.

The `pycgh.datatypes` module contains a set of object-oriented classes which implement some basic data structure useful to analyze aCGH data and to build complete experiments. The main class is the `ArrayCGH` which models a generic aCGH. The instantiation of the class needs a small number of parameters with information related to the probes and reference/test signals. The class also exposes the possibility to save and load aCGH objects in a custom binary file format. The `CytoStructure` class is a map-like collection of 24 `ChromosomeStruture` objects. Each one also contains information about the cytogenetic structure (bands and their coordinates) of a specific chromosome. The `CytoStructure` class acts as a parser of file containing needed informations in UCSC format[6]. Moreover, the `DataTable` class encapsulates the concepts of *collection of homogeneous labeled samples* described by a fixed number of named variables. It is a representation of a two-entry table, where on the rows we have the different labeled samples and on the columns the different named variables. This class may be useful to handle clinical information related to the samples usually stored as CSV files.

The `pycgh.readers` module currently contains two parsing function `agilent` and `nimblegen` which can respectively read the Agilent and NimbleGen file format and return an `ArrayCGH` object. Using this parsers is possible to generalize the concept of aCGH (via the `ArrayCGH` class) leading to the possibility to analyze with the same toolkit different aCGH platforms. A third parser, namely `ucsc_mapping`, can parse chip design file in UCSC file format.

The `pycgh.plots` module contains some plotting functions, in particular there is the possibility to plot an aCGH `profile` or a `spatial` distribution of the probes across the chip (if the geometric information were given). Moreover, a standard `MA` plot can be generated[7].

Finally, the `pycgh.synth` module contains the `ArrayCGHSynth` class which implements the synthetic model for aCGH data generation presented in Section 3.4.

The code example in the Listings 5.3 shows how to generate and plot a synthetic aCGH. The Figure 5.3 shows the output of the code.

[6] `http://genome.ucsc.edu/`

[7] M is the log-ratio of test and reference signals, while A is the log-product of them.

Listing 5.3: PyCGH example script

```
import pylab as pl # standard plotting library
from pycgh import synth, datatypes, readers, plots

# Read chip definition (file downloaded from UCSC Genome Browser)
chip_design = readers.ucsc_mapping('agilentCgh4x44k.txt.gz',
                                   filter_valid=True)

# Read cytobands information (file downloaded from UCSC Genome Browser)
cs = datatypes.CytoStructure('cytoBandIdeo.txt.gz')

# Create aCGH synthetizer
source = synth.ArrayCGHSynth(geometry=(430, 103),
                             design=chip_design,
                             cytostructure=cs,
                             alterations = {'17q': [(3, .9)]}) # Gain
# Generate a male sample
acgh = source.draw('male')

# Profile plot (top)
pl.subplot2grid((2,2),(0, 0), colspan=2)
plots.profile(acgh)

# Spatial plot (bottom-left)
pl.subplot2grid((2,2), (1, 0))
plots.spatial(acgh)

# MA-plot (bottom-right)
pl.subplot2grid((2,2), (1, 1))
plots.MA(acgh)

# Save to aCGH to file
acgh.save('acgh_file.gz')

pl.show()
```

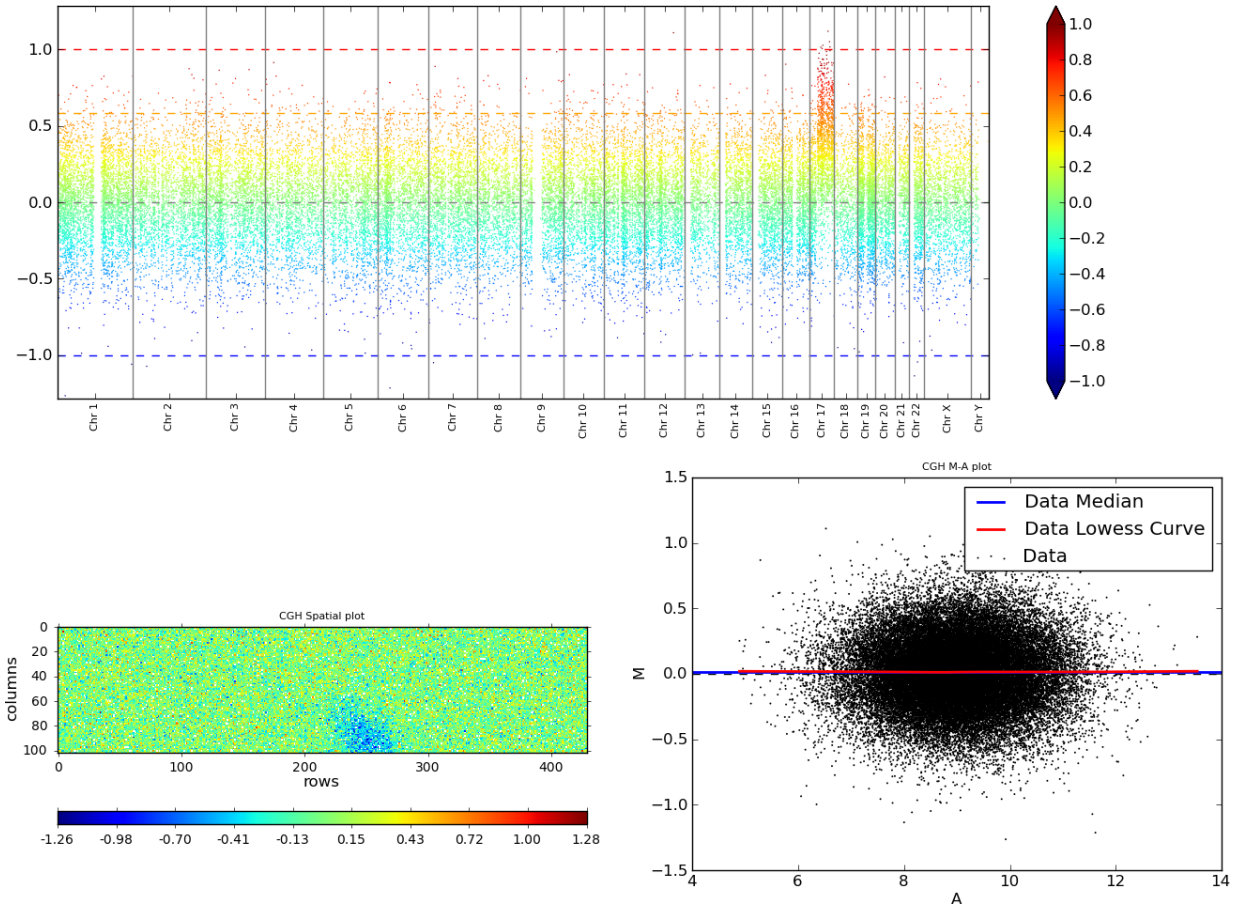


Figure 5.3: Output of the script 5.3. The synthetic aCGH generated has the required gain on the band 17q. It also shows a spatial bias but a good MA-plot (centered and symmetric with respect to 0.0). Note that in this case we did not use high signal noise (default value).

Chapter 6

Conclusions

In this thesis we studied, designed and implemented statistical learning methods for high dimensional genomic data, with a major interest in regularized linear models. The interest is on methods that incorporate (possibly in the penalty term) prior biological knowledge (*e.g.* public databases) and that are able to deal with different data types.

In Chapter 2 we introduced the problem of gene profiling approached with a feature selection method developed by our research group. We described the context, and then we focused on the statistical learning method and on the model assessment framework we adopted. In this context, the contribution of this thesis is also related with the tools developed and publicly distributed. This allows us to implement the proposed and studied methods in a software infrastructure. The developed libraries, L1L2Py, L1L2Signature and PPlus have been presented in Chapter 5.

Chapter 3 was dedicated to the main contribution of this thesis. We presented CGHDL, a novel Dictionary Learning method for the identification of genomic aberration from array-based Comparative Genomic Hybridization (aCGH) data. Currently, we are still working on CGHDL because some practical and algorithmic questions remain open. Regarding the model, we are investigating for the adoption of different penalties on the coefficients matrix, in particular we are testing different hard constraints. We are also working on the *total variation* weighting scheme in order to understand the use of different approaches, taking into account the real distances between clones on the chromosomes. Some advances are also possible with respect to the algorithm and the model selection procedure. Regarding the reproducibility of our results, we are actually including a CGHDL implementation into the PyCGH library presented in Chapter 5.

Our pipelines for oncogensis were presented in Chapter 4. We showed the differences between the standard approach and the dictionary learning based one. Also in this context the interpretability of the data resulting from CGHDL helps for the generations of

valuable tree models for oncogenesis. In this context, some other aspects should be considered for future work. Stability of the trees is one of them. Inspired by the approach adopted to evaluate the stability of the inferred interaction networks in Chapter 2, we already plan to design some experiments in that sense, also using more recent and complex oncogenesis tree models Beerenwinkel et al. (2005).

Bibliography

Alibés, A., Cañada, A., and Díaz-Uriarte, R. (2008). PaLS: filtering common literature, biological terms and pathway information. *Nucleic acids research*, 36(Web Server issue):W364–7.

Ambroise, C. and McLachlan, G. J. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences of the United States of America*, 99(10):6562–6.

Ancona, N., Maglietta, R., Piepoli, A., D'Addabbo, A., Cotugno, R., Savino, M., Liuni, S., Carella, M., Pesole, G., and Perri, F. (2006). On the statistical assessment of classifiers using DNA microarray data. *BMC bioinformatics*, 7:387.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., and Sherlock, G. (2000). Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, 25(1):25–9.

Attouch, H., Bolte, J., Redont, P., and Soubeyran, A. (2010). Proximal Alternating Minimization and Projection Methods for Nonconvex Problems: An Approach Based on the Kurdyka-Łojasiewicz Inequality. *Mathematics of Operations Research*, 35(2):438–457.

Barabási, A.-L., Gulbahce, N., and Loscalzo, J. (2011). Network medicine: a network-based approach to human disease. *Nature reviews. Genetics*, 12(1):56–68.

Baralla, A., Mentzen, W. I., and de la Fuente, A. (2009). Inferring gene networks: dream or nightmare? *Annals of the New York Academy of Sciences*, 1158:246–56.

Barla, A., Filosi, M., Squillario, M., Masecchia, S., Riccadonna, S., Jurman, G., and Furlanello, C. (2011a). The impact of enrichment variability in pathway profiling. In *Workshop on Machine Learning in Computational Biology at NIPS*.

Barla, A., Jurman, G., Visintainer, R., Filosi, M., Riccadonna, S., and Furlanello, C. (2011b). A machine learning pipeline for discriminant pathways identification. In *CIBB 2011*, number 1, pages 1–10.

Barla, A., Masecchia, S., and Squillario, M. (2010). L1-L2 regularization framework for Alzheimer's molecular characterization. In *18th Annual International Conference on Intelligent Systems for Molecular Biology*.

Barla, A., Mosci, S., Rosasco, L., and Verri, A. (2008). A method for robust variable selection with significance assessment. In *European Symposium on Artificial Neural Networks*, volume 1.

Barla, A., Riccadonna, S., Masecchia, S., Squillario, M., Filosi, M., Jurman, G., and Furlanello, C. (2012). Evaluating sources of variability in pathway profiling.

Barrett, T., Troup, D. B., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Muertter, R. N., Holko, M., Ayanbule, O., Yefanov, A., and Soboleva, A. (2011). NCBI GEO: archive for functional genomics data sets–10 years on. *Nucleic acids research*, 39(Database issue):D1005–10.

Beerenwinkel, N., Rahnenführer, J., Däumer, M., Hoffmann, D., Kaiser, R., Selbig, J., and Lengauer, T. (2005). Learning multiple evolutionary pathways from cross-sectional data. *Journal of Computational Biology*, 12(6):584–598.

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: Statistical Methodology*, 57(1):289–300.

Bilke, S., Chen, Q.-R., Westerman, F., Schwab, M., Catchpoole, D., and Khan, J. (2005). Inferring a tumor progression model for neuroblastoma from genomic data. *Journal of clinical oncology : official journal of the American Society of Clinical Oncology*, 23(29):7322–31.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.-U. (2006). Complex networks: Structure and dynamics. *Physics reports*, 424(4):175–308.

Bolstad, B. M. (2004). *Low-level Analysis of High-density Oligonucleotide Array Data: Background, Normalization and Summarization*. PhD thesis, University of California, Berkeley.

Brodeur, G., Pritchard, J., Berthold, F., Carlsen, N., Castel, V., Castelberry, R., De Bernardi, B., Evans, A., Favrot, M., and Hedborg, F. (1993). Revisions of the international criteria for neuroblastoma diagnosis, staging, and response to treatment. *J. Clin. Oncol.*, 11(8):1466–1477.

Buchanan, M., Caldarelli, G., De Los Rios, P., Rao, F., and Vendruscolo, M. (2010). *Networks in Cell Biology*. Cambridge University Press.

Chan, T. F. and Wong, C. (2000). Convergence of the alternating minimization algorithm for blind deconvolution. *Linear Algebra and its Applications*, 316(1-3):259–285.

Chen, H.-I. H., Hsu, F.-H., Jiang, Y., Tsai, M.-H., Yang, P.-C., Meltzer, P. S., Chuang, E. Y., and Chen, Y. (2008). A probe-density-based analysis method for array CGH data: simulation, normalization and centralization. *Bioinformatics (Oxford, England)*, 24(16):1749–56.

Cleveland, W. S., Devlin, S. J., and Grosse, E. (1988). Regression by local fitting. *Journal of Econometrics*, 37(1):87–114.

Coco, S., Theissen, J., Scaruffi, P., Stigliani, S., Moretti, S., Oberthuer, A., Valdora, F., Fischer, M., Gallo, F., Hero, B., Bonassi, S., Berthold, F., and Tonini, G. P. (2012). Age-dependent accumulation of genomic aberrations and deregulation of cell cycle and telomerase genes in metastatic neuroblastoma. *International journal of cancer. Journal international du cancer*.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley-Interscience.

Curry, B., Ghosh, J., and Troup, C. (2008). Normalization of Array CGH Data. In Stafford, P., editor, *Methods in Microarray Normalization*, chapter 10, pages 233 – 244.

Daruwala, R.-S., Rudra, A., Ostrer, H., Lucito, R., Wigler, M., and Mishra, B. (2004). A versatile statistical analysis algorithm to detect genome copy number variation. *Proceedings of the National Academy of Sciences of the United States of America*, 101(46):16292–7.

Davies, J. J., Wilson, I. M., and Lam, W. L. (2005a). Array CGH technologies and their applications to cancer genomes. *Chromosome research: an international journal on the molecular, supramolecular and evolutionary aspects of chromosome biology*, 13(3):237–48.

Davies, R. J., Miller, R., and Coleman, N. (2005b). Colorectal cancer screening: prospects for molecular stool analysis. *Nature reviews. Cancer*, 5(3):199–209.

De Mol, C., De Vito, E., and Rosasco, L. (2009a). Elastic-Net Regularization in Learning Theory. *Journal of Complexity*, 25(2):201–230.

De Mol, C., Mosci, S., Traskine, M., and Verri, A. (2009b). A regularized method for selecting nested groups of relevant genes from microarray data. *Journal of computational biology: a journal of computational molecular cell biology*, 16(5):677–90.

de Ronde, J. J., Klijn, C., Velds, A., Holstege, H., Reinders, M. J. T., Jonkers, J., and Wessels, L. F. A. (2010). KC-SMARTR: An R package for detection of statistically significant aberrations in multi-experiment aCGH data. *BMC research notes*, 3:298.

De Smet, R. and Marchal, K. (2010). Advantages and limitations of current network inference methods. *Nature reviews. Microbiology*, 8(10):717–29.

Desper, R., Jiang, F., Kallioniemi, O.-P., Moch, H., Papadimitriou, C. H., and Schäffer, A. A. (1999). Inferring Tree Models for Oncogenesis from Comparative Genome Hybridization Data. *Journal of Computational Biology*, 6(1):37–51.

Desper, R., Jiang, F., Kallioniemi, O.-P., Moch, H., Papadimitriou, C. H., and Schäffer, A. A. (2000). Distance-based reconstruction of tree models for oncogenesis. *Journal of computational biology : a journal of computational molecular cell biology*, 7(6):789–803.

Di Camillo, B., Toffolo, G., and Cobelli, C. (2009). A gene network simulator to assess reverse engineering algorithms. *Annals of the New York Academy of Sciences*, 1158:125–42.

Diskin, S. J., Li, M., Hou, C., Yang, S., Glessner, J., Hakonarson, H., Bucan, M., Maris, J. M., and Wang, K. (2008). Adjustment of genomic waves in signal intensities from whole-genome SNP genotyping platforms. *Nucleic acids research*, 36(19):e126.

Dubitzky, W., Granzow, M., and Berrar, D. P. (2007). *Fundamentals of Data Mining in Genomics and Proteomics*. Springer.

Dupuy, A. and Simon, R. M. (2007). Critical review of published microarray studies for cancer outcome and guidelines on statistical analysis and reporting. *Journal of the National Cancer Institute*, 99(2):147–57.

Edgar, R., Domrachev, M., and Lash, A. E. (2002). Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic acids research*, 30(1):207–10.

Elad, M. and Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745.

Evgeniou, T., Poggio, T., Pontil, M., and Verri, A. (2002). Regularization and statistical learning theory for data analysis. *Computational Statistics & Data Analysis*, 38(4):421–432.

Faith, J. J., Hayete, B., Thaden, J. T., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J. J., and Gardner, T. S. (2007). Large-scale mapping and validation of

Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS biology*, 5(1):e8.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A Library for Large Linear Classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Fearon, E. R. and Vogelstein, B. (1990). A genetic model for colorectal tumorigenesis. *Cell*, 61(5):759–67.

Fitzgerald, T. W., Larcombe, L. D., Le Scouarnec, S., Clayton, S., Rajan, D., Carter, N. P., and Redon, R. (2011). aCGH.Spline - An R Package for aCGH Dye Bias Normalisation. *Bioinformatics (Oxford, England)*.

Fridlyand, J., Snijders, A. M., Pinkel, D., Albertson, D. G., and Jain, A. N. (2004). Hidden Markov models approach to the analysis of array CGH data. *Journal of Multivariate Analysis*, 90(1):132–153.

Furlanello, C., Serafini, M., Merler, S., and Jurman, G. (2003). Entropy-based gene ranking without selection bias for the predictive classification of microarray data. *BMC bioinformatics*, 4(1):54.

Gentleman, R., Carey, V. J., Huber, W., Irizarry, R. A., and Dudoit, S. (2005). *Bioinformatics and computational biology solutions using R and Bioconductor*. Springer.

Girosi, F., Jones, M., and Poggio, T. (1995). Regularization Theory and Neural Networks Architectures. *Neural Computation*, 7(2):219–269.

Golub, T. R., 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(5439):531–537.

Grimaldi, M., Visintainer, R., and Jurman, G. (2011). RegnANN: Reverse Engineering Gene Networks using Artificial Neural Networks. *PloS one*, 6(12):e28646.

Guo, X., Yanna, Ma, X., An, J., Shang, Y., Huang, Q., Yang, H., Chen, Z., and Xing, J. (2011). A meta-analysis of array-CGH studies implicates antiviral immunity pathways in the development of hepatocellular carcinoma. *PloS one*, 6(12):e28404.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, page 1157.

Hale, E. T., Yin, W., and Zhang, Y. (2008). Fixed-Point Continuation for l1-Minimization: Methodology and Convergence. *SIAM Journal on Optimization*, 19(3):1107–1130.

He, F., Balling, R., and Zeng, A.-P. (2009). Reverse engineering and verification of gene networks: principles, assumptions, and limitations of present methods and future perspectives. *Journal of biotechnology*, 144(3):190–203.

Hilario, M. and Kalousis, A. (2008). Approaches to dimensionality reduction in proteomic biomarker studies. *Briefings in bioinformatics*, 9(2):102–18.

Horvath, S. (2011). *Weighted Network Analysis: Applications in Genomics and Systems Biology*. Springer.

Hsu, L., Self, S. G., Grove, D., Randolph, T., Wang, K., Delrow, J. J., Loo, L., and Porter, P. (2005). Denoising array-based comparative genomic hybridization data using wavelets. *Biostatistics (Oxford, England)*, 6(2):211–26.

Huang, D. W., Sherman, B. T., and Lempicki, R. A. (2009). Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic acids research*, 37(1):1–13.

Hupé, P., Stransky, N., Thiery, J.-P., Radvanyi, F., and Barillot, E. (2004). Analysis of array CGH data: from signal ratio to gain and loss of DNA regions. *Bioinformatics (Oxford, England)*, 20(18):3413–22.

Ioannidis, J. P. A. (2005). Why most published research findings are false. *PLoS medicine*, 2(8):e124.

Ioannidis, J. P. A., Allison, D. B., Ball, C. A., Coulibaly, I., Cui, X., Culhane, A. C., Falchi, M., Furlanello, C., Game, L., Jurman, G., Mangion, J., Mehta, T., Nitzberg, M., Page, G. P., Petretto, E., and van Noort, V. (2009). Repeatability of published microarray gene expression analyses. *Nature genetics*, 41(2):149–55.

Ipsen, M. and Mikhailov, A. S. (2002). Evolutionary reconstruction of networks. *Phys Rev E*, 66(4).

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., and Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics (Oxford, England)*, 4(2):249–64.

Jong, K., Marchiori, E., van der Vaart, A., Chin, S.-F., Carvalho, B., Tijssen, M., Eijk, P. P., van den Ijssel, P., Grabsch, H., Quirke, P., Oudejans, J. J., Meijer, G. A., Caldas, C., and Ylstra, B. (2007). Cross-platform array comparative genomic hybridization meta-analysis separates hematopoietic and mesenchymal from epithelial tumors. *Oncogene*, 26(10):1499–506.

Jurman, G., Visintainer, R., and Furlanello, C. (2011). An introduction to spectral distances in networks. *Frontiers Artificial Intelligence Appl*, 226:227—-234.

Jurman, G., Visintainer, R., Riccadonna, S., Filosi, M., and Furlanello, C. (2012). A glocal distance for network comparison.

Kallioniemi, A., Kallioniemi, O.-P., Sudar, D., Rutovitz, D., Gray, J. W., Waldman, F., and Pinkel, D. (1992). Comparative genomic hybridization for molecular cytogenetic analysis of solid tumors. *Science (New York, N.Y.)*, 258(5083):818–21.

Kanehisa, M. and Goto, S. (2000). KEGG: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30.

Khojasteh, M., Lam, W. L., Ward, R. K., and MacAulay, C. (2005). A stepwise framework for the normalization of array CGH data. *BMC bioinformatics*, 6:274.

Kowalski, M. and Torrésani, B. (2009). Sparsity and persistence: mixed norms provide simple signal models with dependent coefficients. *Signal, Image and Video Processing*, 3(3):251–264.

Krona, C., Carén, H., Sjöberg, R.-M., Sandstedt, B., Laureys, G., Kogner, P., and Martinsson, T. (2008). Analysis of neuroblastoma tumour progression; loss of PHOX2B on 4p13 and 17q gain are early events in neuroblastoma tumourigenesis. *International Journal of Oncology*, 32(3):575.

Lai, W., Choudhary, V., and Park, P. J. (2008). CGHweb: a tool for comparing DNA copy number segmentations from multiple algorithms. *Bioinformatics (Oxford, England)*, 24(7):1014–5.

Lai, W. R., Johnson, M. D., Kucherlapati, R., and Park, P. J. (2005). Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data. *Bioinformatics (Oxford, England)*, 21(19):3763–70.

Leek, J. T., Scharpf, R. B., Bravo, H. C., Simcha, D., Langmead, B., Johnson, W. E., Geman, D., Baggerly, K., and Irizarry, R. A. (2010). Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature reviews. Genetics*, 11(10):733–9.

Liang, W. S., Dunckley, T., Beach, T. G., Grover, A., Mastroeni, D., Walker, D. G., Caselli, R. J., Kukull, W. A., McKeel, D., Morris, J. C., Hulette, C., Schmechel, D., Alexander, G. E., Reiman, E. M., Rogers, J., and Stephan, D. A. (2007). Gene expression profiles in anatomically and functionally distinct regions of the normal aged human brain. *Physiological genomics*, 28(3):311–22.

Liang, W. S., Reiman, E. M., Valla, J., Dunckley, T., Beach, T. G., Grover, A., Niedzielko, T. L., Schneider, L. E., Mastroeni, D., Caselli, R., Kukull, W., Morris, J. C., Hulette, C. M., Schmechel, D., Rogers, J., and Stephan, D. A. (2008). Alzheimer's disease is associated with reduced expression of energy metabolism genes in posterior cingulate

neurons. *Proceedings of the National Academy of Sciences of the United States of America*, 105(11):4441–6.

Liu, J., Bandyopadhyay, N., Ranka, S., Baudis, M., and Kahveci, T. (2009). Inferring progression models for CGH data. *Bioinformatics (Oxford, England)*, 25(17):2208–15.

Ma, S. and Huang, J. (2008). Penalized feature selection and classification in bioinformatics. *Briefings in Bioinformatics*, 9(5):392–403.

MAQC Consortium (2010). The MicroArray Quality Control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models. *Nature biotechnology*, 28(8):827–38.

Margolin, A. A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Dalla Favera, R., and Califano, A. (2006). ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC bioinformatics*, 7 Suppl 1:S7.

Marioni, J. C., Thorne, N. P., Valsesia, A., Fitzgerald, T. W., Redon, R., Fiegler, H., Andrews, T. D., Stranger, B. E., Lynch, A. G., Dermitzakis, E. T., Carter, N. P., Tavaré, S., and Hurles, M. E. (2007). Breaking the waves: improved detection of copy number variation from microarray-based comparative genomic hybridization. *Genome biology*, 8(10):R228.

Masecchia, S., Barla, A., Salzo, S., and Verri, A. (2013a). Dictionary Learning improves subtyping of breast cancer aCGH data. In *IEEE Engineering in Medicine and Biology Society*.

Masecchia, S., Barla, A., Squillario, M., Coco, S., Verri, A., and Tonini, G. P. (2012a). Inferring oncogenetic tree-models from aCGH of metastatic neuroblastoma. In *European Conference on Computational Biology*.

Masecchia, S., Barla, A., Squillario, M., Coco, S., Verri, A., and Tonini, G. P. (2012b). Inferring oncogenic tree-models from aCGH. In *Intelligent Data Analysis in bioMedicine And Pharmacology*.

Masecchia, S., Salzo, S., Barla, A., and Verri, A. (2013b). A dictionary learning based method for aCGH segmentation. In *European Symposium on Artificial Neural Networks*.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et biophysica acta*, 405(2):442–51.

Mei, T. S., Salim, A., Calza, S., Seng, K. C., Seng, C. K., and Pawitan, Y. (2010). Identification of recurrent regions of Copy-Number Variants across multiple individuals. *BMC bioinformatics*, 11:147.

Miecznikowski, J. C., Gaile, D. P., Liu, S., Shepherd, L., and Nowak, N. (2011). A new normalizing algorithm for BAC CGH arrays with quality control metrics. *Journal of biomedicine & biotechnology*, 2011:860732.

Mosci, S., Rosasco, L., Santoro, M., Verri, A., and Villa, S. (2010). Solving Structured Sparsity Regularization with Proximal Methods. In Balcázar, J. L., Bonchi, F., Gionis, A., and Sebag, M., editors, *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010*, volume 6322 of *Lecture Notes in Computer Science*, pages 418–433–433, Berlin, Heidelberg. Springer Berlin Heidelberg.

Navin, N., Krasnitz, A., Rodgers, L., Cook, K., Meth, J., Kendall, J., Riggs, M., Eberling, Y., Troge, J., Grubor, V., Levy, D., Lundin, P., Må nér, S., Zetterberg, A., Hicks, J., and Wigler, M. (2010). Inferring tumor progression from genomic heterogeneity. *Genome research*, 20(1):68–80.

Nemenman, I., Escola, G. S., Hlavacek, W. S., Unkefer, P. J., Unkefer, C. J., and Wall, M. E. (2007). Reconstruction of metabolic networks from high-throughput metabolite profiling data: in silico analysis of red blood cell metabolism. *Annals of the New York Academy of Sciences*, 1115:102–15.

Neuvial, P., Hupé, P., Brito, I., Liva, S., Manié, E., Brennetot, C., Radvanyi, F., Aurias, A., and Barillot, E. (2006). Spatial normalization of array-CGH data. *BMC bioinformatics*, 7(1):264.

Newman, M. (2010). *Networks: An Introduction*. Oxford University Press, USA.

Newman, M. E. (2003). The structure and function of complex networks. *SIAM review*, 45(2):167–256.

Nowak, G., Hastie, T., Pollack, J. R., and Tibshirani, R. (2011). A fused lasso latent feature model for analyzing multi-sample aCGH data. *Biostatistics (Oxford, England)*.

Olshen, A. B., Venkatraman, E. S., Lucito, R., and Wigler, M. (2004). Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics (Oxford, England)*, 5(4):557–72.

Picard, F., Lebarbier, E., Hoebeke, M., Rigaill, G., Thiam, B., and Robin, S. (2011). Joint segmentation, calling, and normalization of multiple CGH profiles. *Biostatistics Oxford England*, pages 1–16.

Pique-Regi, R., Monso-Varona, J., Ortega, A., Seeger, R. C., Triche, T. J., and Asgharzadeh, S. (2008). Sparse representation and Bayesian detection of genome copy number alterations from microarray data. *Bioinformatics (Oxford, England)*, 24(3):309–18.

Pique-Regi, R., Ortega, A., and Asgharzadeh, S. (2009). Joint estimation of copy number variation and reference intensities on multiple DNA arrays using GADA. *Bioinformatics (Oxford, England)*, 25(10):1223–30.

Pollack, J. R., Sø rlie, T., Perou, C. M., Rees, C. A., Jeffrey, S. S., Lonning, P. E., Tibshirani, R., Botstein, D., Bø rresen Dale, A.-L., and Brown, P. O. (2002). Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors. *Proceedings of the National Academy of Sciences of the United States of America*, 99(20):12963–8.

Potkin, S. G., Guffanti, G., Lakatos, A., Turner, J. A., Kruggel, F., Fallon, J. H., Saykin, A. J., Orro, A., Lupoli, S., Salvi, E., Weiner, M., and Macciardi, F. (2009). Hippocampal atrophy as a quantitative trait in a genome-wide association study identifying novel susceptibility genes for Alzheimer's disease. *PloS one*, 4(8):e6501.

Price, T. S., Regan, R., Mott, R., Hedman, A., Honey, B., Daniels, R. J., Smith, L., Greenfield, A., Tiganescu, A., Buckle, V., Ventress, N., Ayyub, H., Salhan, A., Pedraza-Diaz, S., Broxholme, J., Ragoussis, J., Higgs, D. R., Flint, J., and Knight, S. J. L. (2005). SW-ARRAY: a dynamic programming solution for the identification of copy-number changes in genomic DNA using array comparative genome hybridization data. *Nucleic acids research*, 33(11):3455–64.

Radmacher, M. D., Simon, R., Desper, R., Taetle, R., Schäffer, A. A., and Nelson, M. A. (2001). Graph models of oncogenesis with an application to melanoma. *Journal of theoretical biology*, 212(4):535–48.

Rosenzweig, B. A., Pine, P. S., Domon, O. E., Morris, S. M., Chen, J. J., and Sistare, F. D. (2004). Dye bias correction in dual-labeled cDNA microarray gene expression measurements. *Environmental health perspectives*, 112(4):480–7.

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

Scherzer, C. R., Eklund, A. C., Morse, L. J., Liao, Z., Locascio, J. J., Fefer, D., Schwarzschild, M. A., Schlossmacher, M. G., Hauser, M. A., Vance, J. M., Sudarsky, L. R., Standaert, D. G., Growdon, J. H., Jensen, R. V., and Gullans, S. R. (2007). Molecular markers of early Parkinson's disease based on gene expression in blood. *Proceedings of the National Academy of Sciences of the United States of America*, 104(3):955–60.

Schleiermacher, G., Michon, J., Huon, I., D'Enghien, C. D., Klijanienko, J., Brisse, H., Ribeiro, A., Mosseri, V., Rubie, H., Munzer, C., Thomas, C., Valteau-Couanet, D., Auvrignon, A., Plantaz, D., Delattre, O., and Couturier, J. (2007). Chromosomal CGH identifies patients with a higher risk of relapse in neuroblastoma without MYCN amplification. *British journal of cancer*, 97(2):238–46.

Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464.

Sharan, R. and Ideker, T. (2006). Modeling cellular machinery through biological network comparison. *Nature biotechnology*, 24(4):427–33.

Sibson, R. (1973). SLINK: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34.

Sokal, R. R. and Rohlf, F. J. (1962). The Comparison of Dendrograms by Objective Methods. *Taxon*, 11(2):33.

Squillario, M., Masecchia, S., and Barla, A. (2010). Functional characterization of Parkinson by high-throughput data analysis with l1l2 regularization. In *9th European Conference on Computational Biology ECCB10*, number 1.

Squillario, M., Masecchia, S., Zycinski, G., and Barla, A. (2011). Uncovering Candidate Biomarkers for Alzheimer's and Parkison's Diseases with Regularization Methods and Prior Knowledge. In *10th International Conference on Alzheimer's and Parkinson's Disease*.

Squillario, M., Zycinsky, G., Masecchia, S., Verri, A., and Barla, A. (2012). Analysis of a Parkinson dataset: comparison between KDVS and the standard pipeline. In *European Conference on Computational Biology*, volume 3.

Staaf, J., Jönsson, G., Ringnér, M., and Vallon-Christersson, J. (2007). Normalization of array-CGH data: influence of copy number imbalances. *BMC genomics*, 8(1):382.

Strogatz, S. H. (2001). Exploring complex networks. *Nature*, 410(6825):268–76.

Subramanian, A., Shackney, S., and Schwartz, R. (2012). Inference of tumor phylogenies from genomic assays on heterogeneous samples. *Journal of biomedicine & biotechnology*, 2012:797812.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–50.

Tian, Z. and Kuang, R. (2010). Integrative classification and analysis of multiple array-CGH datasets with probe alignment. *Bioinformatics (Oxford, England)*, 26(18):2313–20.

Tian, Z., Zhang, H., and Kuang, R. (2012). Sparse Group Selection on Fused Lasso Components for Identifying Group-specific DNA Copy Number Variations. In *Proc. of IEEE International Conference on Data Mining (ICDM)*.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society - Series B: Statistical Methodology*, 67(1):91–108.

Tibshirani, R. and Wang, P. (2008). Spatial smoothing and hot spot detection for CGH data using the fused lasso. *Biostatistics (Oxford, England)*, 9(1):18–29.

van Houte, B. P. P., Binsl, T. W., Hettling, H., and Heringa, J. (2010). CGHnormaliter: a Bioconductor package for normalization of array CGH data with many CNAs. *Bioinformatics (Oxford, England)*.

van Houte, B. P. P., Binsl, T. W., Hettling, H., Pirovano, W., and Heringa, J. (2009). CGHnormaliter: an iterative strategy to enhance normalization of array CGH data with imbalanced aberrations. *BMC genomics*, 10:401.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley-Interscience.

Vert, J.-P. and Bleakley, K. (2010). Fast detection of multiple change-points shared by many signals using group LARS. *Advances in Neural Information Processing Systems 23*, 1:1–9.

Vicente Miranda, H. and Outeiro, T. F. (2010). The sour side of neurodegenerative disorders: the effects of protein glycation. *The Journal of pathology*, 221(1):13–25.

Villa, S., Salzo, S., Baldassarre, L., and Verri, A. (2012). Accelerated and inexact forward-backward algorithms. *Optimization Online*, pages 1–29.

Vogelstein, B., Fearon, E. R., Hamilton, S. R., Kern, S. E., Preisinger, A. C., Leppert, M., Nakamura, Y., White, R., Smits, A. M., and Bos, J. L. (1988). Genetic alterations during colorectal-tumor development. *The New England journal of medicine*, 319(9):525–32.

von Bernhardi, R., Tichauer, J. E., and Eugenín, J. (2010). Aging-dependent changes of microglial cells and their relevance for neurodegenerative disorders. *Journal of neurochemistry*, 112(5):1099–114.

Wang, H. J. and Hu, J. (2010). Identification of Differential Aberrations in Multiple-Sample Array CGH Studies. *Biometrics*, 67(2):353–62.

Wang, P., Kim, Y., Pollack, J. R., Narasimhan, B., and Tibshirani, R. (2005). A method for calling gains and losses in array CGH data. *Biostatistics (Oxford, England)*, 6(1):45–58.

Wang, Y., Wang, S., and Zinn, A. R. (2007). rSWTi: A Robust Stationary Wavelet Denoising Method for Array CGH Data. In *2007 IEEE 7th International Symposium on BioInformatics and BioEngineering*, pages 1066–1070. IEEE.

Willenbrock, H. and Fridlyand, J. (2005). A comparison study: applying segmentation to array CGH data for downstream analyses. *Bioinformatics (Oxford, England)*, 21(22):4084–91.

Ylstra, B., van den Ijssel, P., Carvalho, B., Brakenhoff, R. H., and Meijer, G. A. (2006). BAC to the future! or oligonucleotides: a perspective for micro array comparative genomic hybridization (array CGH). *Nucleic acids research*, 34(2):445–50.

Yu, W., Wulf, A., Liu, T., Khoury, M. J., and Gwinn, M. (2008). Gene Prospector: an evidence gateway for evaluating potential susceptibility genes and interacting risk factors for human diseases. *BMC bioinformatics*, 9:528.

Zhang, B., Kirov, S., and Snoddy, J. (2005). WebGestalt: an integrated system for exploring gene sets in various biological contexts. *Nucleic acids research*, 33(Web Server issue):W741–8.

Zhang, Q., Ding, L., Larson, D. E., Koboldt, D. C., McLellan, M. D., Chen, K., Shi, X., Kraja, A., Mardis, E. R., Wilson, R. K., Borecki, I. B., and Province, M. A. (2010). CMDS: a population-based method for identifying recurrent DNA copy number aberrations in cancer from high-resolution data. *Bioinformatics*, 26(4):464–469.

Zycinski, G. (2012). *Applying Data Integration into Reconstruction of Gene Networks from Microarray Data*. PhD thesis, University of Genoa, DIBRIS.

Zycinski, G., Barla, A., Squillario, M., Sanavia, T., Di Camillo, B., and Verri, A. (2013). Knowledge Driven Variable Selection (KDVS) - a new approach to enrichment analysis of gene signatures obtained from high-throughput data. *Source code for biology and medicine*, 8(1):2.

Zycinski, G., Barla, A., and Verri, A. (2011). SVS: data and knowledge integration in computational biology. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2011:6474–8.

Zălinescu, C. (2002). *Convex Analysis in General Vector Spaces*. World Scientific.