UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sede Amministrativa: Università degli Studi di Padova

Dipartimento di Scienze Statistiche
SCUOLA DI DOTTORATO DI RICERCA IN SCIENZE STATISTICHE
CICLO XXVII

THREE TOPICS IN OMICS RESEARCH

Direttore della Scuola: Prof. Monica Chiogna

Supervisore: Prof. Alessandra R. Brazzale

Co-supervisore: Prof. Sorin Drăghici, Prof. Silvio Bicciato

Dottorando: PAOLA TELLAROLI

January 31, 2015

"*If you have an apple and I have an apple and we exchange apples then you and I will still each have one apple. But if you have an idea and I have an idea and we exchange these ideas, then each of us will have two ideas.*"

George Bernard Shaw (1856 – 1950), Irish writer

Riassunto

Il titolo piuttosto generico di questa tesi è dovuto al fatto che sono stati indagati diversi aspetti di fenomeni biologici. La maggior parte di questo lavoro è stato rivolto alla ricerca dei limiti di uno degli strumenti essenziali per l'analisi di dati di espressione genica: l'analisi dei gruppi. Esistendo diverse centinaia di metodi di raggruppamento, chiaramente non c'è carenza di algoritmi di analisi dei gruppi, ma, allo stesso tempo, alcuni quesiti fondamentali non hanno ancora ricevuto risposte soddisfacenti. In particolare, presentiamo un nuovo algoritmo di analisi dei gruppi per dati statici ed una nuova strategia per il raggruppamento di dati temporali di breve lunghezza. Infine, abbiamo analizzato dati provenienti da una tecnologia relativamente nuova, chiamata Cap Analysis Gene Expression, utile per l'analisi dei promotori su tutto il genoma e ancora in gran parte inesplorata.

Abstract

The rather generic title of this Thesis is due to the fact that several aspects of biological phenomena have been investigated. Most of this work was addressed at the investigation of the limitations of one of the essential tools for analyzing gene expression data: cluster analysis. With several hundred of clustering methods in existence, there is clearly no shortage of clustering algorithms but, at the same time, satisfactory answers to some basic questions are still to come. In particular, we present a novel algorithm for the clustering of static data and a new strategy for the clustering of short-length time-course data. Finally, we analyzed data coming from Cap Analysis Gene Expression, a relatively new technology useful for the genome-wide promoter analysis and still mostly unexplored.

Acknowledgements

I would like to thank my supervisor, Alessandra R. Brazzale, for her support and advice, and for giving me the opportunity to work in the field I was interested in. I furthermore acknowledge the financial support by Fondazione CARIPARO (grant n. 112419/11).

When I arrived in Detroit I couldn't imagine to find such a great person and mentor as Sorin Drăghici: I was very lucky to meet him and having the chance to work with him for one year. I am very grateful for him letting me understand how a great team and great professor work.

I am infinitely grateful to Michele Donato for all the help and psychological support, but above all for giving me the passion for research: without him I wouldn't have finished my PhD.

A vote of thanks goes also to Silvio Bicciato, Guidantonio Malagoli Tagliazucchi, Nathalie Chèvre, Myriam Borgatta, and Patrice Waridel for sharing with me this data and for all the motivating discussions.

Thanks to the XXVII cycle of the PhD School in Statistics for having shared a lot of funny and stressful moments.

A special mention goes to the ISBL group members for letting me be part of their big family and letting me learn so many amazing things. I still feel the thrill of the instant meetings and I will never forget that every Monday could be a bloody Monday!

Last but not least, thanks to all my friends, for always being there for me. A special thank to Emanuele.

Thank you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The rather generic title of this Thesis is due to the fact that several aspects of biological phenomena have been investigated. In fact, the term "omics" is a neologism that refers to several fields of study in biology ending in *-omics*, such as genomics and proteomics. *Genomics* is the discipline that analyzes the function and structure of the genome, i.e. the entire set of genes in an organism. Similarly, *proteomics* is the study of the proteome, i.e. the set of all proteins in a given organism. Following this rule, while the suffix "-ome" represents the set of all entities of a particular type, the suffix "-omics" indicates the field of research which studies them.

Generally speaking, gene expression studies aim to quantify the gene products in order to explain the molecular processes underlying life phenomena. Nowadays, high-throughput technologies allow the collection of biological information in extremely large quantities, leading to the challenge of analyzing, interpreting and understanding all data that are being produced.

The first line of research we investigated is one of the essential tools for analyzing gene expression data: cluster analysis. It has been shown that genes and proteins of similar function cluster together (Eisen et al., 1998; Spellman et al., 1998; Barenco et al., 2006; Tomancak et al., 2002; Straume, 2004; Heyer et al., 1999), and clustering methods have been used to solve many problems of biological nature. One of the most interesting of these problems is related to *disease subtyping*, i.e. the stratification of patients in terms of underlying disease characteristics in order to direct the patient to more personalized therapies. With several hundreds of clustering methods in existence (Milligan & Cooper, 1987), there is clearly no shortage of clustering algorithms but, at the same time, satisfactory answers to some basic questions are still to come. Four of the

most common limitations of the many available clustering methods are i) the lack of a proper strategy to deal with outliers, ii) the need of a good estimate of the number of clusters to obtain reasonable results, iii) the lack of a method able to suggest that partitioning that specific data is not appropriate, and iv) the dependence of most of the available methods on the initialization.

The second line of research we investigated was the clustering of short-length time-course data. Thanks to advanced technologies we are now able to observe the evolution of the gene expression over time. Clustering of time-course gene expression data is a field even more challenging than the clustering of static data, because of the time-dependence of observations and the small number of time points usually available. The direct application of standard clustering methods to time series experiments is difficult because they typically assume that the observation for each gene across time is independent. This assumption holds when expression measures are taken from independent biological samples, such as different subjects or different experimental conditions. However, it is not valid when observations are directly related to previous time points. Standard distance measures currently used for clustering, such as Euclidean distance or one minus Pearson correlation, are invariant with respect to the order of observation. That is, even if the temporal order of a pair of series is permuted, their distance will not change. Consequently, methods which can preserve the time sequence and time dependence of observed data are needed for cluster analysis of time-course data. Furthermore, short time series are common because multiple arrays are expensive and it is often prohibitive to obtain large quantities of biological material. Although little work has been done in cluster analysis of time-course data compared with those focusing on static data, there seems to be a trend of increased activity (Warren Liao, 2005).

The last important topic in Omics investigated in this Thesis is the analysis of data coming from Cap Analysis Gene Expression (CAGE) technology (Shiraki et al., 2003; Carninci et al., 2006), a relatively new method useful for the genome-wide promoter analysis. Whereas for microarray technology a wide literature of statistical methods is available, the literature addressed at CAGE data analysis is still poor, making it a very interesting field to explore.

After an overview of the Thesis in Chapter 1, Chapter 2 gives a brief description of the biological and technological background useful for understanding the problems and the terms used in the rest of the Thesis. Chapter 3 is addressed at the description of the four datasets from real experiments and a simulated dataset. In Chapter 4 we introduce and validate a novel algorithm for the clustering of static data that overcomes the limitations of common clustering methods. Chapter 5 faces the problem of clustering short-length time-course data and proposes

a possible solution. In Chapter 6 a CAGE data analysis is presented. Finally, we conclude in Chapter 7 with a general discussion and some possibilities for future work.

1.2 Contribution of this thesis

The contribution of this Thesis may be summarized as follows:

- Definition and validation of a novel algorithm for the clustering of static data that overcomes the main limitations of common clustering methods.
- Implementation of the R package `CrossClustering` for the clustering of highly noisy data.
- Definition of a novel method for clustering short-length time-course data.
- Analysis of CAGE data using current methods of analysis.

Chapter 2

Biological background

This chapter is dedicated to the introduction of some basic biological notions. In Section 2.1 we will introduce the important concepts of gene expression, differential expression between conditions, and promoter analysis, while in Section 2.2 microarray technologies are briefly presented.

2.1 Differential expression

The genetic program of all living beings is encoded in the deoxyribonucleic acid (DNA), which is contained in each cell of the organism and has the form of a double helix that looks like a twisted long ladder. The sides of the "ladder" are the strands and consist of two nucleotides joined in the middle by hydrogen bonds. These nucleotides are complementary bases: every Adenine (base A) has a Thymine (base T) partner and every Guanine (base G) has a Cytosine partner (base C) on the other side. Two bases form a base pair (bp) and it is the unit for measuring the length of the DNA. In the order of these bases there are the exact instructions required to create a particular organism with its unique traits. Two strands are called complementary if for any base on one strand, the other strand contains the complement of this base. Two complementary single-stranded DNA chains that come into close proximity react to form a stable double helix in a process known as *hybridization*. Conversely, a double-stranded DNA can be split into two complementary, single-stranded chains in a process called *denaturation*.

The DNA in the human genome is arranged into 23 distinct chromosomes. Each chromosome contains many genes, which encode instructions useful for the cell in order to know how much proteins should be produced, when and how.

Most of the cells of an organism contain the same DNA but, obviously, an eye cell is dif-

ferent from a lung cell, for example. This is possible because not all genes are expressed in the same way in all cells. The cell differentiation is given by changes in gene expression, which control the production of proteins. For this reason, studying the different levels of expression of a gene in different tissues can explain why the tissues are different. In the same way, studying the differential expression (DE, i.e., the relative expression of each gene between conditions of interest) of genes in the same tissue coming from different conditions can help us understand the differences between these conditions (e.g., cancer vs. normal tissue). The Central Dogma of Molecular Biology (Crick et al., 1970) essentially states that the flow of genetic information goes from DNA to RNA (ribonucleic acid) to proteins (Alberts et al., 2013). The mechanism by which the information coded into the DNA sequence is converted into an RNA sequence is called *transcription* and it begins at DNA sites called *promoters*. One of the different functional types of RNAs products is the messenger RNA (mRNA), which carries the instructions for making proteins in the process called *translation* (Draghici, 2012). Anyway, as in many aspects of life science, there are special cases in which the Central Dogma is not applied. For instance, the *reverse transcription*, which is the process by which genetic information is transcribed from RNA into new DNA, and that will be discussed again later (Section 6.1).

Proteins are long chains of amino acid molecules that have a crucial role in all life processes. A gene is active, or expressed, if the cell makes the gene product (e.g., protein) encoded by the gene. If a lot of gene product is produced, the gene is said to be highly expressed. If no gene product is produced, the gene is not expressed.

The cell control the amount of gene products that is produced at any stage of the process that leads from a gene to a functional protein. During transcription, for example, this regulation can happen through transcription factors (proteins that bind to the DNA) as repressors or activators, or through regulatory elements (sites on the DNA), such as promoters and enhancers.

2.2 Microarrays

Microarrays are tools for gene expression analysis that allow the analysis of thousands of genes at the same time. A microarray is a solid surface (such as glass, plastic or nylon) on which single-stranded DNAs are deposited in correspondence to microscopic DNA probes, forming an array. The array is then washed with a solution containing a *target* DNA containing sequences complementary to those of the deposited DNA, allowing the hybridization process. In order to interpret the experiment it is essential to understand which DNA material is used to hybridize on

the array, and this can be easily done labeling with a fluorescent dye the target DNA in a reverse transcription procedure. A subsequent illumination with an appropriate source of light provides an image of the array, where the intensity of each spot is related to the amount of mRNA present in the tissue and, as a consequence, to the amount of protein produced by the specific gene. After an image-processing step, the intensities are translated in a large number of expression values ready to be analyzed.

Microarrays have been used successfully in a wide range of applications and, after a large number of papers reporting validated results obtained using this technology, are recognized as accurate, precise and reliable tools for the gene expression data analysis.

Chapter 3

Data

In this chapter we will introduce the case studies and the simulation scenario used in this Thesis. For each dataset, we will briefly describe the underlying experimental design or how we simulated it. The order in which datasets are presented follows the order in which they will appear throughout this work.

3.1 Fake data

Our simulation was set up as follows: we generated 100 fake data sets, each representing the expression profiles of 2,000 genes for 5 samples. The expression profile of each gene was chosen among five distinct *behaviors*: constant at a positive value (500 genes), increasing (250 genes), decreasing (700 genes), oscillating (300 genes), and convex (100 genes). We added random noise from $\mathcal{N}(\mu = 0, \sigma = 0.2)$, $\mathcal{N}(\mu = 0, \sigma = 0.5)$, $\mathcal{N}(\mu = 0, \sigma = 1)$, $\mathcal{N}(\mu = 0, \sigma = 1.5)$ to each of the 100 data sets, resulting in four groups of 100 data sets. An additional set of 150 gene profiles were drawn from a Uniform distribution between 0 and the maximum simulated value present in the dataset, and they were added to each of the 400 data sets. The graphical representation of these behaviors is shown in Fig. 3.1.

We performed a principal component analysis (PCA) on the data, and plotted the first three principal components (cumulative percentage of variance explained 89.8%). In Fig. 3.2 each gene has been plotted as a point in the three-dimensional space corresponding to the first three principal components, denoted with PC1, PC2, and PC3. The figure shows how categories are well defined even when $\sigma = 1.5$, while outlier genes are distributed uniformly around the center.

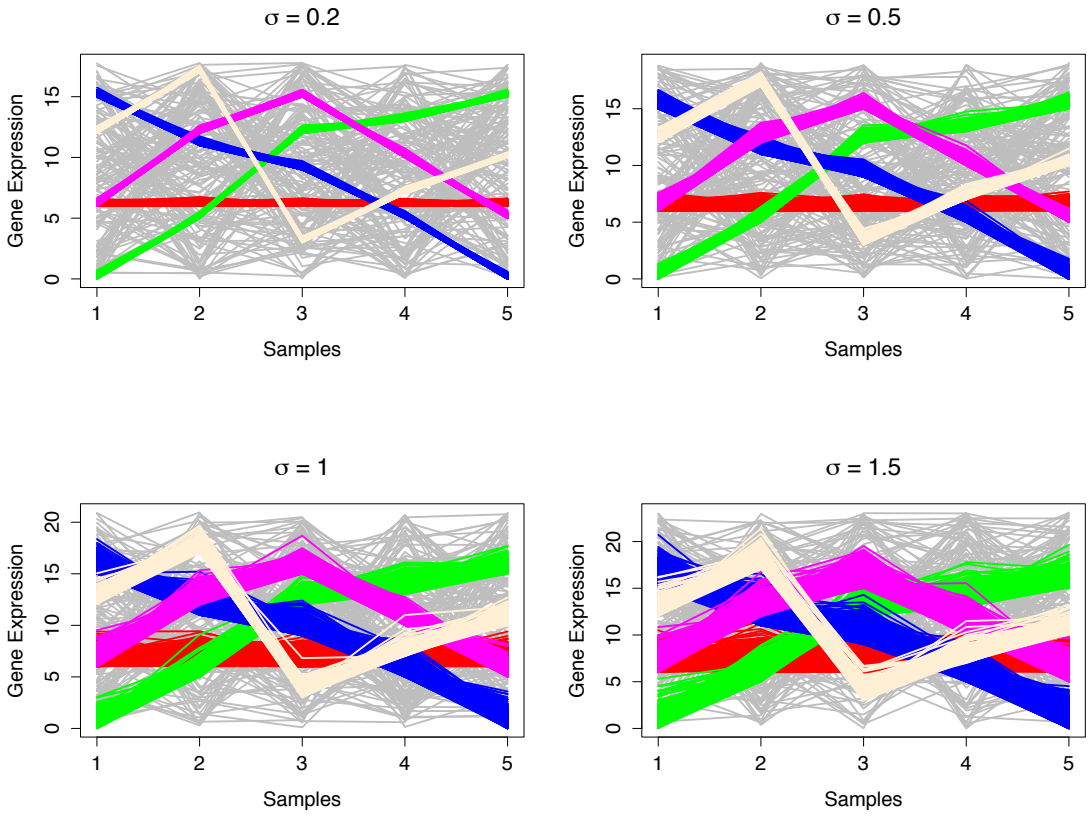

Figure 3.1: Graphical representation of the five behaviors (constant, increasing, decreasing, oscillating, and convex) when $\sigma = 0.2, 0.5, 1, 1.5$ over five samples in one of the 100 simulated datasets. Each color represents a different group of profiles. Gray lines in the background represent 150 outlier profiles, drawn from a Uniform distribution between 0 and the maximum simulated value present in the dataset.

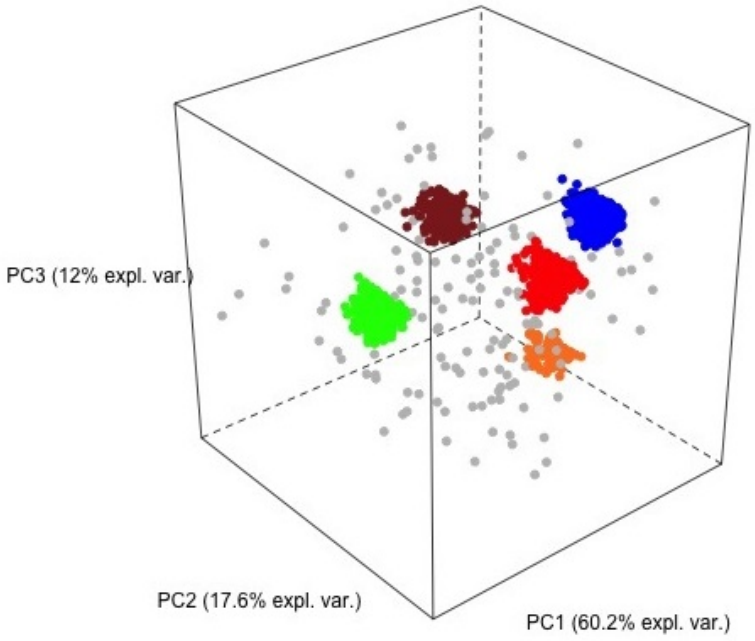

Figure 3.2: A Principal Component Analysis plot of one simulation of the data with the highest variance ($\sigma = 1.5$) on the first three principal components. Each group of points corresponds to one of the five prototypes used to define the general behavior of the genes. The gray points represent the 150 completely random genes added to the set. The cumulative percentage of variance explained is 89.8%. Categories are well defined even when the variability added to the data is high.

3.2 Brain tumors

Central nervous system embryonal tumors (CNSET) are a group of tumors characterized by high heterogeneity. The understanding of the biological mechanisms underlying CNSET is still limited (Pomeroy et al., 2002). Although the classification of these tumors based on histopathological appearance is still debated, they are usually divided in: medulloblastoma (MD), CNS primitive neuroectodermal tumors (PNET), rhabdoid tumors (Rhab), and malignant glioma (Mglio).

The public dataset used by Pomeroy et al. (2002) (available for free download at `http://www.broadinstitute.org/MPR/CNS/`) contains $5,299$ gene expression profiles of 42 samples: 10 MD, 10 Rhab, 8 PNET, 10 Mglio, and 4 normal human cerebella (Ncer). The dataset has been $\log_2$ transformed and scaled to mean zero and variance one.

3.3 Breast cancer

More than 1.7 million new cases of breast cancer occurred among women worldwide in 2012 (International Agency for Research on Cancer and others, 2014), making breast cancer the most common cancer in women worldwide. The incidence of breast cancer in women in 2011 (most recent data available) was of 124.3 per $100,000$, while the mortality was of 21.5 per $100,000$ (Howlader et al., 2013). Increasing evidence suggests that breast cancer can be classified in multiple subtypes based on the kind of treatment, level of aggressivity, risk factors, and survival rates. Depending on the number of biological markers (proteins associated with mechanisms underlying the disease), most studies divide breast cancer into four major molecular subtypes: luminal A, luminal B, triple negative/basal-like (TN), and HER2 over-expression (approximate prevalences of, respectively, 40%, 20%, 20%, and 15%). The remaining cases are less common and often listed as unclassified. The public dataset GSE38888 from the Gene Expression Omnibus database (Barrett et al., 2005) describes the expression profiles of $719,690$ probesets and 30 samples, classified in two subtypes: 16 luminal and 14 TN (Vollebergh et al., 2014).

3.4 Daphnia pulex

Daphnia pulex is a freshwater crustacean commonly used for environmental monitoring of pollutants around the world. It is the established model species for toxicological studies (Shaw et al., 2008) because daphnids serve as an important source of food for fish and other aquatic organisms, they are strongly sensitive to changes in water chemistry, are simple and inexpensive

to raise in an aquarium, and are able to inhabit very different environments throughout the world.

Borgatta et al. (2014) exposed *Daphnia pulex* to the most commonly used drug against breast cancer: Tamoxifen (Goetz et al., 2007; Kisanga et al., 2005; Zheng et al., 2007). Tamoxifen is the most important drug worldwide for the prevention and treatment of hormone receptor positive breast cancer and its importance is due also to the fact that it is the only hormonal agent approved by the US Food and Drug Administration for the prevention of breast cancer (Fisher et al., 2005), the treatment of ductal carcinoma in situ (Fisher et al., 2002), and the treatment of pre-menopausal breast cancer (International Breast Cancer Study Group and others, 2006).

The dataset has been supplied by the Protein Analysis Facility, Center for Integrative Genomics of the University of Lausanne. Daphnids were treated with two doses (low and high) of Tamoxifen dissolved in water and dimethyl sulfoxide (DMSO). In addition, there were two groups of daphnids without toxic treatment: an untreated group exposed only to water and a group treated only with the drug administration vehicle, DMSO. In total there were four experimental groups denoted by the labels: M4 (water), DMSO (dimethyl sulfoxide and water), C1 (low concerntration), and C2 (high concentration). Measurements are taken at two time instants, representing two stages of development of the *Daphnia*: at day 2, when it is a neonate, and at day 7 (actually between 6 and 8 days), after the first laying of eggs. Every experiment has been made in 2 replicates we will refer to as a and b. After merging the data coming for each experiment, we decided to keep only those proteins with no missing values, resulting in a dataset composed by $2,720$ proteins.

3.5 CAGE

Haematopoiesis is an ideal model for the study of multi-lineage differentiation in humans. Haematopoietic Progenitor Stem Cells (HPSC) are the blood cells that give rise to all the other blood cells and have a known functional heterogeneity. This experimental design will compare HPSC against erythroblasts (also called normoblast) and myeloblasts, which are more differentiated cells, as can be seen in Fig. 3.3. Our dataset, supplied by the Department of Life Sciences, University of Modena and Reggio Emilia, is composed of 16,491 CAGE peaks. For each of the three samples, HPSC, erythroblasts, and myeloblasts, we know the identification number, the chromosome on which it is located, the location of the start and end of the peak, the positive or negative strand of DNA in the chromosome (1 if positive, 0 if negative), raw peak values on the three cell lines, and the name of the entrez gene in close proximity to the peak.

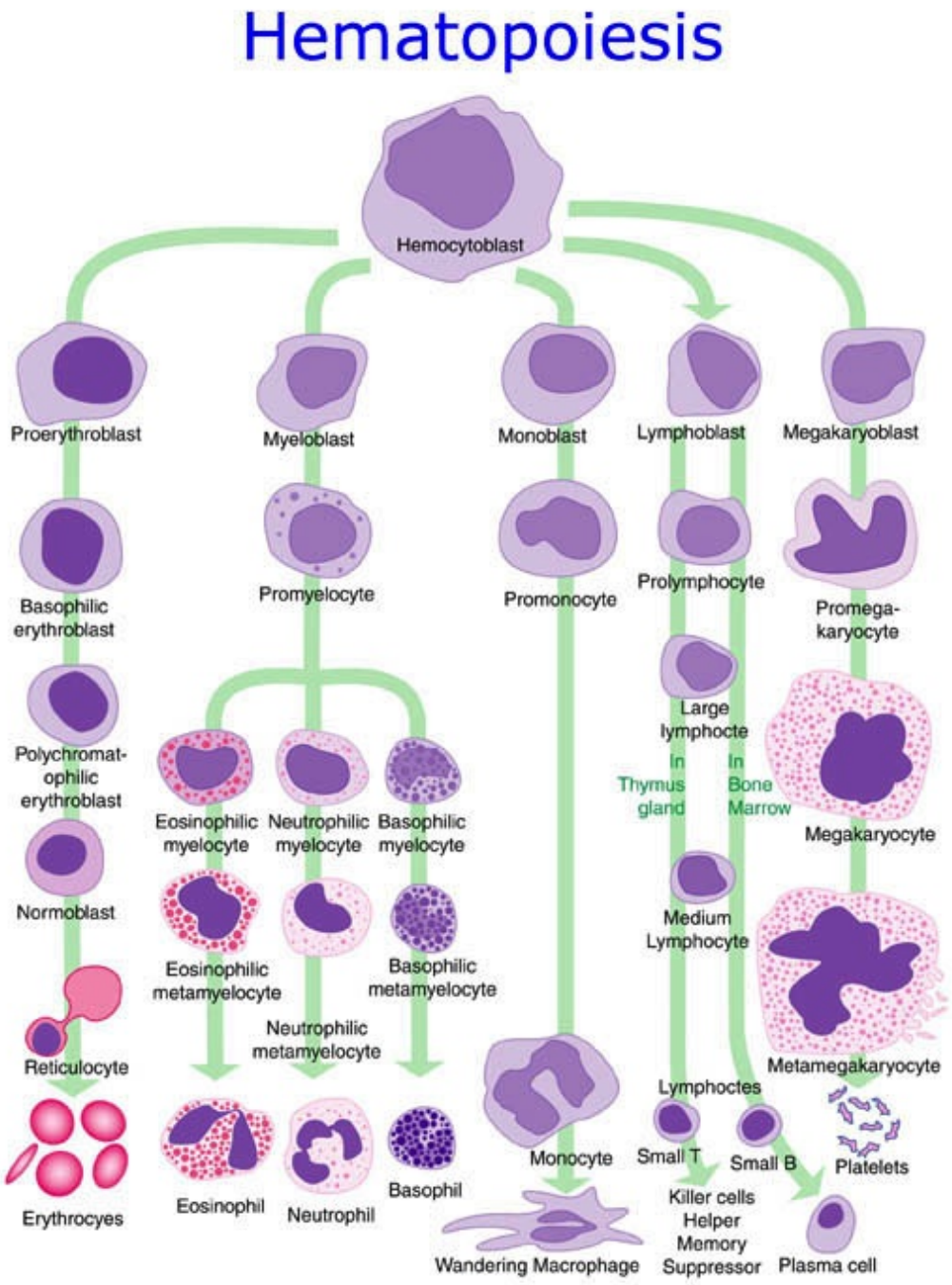

Figure 3.3: Graphical representation of the hematopoiesis. Hematopoietic stem cells (HPSC) have the potential to differentiate into various progenitor cells that eventually commit to further maturation along specific pathways. The end results of these event is the continuous production of sufficient number of cells of all lineages (red and white blood cells). Source: `http://www.auburn.edu/academic/classes/zy/hist0509/html/Lec05Bnotes-cart_bone_bloo.html`

Chapter 4

Cluster analysis of static data

4.1 Literature overview and motivation

Clustering is the process of assigning elements into a number of groups (clusters) such that elements in the same cluster are more similar than elements in different clusters. Clustering has been applied in a wide variety of fields, ranging from medical sciences, economics, computer sciences, engineering, social sciences, to earth sciences (Hartigan, 1975; Everitt et al., 2001), reflecting its important role in scientific research. With several hundred clustering methods in existence (Milligan & Cooper, 1987), there is clearly no shortage of clustering algorithms but, at the same time, satisfactory answers to some basic questions are still to come.

Clustering methods are nowadays essential tools for the analysis of gene expression data, becoming routinely used in many research projects (Drăghici, 2011). Many papers have shown that genes or proteins of similar function cluster together (Eisen et al., 1998; Spellman et al., 1998; Barenco et al., 2006; Tomancak et al., 2002; Straume, 2004; Heyer et al., 1999), and clustering methods have been used to solve many problems of biological nature. One of the most interesting of these problems is related to *disease subtyping*, i.e. the stratification of different patients in terms of underlying disease characteristics. This is extremely important in the drug development process, in which the correct identification of the subgroup of patients who are most likely to respond to the drug may be needed in order to get the drug approved by FDA. Also, ultimately, disease subtyping is expected to be the key for personalized therapies.

A widely used type of clustering is K-means (Hartigan & Wong, 1979; MacQueen et al., 1967; Forgy, 1965), the best known squared error-based clustering algorithm (Xu et al., 2005). This method consists in initializing a number of random centroids, one for each cluster, and then

associating each element to the nearest centroid. This procedure is repeated until the locations of the centroids do not change anymore.

Many of the most widely used clustering methods, including K-means, require the estimation of the most appropriate number of clusters for the data. Ideally, the resulting clusters should not only have good properties (compact, well-separated, and stable), but also give biologically meaningful results. This is an issue that derives from the more general problem of defining the term "cluster" (Milligan & Cooper, 1987) and has been extensively treated in the literature (Milligan & Cooper, 1985). Furthermore, K-means is not a deterministic method, because the results are dependent on the initialization of the algorithm and can change between successive runs. The same not-deterministic property is shared by SOM (Kohonen, 1995), a neural network clustering method, which, even if it does not need the number of clusters to be defined *a priori*, requires the user to specify the maximum number of clusters.

Another very popular clustering methods category is *hierarchical clustering* (HC), where the term *hierarchical* refers to the relation between clusters, which are nested according to a pairwise distance matrix. Results obtained with HC methods are usually represented with a dendrogram, a tree diagram where the height of the vertical lines is proportional to the difference between each pair of elements or clusters. HC methods can be divided into *agglomerative* or *divisive* methods. Agglomerative clustering (also called *bottom up*) starts with each element belonging to one different cluster (singleton cluster). Following the initialization, a series of *merge* operations groups pairs of clusters based on a predefined distance metric, until only one cluster remains, containing all elements. Divisive clustering (also called *top down*) proceeds in the opposite way. At the beginning all elements belong to the same cluster and the algorithm successively divides it until all clusters are singleton clusters. Based on different definitions of the distance between two clusters, there are many agglomerative clustering algorithms. Two of the simplest and most popular methods include Complete-linkage (CL) (Sørensen, 1948) and Ward's minimum variance criterion (Ward Jr, 1963) henceforth referred to as "Ward". The former uses the greatest among the distances of each pair of elements in a cluster to define the inter-cluster distance. The latter assumes that a cluster is represented by its centroid (the location that corresponds to the means of the coordinates in the multivariate space) and measures the proximity between two clusters in terms of the increase in the Sum of Squared Error (SSE) that results from merging two clusters.

One advantage of HC is that the number of clusters is not required as a parameter. However, this method does not solve this issue completely, as the number of clusters is not determined

exactly. In HC the number of clusters can be chosen after the inspection of the dendrogram, by cutting it at a certain depth or choosing a number of clusters based on intuition (see Chap. 18 in Drăghici (2011)). As a disadvantage, HC has a great computational complexity of the order of $O(n^2)$ where n is the number of data points to be clustered (Xu et al., 2005).

The determination of the number of clusters is a common issue in many clustering methods (Gordon, 1996; Milligan & Cooper, 1985). This problem is often solved by involving the use of indices that measure the quality of clusters according to their compactness and separation. However, a disadvantage shared by most of these indices is that they do not offer any indication whether the data should be clustered at all (Gordon, 1996). The requirement that the number of clusters has to be known in advance is only one of the most common limitations of clustering methods. Many existing methods will assign *all* the input data to some cluster. Even if there exist methods that allow for *overlapping* clustering (an element can simultaneously belong to more than one group), and *partial* clustering (not every element belongs to a cluster), the most widely used methods result in *complete* clustering, where every element is assigned to exactly one cluster, disregarding the possibility that some elements might be *outliers* that do not belong to any real group. There are various applications for which it makes little or no sense to force all data items to belong to some group, and doing so inevitably leads to poorly-coherent clusters. For example, genetic data often contain a significant amount of noise, and complete clustering leads to the presence of many outliers among the genetic entities that are clustered.

4.2 Cross-clustering

The CC algorithm proposed here overcomes four of the most common limitations of the existing clustering algorithms. First, it does not necessarily group all the elements into clusters, thus falling in the category of partial clustering methods. The basic assumption made in this algorithm is that the data points that greatly deteriorate the quality of the clusters represent noise, and thus should not be included in the final clustering. Second, the algorithm automatically identifies the optimal number of clusters. Third, CC allows the possibility of obtaining one cluster as result, suggesting that partitioning that specific data may not be appropriate. Fourth, CC is a deterministic algorithm that does not depend on any initialization, and always produces reproducible results. These results are obtained by combining the two basic principles of Ward and CL.

As Ward attempts to minimize the sum of the squared distances of points from their cluster

centroids, it is able to build well-separated clusters and to provide a good estimate of the number of clusters. On the other hand, as CL defines the proximity of two clusters as the maximum of the distance between any two points in the two different clusters, it is not optimal in separating clusters appropriately, but it is able to clearly identify outliers.

The parameters required by CC are optional and include a reasonable interval of values for the number of clusters for Ward (n_W), and a maximum number of clusters to be used in CL (n_C). Partitions are computed for both methods for each value in the input range. Then, each partition obtained cutting the Ward dendrogram in correspondence with n_W clusters is combined with each partition resulting from the cutting of CL with a number of clusters n_C, which has to be higher than n_W, allowing for the identification of small and singleton clusters, likely to contain outliers. The algorithm then choses the partitioning yielding the maximum consensus between the two methods, providing the number of clusters and the elements to be considered as outliers. Note that n_W and n_C are there just to reduce the computational effort. In principle, the cluster range can always be set from a minimum of 2 clusters to a maximum number of clusters equal to the number of data points.

4.2.1 The algorithm

As explained above, CC can use two optional parameters in order to reduce the computational complexity. The first one is the range for the number of clusters $I^W = [n_{W_{min}}, \ldots, n_{W_{max}}]$. This range can be chosen by performing Ward and choosing a reasonable interval after the visual inspection of the results though the dendrogram. A set of different partitions $\mathbf{k}_i^W$, where $i = 1, \ldots, \text{length}(I^W)$, is obtained by cutting the tree at the levels associated to each $n_{W_i} \in I^W$. The second parameter is the range for the number of clusters in CL, $I^C = [n_{W_{min}} + 1, \ldots, n_{C_{max}}]$, where $n_{C_{max}}$ is an arbitrary number, greater than $n_{W_{max}}$, but smaller than the number of elements minus one. Also this range can be chosen after the visual inspection of the CL dendrogram. Setting a high $n_{C_{max}}$ allows to isolate those elements which are less likely to belong to a cluster, eliminating them from the final partition. Then, the CC algorithm works as follows:

1. A set of different partitions $\mathbf{k}_j^C$, where $j = 1, \ldots, \text{length}(I^C)$, is obtained by cutting the tree at the levels associated to each $n_{C_j} \in I^C$.

2. For all the possible pairs $\mathbf{k}_j^C$, $\mathbf{k}_i^W$ such that $n_{C_j} > n_{W_i}$ build the contingency table A, where the element a_{rs} represents the number of elements belonging simultaneously to

cluster c_r^W (the r-th cluster obtained with Ward when the number of total clusters is set at n_i^W) and to cluster c_s^C (the s-th cluster obtained with CL when the number of total clusters is set at n_j^C). The requirement $n_{C_j} > n_{W_i}$ if essential in order to isolate as much noise and singleton elements as possible.

3. Permute the columns of matrix A until the sum of elements on the diagonal of A, denoted as $\mathrm{sum}(\mathrm{diag}(A))$, is the largest possible. The term $\max(\mathrm{sum}(\mathrm{diag}(A)))$ represents the maximum amount of overlap between partitions $\mathbf{k}_i^W$ and $\mathbf{k}_j^C$, and it will be denoted henceforth as MO_{ij}.

4. Choose the pair $(i^*, j^*) = \mathrm{argmax_{i,j}} MO(i,j)$. The optimal number of clusters is indicated by $n^*_{W_i}$, and this follows from the ability of Ward to identify well-separated clusters. The optimal partitioning is identified by the consensus clusters obtained by combining $\mathbf{k}_{j^*}^C$ and $\mathbf{k}_{i^*}^W$.

There could be cases in which more than one A is associated with the same maximum number of clustered elements. In order to choose a unique solution, the largest Average Silhouette Width (ASW) is used to select the best combination of $\mathbf{k}_j^C$ and $\mathbf{k}_i^W$.
The Silhouette Width (Rousseeuw, 1987) is defined as the average of the degree of confidence of an element to be in a cluster. For an element i:

$$S(i) = \frac{b_i - a_i}{max(b_i, a_i)},$$

where a_i is average distance between i and elements in the same cluster, and b_i is the average distance between i and elements in the nearest cluster. $S(i)$ lies in $[-1, 1]$ and should be maximized. Taking the average of the $S(i)$ for all objects i belonging to that cluster we obtain the ASW, which allows us to identify weak clusters.

In the case of more than one partition obtaining the same maximum ASW, the algorithm will choose the one with the smaller number of clusters in the Ward algorithm.

4.2.2 Example

We are now going to present an example to clarify the algorithm and the notation. Data in form of a 10×7 matrix were simulated with the rows representing genes and columns identifying samples. The random values of the first two samples are drawn from $\mathcal{N}(\mu = 10, \sigma = 0.1)$, the values of samples 3 and 4 are drawn from $\mathcal{N}(\mu = 20, \sigma = 0.1)$, the values of samples 5 and 6 are drawn from $\mathcal{N}(\mu = 5, \sigma = 0.1)$, while the last sample is designed to be the outlier, with

		CL			
		Cluster 1	Cluster 2	Cluster 3	Cluster 4
Ward	Cluster 1	5	0	0	0
	Cluster 2	0	2	0	4
	Cluster 3	6	0	3	0

Table 4.1: Example of a contingency matrix that will end up with an empty cluster.

values drawn from $\mathscr{U}(0,1)$. We used RStudio (R Core Team, 2014) (version 0.98.507) setting a seed of 123.

The distance chosen for the algorithm is Euclidean, as it is the most commonly used metric (Xu et al., 2005). In this example the interval of plausible number of clusters is easy to set and will be between $n_{W_{min}} = 2$ and $n_{W_{max}} = 5$ for the Ward algorithm, to let the Complete-linkage (CL) have always a higher number of clusters ($n_{C_{min}} = 3$ and $n_{C_{max}} = 6$) and be able to isolate the outlier. This results in 10 contingency tables A with a number of rows varying between 2 and 5, and a number of columns varying between 3 and 6, representing all the possible combinations of Ward and CL partitions. The list of possible combinations is shown in Table 4.2. An example of a combination is shown in Table 4.3, where the number of clusters for the Ward algorithm is set equal to 3 while the CL is set equal to 4. A^* starts as an empty matrix of size $n^*_{W_i} \times n^*_{C_j}$. The algorithm chooses $\max(A)$, puts it in $A^*[1,1]$ and removes the row and column where $\max(A)$ appears, obtaining the sub-matrix A^-. Then, the algorithm sets $A^*[2,2]$ as $\max(A^-)$, removes the row and column where $\max(A^-)$ appears, obtaining a sub matrix of A^-, defined as A^{2-}. Lastly, it sets $A^*[3,3]$ as $\max(A^{2-})$, removes the rows and columns where $\max(\mathrm{A}^{2-})$ appears, and stops, as an empty matrix is obtained. In this case A^* is a 3×3 matrix with all 2 on the diagonal. Cross-clustering (CC) repeats this operation for every combination of number of clusters. In this example we obtain 10 different A^* matrices. The pairs that maximize the sum of the diagonal (representing the maximum number of elements clustered together by both Ward and CL) is obtained for combinations [(3,4), (4,5), (5,6)]. The combination (3,4) is chosen as it yields the largest ASW of 0.84. CC here suggests 3 clusters of the first three pairs of samples, excluding the last sample, which is correctly identified as an outlier.

The algorithm may end in empty clusters, an example is the case of the contingency table in Table 4.1. Here A^* starts as an empty matrix of size 3×3. The algorithm choses $\max(\mathrm{A}) = 6$, it puts it in $A^*[1,1]$ and removes the row and column where $\max(\mathrm{A})$ appears, obtaining the sub-

Ward	CL	Elements classified	ASW
2	3	5	-
2	4	4	-
2	5	3	-
2	6	3	-
3	4	**6**	0.84
3	5	5	-
3	6	4	-
4	5	**6**	0.56
4	6	5	-
5	6	**6**	0.28

Table 4.2: Results from the simulated data at step 5 of the algorithm. The first two columns represent the number of clusters in Ward's minimum variance and CL methods, while the third one shows how many samples have been classified together simultaneously by both algorithms. The last column reports the average Silhouette Width (ASW) computed in case of multiple maximum values of elements classified (6, in this case) in order to choose the optimal combination.

		CL			
		Cluster 1	Cluster 2	Cluster 3	Cluster 4
Ward	Cluster 1	0	2	0	0
	Cluster 2	0	0	2	1
	Cluster 3	2	0	0	0

Table 4.3: Contingency matrix A obtained after step 7 of CC algorithm on simulated data when the number of clusters for Ward algorithm is set to 3 and for CL is set to 4.

matrix A^-. Then, the algorithm sets $A^*[2,2] = \max(\mathrm{A}^-) = 4$, removes the row and column where $\max(\mathrm{A}^-)$ appears, obtaining a sub-matrix A^{2-} which is of dimensions 1×2 and contains only zero values. In the last available cell on the diagonal of A^* we will put $\max(\mathrm{A}^{2-}) = 0$. In this case, such clusters are simply eliminated and the number of clusters will be decreased, also in the case of just one left cluster.

4.3 Comparison criteria

In order to assess the performance of CC we used the adjusted Rand Index (ARI, Hubert & Arabie (1985)), an updated form of the Rand Index (Rand, 1971), which measures the agreement between two partitionings correcting it for chance agreement. This index has an expected value of 0 for independent partitionings and maximal value 1 for identical clusterings. Negative values are possible and indicate less agreement than expected by chance. There are several external indices like the ARI in the literature, such as Hubert (Hubert & Schultz, 1976) and Jaccard (Jaccard, 1912), but they can be sensitive to the number of classes in the partitions or to the distributions of elements in the cluster (Dubes, 1987). The ARI is not affected by any of these issues (Milligan & Cooper, 1986) and has been found to have the most desirable properties in a comparative study of several pairwise clustering agreement criteria (Steinley, 2004), making it the choice as main measure of comparison. In this work, we used the ARI to compare the partitionings obtained with different clustering methods with the reality, in order to measure the quality of the prediction. However, as the ARI compares only partitions of the same length, elements identified to be outliers by CC have been considered as a single cluster. For computing the ARIs we used the R package `clues` (Chang et al., 2010).

An issue in using the ARI on the simulated data (§ 3.1) to score different results is that we are considering the 150 outlier profiles as a single cluster, while they could represent 150 different clusters. Therefore, we analyzed the ability of various methods to cluster elements correctly. Assume that we are analyzing a dataset with a specific method M that produces a partitioning of the genes into clusters. Given a *real* cluster C, assume that the majority of elements of C is clustered together by M in a cluster X. We then identify X as being representative of C. Given these assumptions, we can define:

- True Positives: genes belonging to cluster C clustered together by M in the cluster with the largest number of genes actually coming from cluster C.
- False Positives: genes not belonging to cluster C clustered together by M in the cluster

with the largest number of genes actually coming from cluster C.

- False Negatives: genes belonging to cluster C not clustered by M in the cluster with the largest number of genes actually coming from cluster C.
- True Negatives: genes not belonging to cluster C not clustered by M in the cluster with the largest number of genes actually coming from cluster C.

Once these categories are defined, we can plot, for each method, the Receiver Operating Characteristic (ROC) curve, to study their ability to identify correct cluster memberships. The measure used to summarize the performance is the Area Under the Curve (AUC). The AUC combines sensitivity and specificity, where sensitivity measures the proportion of actual positives that are correctly identified as such, while specificity measures the proportion of negatives that are correctly identified as such. AUC for each of the $K = 6$ clusters (considering the outliers as a single cluster) has been computed with the R package `pROC` (Robin et al., 2011).

In order to assess the performance of CC, we compared it with the methods on which it is based, as well as with a method that attempts to find the correct number of clusters and to identify outliers, the DBSCAN method (Ester et al., 1996). We performed these comparison on a set of simulated data. Furthermore, we also compared CC with Ward, CL, DBSCAN, K-means and SOM on two real datasets, showing that CC is able to obtain meaningful results that represent the underlying partitioning of the data.

4.4 Simulation study

We applied the Ward and CL algorithms on the simulated data (§ 3.1), using the Euclidean distance. The number of clusters k is chosen as follows. We computed the clusterings for each k in the interval $[2, 20]$ and we used the value of k for which the ASW was maximum. This represents a reasonably good choice for k. The only parameters to be set in CC are the interval for the number of clusters in Ward, in this case set to $k \in [2 : 19]$, and the maximum number of clusters in the CL algorithm, which is set to 20. These parameters were used in all the applications of CC throughout this work.

DBSCAN determines automatically the number of clusters and classifies points in low-density regions as noise. The issue with this algorithm is that it needs the user to determine two parameters, ϵ and *MinPts*. A point is a *core point*, i. e. is in the interior of a cluster, if it has more than a specified number of points (*MinPts*) within a radius of ϵ, the reachability index (a

sort of radius essential in order to compute the density). The criterion suggested by the authors (Ester et al., 1996) for the choice of the parameters only works for two-dimensional datasets. The idea is that points in a cluster are roughly at same distance from their n^{th} nearest neighbor, while the distance from noise points is higher. The suggested n for two-dimensional data is 4. Therefore, plotting the sorted distances, sorted in descending order, of each point from its n^{th} nearest neighbor gives hints concerning the proximity of the elements in the data. A threshold point p should be chosen to be the first one in the first "valley" of the sorted distances: all the points on the left of the threshold are considered to be noise, while all other points are assigned to some cluster. The parameters are then set such as $\epsilon = dist(p)$ and $MinPts = n$. As it is not practical to use this criterion on 100 simulations, here we set $\epsilon \in [0.1, 10]$ and *MinPts* equal 5, the default value of the R package `fpc` (Hennig, 2014).

Fig. 4.1 shows that the CC algorithm has a good performance if compared to the methods that constitute it, resulting in values of ARIs always higher than those coming from CL and very similar to Ward results. Fig. A.1 in Appendix A shows a very similar performance also when the chosen distance is Chebychev.

Fig. 4.2 shows boxplots of the AUC for each of the six clusters over 100 simulations, with varying σ in the added noise, for CC, CL, and Ward. The last group represents the outliers. Fig. 4.3 shows boxplots of the average AUC over the six clusters for 100 simulations, for various values of ϵ. The comparisons show that CC obtains an higher AUC in every cluster and with each σ with respect to the methods that constitute it, proving its ability in detecting the correct clusters and in identifying the outliers. Figs. A.2 to A.5 in Appendix A show that CC obtains higher AUCs for each cluster with respect to Ward and CL also when the distance chosen is Chebychev. The comparison with DBSCAN shows a high variability of this method in the resulting AUC, showing its sensitivity to parameter choice and to σ, with performance decreasing as σ increases.

4.4.1 Number of clusters

We compared the ability of CC to detect the correct number of clusters. In order to do so, we compared the results of CC with several well established methods: CH (Caliński & Harabasz, 1974), Silhouette Width (Rousseeuw, 1987), Dunn Index (Dunn, 1974), Beale Index (Beale, 1969), C-Index (Hubert & Levin, 1976), Duda Index (Duda & Hart, 1973), H (Hartigan, 1975), KL (Krzanowski & Lai, 1988), Gap (Tibshirani et al., 2001), and Jump (Sugar & James, 2003). A good summary of these methods is given by (Gordon, 1996), while Milligan & Cooper (1985) conducted a simulation study of the performance of 30 decision rules. These

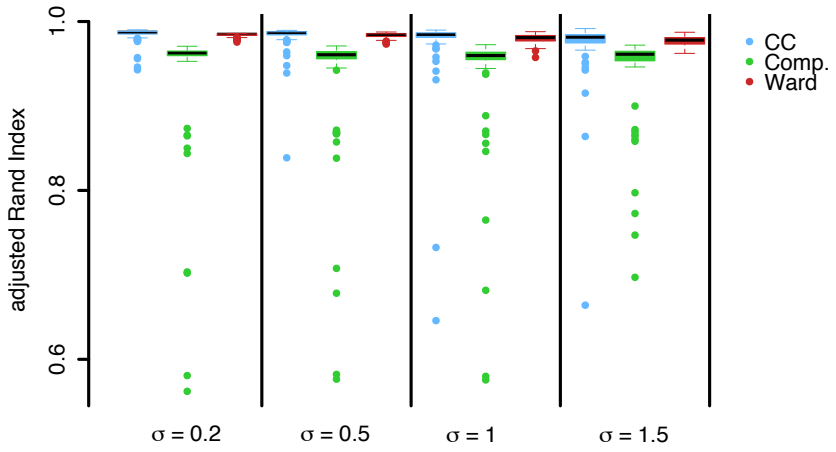

Figure 4.1: Boxplots of the Adjusted Rand Index resulting from Cross-clustering (CC), Complete-linkage (CL) and Ward's algorithm (all with Euclidean distance) with $\sigma = 0.2, 0.5, 1, 1.5$ on simulated data. The ARI here is used to measure the agreement between the obtained partition with each method and the real partition (the higher, the better), proving that CC performs at least as well as the two methods that constitute it.

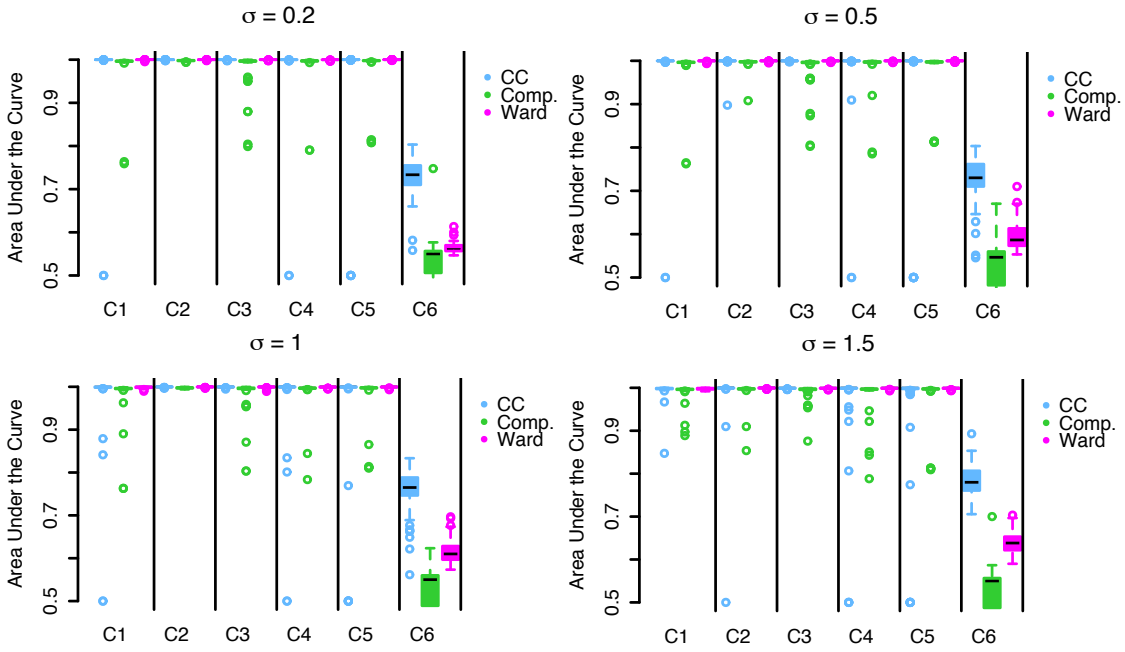

Figure 4.2: Boxplots of the Area Under the Curve (AUC) in each of the $K = 6$ simulated clusters for $S = 100$ simulations with Cross-clustering (CC), Complete-linkage (CL) and Ward algorithm with $\sigma = 0.2, 0.5, 1, 1.5$. High AUC values represent high true positive rates and low false positive rates, thus, the higher the better. In the first clusters the performance is approximately the same, while in the last group, which is the outliers one, CC performs better.

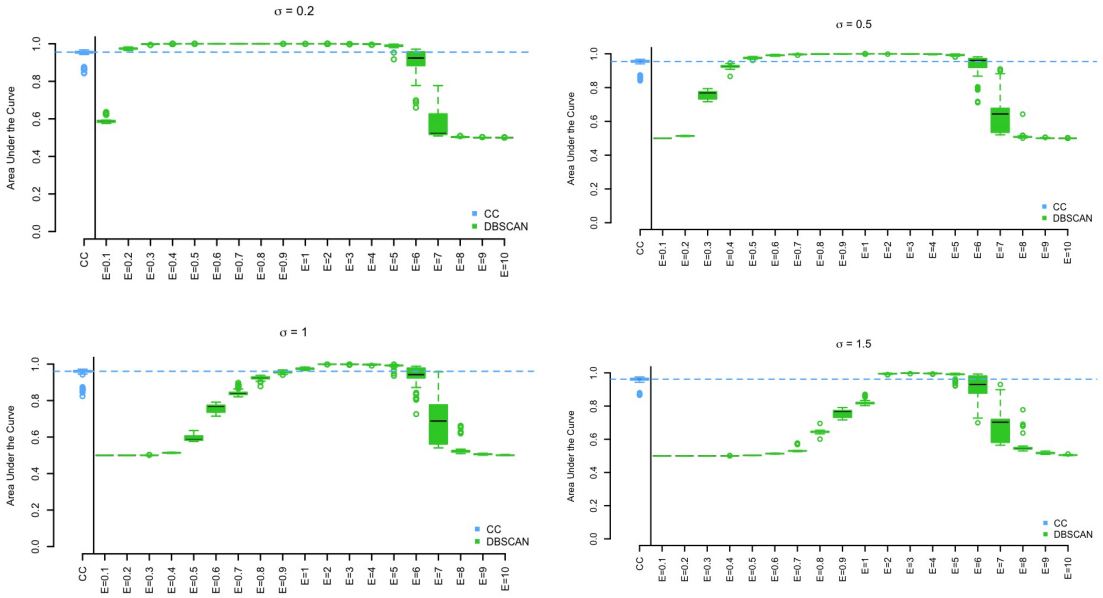

Figure 4.3: Boxplots of average Area Under the Curve (AUC) over the $K = 6$ simulated groups coming from Cross-clustering (CC) and DBSCAN where ϵ is changing ($\epsilon \in [0.1, 10]$) and the minimum number of points within ϵ is set equal 5 (as by default in the R package used), when $\sigma = 0.2, 0.5, 1, 1.5$. High values represent high true positive rates and low false positive rates, thus, the higher the better. The horizontal line represents CC's median value, which is higher than most of the values resulting from DBSCAN. This figure shows how sensitive DBSCAN is to the choice of parameters.

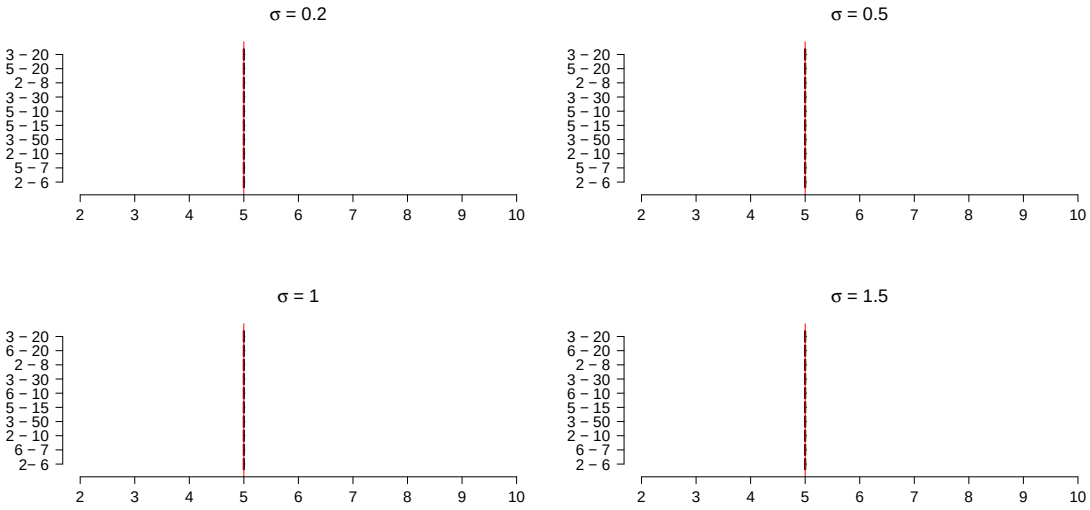

Figure 4.4: Boxplots of the number of clusters found by the CC algorithm with ten different intervals I^W and with $\sigma = 0.2, 0.5, 1, 1.5$. The vertical line represents the correct number of clusters, $k = 5$. Labels on the y-axis represent the lower and upper boundaries of each interval. The x-axis has been shortened for reasons of space, but x can take values up to 50. The maximum number of clusters for the CL is always set equal 99. In this case CC was always able to detect the actual number of clusters.

methods were combined with Ward, CL, and K-means. The results of such comparisons are reported in Figs. A.6 to A.17 and in Table A.1 in Appendix A. CC always detected the correct number of clusters, while the other methods showed variability in the results.

4.4.2 Sensitivity analysis

We wanted also to assess how stable the method is with regard to the input parameters ($n_{W_{min}}$, $n_{W_{max}}$, and $n_{C_{max}}$). Therefore, we performed CC with 10 different pairs of values for the boundaries of I^W, while setting a high and fixed $n_{C_{max}} = 99$ on the simulated data after the removal of outliers (a representation of the intervals can be seen in Fig. A.18 in Appendix A). Results in terms of number of clusters identified by the CC are shown in Fig. 4.4 and prove the robustness of the method, which is always able to identify the correct number of clusters ($K = 5$) also when I^W was very large or asymmetric, and independently from σ.

4.5 Case studies

4.5.1 Brain tumors dataset

Similarly to the analysis performed on simulated data, we applied the ASW to the brain tumors dataset (§ 3.1) in order to choose the number of clusters while comparing the results obtained by applying CC, Ward, CL, DBSCAN, K-means, and SOM. We applied SOM with a rectangular topology and all the default parameters of the R package `kohonen` (Wehrens & Buydens, 2007): Euclidean distance, number of training iterations $rlen = 100$, learning rate (amount of change) α decreasing linearly from 0.05 to 0.01, the radius of the neighborhood starting with a value that covers $2/3$ of all unit-to-unit distances, and initial values for each node chosen randomly without replacement from the data. In order to choose the dimensions of the grid, we applied SOM with 70 combinations of dimensions (see Table A.2 in Appendix A) and we computed the ASW for each combination.

Furthermore, we performed the Fisher exact test for count data in order to test if any cluster was over-represented in any particular subtype. All the p-values obtained were corrected for multiple testing with the BH method (Benjamini & Hochberg, 1995).

We applied Ward and CL to the data using the same parameters used for the simulation study. In order to address the dependence of K-means on the initialization of the parameters, we ran the algorithm 10 times, every time choosing k in the interval $[2, 20]$ that maximizes the ASW.

Classical approaches performed poorly, yielding ARI values ranging from 0.003 to 0.19, the highest value being obtained by K-means. In terms of number of clusters, the ASW criterion for Ward, CL, and SOM identified two clusters (maximum ASW of 0.19 in each method), while K-means resulted in three clusters (maximum ASW of 0.17). Among the two clusters identified by Ward, one cluster represented the subtype Rhab (p-value of $3.81e^{-06}$). CL reported two clusters, one of which represented the Rhab subtype (p-value of 0.0003). K-means reported three clusters, one of which represented the PNET subtype (p-value of 0.0005). None of the clusters reported by SOM was over-represented in any subtype, as expected from the low ARI. In contrast, CC obtained an ARI of 0.64, identifying nine clusters and one sample as an outlier. Although CC identified more than five clusters, four of them almost perfectly represented MD, MGlio, Ncer, and Rhab subtypes (p-values, respectively, of $9.53e^{-06}, 1.94e^{-05}, 1.12e^{-04}, 9.53e^{-06}$), as it is shown in Fig. 4.5. The PNET subtype was represented by six of CC clusters. This particular subtype is defined by the World Health Organization (WHO) with useful guidelines for diagnosis its heterogeneous histological characteristics and malignancy grade (Louis et al.,

2007). All the contingency tables reporting both the real classification in subtypes and the clusters identified by each method are shown in Tables A.3 to A.7. A summarization of the results can be found in Table 4.4, while dendrograms for the HC methods can be seen in Figs. A.19 and A.20 in Appendix A. Lastly, we applied DBSCAN trying different types of n-th nearest neighbors. Setting the value of ϵ and $MinPts$ to each of the pairs obtained for different values of n ($\epsilon = 25.50, MinPts = 4; \epsilon = 27.41, MinPts = 5; \epsilon = 27.12, MinPts = 6; \epsilon = 27.65, MinPts = 7$) always led to a unique non-noise cluster (2 outliers and 40 elements grouped in a single cluster).

	Results	
Method	# Clusters	ARI
Ward	2	0.14
CL	2	0.18
K-means	3	0.19
SOM	2	0.003
CC	9 + 1	**0.64**

Table 4.4: Summarization of results on brain tumors dataset with Euclidean distance. The table reports every method applied with the resulting number of clusters and adjusted Rand Index (ARI) (the higher the better). The actual number of subtypes in the dataset was five. CC identified nine clusters plus one cluster containing one outlier, leading to the highest ARI. None of the other methods was able to identify the correct number of clusters. Although CC identified more than five clusters, four of them almost perfectly represented MD, MGlio, Ncer, and Rhab subtypes. The PNET subtype was represented by six of CC clusters, but it is well known for having an high heterogeneity.

4.5.2 Breast cancer dataset

A second disease subtyping analysis was performed on the breast cancer dataset (§ 3.3) similarly to the comparison of the previous Section. Classical approaches performed poorly, obtaining ARI values ranging from 0.04 to 0.1, the highest value being obtained by CL. In terms of number of clusters, the ASW criterion for Ward, CL, and K-means identified 11 clusters (maximum ASW of 0.24, 0.24, and 0.25 respectively), while SOM resulted in 18 clusters (maximum ASW of 0.18), out of which two were empty. DBSCAN detected one cluster containing 28 of the 30 elements. In contrast, CC obtained an ARI of 0.63, showing great agreement with the

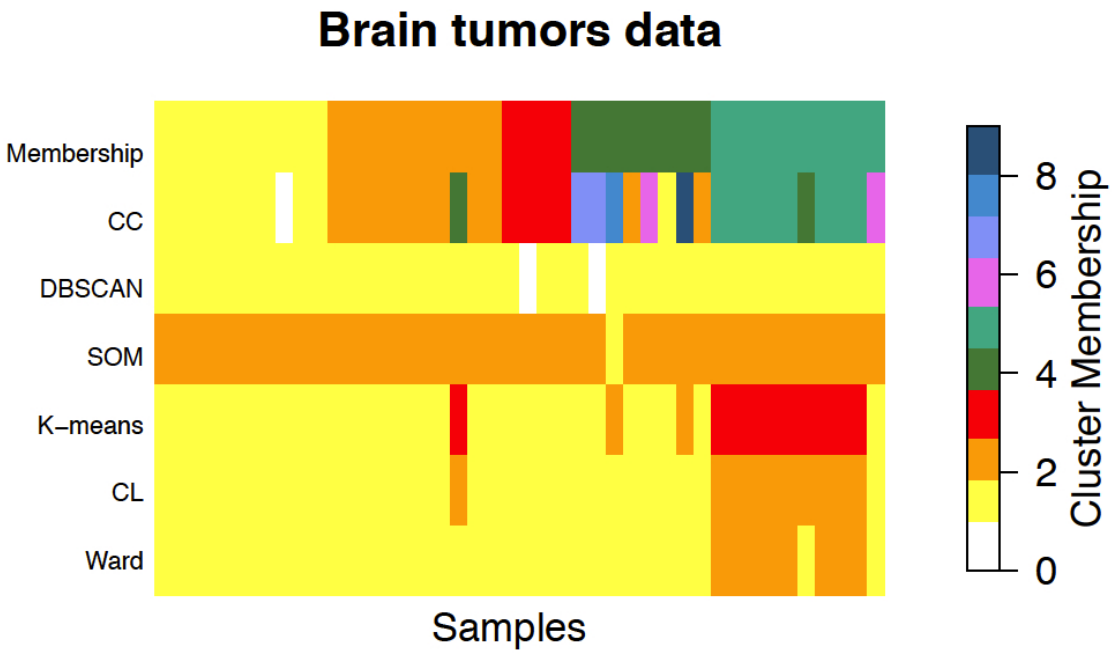

Figure 4.5: Graphical representation of the true membership (first row) of the 42 samples in the brain tumors dataset, compared with the memberships resulting from CC, DBSCAN, SOM, K-means, CL, and Ward. The five subtypes, in the order in which are sorted in the membership row of the image, are: medulloblastoma (MD), malignant gliomas (MGlio), normal human cerebella (Ncer), primitive neuroectodermal tumors (PNET), and atypical teratoid/rhabdoid tumors (Rhab). Different colors represent only the index of the cluster given by each method. The white color represents outliers, only detected by CC and DBSCAN. Classical approaches performed poorly, obtaining ARI values ranging from 0.003 to 0.19, the highest value being obtained by K-means. In terms of number of clusters, the ASW criterion for Ward, CL, and SOM identified two clusters (maximum ASW of 0.19 in each method), while K-means resulted in three clusters (maximum ASW of 0.17). In contrast, CC obtained an ARI of 0.64, identifying nine clusters and one sample as an outlier. Although CC identified more than five clusters, four of them almost perfectly represented four of the real subtypes while PNET, a subtype known to present heterogeneous histological characteristics, was fragmented in six clusters. Furthermore, it is important to notice that, while CC requires a loose set of parameters (a range where the real number of clusters has to be found), K-means require the correct number of clusters, to be found with one of the many techniques available, and SOM requires two parameters whose choice is not easy.

ground truth, and identifying correctly the number of clusters. Two out of the 30 elements were considered outliers. A graphical representation of the results is shown in Fig. 4.6, while a summarization of the results can be found in Table 4.5, and dendrograms for the HC methods can be seen in Figs. A.21 and A.22 in Appendix A.

CC was the only method to yield clusters that were over-represented in elements belonging to real subtypes. The two clusters identified by CC represented the two cancer subtypes, luminal with a p-value of $2.81e^{-05}$, TN with a p-value of $3.28e^{-05}$. All the contingency tables reporting both the real classification in subtypes and the clusters identified by each method are reported in Tables A.8 to A.12 in Appendix A.

	Results	
Method	# Clusters	ARI
Ward	11	0.09
CL	11	0.10
K-means	11	0.04
SOM	18	0.07
CC	2 + 1	**0.63**

Table 4.5: Summarization of results on breast cancer dataset with Euclidean distance. The table reports every method applied with the resulting number of clusters and adjusted Rand Index (ARI, the higher the better). The actual number of subtypes in the dataset was two. CC identified two clusters plus one cluster containing two outliers, leading to the highest ARI.

4.6 R package

The Cross-clustering algorithm was implemented in the numerical computing environment R and will be submitted to CRAN with the name `CrossClustering`. The R code can be found in Appendix B.

Description

The function `CrossClustering` performs the Cross-clustering algorithm. This method combines the Ward's minimum variance and Complete-linkage algorithms, providing automatic estimation of a suitable number of clusters and identification of outlier elements.

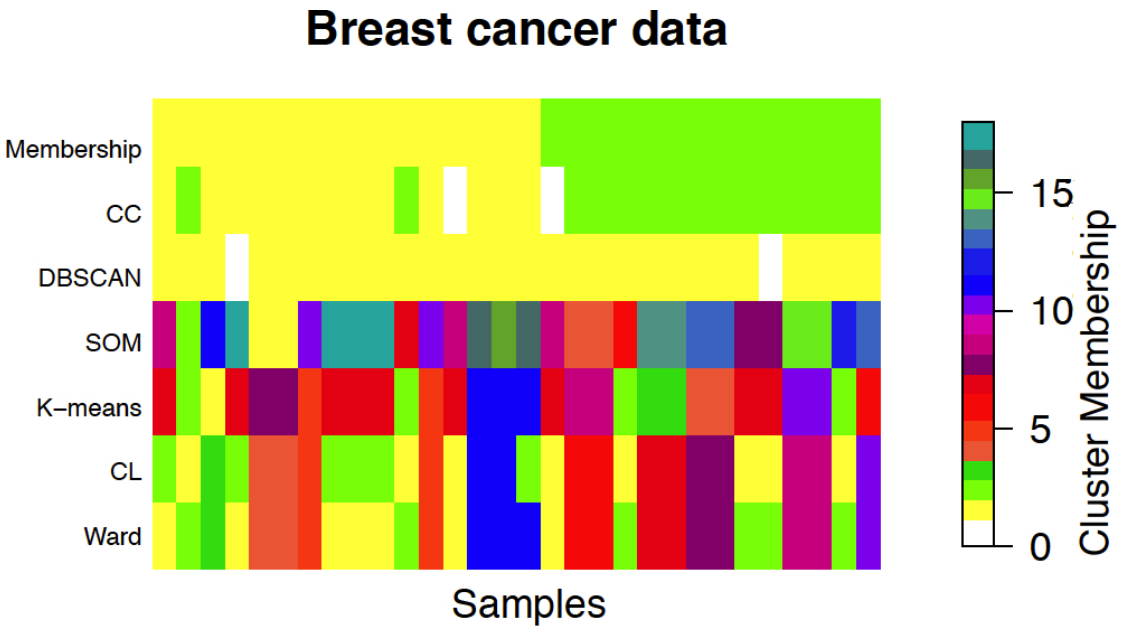

Figure 4.6: Graphical representation of the true membership (first row) of the 30 samples in the breast cancer dataset, compared with the memberships resulting from Cross-clustering (CC), DBSCAN, SOM, K-means, Complete-linkage (CL), and Ward. There are two subtypes: luminal and triple negative (TN). The yellow color represents the luminal subtype, the green color represents the TN subtype, while white color represents outliers, only detected by CC and DBSCAN. Different colors represent only the index of the cluster given by each method. Classical approaches performed poorly, obtaining ARI values ranging from 0.04 to 0.1, the highest value being obtained by CL. In terms of number of clusters, the ASW criterion for Ward, CL, and K-means identified 11 clusters (maximum ASW of 0.24, 0.24, and 0.25 respectively), while SOM resulted in 18 clusters (maximum ASW of 0.18), out of which two were empty. DBSCAN detected one cluster containing 28 of the 30 elements. In contrast, CC obtained an ARI of 0.63, showing great agreement with the ground truth, and identifying correctly the number of clusters. Two out of the 30 elements were considered outliers. Furthermore, it is important to notice that, while CC requires a loose set of parameters (a range where the real number of clusters has to be found), K-means require the correct number of clusters, to be found with one of the many techniques available, and SOM requires two parameters whose choice is not easy.

Usage

CrossClustering(d, k.w.min = 2, k.w.max, k.c.max, out = TRUE)

Arguments

- d: a dissimilarity structure as produced by the function `dist`.
- k.w.min: minimum number of clusters for the Ward's minimum variance method. Default value is 2.
- k.w.max: maximum number of clusters for the Ward's minimum variance method.
- k.c.max: maximum number of clusters for the Complete-linkage method. It can not be equal or greater than the number of elements to cluster.
- out: logical. If `TRUE` (default) outliers must be searched for.

Value

A list of objects describing characteristics of the partitioning as follows:

- Optimal.cluster: number of clusters.
- Cluster.list: a list of clusters; each element of this lists contains the indices of the elements belonging to the cluster.
- Silhouette: the average silhouette width over all the clusters.
- n.total: total number of input elements.
- n.clustered: number of input elements that have actually been clustered.

Example

```
### Generate simulated data
toy <- matrix( NA, nrow = 10, ncol = 7 )
colnames( toy ) <- paste( "Sample", 1:ncol( toy ), sep= " " )
rownames( toy ) <- paste( "Gene", 1:nrow( toy ), sep = " " )
set.seed( 123 )
toy[ ,1:2 ] <- rnorm( n = nrow( toy )*2, mean = 10, sd = 0.1 )
```

```
toy[ ,3:4 ] <- rnorm( n = nrow( toy )*2, mean = 20, sd = 0.1 )
toy[ ,5:6 ] <- rnorm( n = nrow( toy )*2, mean = 5, sd = 0.1 )
toy[ ,7 ] <- runif( n = nrow( toy ), min = 0, max = 1 )

### toy is transposed as we want to cluster samples
### (columns of the original matrix)
d <- dist( t( toy ), method = "euclidean" )

### Run CrossClustering
toyres <- CrossClustering( d, k.w.min = 2, k.w.max = 5,
                               k.c.max = 6, out = TRUE)
```

4.7 Discussion

The CC algorithm provides an intuitive and easily implementable approach for clustering of gene expression data. CC presents several advantages over existing methods: i) it does not require *a priori* knowledge on the number of clusters, ii) it leaves outlier elements unassigned, and iii) it can yield only one cluster as result, suggesting that data should not be clustered at all. Furthermore, even though in this paper we only report results obtained on biological data, CC can be successfully applied on any type of data, as demonstrated by the results obtained on the simulated data. The only (unavoidable) weakness of CC is its quadratic computational complexity, inherited from the methods that constitute it. However, the computational complexity of CC is not worst than that of Ward and CL.

We compared CC with the most widely used clustering methods and CC consistently obtained partitions closer to the reality than the results obtained with these methods. As the distance is a key element in cluster analysis, we performed our comparisons using both Euclidean and Chebychev distances in order to check how sensible results are to the choice of the distance, and in both cases CC performed better than the methods it was compared with. Furthermore, CC proved to be an useful tool for detecting a suitable number of clusters in the data, better of most of the well established criteria proposed in the literature. When compared to DBSCAN, CC showed a good performance and more robustness, while DBSCAN was highly sensible to parameters choice and data variability. Furthermore, in DBSCAN the only available approach to parameters choice works only for two dimensional datasets. When applied on two real publicly

available datasets, CC was able to identify subtypes better than the other approaches. K-means and SOM produced different results in different runs as the two methods are strictly dependent on the initialization of the algorithm, yielding results ranging from very good to very poor. In contrast, the results of CC were stable, as it does not involve any random initialization.

Chapter 5

Clustering of short-length time-course data

5.1 Motivation

Many fundamental biological processes are dynamic in nature, thus they can be understood only by taking into account the factor *time*. As mRNA is transcribed continuously, also new proteins are generated in a continuous way. Some of these proteins are transcription factors, which regulate the expression of other genes initiating or repressing the transcription process. When cells are faced with a condition, such as starvation (Natarajan et al., 2001), infection (Nau et al., 2002), and stress (Gasch et al., 2000), they react to the new condition. In these cases, only the study of the gene or protein expression over time can determine the complete set of genes or proteins that are expressed under a particular condition and the interaction between them.

The output of an experiment that collects gene product measurements over time is known as *time-course* data, in contrast to static data, which consists in the snapshot of these measurements. Gene expression time-course experiments were used to investigate successfully many phenomena, for instance the cell cycle (Spellman et al., 1998; Cho et al., 1998; Zhu et al., 2000; Pramila et al., 2002; Whitfield et al., 2002), cell signaling (Barenco et al., 2006), circadian rhythms (Storch et al., 2002; Panda et al., 2002; Straume, 2004), and developmental processes (Tomancak et al., 2002; Arbeitman et al., 2002; Kim et al., 2001; Ivanova et al., 2002). This kind of experiments are potentially very useful also in knockout studies, where a gene is deleted from the genome in order to determine the downstream effects of this knockout gene and build a genetic interaction network (Zhu et al., 2000; Pramila et al., 2002; Gasch et al., 2000). Fur-

thermore, identifying genes that act in response to a certain infectious disease is a key issue in order to develop drugs to fight these diseases (Nau et al., 2002; Xu et al., 2002; Whitfield et al., 2002). Progress in the development of high throughput protein level assays (Gygi et al., 2000, 1999) allows similar techniques to be used also with protein expression data (Aach & Church, 2001).

As it is nowadays well known, grouping together genes or proteins that express in a similar fashion over time can identify which are likely to be co-regulated by the same transcription factors (Eisen et al., 1998; Quackenbush, 2001; Slonim, 2002). However, the cluster analysis of time-course data is more challenging than the cluster analysis of static data because of the unique features that characterize them.

First of all, even if the cost of high throughput technology experiments continues to fall (Kiddle et al., 2010), the prediction made by Ernst et al. (2005), who claimed that short time series (8 time points or fewer) would remain prevalent in future, still seems reasonable. In fact, time-course experiments require multiple arrays and in many cases each experiment is repeated at least once, making it still prohibitive to obtain large quantities of biological material. For this reason, in this chapter we are going to focus on short-length time-course data. On such type of data conventional techniques such as Fourier analysis, autoregressive, and moving-average modeling may not be applicable. Furthermore, these classic autocorrelation approaches generate bias when applied to short time-course data (Arnau & Bono, 2001). Short-length time series can also be extracted from one long time series in the form of a sliding window, though it is well known now that clustering this type of time series can easily lead to meaningless results (Keogh & Lin, 2005).

The second issue is that, when we cluster, we implicitly assume that the observations for each gene or protein are independent and identically distributed (*iid*), and also invariant with respect to the order of the observations. This means that if the order of the observations in two sequences is permuted, the chosen distance metric will not change, and so will not the resulting clustering. In time-course data we can not disregard temporal information because each observation depends on its past and the expression levels exhibit strong autocorrelation between successive time points. This was demonstrated in the classic paper of Eisen et al. (1998), who observed that the biologically meaningful clusters obtained by hierarchical clustering of *S. cerevisiae* microarray time series data disappeared when the observations were randomly permuted.

Furthermore, time-course experiments involve measuring the expression levels of thousands of genes or proteins repeatedly through time and result in extremely high-dimensional data,

where patterns could be recognized randomly.

Lastly, different time series can be sampled at different time points or even non uniformly as in Spellman et al. (1998); Chu et al. (1998); Eisen et al. (1998), leading to the problem of missing data. For example, some commonly used time-points are 0, 4, 12, 24, and 48 hours. Values may not have been observed also because of errors in the experiment that lead to corrupted or missing data. Unfortunately, the interpolation is difficult on this kind of data and leads to poor estimates because of the low number of replicates and the high noise.

5.2 Literature overview

Even if popular methods such as hierarchical clustering, K-means clustering, and self-organizing maps (SOM) do not address any of the above problems, much of the early work on analyzing time-course expression experiments used methods developed originally for static data (Spellman et al., 1998; Friedman et al., 2000; Zhu et al., 2000; Troyanskaya et al., 2001). As a proof of this, in a comparative analysis of clustering methods for gene expression time-course data (Costa et al., 2004) five clustering methods were compared and none of them was build expressly for time-course data (agglomerative hierarchical clustering, Eisen et al. (1998)), Cluster Identification via Connective Kernels (CLICK, Sharan & Shamir (2000)), dynamic clustering (Costa et al., 2002), K-means (Tavazoie et al., 1999), and SOM (Tamayo et al., 1999)).

More recently, several specially tailored time-course gene expression data clustering algorithms were presented, allowing researchers to fully utilize these data by taking advantage of their unique features. Möller-Levet et al. (2003) proposed the short time series (STS) distance and incorporated it in the fuzzy clustering algorithm, where each element is given a degree of membership to each set. This algorithm assigns each gene to predefined profiles, whose number grows exponentially in the number of time points and for this reason is only appropriate for very short time series. Höppner & Klawonn (2009) implemented in the fuzzy clustering the Cross Correlation distance, which can help to overcome the missing alignment of the series due to the fact that two identical time series, one of them shifted by δ in time, may appear uncorrelated under the Pearson coefficient. The resulting cross-correlation clustering can be applied to short time series and to time series subsequences. Lu et al. (2004) and Zhao et al. (2001) used predefined profiles to identify cycling yeast genes. Thus, unlike the method we are going to present, these approaches require a prior knowledge of the shape of the curve to fit. Peddada et al. (2003) suggested a method in which genes are assigned to the profile for which they best

match. Unlike our method, their algorithm requires to specify the set of profiles in which one is interested. Ernst et al. (2005) proposed a method specific for short time series, which assigns genes probabilistically to preselected sub-patterns which were generated independently of data.

Although spline methods are not recommended for fewer than five time-points, a popular approach is the use of splines as basis functions (Heard et al., 2005, 2006). Splines are continuous functions consisting of piecewise functions, which are connected to each other at knots (Friedman & Silverman, 1989). De Hoon et al. (2002) proposed a strategy that uses the maximum likelihood method together with Akaike's Information Criterion (AIC) (Akaike, 1998) to fit linear splines. Unlike our method, this strategy works only if several replications are available. The smoothing spline clustering (SSC) proposed by Ma et al. (2006) used a mixed-effect smoothing spline model and the rejection-controlled EM algorithm (Dempster et al., 1977). The algorithm handles missing data, determines the number of clusters in the data, and has been validated on datasets with a minimum of 6 time points. Bar-Joseph et al. (2003) used cubic B-spline estimations to represent time-series gene expression profiles as continuous curves and TimeFit, a model-based clustering algorithm to perform class assignment. Using B-splines, unobserved data can be estimated, allowing for missing and non uniformly sampled data. However, the clustering algorithm requires the number of clusters as an input and is appropriate only for long time series. Even when only two spline segments are used, this algorithm requires the estimation of five parameters for each gene, causing overfitting in short time series. Coffey et al. (2014) proposed the linear mixed effects model representation of penalized spline smoothing to cluster the gene expression profiles. This approach estimates using the Bayesian Information Crietrion (BIC) the number of clusters and allows for missing and irregularly sampled data. However it has been validated on datasets with at least 10 time points.

D'Urso & Maharaj (2012) proposed a clustering method based on the combination of univariate and multivariate wavelet features, but it has been validated only on long time series. Phang et al. (2003) developed a non-parametric clustering algorithm using only the direction of change from one time point to the next.

Kim & Kim (2007) proposed a difference-based clustering algorithm that discretizes a gene profile into a pattern that summarizes the first and second order differences representing direction and rate of change, respectively. It can be used only if replicates are available, it identifies the number of clusters, allows unevenly distributed time points and has been validated on datasets with at least 4 time points.

Cooke et al. (2011) used an extension of the Bayesian Hierarchical Clustering algorithm

(Heller & Ghahramani, 2005; Savage et al., 2009). This approach is able to identify the number of clusters, allows non-uniformly sampled time points, replicates, and models outlying observations by assigning them to biologically relevant clusters that share biological functions with the other cluster members. However, it has been validated only on datasets with at least 10 time points. Hensman et al. (2013) used GP regression but they combined it with a hierarchical structure to account for covariance between biological replicates. Their method allows the presence of missing and irregularly sampled data, detects an optimal number of clusters, and claims to be well suited for short time series. Another model-based clustering has been proposed by Ciampi et al. (2012) and uses a mixture or regressions with components in the Extended Linear Mixed Models (Pinheiro & Bates, 2000). It can be used in presence of missing and irregularly sampled data, it provides the number of clusters and has been successfully applied on data with 6 time points.

Although all the scientific efforts, at the moment it still does not exists a clustering method that is well suited for short-length time course data and simultaneously takes into account the temporal information, the possibility of missing values and outliers, the lack of replicates, while being able to identify a reasonable number of clusters, and to suggest if clustering a particular set of data makes sense. This is the reason why we developed a clustering algorithm that aims to overcome these limitations.

5.3 Daphnia pulex dataset

Our interest in *time-course* data, stems from a collaboration with the University of Lausanne (Institute of Earth Surface Dynamics and Protein Analysis Facility) who provided us with a dataset coming from an ecotoxicoproteomic study (§ 3.4) that wanted to analyze the response of proteins in organisms under stress. This is a protein expression dataset but this makes no difference for our statistical analysis, as methods used for clustering gene expression time-course data can be used in the same way also on protein expression time-course data. The aims of this ecotoxicoproteomic study were to assess the effects of tamoxifen in *Daphnia pulex*, an aquatic invertebrate species, at the organism and proteomic levels. In the experiment performed by Borgatta et al. (2014) daphnids were exposed to tamoxifen. Daphnids were chosen as subject of the experiment because this water flea is easy to manipulate in laboratory, reproduces quickly, produces clones, but most prominently is at the base of the food chain. Tamoxifen is a widely prescribed anticancer drug worldwide and an endocrine disruptor which causes side effects in

humans and has very powerful metabolites. Also, tamoxifen effects on aquatic species are little studied.

A first analysis was carried out by Borgatta et al. (2014) who investigated the impact of tamoxifen on daphnids at the biochemical level performing local-pooled-error (LPE) (Jain et al., 2003; Zhang et al., 2006) estimates of $\log_2$ fold-changes. Their study analyzed the differences in protein expression of daphnids resulting from acute (2 days) and middle-term (6 to 8 days) exposures to this anticancer drug. During the experiment, tamoxifen was first diluted in the solvent (DMSO), because of its poor solubility. Because of the solvent, an additional control was prepared with the solvent diluted in M4. We built upon Borgatta et al. (2014) using a different approach, that is with the two-step regression method `maSigPro`, with the aim of answering two further questions: i) whether the solvent used induces protein modifications in the treated daphnids, and ii) whether protein changes provide suitable biomarkers for an early detection of drug-related stress of daphnids.

5.3.1 Pre-processing

We removed background noise and normalized our dataset using the *variance stabilizing method* (`vsn`) proposed by Huber et al. (2002). The advantage of this combined approach is that information across arrays can be shared to estimate the background correction parameters, which are otherwise estimated separately for each array. Fig. 5.1 represents an important tool for assessing whether the `vsn` fit worked: by representing the ranked means, m_i, versus the empirical standard deviations σ_i, it proves that the distribution of σ_i is concentrated on small values and there is no significant trend of these values as a function of the means. In Fig. A.23 of Appendix A boxplots of raw and normalized values are reported for every experimental conditions, while in Fig. A.24 of Appendix A the output data of the normalization are reported in violin plots, which are combinations of boxplots and kernel density plots. Both figures show how the normalization aligned all the values.

5.3.2 Quality control

We checked the quality of the experiment performing a cluster analysis on the replicates using hierarchical clustering with the Euclidean distance. Fig. 5.2 represents the heatmap showing that replicates are more similar at day 7 than at day 2, and identifying sample DMSO.2a as an outlier in the dendrogram. The Pearson correlation of the normalized intensities was always higher than 0.95 for all duplicates ($0.9541 - 0.9855$) but DMSO at day 2 (0.93), showing a good

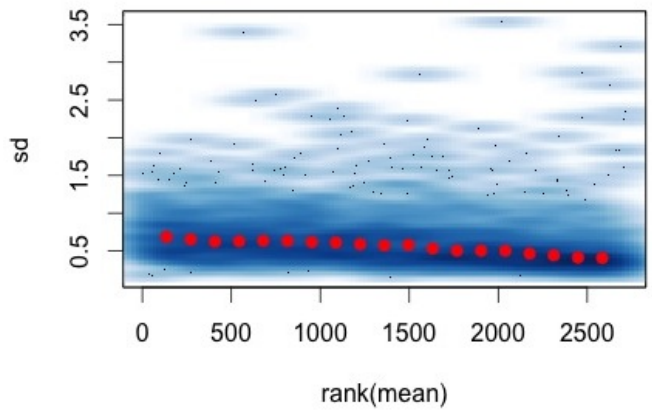

Figure 5.1: Plot of the relationship between standard deviations (sd) and ranked means of the 2720 proteins. For each feature, the plot shows the empirical standard deviation σ_i of the normalized and $glog$-transformed protein levels on the y-axis versus the rank of the mean m_i on the x-axis. The red dots, connected by lines, show the running median estimator of the standard deviation. If there is no variance-mean dependence, then the line formed by the red dots should be approximately horizontal.

reproducibility of replicated experiments and justifying our suspects about this sample.

5.3.3 Outlier detection

The Bland-Altman plot (Altman & Bland, 1983; Martin Bland & Altman, 1986) is a visualization tool used in analyzing the agreement between two different assays in which every point is represented on the graph by the mean of the two measurements on the x-axis and the difference between them on the y-axis. We consider as outlier those values falling outside the boundaries on $\pm$ two standard deviations of all the differences at that specific day. In Fig. 5.3 we see how the DMSO samples at day 2 have a larger number of outliers than all the other replicates at both days, inducing us to leave this sample out from the analysis.

5.3.4 `maSigPro` analysis

The R package `maSigPro` (Conesa et al., 2006) implements a method tailored for the analysis of time-course experiments, it follows a two-step regression strategy to find significant temporal expression changes and differences among experimental groups and then clusters the selected

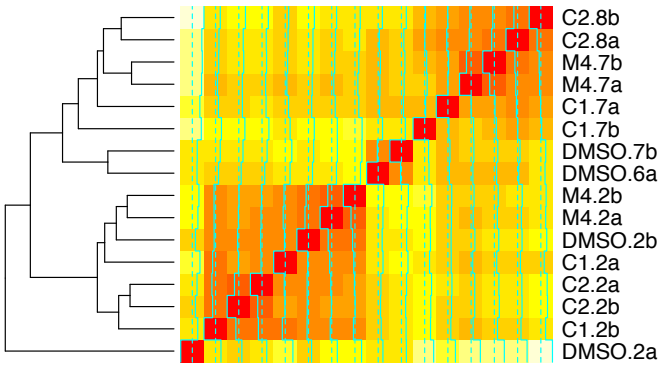

Figure 5.2: Heatmap and dendrogram resulting from the hierarchical clustering on Euclidean distance of the replicates for quality control. From the dendrogram comes the suspect that the biological sample DMSO.2a is an outlier, showing high discordance with its replicate.

proteins with classical clustering algorithms.

Assume that we have I experimental groups ($i = 1, \ldots, I$) identified by a qualitative variable, evaluated at J time points ($j = 1, \ldots, J$). Assume that protein expression is measured for N proteins in R_{ij} replicates. Let y_{ijr} denote the normalized and transformed expression value of each protein under condition ijr. We define $I-1$ dummy variables D to distinguish between each group and the reference group. To explain the evolution of y along time t we consider the following polynomial model, which includes simple time effects and the interactions between dummies and time:

$$y_{ijr} = \beta_0 + \beta_1 D_{1ijr} + \cdots + \beta_{(I-1)ijr} D_{(I-1)ijr} + \\ + \delta_0 t_{ijr} + \delta_1 t_{ijr} D_{1ijr} + \cdots + \delta_{(I-1)} t_{ijr} D_{(I-1)ijr} + \epsilon_{ijr},$$

where:

1. β_0, δ_0 are the regression coefficients corresponding to the reference group;
2. β_i, δ_i are the regression coefficients that account for specific linear differences between the $(i+1)$-th group profile and the reference group profile;
3. ϵ_{ijr} is the random variation associated with each protein owing to all sources other than those that have already been incorporated into the model.

We compute a regression fit for each protein, which involves the estimation of the coefficients β and δ of the described general regression model for each protein using Ordinary Least Squares,

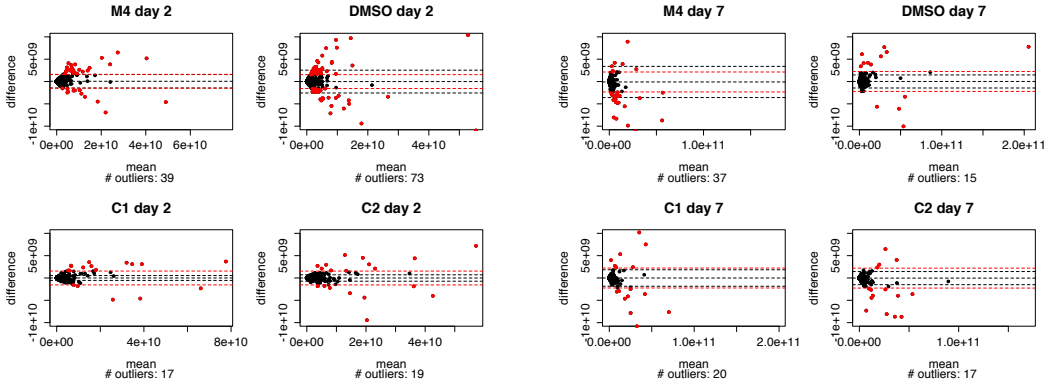

Figure 5.3: Bland-Altman plots for day 2 and 7. Black boundaries are at $\pm$ two standard deviations of the difference between the two replicates from the same experiment, while the red bandwidths are computed with a pooled standard deviation calculated on all differences at the specific day.

and we test:

$$H_0 : \beta_1 = \ldots = \beta_{I-1} = \delta_0 = \delta_1 = \ldots = \delta_{I-1}$$

against

$$H_1 : \exists i \mid \beta_i \neq 0 \vee \delta_i \neq 0, (i = 1, \ldots, I-1).$$

In order to test this hypothesis we compare the full and the reduced models with an ANOVA for nested models, where the reduced model is $y_{ij} = \beta_0 + \epsilon_{ij}$. P-values are then corrected for multiple comparison by applying the False Discovery Rate (Benjamini & Hochberg, 1995). We consider those proteins excluded from the first step as "flat", i.e. behaving with no change both in time and among groups. Once significant proteins have been found, the next step is to apply a variable selection strategy. This is done by inspecting, through backward stepwise regression, the regression coefficients of the protein models not excluded at the first step. The backward method starts by fitting the full model and removing the variable with the largest p-value and fitting the model until all the p-values are significant ($\alpha < 0.05$). The resulting significant coefficients are those that indicate the differences in the profiles. We eliminated those proteins with a low goodness of fit ($R^2 < 0.60$).

In the last step of the analysis, the identified proteins are clustered in order to identify similar behaviors in time. In our analysis we used the Complete-linkage algorithm associated with the

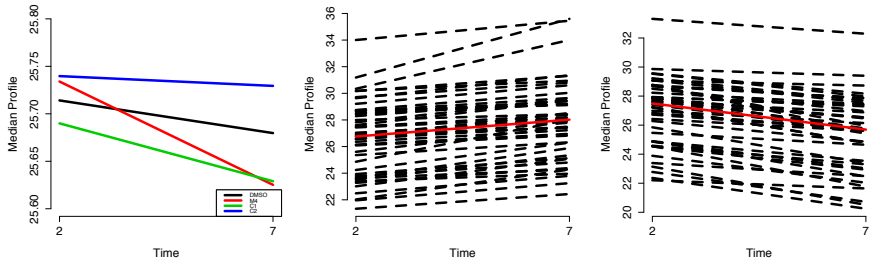

Figure 5.4: Left panel: median profiles of the 1,903 proteins showing no significant trend excluded at the first selection step of `maSigPro` for each experimental condition. The absolute median change in time is less than 0.05 (IQR: < 0.1). Middle and right panels: median profiles of the 94 proteins excluded at the second selection step of `maSigPro` computed over the four experimental groups at each time point, splitted in increasing (middle) and decreasing (right). These proteins only change in time, with a median change (red solid line) of 1.19 (IQR: 0.87) and of 1.42 (IQR: 1.21) for, respectively, the upwards and downwards trends.

correlation distance measure. Finally, statistical over-representation tests of significant proteins were carried out for GO (Ashburner et al., 2000; Consortium et al., 2001) categories against the whole *D. pulex* genome using PANTHER tool (`www.pantherdb.org`), and applying Bonferroni's correction for multiple testing with a p-value threshold of 0.05.

5.3.5 Results

In our study there are four experimental groups ($i = 1, \ldots, 4$), two time points ($j = 1, 2$), two replicates ($r = 1, 2$) for each case ij, and $N = 2,720$ proteins available for statistical analysis. Consequently, we defined three dummy variables D_{M4}, D_{C1}, D_{C2} to introduce into the model the experimental groups as described in the previous section. As we have only two time points, only linear time effects and their interactions with the dummies were modeled.

After the first selection step, only proteins showing statistically significant coefficients between the reference group (DMSO) and any other experimental group and/or time will be kept. Our analysis excluded $1,903$ proteins at this step, which exhibit a flat behavior over time and among comparisons, i.e. C7DR07, E9FQM3, E9FQN6, E9FQP0, E9FQQ8, E9FQR4 etc. The behavior of proteins excluded from the analysis at the first selection step is represented in the left panel of Fig. 5.4. Table 5.1 reports the coefficients of the first five proteins selected after the first step of `maSigPro`. For example, the protein "A1C331" passed the first selection because

Protein	β_0	$\beta_{M4vsDMSO}$	$\beta_{C1vsDMSO}$	$\beta_{C2vsDMSO}$	β_t	$\beta_{t\cdot M4}$	$\beta_{t\cdot C1}$	$\beta_{t\cdot C2}$
A1C331	26.688	0	0	0	0.479	0	0	0
E9FQR8	29.529	-0.32	0.431	-0.534	-0.046	0	-0.227	0
E9FQS5	27.622	0	0	-0.560	0.143	0	0	0
E9FQT2	23.668	1.615	0.526	0	0.210	-0.395	0	0
E9FQT5	26.100	0	0	0	0	-0.212	-0.39	-0.333
...	...	...	...	...	...	...	...	...

Table 5.1: Coefficients of the first 5 proteins among the 817 selected after the first step of `maSigPro`. By analyzing the full table of coefficients we can understand the behavior of proteins. For instance, the first protein passed the first selection step because time has a significant effect on his variation. However, as there is no change in experimental conditions this protein it will be excluded at the second step. The second and fifth proteins will be considered as significant in each comparison, while the third and fourth only in some comparisons.

time has a significant effect on his variation, though the experimental conditions do not and in fact will be excluded in the second step. The second protein "E9FQR8" shows a strong influence of each experimental group against the reference one and also a negative interaction between C1 and time, in fact will be considered significant in each group comparison and kept by both selection steps. While the third protein named "E9FQS5" will be selected at the first selection step but will be identified as significant only in the comparison C2 vs. DMSO. Finally, "E9FQT2" will be considered significant in the comparisons of DMSO against M4 and C1, while "E9FQT5" will be considered significant in each comparison because of the interactions with the time.

After the second step we selected 723 proteins which vary significantly over time and across experimental condition at a FDR significance level of $\alpha = 0.01$ and R-squared threshold equal to 0.6. The FDR provides the expected number of false positives among the selected genes, while the R-squared criterion selects only those proteins statistically well modeled.

It resulted that 94 proteins presented a change only due to time, whose time profile can be visualized in the middle and right panels of Fig. 5.4. In Fig. 5.5 we show how the 723 proteins with a significant expression change can be grouped according to the discovered significant profile differences between experimental groups: only 332 change significantly in all three comparisons. 559 proteins resulted with a statistically significant difference between the experimental groups M4 and DMSO, proving that DMSO has an effect on proteins, while 56 proteins resulted with a statistically significant difference in both the experimental groups C1 and C2 when compared with DMSO, identifying potential biomarkers. Of the 189 proteins identified as significant by Borgatta et al. (2014), 80 (42%) were identified by `maSigPro`.

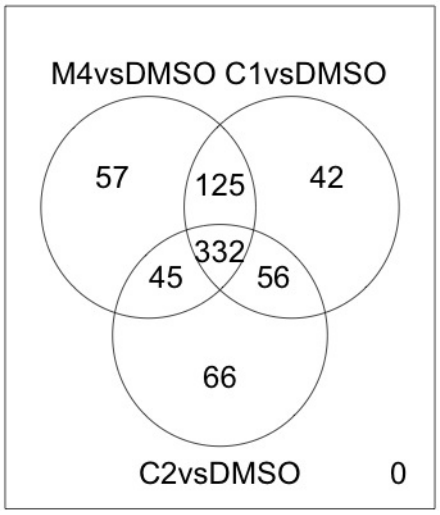

Figure 5.5: Venn diagram of the 723 significant proteins which passed both selection steps for each significant comparison. Only 332 proteins change significantly in all the three comparisons, while others are comparison-specific or shared only by two out of the three possible comparisons. The 559 proteins with a statistically significant difference between the experimental groups M4 and DMSO highlight a potential confounding effect of the solvent, while 56 proteins with a statistically significant difference in both the experimental groups C1 and C2 when compared with DMSO represent possible biomarkers.

Each of these groups was then subjected to clustering, and the number of clusters has been set equal three for each comparison because we aim to find three major groups: increasing over time, decreasing over time, and all the remaining cases. Dendrograms for the cluster analysis of the three comparison groups are reported in Fig. A.25 in Appendix A. Fig. 5.6 shows the experiment-wide protein expression profiles and is useful to evaluate the homogeneity of the obtained clusters, whereas Fig. 5.7 represents the median profile by groups of each cluster and is useful to analyze the actual profile differences between groups. From the median profiles we can always recognize an increasing and decreasing trend in time and also a third group with a less defined structure, though with strong differences between groups. Figs. A.26 – A.28 in Appendix A show the heatmaps of the significant proteins; note that less contamination is present at day 7. Fig. A.29 in Appendix A reports the heatmap of the proteins excluded from the analysis.

Among those 723 proteins, 649 were annotated with PANTHER database. Statistical overrepresentation tests of the annotated proteins were carried out for Gene Ontology (GO) categories in Biological Process (BP) and Molecular Function (MF) classes. In the BP and MF classes, respectively 44 and 20 GO categories showed significant enrichment of tested proteins compared to the whole *D. pulex* genome. In particular, the first ten MF classes identified as overrepresented are: catalytic activity, oxidoreductase activity, structural constituent of ribosome, structural molecule activity, binding, aminoacyl-tRNA ligase activity, translation regulator activity, translation factor activity, lyase activity, and RNA binding. While the ten most overrepresented BP classes are: metabolic, primary metabolic, and protein metabolic processes, translation, carbohydrate metabolic process, cellular amino acid metabolic process, proteolysis, regulation of translation, cellular process, and nuclease-containing compound metabolic process.

5.3.6 Conclusion

In this section we analyzed the response of proteins in organisms under stress, dividing the population in four experimental groups and measuring their protein production at day 2 and 7. We used a two-step regression strategy to identify significantly differentially expressed proteins and to study their behavior, finding that only 723 proteins out of the $2,720$ initially considered have to be retained, as they change in time and across conditions. We grouped proteins on the basis of their profile in three clusters, finding that there was always an increasing, a decreasing, and a less well defined pattern over time.

These results are at the beginning of a deeper biological investigation about function and

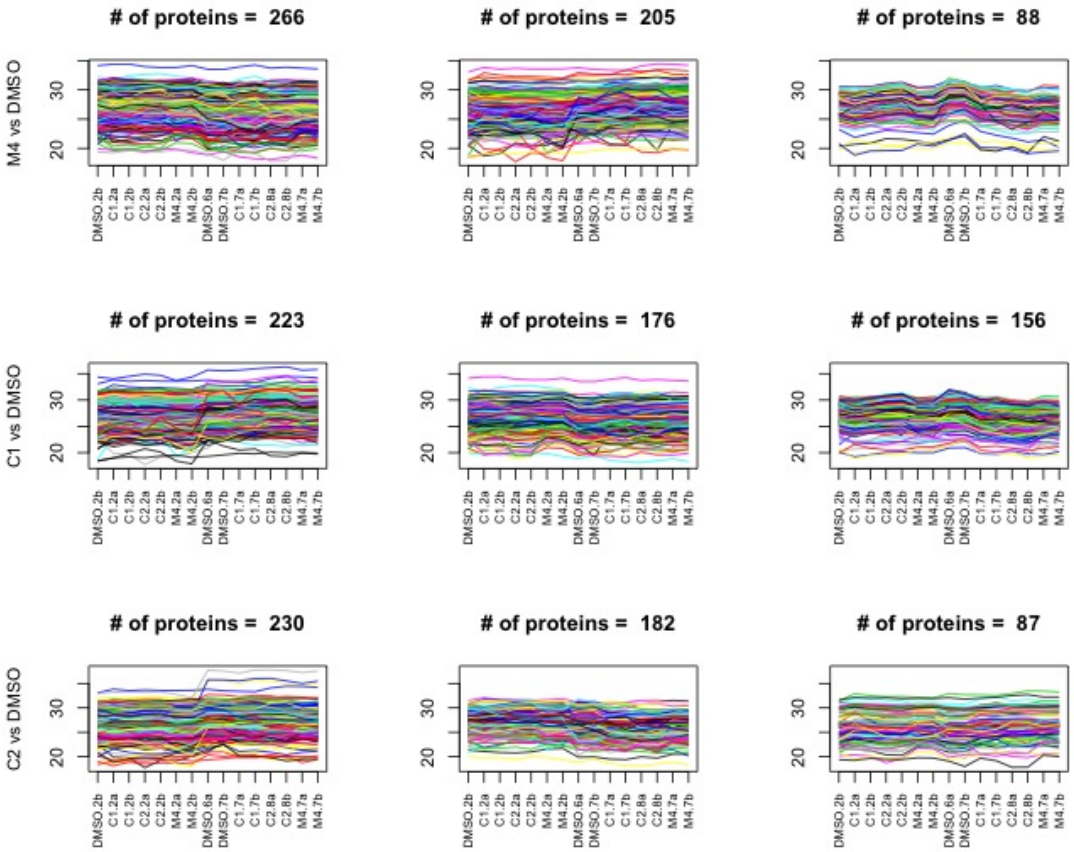

Figure 5.6: Data visualization by clusters obtained with `maSigPro` for each comparison on the significant proteins obtained after the two-step regression. The cluster analysis has been performed using Pearson's correlation as distance measure and Complete-linkage as algorithm. In each of the 9 panels the number of proteins contained in the cluster is always reported, as well as the name of the sample on the x-axis. Every row reports the names of the experimental groups compared.

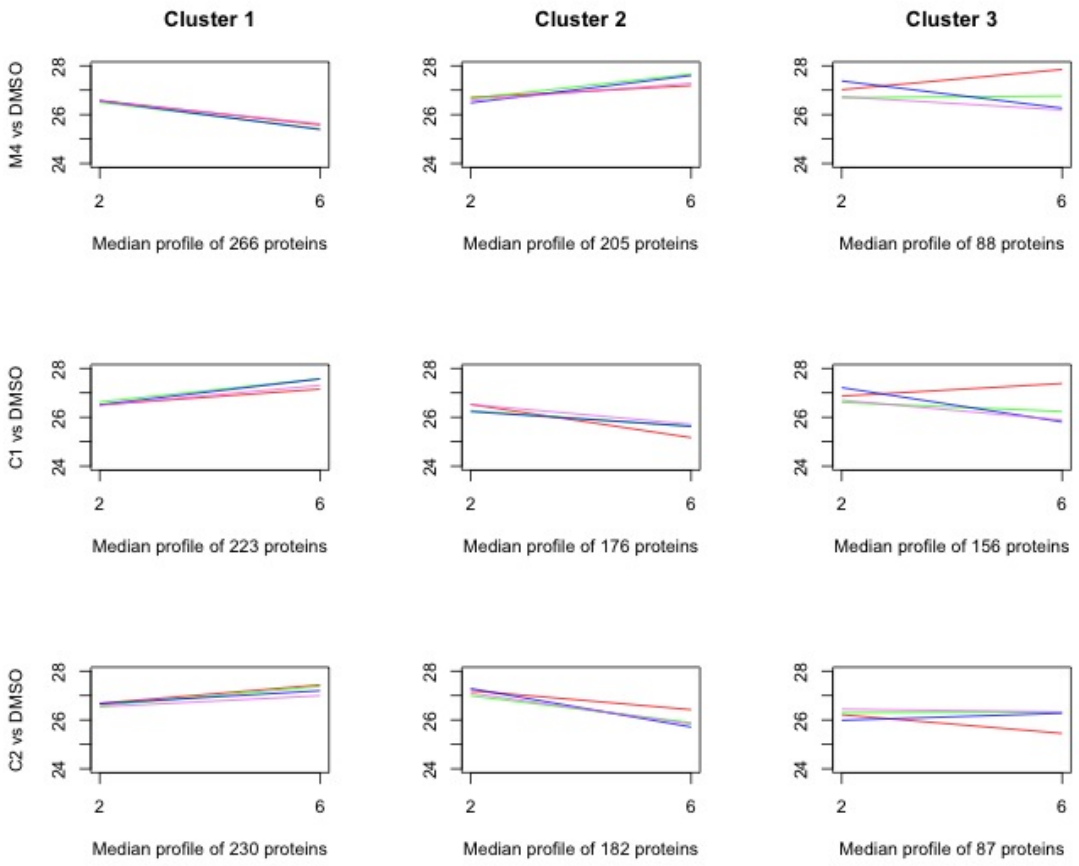

Figure 5.7: Median expression profiles of the selected proteins splitted in three clusters and significant profile differences between experimental groups obtained with `maSigPro`. The color of the line indicates the experimental group of belonging. Red lines indicate DMSO profiles, green lines represent C1 profiles, blue lines C2 profiles, while the violet ones represent M4 profiles. The cluster analysis has been performed applying Complete-linkage on Pearson's correlation. Under each panel the number of proteins contained in the cluster is reported. Every row reports the names of the experimental groups compared. We can always recognize an increasing and decreasing trend in time and also a third group with a less defined structure, though with strong differences between groups.

meaning of these patterns, and a comparison between our results and those coming from Borgatta et al. (2014) needs to be performed in order to confirm the biological results and to find differences between the two methods. However, with only two time points available we could only model a linear regression, which may not be enough in order to detect all the significant differences between time points and experimental groups.

The most important result for our purposes was that analyzing these data we realized some major issues of using `maSigPro` that motivated us to focus on the clustering of short-length time-course data: i) results are strongly sensible to the polynomial degree chosen, ii) `maSigPro` does not suggest a suitable number of clusters, iii) using classical clustering approaches it does not take into account the temporal information, considering time points as uncorrelated, iv) in very short-length time-course data (as in our case, when only two time points are available) it is only possible to use a linear regression model that may not be enough for detecting all the significant differences, v) missing data are not allowed, vi) the model is built to detect flat but not outlying behaviors, vii) the model is not able to suggest if clustering that particular set of data is appropriate or not. In the next section we are going to present our proposal, which aims to overcome these limitations.

5.4 Cross-clustering

In order to cluster short-length time-course data, we investigate an approach that considers the temporal evolution of gene expressions as curves, fitting them by particular splines. After estimating the coefficients of these non parametric functions, we identify groups of genes applying the Cross-clustering algorithm on them. Using splines we can handle missing data and take into account the temporal information contained in the data even in case of no replicates. Applying the Cross-clustering algorithm we are able to automatically identify the number of clusters, detect outliers, and suggest only one cluster in the case in which clustering a particular set of data is not appropriate. Although linear models could be used for this purpose, they are often too restrictive to capture the underlying phenomenon. Indeed, the flexibility of tools such as splines permits to reach satisfactory results with a limited number of coefficients.

5.4.1 B-splines

Following the proposal of Abraham et al. (2003), we suggest to use B-splines to fit the curves. B-splines have the great computational advantage of being estimated from recursive equations

(Wasserman, 2006). We assume that the measurements of the curves G_i for gene $i = 1, \ldots, n$, contain errors which can be thought as added to the underlying smooth evolution of gene expression. The model can be written as follows:

$$y_{it} = G_i(x_t) + \epsilon_{it},$$

where $t = 1, \ldots, T$ is the time point, x_t is the gene expression at time t, and ϵ_{it} are independent random errors. The aim of the procedure is to remove this noisy part, focusing on the smooth part of interest. After summarizing each curve by a few coefficients, which capture the smooth part of the process with enough flexibility, we will have to cluster these coefficients.

First, we fit each observation $\{x_t, y_{it}\}_{t=1}^{T}$ by a regression spline function in order to estimate G_i. Let $x \in [a, b]$ and let $(\xi_0 =)a < \xi_1 < \xi_2 < \cdots < \xi_K < b(= \xi_{K+1})$ be a subdivision of K distinct points on $[a, b]$; these points are called *knots*. The spline function $s(x)$ is a polynomial of degree d (or order $d + 1$) on any interval $[\xi_{i-1}, \xi_i]$, and has $d - 1$ continuous derivatives on the open interval (a, b). For a fixed sequence of knots $\xi = (\xi_1, \xi_2, \cdots, \xi_K)$ these splines are linear space functions with $K + d + 1$ free parameters. A useful basis $(B_1, \cdots, B_{K+d+1})$, for this linear space is given by B-splines (Curry et al., 1966). We can write a spline as follows:

$$s(x, \beta) = \sum_{l=1}^{K+d+1} \beta_l B_l(x),$$

where $\beta = (\beta_1, \cdots, \beta_{K+d+1})'$ is the vector of spline coefficients and the symbol $'$ indicates transposition.

In many applications, a linear combination of third-degree B-splines is flexible enough to investigate the non-linear evolution over time of several phenomena. With fixed knots, the least-squares spline approximation is equivalent to a linear problem. Let $\{x_t, y_{it}\}_{t=1}^{T}$ be a regression-type data set of T measurements of the curve G_i ranging over $[a, b] \times \mathbb{R}$. As we use the same degree and vector of knots, the same basis functions $(B_1, \cdots, B_{K+d+1})$ are used for all the curves. Thus, each coordinate $\hat{\beta}^i$ has the same meaning for each curve G^i. Denote by $B = \{B_l(x_t)\}_{t=1,\cdots,T}^{l=1,\cdots,K+d+1}$ the corresponding $T \times \{K + d + 1\}$ matrix of sampled basis function, and by y_i the vector $(y_{i1}, \cdots, y_{iT})'$. The spline coefficients are estimated by:

$$\hat{\beta}^i := \arg\min_{\beta^i} \frac{1}{T} \sum_{t=1}^{T} \left[\left(y_{it} - s(x_t, \beta^i) \right) \right]^2 = [(B^i)' B^i]^{-1} (B^i)' y^i,$$

where $[B'B]^{-1}$ is the inverse of $B'B$. Then, $\hat{s}^i(x) := s(x, \hat{\beta}^i)$ estimates $G^i(x)$. The set of curves $\{G^1, \cdots, G^n\}$ is summarized by $\{\hat{\beta}^1, \cdots, \hat{\beta}^m\}$, a set of vectors in $\mathbb{R}^{K+d+1}$.

In our analysis we used the `bs` function of the R package `splines`.

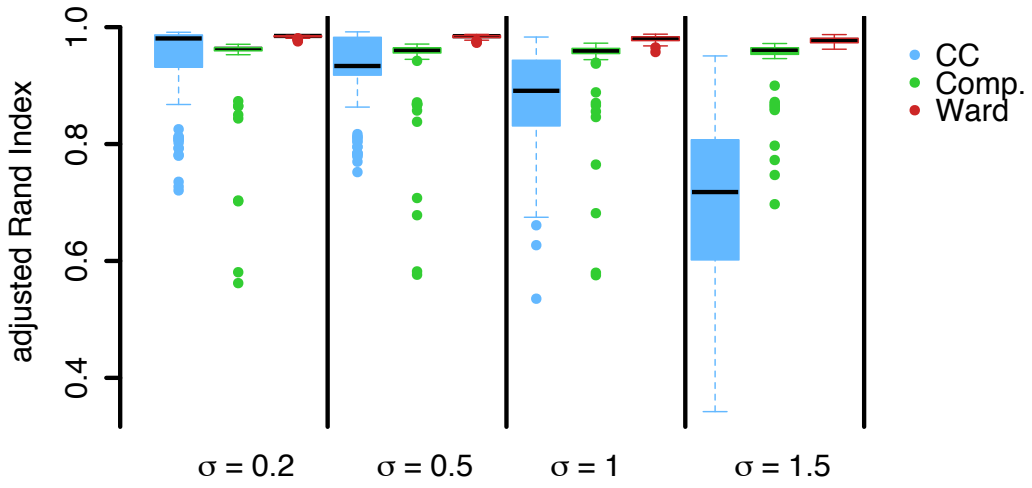

Figure 5.8: Boxplots of adjusted Rand Index (ARI) resulting from Cross-clustering (CC) on B-spline coefficients, Complete-linkage (CL) and Ward's algorithm with $\sigma = 0.2, 0.5, 1, 1.5$ and Euclidean distance on simulated data. The ARI measures the agreement between two partitions (the higher, the better). This figure proves that CC does not perform better than reference methods, as the distribution of ARIs over 100 simulations has a large variability and is strongly sensible to the increase of σ.

5.4.2 Numerical assessment

We applied the CC, Ward, and CL algorithms on the simulated data (§ 3.1), now interpreting the five measurements as time points instead of samples. The choices of the number of clusters, CC parameters, and distance metric are made as before (§ 4.4).

Results in terms of ARI are shown in Fig. 5.8, where CC is compared with the methods that constitute it. This plot shows that CC applied on B-splines coefficients does not perform better than the reference methods, as the distribution of ARIs over 100 simulations has a large variability and is strongly sensible to the increase of the variability in the data. Fig. A.30 in Appendix A shows a very similar performance also when the chosen distance metric is Chebychev.

In order to compare also sensitivity and specificity, Fig. 5.9 reports results regarding the

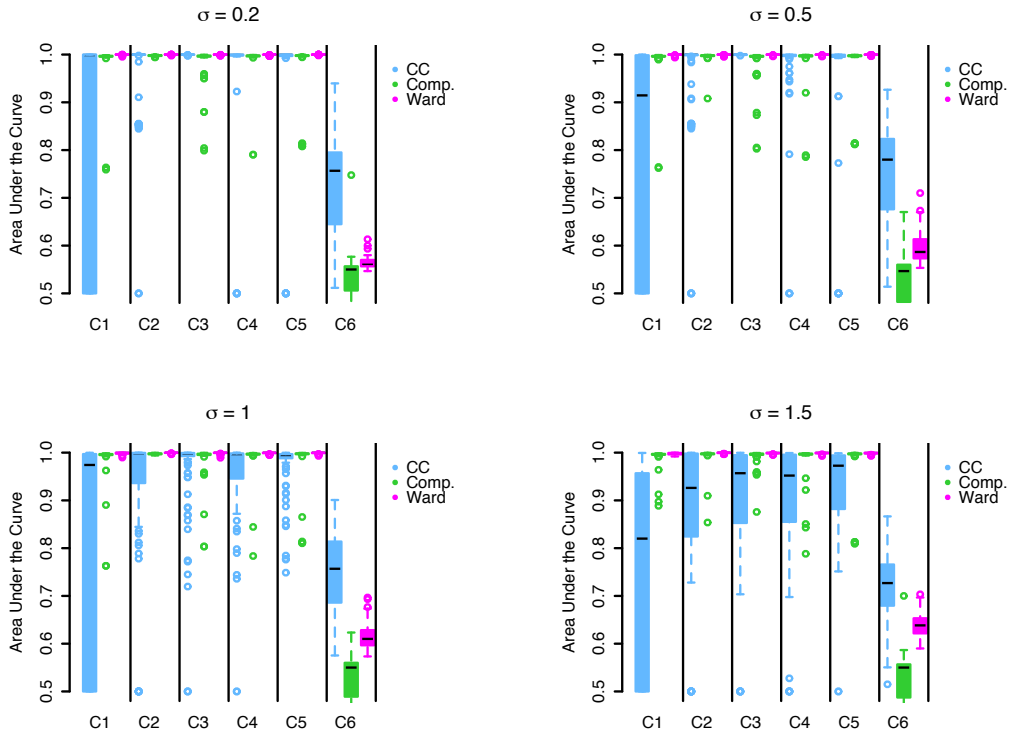

Figure 5.9: Boxplots of the Area Under the Curve (AUC) resulting from Cross-clustering (CC) on B-spline coefficients, Complete-linkage (CL) and Ward's algorithm with $\sigma = 0.2, 0.5, 1, 1.5$ and Euclidean distance on each of the six clusters of the simulated data. The last cluster is the outliers one. The AUC is a measure of sensitivity and specificity (the higher, the better). This figure proves that CC does not perform better than reference methods, as the distribution of AUCs over 100 simulations has a large variability and is strongly sensible to the increase of σ.

AUCs in each cluster. This figure proves that CC does not perform better than the reference methods, as its distribution of AUCs over 100 simulations is generally lower than others, has a large variability and is strongly sensible to the increase of the variability in the data. Fig. A.31 in Appendix A shows a similar performance also when the chosen distance metric is Chebychev.

5.5 Discussion

Clustering short-length time-course data is a highly challenging goal, as many issues has to be taken into account. Recently, many scientific efforts have been dedicated to this purpose

but none of the proposed methods so far is well suited for short-length time-course data and simultaneously takes into account the temporal information, the possibility of missing values and outliers, the lack of replicates, is able to identify a reasonable number of clusters, and to suggest if clustering a particular set of data makes sense. We started from analyzing a very short-length time-course data (only two time points) with a very well-known analysis tool, `maSigPro`, and discovered its weaknesses. This motivated us to assess whether the CC algorithm of Chapter 4 may overcome all the discovered limitations.

Our proposal was to apply the CC algorithm, presented in Section 4.2 for static data, on B-spline coefficients. Using splines we can handle missing data and unevenly sampled observations, also taking into account the temporal information contained in the data even in a case of no replicates. Using the CC algorithm we are able to automatically identify the number of clusters, detect outliers, and suggest only one cluster in the case in which clustering a particular set of data is not appropriate.

The validation of CC applied on B-spline coefficients on simulated data resulted in a worst performance than the methods that constitute CC, with distributions of ARIs and AUCs varying widely and strongly sensible to the increase of data variability. In future some modifications of our proposal could be considered, such as using different types of splines or applying the CC algorithm on specially tailored distances for short-length time-course data, such as the STS and Cross Correlation distances.

Chapter 6

CAGE data analysis

6.1 Literature overview and motivation

Cap-Analysis Gene Expression (CAGE) is a technology recently developed by Carninci and colleagues (Shiraki et al., 2003; Carninci, 2006), based on the Cap Trapper method (Carninci et al., 1996), which allows the genome-wide survey of promoters and quantification of gene activity.

Briefly, small fragments (usually 20–21 nucleotides long) from the very beginning of mRNA are extracted, reverse-transcribed to DNA, PCR amplified and sequenced. The output of CAGE is a set of short nucleotide sequences (called *tags*). The first step in the analysis of deep-sequencing expression data is the mapping of the (short) reads to the genome from which they derive. CAGE tags are usually mapped to a reference genome using a novel alignment-algorithm called *Kalign2* (Lassmann et al., 2009) that maps tags in multiple passes. Once the RNA sequence reads or CAGE tags have been mapped to the genome, is obtained a collection of positions for which at least one tag is present. The CAGE tags whose genomic mapping overlapped by at least 1bp can be group using clustering algorithms and then generate a list of discrete genomic regions called *peaks*.

CAGE is a unique tool in the analysis of transcriptional regulatory networks. The analysis of promoters is essential to understand the functioning's of gene networks which regulate life phenomena at a molecular level. There are several advantages that deep-sequencing offers. One of these is that large scale full-length cDNA sequencing efforts have made it clear that most of the genes are transcribed in different isoforms using alternative TSSs. With the CAGE technology it is now possible to locate an exact TSS in the genome and also to quantify the expression level of

each individual TSS (Maeda et al., 2008). In the Functional Annotation of Mouse (FANTOM) 3 study, this technology was used to map TSSs in the mouse genome, leading to the identification of more than 230,000 promoters (Carninci, 2006).

However, whereas for microarrays technology a wide bulk of statistical methods is available, CAGE technology is relatively new and the corresponding statistical methodology is still quite poor. For instance, so far there is only one R package designed to manipulate, analyze and visualize CAGE data: `CAGEr` (Haberle et al., 2013) as compared to the extensive Bioconductor project. This is the reason why we decided to focus on the analysis of CAGE data in collaboration with the Center for Genome Research of the University of Modena and Reggio Emilia.

6.2 Preliminary analyses

6.2.1 Normalization

Normalization strategies aim at removing the effect of unknown or irrelevant sources of variability from the raw data. When data are normalized, they are converted to quantities that can be compared among different types of experiments and platforms. While the principle of normalization remains the same, each expression platform has its own set of background noise to be corrected for.

The simplest method proposed in literature for the normalization of CAGE data is the "tags per million" (TPM) normalization (Carninci, 2009). We first extract the CAGE tags from the raw sequence reads and count how often each tag occurs, these absolute tags are not directly comparable because of the dependence on the total number of reads. For this reason tag counts are expressed in TPM, which is defined as the expected count for a particular tag if we had extracted one million raw CAGE tags:

$$TMP = 1,000,000 \times \frac{AbsoluteTagCount}{TotalTagCount}.$$

The relative error due to sampling noise is approximately

$$RelativeError = \frac{1}{\sqrt{AbsoluteTagCount}}.$$

This approach was however criticized because it does not take account of possible systematic variations among samples, due, for example, to the quality of the RNA, to the variation in library production, or even to biases of the sequencing technology used. In order to address this issue,

Balwierz et al. (2009) proposed a new normalization strategy based on the fact that many CAGE datasets follow a power-law distribution.

A quantity x obeys a power-law distribution if it is drawn from a probability distribution $p(x)$ of the form:

$$p(x) \propto x^{-\alpha}, \tag{6.1}$$

with $x > 0$ and where $\alpha > 1$ is a fixed parameter of the distribution known as the *exponent* or *scaling parameter*, which typically lies in the range $2 < \alpha < 3$. As this probability distribution diverges to infinity as $x \to 0$, it cannot hold for all $x \geq 0$, but there must be a lower bound $x_{min} > 0$. It is often useful to consider as reference the complementary (or reverse) cumulative distribution function (CCDF) of a power-law distribution, which is defined to be $S(x) = P(X > x) = 1 - F(x)$. If we take the logarithm of both sides of Eq. (6.1) we have that

$$\ln p(x) = \alpha \ln x + \text{constant}, \tag{6.2}$$

implying that it is represented as a straight line on a doubly logarithmic plot. Eq. (6.2) is fully determined by the slope α and the total number of tags T, which together with α determines the value of the constant. Even if this method is known for generating systematic errors quite often, a first simple way to check the suitability of the power-law distribution for our data (§ 3.5) is to plot the CCDFs on doubly logarithmic axes. In Fig. 6.1 we can see that the empirical CCDFs of our three samples on a logarithmic scale look similar though very different from the oblique grey dashed line they are supposed to follow. For this reason we decided to investigate further if the assumption of a power-law distribution is appropriate in our case.

First of all, we had to estimate the lower bound of the support of the power-law distribution, x_{min}, in order to consider only those points for which the reference distribution is valid. We have chosen the value of $\hat{x}_{min}$ that minimizes the maximum distance between the CCDFs of the data and the fitted model, as recommended by Clauset et al. (2009):

$$D = \max_{x \geq x_{min}} \mid \hat{S}(x) - S(x; \hat{\alpha}) \mid,$$

where $\hat{S}(x)$ is the empirical CCDF of the data for the observations with value at least x_{min}, $S(x; \hat{\alpha})$ is the CCDF for the power-law model that best fits the data in the region $x > x_{min}$, and D is the Kolmogorov-Smirnov (KS) statistic.

Second, we estimated α with the method of maximum likelihood, which gives accurate parameter estimates for large sample sizes (Cox & Barndorff-Nielsen, 1994; Wasserman, 2004).

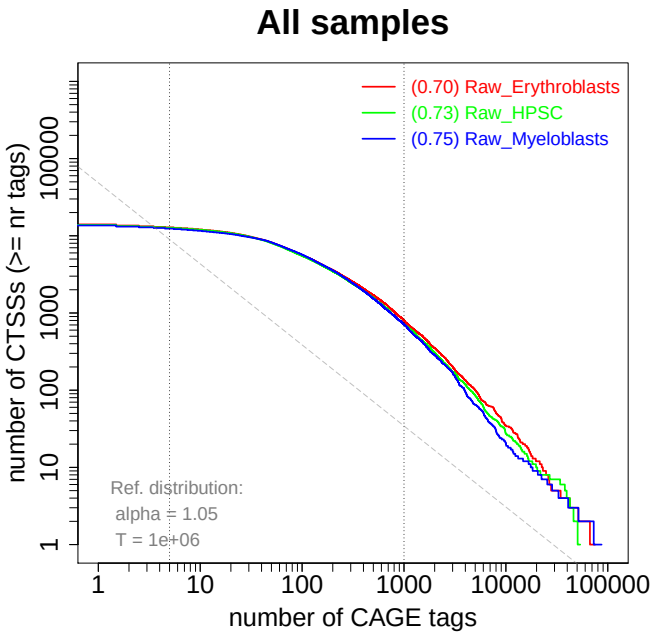

Figure 6.1: Complementary cumulative distribution functions for the number of different CAGE detected transcription starting sites (CTSSs) positions in our CAGE dataset that have at least a given number of tags mapping to them. Both axes are on a logarithmic scale. The x-axis represents the number of CAGE tags, while on the y-axis are reported the number of TSSs that are supported by at least that number of tags. The three curves correspond to the distributions of the three samples. Values in brackets are values of $\hat{\alpha}$ estimated with the maximum likelihood method for each sample. Grey vertical dashed lines determine the specified range in which the power-law distribution has been fitted (default values of the range from 5 to 1000). The oblique gray line indicates the referent power-law distribution, which is defined based on the parameters written in the bottom left part of the plot: in this case the slope (α) in the log-log representation is 1.05, while the total number of tags T is $1,000,000$.

	α	x_{min}	logLik	KS.stat	KS.p
1	2.45	1681	-3436.08	0.03	0.95
2	2.42	1022	-5780.32	0.03	0.40
3	2.47	2296	-2682.99	0.03	0.99

Table 6.1: Results of the Kolmogorov-Smirnov test performed with the R package `igraph` on the raw data. The parameter α is the exponent of the fitted power-law distribution , x_{min} the lower bound of the support (only values larger that x_{min} were used from the data), logLik is the log-likelihood calculated for the fitted parameters, KS.stat the test statistic (smaller scores denote a better fit), while KS.p is the p-value resulting from the test.

Finally, as with these two estimated parameters we still do not know if the power-law is a good model for the data, we needed to perform a goodness-of-fit test, which generates a p-value that quantifies the plausibility of the hypothesized power-law distribution. For testing this hypothesis we performed a KS test. In Tab. 6.1 we show the resulting p-values of the tests and the estimated parameters. The null hypothesis that the data were drawn from the fitted power-law distribution can never be rejected, but it is important to notice that $\hat{x}_{min}$ is always very high, while our data have a lot of zero values. Otherwise, if we set x_{min} equal to 1, the p-values are always equal to zero, suggesting that our data are not power-law distributed. For this reason the normalization of raw CAGE tag counts was performed with the simple TPM.

An analysis of the correlation between normalized samples was then performed on the data. Fig. 6.2 shows an higher correlation between HPSC and Erythroblasts samples ($r = 0.90$), than in other comparisons.

6.2.2 Clustering

TSSs localized in close proximity to each other lead to a larger promoter region, i.e. a set of transcripts that are regulated by the same promoter elements. For this reason it is useful to spatially cluster TSSs into larger units called tag clusters (TCs), defined as genomic regions of overlapping tags (at least 1 base pair) that map to the same strand of a chromosome and that correspond to individual promoters.

We performed a simple distance-based clustering in which two neighbouring TSSs are joined together if they are closer than some specified distance, that we first fixed at 20bp. Before clustering we filtered out low-fidelity TSSs, i.e. the ones supported by less than 2 normalized tag counts in all samples. The final set of TCs will exclude clusters with only one TSS, the so-

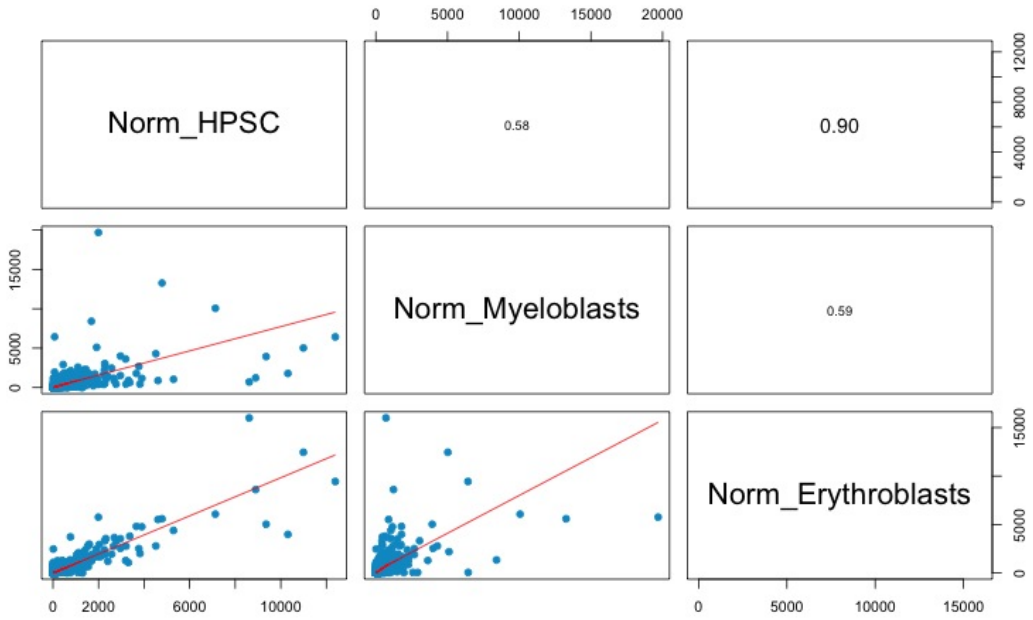

Figure 6.2: Pairwise scatter plots (lower panels) and Pearson's correlation (upper panels) of normalized CAGE tag counts per TSS. The highest correlation resulted to be the one between HPSC and Erythroblasts samples ($r = 0.90$).

called *singletons*, unless the signal (the transcription initiation as measured by CAGE) is larger than 5. As we use a distance of 20bp, only clusters with one TSS were obtained; we changed the maximum distance to 50bp and to 100 bp, obtaining a maximum number of TSSs clustered together of 2 and 4, respectively.

We then decided to use the so-called "Paraclu algorithm", a parametric clustering method based on the density of the signal (Frith et al., 2008). Here the minimal stability of the cluster, defined as the ratio between maximal and minimal density value for which a cluster is maximal scoring, was set equal 1. We obtained a clustering with a maximal counts of the number of TSSs within the cluster of 4.

As TCs are created for each sample and they might not coincide perfectly within the same promoter region, we aggregated TCs into a set of consensus clusters (promoter). We can do this aggregating TCs from all samples into a single set of non-overlapping consensus clusters. The expression clustering at the level of entire promoter was performed using SOM and K-means only on promoters with normalized CAGE signal ≥ 15 in at least one sample.

In order to decide which number of clusters was the more reasonable in our case for K-means, we let it vary from 2 to 20 and computed the average silhouette width (ASW) for each of the resulting partitions, choosing the number of clusters that maximized it. Fig. 6.3 and 6.4 show the ASW resulting from the K-means algorithm applied on CAGE detected transcription start sites (CTSSs) and on consensus clusters, respectively. The maximum ASW value is achieved in both cases with 2 clusters, but it is very low, being around 0.10. In SOM the dimensions of the grid has to be set. In Tab. 6.2 results coming from different dimensions (xDim, yDim) of the grid are shown and seem to suggest that no clustering is appropriate. From all the clustering performed it is clear that our data have a high variability, making the choice of a good clustering rather challenging.

6.3 Comparisons

6.3.1 Comparison across chromosomes

It is interesting to analyze how CAGE peaks are distributed across chromosomes. As it is well known that chromosomes have different lengths, Fig. A.32 in Appendix A reports the relative frequencies of CAGE peaks compared to the length of each chromosome. The chromosome with the largest number of peaks is chromosome 19, followed by chromosome 7, while the mitochondrial chromosome (*chrM*) and the sex chromosome X have very low relative frequencies.

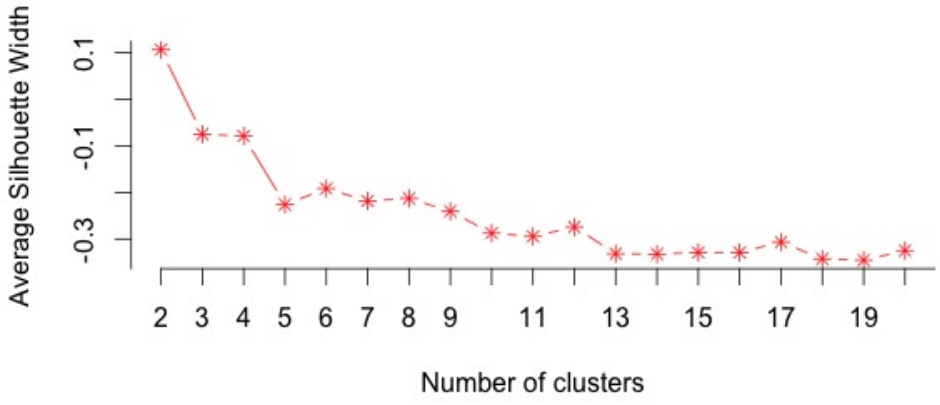

Figure 6.3: Average silhouette width (ASW) for changing the number of clusters (from 2 to 20) for the K-means algorithm applied on the CAGE detected transcription start sites (CTSSs).

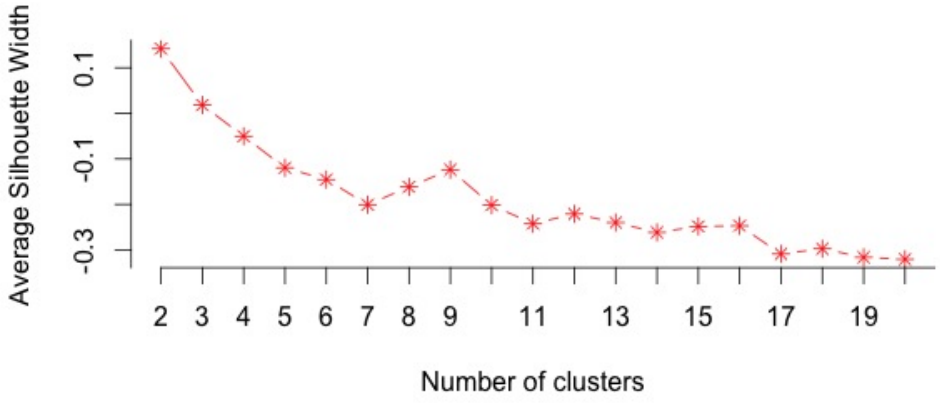

Figure 6.4: Average silhouette width (ASW) for changing the number of clusters (from 2 to 20) for the K-means algorithm applied on the consensus clusters.

x	y	CTSS	consensusClusters
2	2	-0.1507085	-0.1218571
3	3	-0.2717451	-0.267085
4	4	-0.3589982	-0.3509756
5	5	-0.3686897	-0.4163353
1	2	0.03543591	0.04197137
1	3	-0.08423169	-0.08322987
2	3	-0.2258189	-0.3020011
2	4	-0.295827	-0.2807868
3	2	-0.2468775	-0.2646895

Table 6.2: Average silhouette width (ASW) for partitions resulting from the SOM algorithm, with different dimensions (x, y) of the provided grid, both applied on CAGE detected transcription start sites (CTSSs) and consensus clusters.

We performed a two-sided test for equal proportions, and we rejected the null hypothesis at the significance level $\alpha = 0.05$ ($\chi^2_{23} = 3659.84$, p-value $< 2.2e^{-16}$).

6.3.2 Comparisons among cell lines

Identification of differentially expressed CAGE promoters in three cell lines

We performed a χ^2 test on each peak for every chromosome, under the null hypothesis of equal proportions of counts forming the peak in different cellular lines. We eliminated peaks with counts too low (< 10) to avoid a possible inaccurate χ^2 approximation. All p-values were corrected for multiple testing with the BH method (Benjamini & Hochberg, 1995) and the significance level was set to $\alpha = 0.05$. We visualized the profile of each of the 24 chromosomes for the three cell lines (HPSC/Myelo/Erythro) at a time. Plots are available in Appendix A. As we can see in Tab. A.13, for each chromosome we reject $H_0 : p_1 = p_2 = p_3$ in about the 40% of the comparisons.

Identification of differentially expressed CAGE promoters in two cell lines

As there are significant differences in the number of peaks for each chromosome across the three cellular lines, we investigated further this aspect testing the null hypothesis of equal proportions of counts forming the peak in each pair of cell lines. Tab. A.14 in Appendix shows the percentage

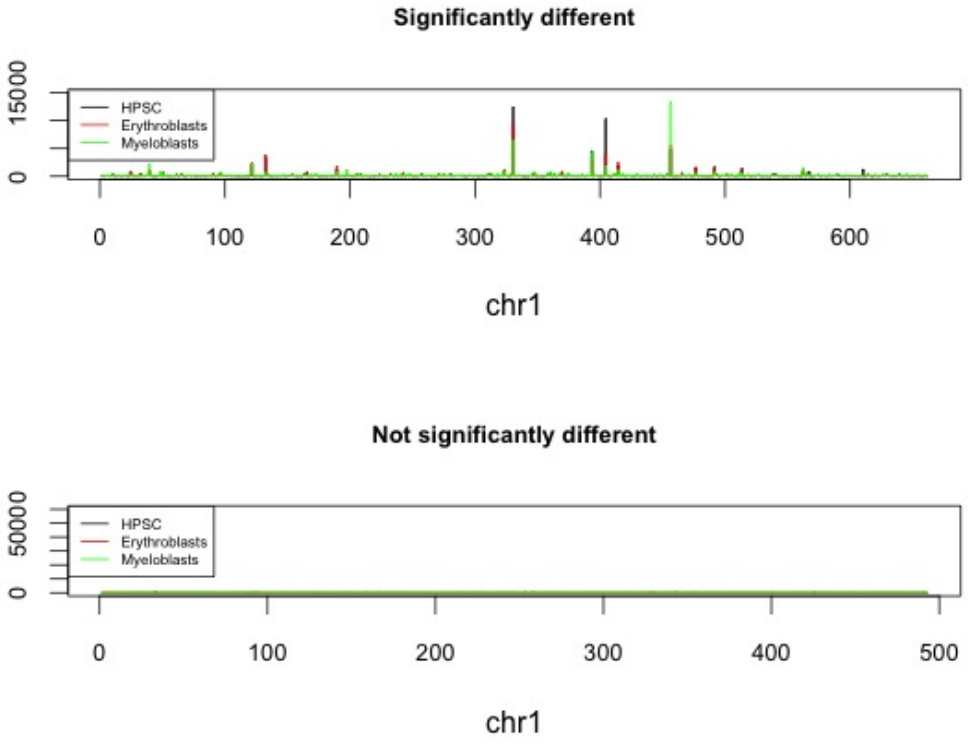

Figure 6.5: Graphical representation of peaks in chromosome 1 in the three cellular lines. Top: peaks detected as significantly different; bottom: peaks detected as not significantly different by the χ^2 test. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

of null hypothesis $H_0 : p_1 = p_2$ rejected, which is always smaller in the comparison between HPSC and Erythroblas than in the others, as expected after our correlation analysis.

Audic & Claverie (1997) proposed a Bayesian method based on the Poisson distribution that can be used in our case. They proved that the probability of observing y number of tags in condition B, given that x tags were observed in condition A, given that N_A and N_B are the total number of raw tags for conditions A and B, respectively, is:

$$P(y \mid x) = \left(\frac{N_B}{N_A}\right)^y \frac{(x+y)!}{x!y!\left(1+\frac{N_B}{N_A}\right)^{x+y+1}},$$

under the null hypothesis of equal expression of the considered gene in the two conditions. The smaller the probability, the more differentially expressed is the gene over the two conditions.

We computed the probabilities for every peak and we adjusted then with FDR for multiple comparisons separately for each comparison. Tab. A.15 in Appendix A reports how many peaks were found significantly different ($p < 0.05$) in each comparison and for every chromosome with the Audic and Claverie test, while Tab. A.16 in Appendix A reports the intersection between these results and those coming from the proportion tests.

6.3.3 Power analysis

We performed a power analysis in order to discover which sample size is able to detect a significative difference with a certain power of the test. For this analysis the R package `pwr` was used.

In the comparison among three cellular lines we fixed the power of the test to $1 - \beta = 0.8$, and the type I error probability $\alpha = 0.05$. We also define the effect size w as:

$$w = \sqrt{\sum_{i=1}^{m} \frac{(p_{0i} - p_{1i})^2}{p_{0i}}},$$

where p_{0i} is the cell probability in the i-th cell under H_0 while p_{1i} is the cell probability in the i-th cell under H_1. According to Cohen (2013), who suggested to consider values of the effect size $w = (0.1, 0.3, 0.5)$ for, respectively, small, medium, and large effect sizes, we let w change in the range $[0.1, 1]$.

In the pairwise cell lines comparisons α and β were fixed as before, while the effect size h was defined as:

$$h = 2arcsin(\sqrt{p_1}) - 2arcsin(\sqrt{p_2}).$$

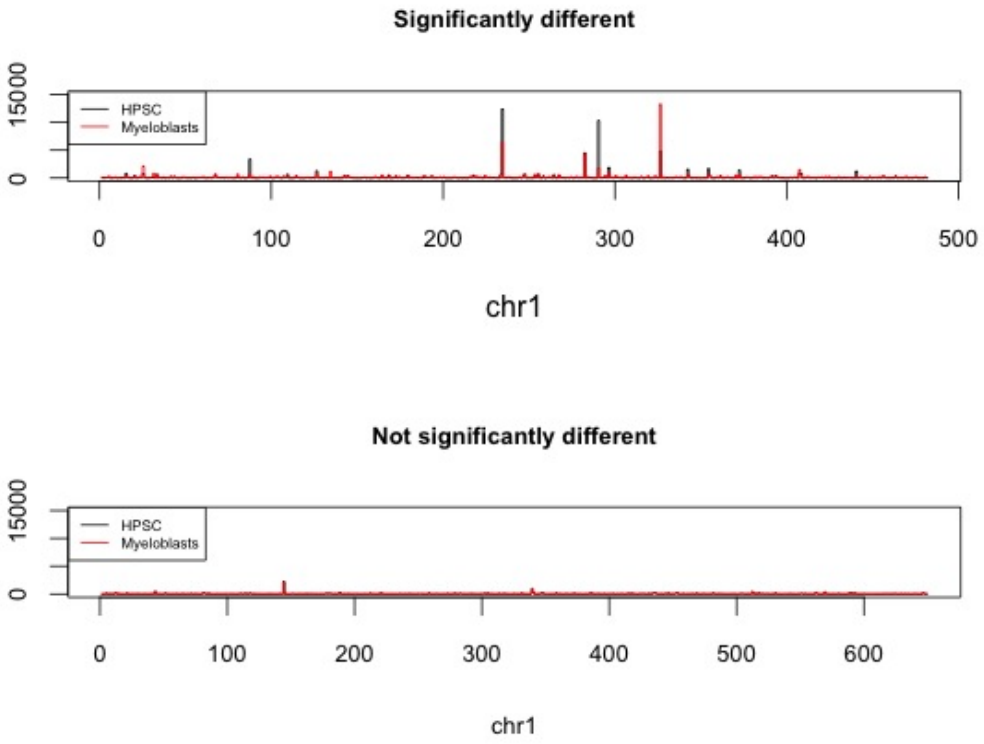

Figure 6.6: Graphical representation of peaks in chromosome 1 in the cellular line HPSC versus Myeloblasts, divided by peaks considered significantly different by the test of equal proportions. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

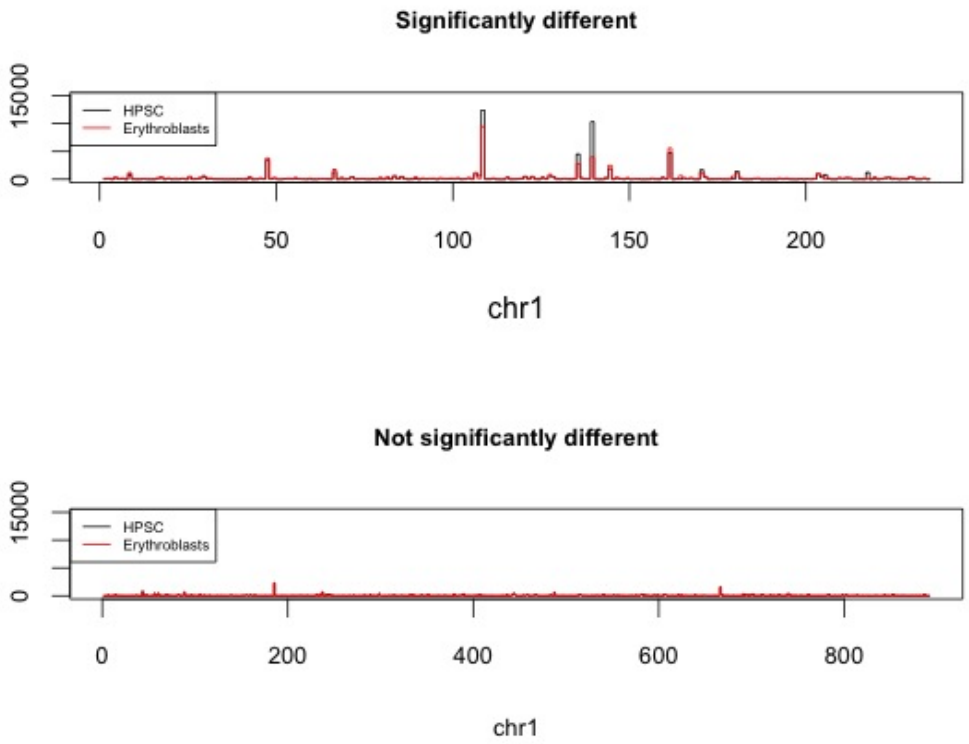

Figure 6.7: Graphical representation of peaks in chromosome 1 in the cellular line HPSC versus Erythroblasts, divided by peaks considered significantly different by the test of equal proportions. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

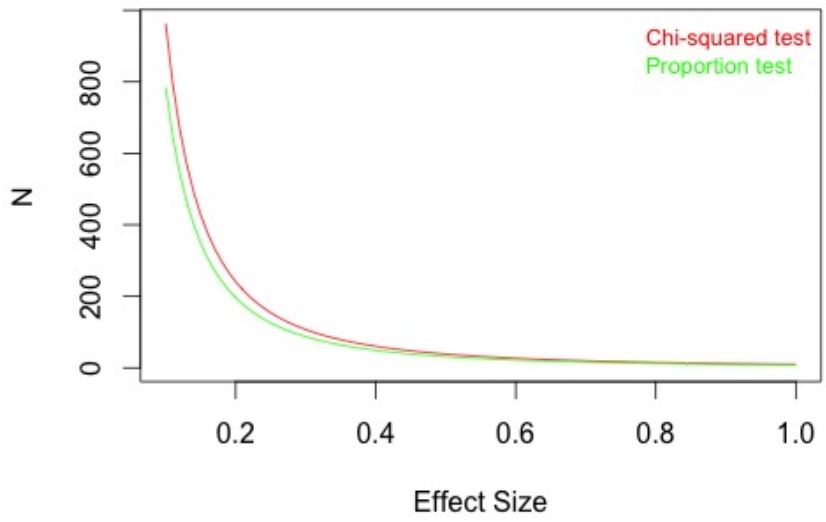

Figure 6.8: Power analysisfor the χ^2 test with 2 degrees of freedom used to compare three cellular lines at a time and for the two-sided test of proportions used to compare two cellular lines at a time. The power was fixed to $1 - \beta = 0.8$ and the type I error probability $\alpha = 0.05$, while the effect size was changing between 0.1 and 1.

Fig. 6.8 reports a graphical representation of the behavior of the sample sizes when the effect size changes. Figs. A.33 to A.36 in Appendix A report for each comparison the boxplots of detected effect sizes. The powers of these tests are really low, as they need a very high sample size (964 in χ^2 test, 785 in proportion test) in order to detect a small effect size. This explains why in the comparison between HPSC and Myeloblasts on chromosome 1 the smallest effect size we were able to detect is $h = 0.02$ with a number of observation equal to $18,820$, while with the smallest number of observation that we have, which is 10, the effect size detected was $h = 0.86$. Analogously, in the comparison on three cellular lines we can detect an effect size of 0.018 with a number of observations equal to $28,257$, while with the smallest number of observation that we have, which is 15, we detected a peak because the effect size was equal to 0.797.

6.4 Results and discussion

We analyzed three different cellular lines using data coming from the CAGE technology in order to validate the methods so far designed for them.

As a goodness-of-fit test suggested that our data are not power-law distributed, we normalized the data using the TPM normalization method. Normalized data were then clustered in order to identify groups of transcripts regulated by the same promoter. These tag clusters were then aggregated into a set of consensus clusters. These two levels of clustering were performed each with two different clustering methods and with a sensitivity analysis for the choice of the number of clusters, but none of the resulting clusterings led to a meaningful clustering. We analyzed how peaks were distributed across chromosomes, discovering very different relative proportions, which was also confirmed by a χ^2 test. The analysis of the differentially expressed CAGE peaks for each pair of cellular lines allows us to identify specific CAGE promoters used in HPSC and Erythroblasts cells. This study helped to better understand the underlying mechanisms of regulation of gene-expression during the hematopoiesis, still unclear.

In this section we analyzed CAGE data using current methods of analysis and with the only available R package specifically tailored for this type of analysis, `CAGEr`. The main contribution of this section lays in the application for the first time on these kind of data of some of the knowledge previously developed in clustering, the use of classical tests for the comparison of peaks across chromosomes and among cell lines, as well as the investigation of possible power issues due to a low number of tag counts. The data we used showed a high variability, making the choice of a good clustering challenging, and a power issue was detected. A biological interpretation of the results is necessary, in order to biologically validate our work and decide which direction has to be taken in future.

Chapter 7

Conclusion

In this Thesis, we have shed some light on three current problems related to Omics research.

In Chapter 4, we reported the most common limitations of the existing clustering algorithms for static data, proposing and successfully validating a novel solution, Cross-clustering (CC), which overcomes these limitations. The CC algorithm bears several advantages over the existing methods: i) it does not require *a priori* knowledge on the number of clusters, ii) leaves outlier elements unassigned, and iii) may results in only one cluster, thus suggesting that data should not be clustered at all. Furthermore, CC is a deterministic method and, though we tested it only on biological data, it is not restricted to such type of application. We compared CC with the most widely used clustering methods both on simulated and real datasets, always obtaining partitions closer to the reality than the results obtained with the other methods. As the distance is a key element in cluster analysis, we performed our comparisons using both the Euclidean and Chebychev distances in order to check how sensible results are to the choice of the distance; in both cases CC performed better than the methods it was compared to. We tested the ability of CC to automatically detect a suitable number of clusters, and it proved to be better than most of the well established criteria proposed in literature. Finally, we tested how stable CC is with regard to the input parameters. It proved to be a robust method, always being able to identify the correct number of clusters also when the given intervals of the input parameters were very large or asymmetric, independently from the variability in the data. An R package with the implementation of the CC algorithm will soon be available.

In Chapter 5, we discussed the most common limitations of the existing clustering algorithm for short-length time-course data. Hence, after the analysis of a very short-length time-course dataset using a well established analysis tool for this kind of application, `maSigPro` (Conesa

et al., 2006), we proposed a novel strategy for overcoming these limitations consisting in the application of the CC algorithm on B-spline coefficients. In particular, this strategy was supposed to take into account the temporal information, the possibility of missing values and outliers, the lack of replicates, having at the same time the advantages of the CC algorithm. We validated our proposal on simulated data. The results in terms of Adjusted Rand Indices and Area Under the Curve indicated that future efforts need to be made in trying different variations of the strategy, such as applying CC on different types of splines or on specially tailored distances for short-length time-course data, as for instance the STS (Möller-Levet et al., 2003) and Cross Correlation distance (Höppner & Klawonn, 2009).

In Chapter 6, we presented the Cap Analysis Gene Expression technology (Carninci et al., 1996), which has been recently developed and allows the genome-wide survey of promoters. The literature regarding CAGE data analysis is still poor, so we performed an exploratory evaluation of the available methods. The main contribution of this chapter lays in the application for the first time to these kind of data of clustering strategies and classical tests. The data we used showed a high variability, making the choice of a good clustering challenging, and a power issue was detected.

Appendix A

Supplementary material

A.1 Chapter 4

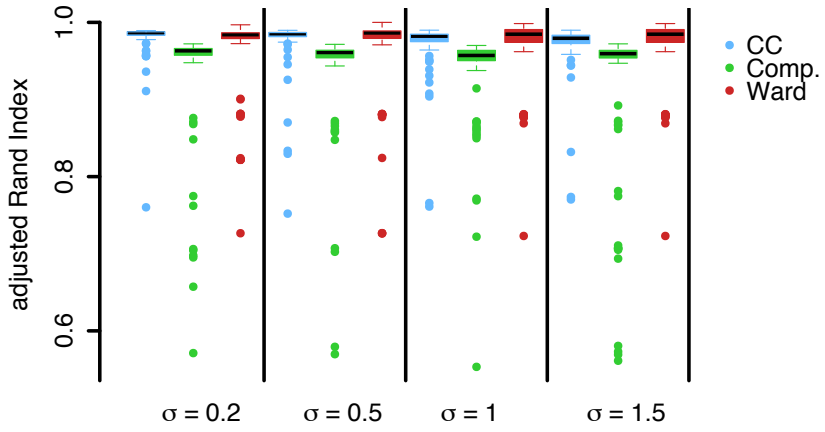

Figure A.1: Boxplots of the Adjusted Rand Index (ARI) resulting from CC, CL, and Ward's algorithm (all with Chebychev distance in order to check if the choice of the distance undermines the quality of results) with $\sigma = 0.2, 0.5, 1, 1.5$ for the simulated data. The ARI here is used to measure the agreement between the partition obtained with each method and the real partition, proving that CC performs at least as well as the two methods that constitute it.

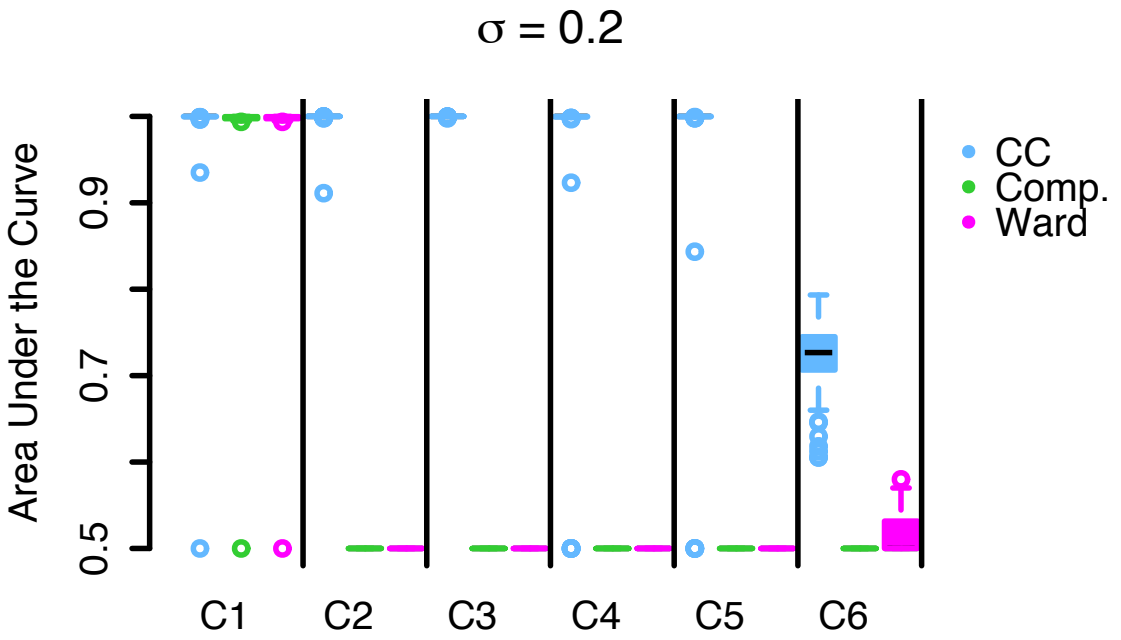

Figure A.2: Boxplots of the Area Under the Curve (AUC) in each of the $K = 6$ simulated clusters for $S = 100$ simulations with CC, CL, and Ward algorithm when $\sigma = 0.2$ using the Chebychev distance in order to check if the choice of the distance undermines the quality of results. High AUC values represent high true positive rates and low false positive rates, thus, the higher the better. In the first cluster the performance is approximately the same, while in the other groups CC performs better. The last cluster includes the outliers.

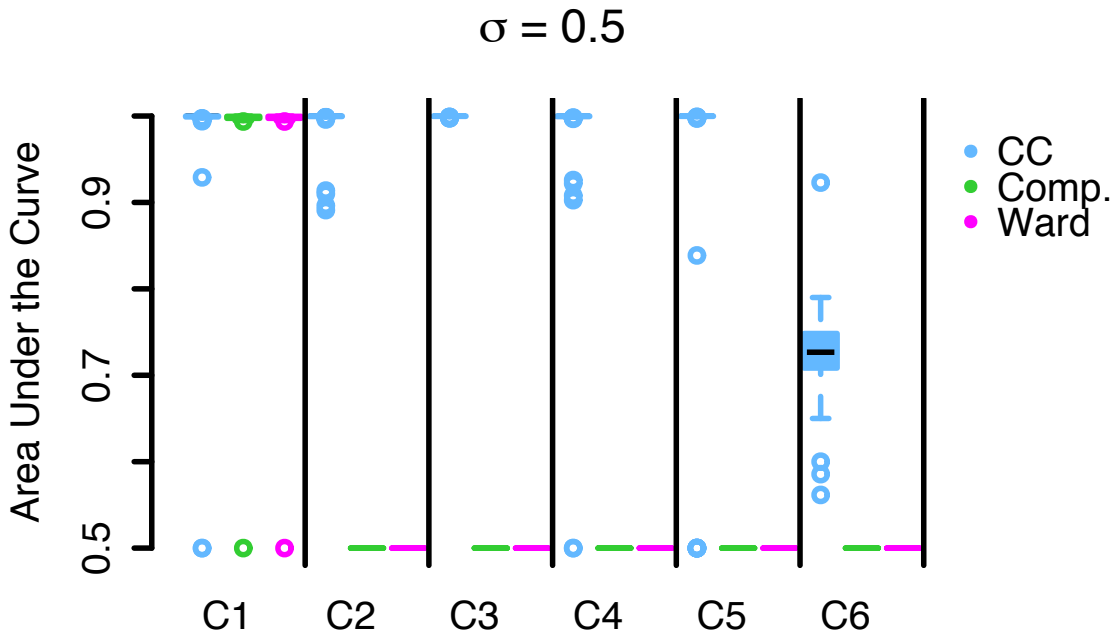

Figure A.3: Boxplots of the Area Under the Curve in each of the $K = 6$ simulated clusters for $S = 100$ simulations with CC, CL, and Ward algorithm when $\sigma = 0.5$ using the Chebychev distance. The last group is the outliers one. As the CC's AUC is always at least as good than as the competitors one, we can conclude that CC performs better than its competitors in terms of sensitivity and specificity.

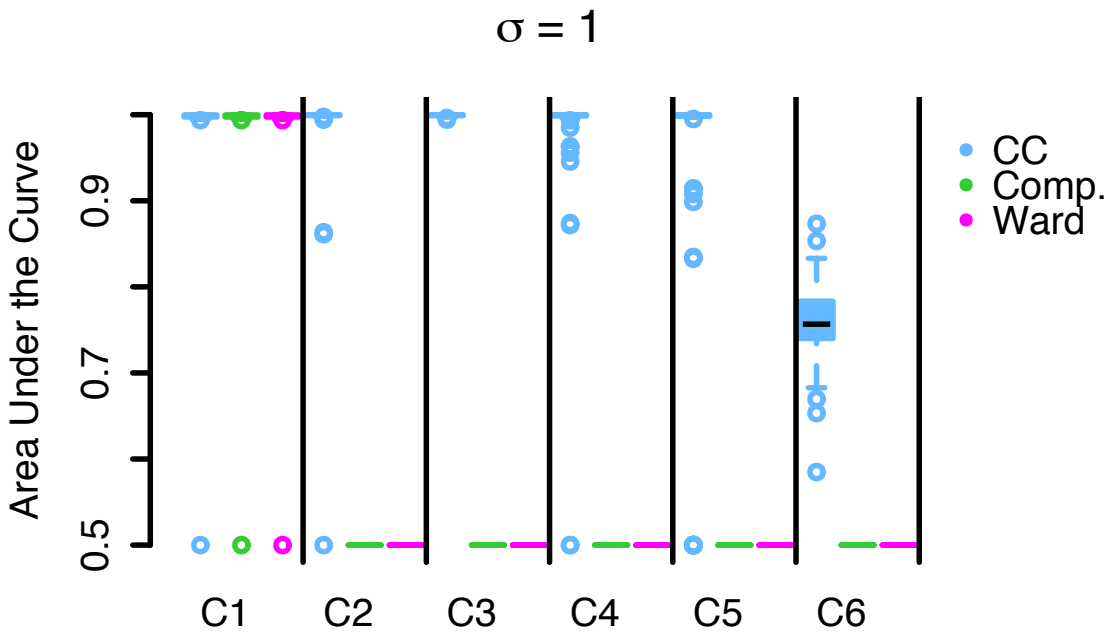

Figure A.4: Boxplots of the Area Under the Curve in each of the $K = 6$ simulated clusters for $S = 100$ simulations with CC, CL, and Ward algorithm when $\sigma = 1$ using the Chebychev distance in order to check if the choice of the distance undermines the quality of results. High AUC values represent high true positive rates and low false positive rates, thus, the higher the better. In the first cluster the performance is approximately the same, while in the other groups CC performs better. The last group includes the outliers.

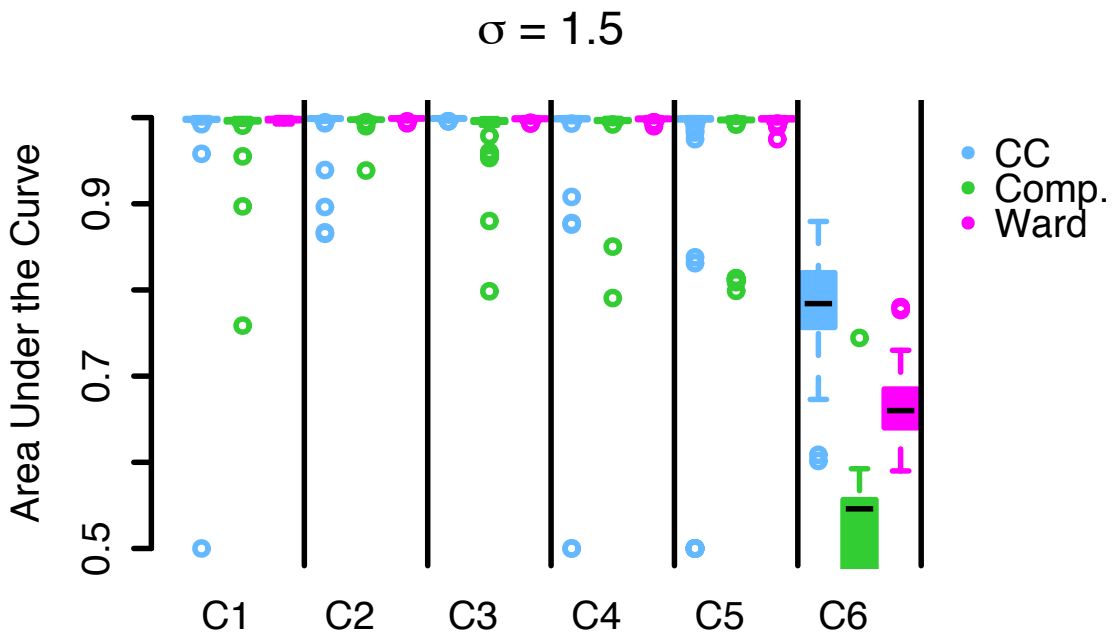

Figure A.5: Boxplots of the Area Under the Curve in each of the $K = 6$ simulated clusters for $S = 100$ simulations with CC, CL, and Ward algorithm when $\sigma = 1.5$ using the Chebychev distance in order to check if the choice of the distance undermines the quality of results. High AUC values represent high true positive rates and low false positive rates, thus, the higher the better. In the first clusters the performance is approximately the same, while in the other groups CC performs better. The last group includes the outliers.

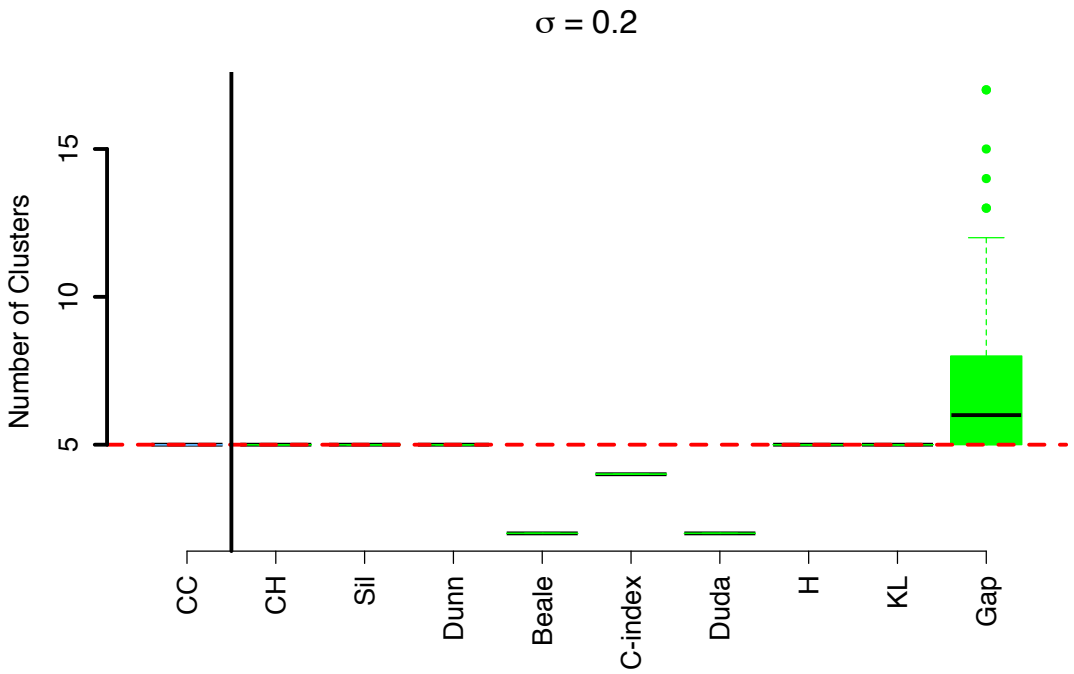

Figure A.6: Boxplots reporting the number of clusters detected on simulated data with $\sigma = 0.2$ by the CC and CL algorithms using different methods, respectively: CH, average Silhouette Width, Dunn index, Beale, C-index, Duda index, H, KL, and Gap. The horizontal red line represents the actual number of clusters, $K = 5$. Here CC proved to be one of the best performing methods, always being able to detect the actual number of clusters in 100 simulations. Beale, C-index, and Duda consistently missed the actual number of clusters, while Gap has high variability with median far from reality.

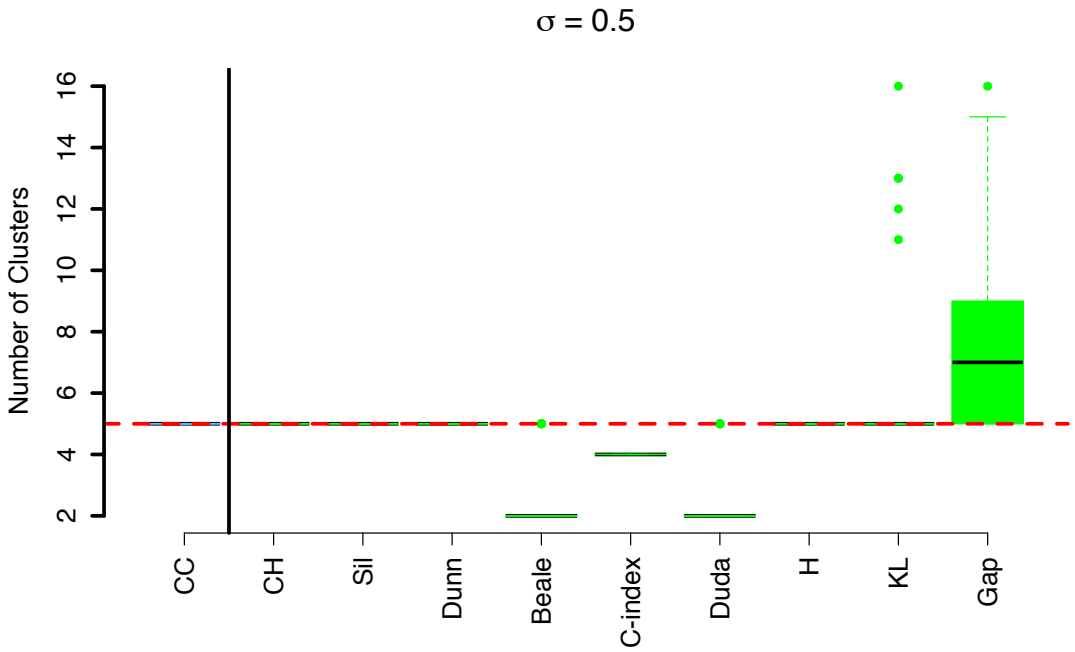

Figure A.7: Boxplots reporting the number of clusters detected on simulated data with $\sigma = 0.5$ by the CC and CL algorithms using different methods, respectively: CH, average Silhouette Width, Dunn index, Beale, C-index, Duda index, H, KL, and Gap. The horizontal red line represents the actual number of clusters, $K = 5$. Here CC proved to be one of the best performing methods, always being able to detect the actual number of clusters in 100 simulations. Beale, C-index, and Duda consistently missed the actual number of clusters, while Gap has high variability with median far from reality.

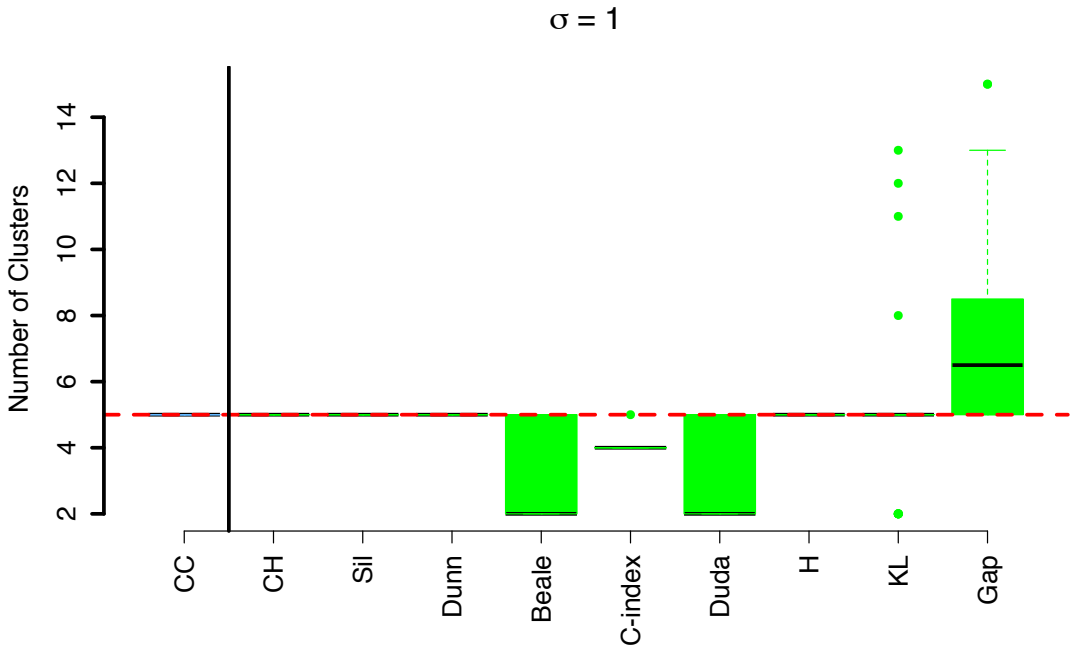

Figure A.8: Boxplots reporting the number of clusters detected on simulated data with $\sigma = 1$ by the CC and CL algorithms using different methods, respectively: CH, average Silhouette Width, Dunn index, Beale, C-index, Duda index, H, KL, and Gap. The horizontal red line represents the actual number of clusters, $K = 5$. Here CC proved to be one of the best performing methods, always being able to detect the actual number of clusters in 100 simulations. C-index consistently missed the actual number of clusters, while Beale, Duda, and Gap have high variability with median far from reality. At the increase of σ even KL starts performing badly.

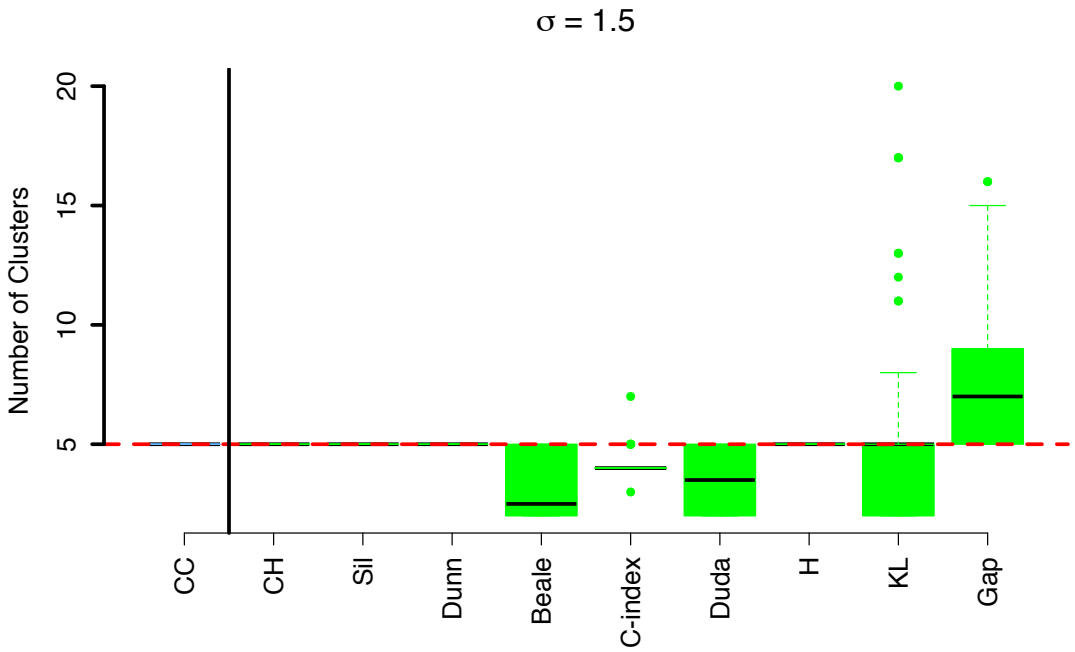

Figure A.9: Boxplots reporting the number of clusters detected on simulated data with $\sigma = 1.5$ by the CC and CL algorithms using different methods, respectively: CH, average Silhouette Width, Dunn index, Beale, C-index, Duda index, H, KL, and Gap. The horizontal red line represents the actual number of clusters, $K = 5$. Here CC proved to be one of the best performing methods, always being able to detect the actual number of clusters in 100 simulations. C-index consistently missed the actual number of clusters, while Beale, Duda, and Gap have high variability with median far from reality. At the increase of σ even KL starts performing badly.

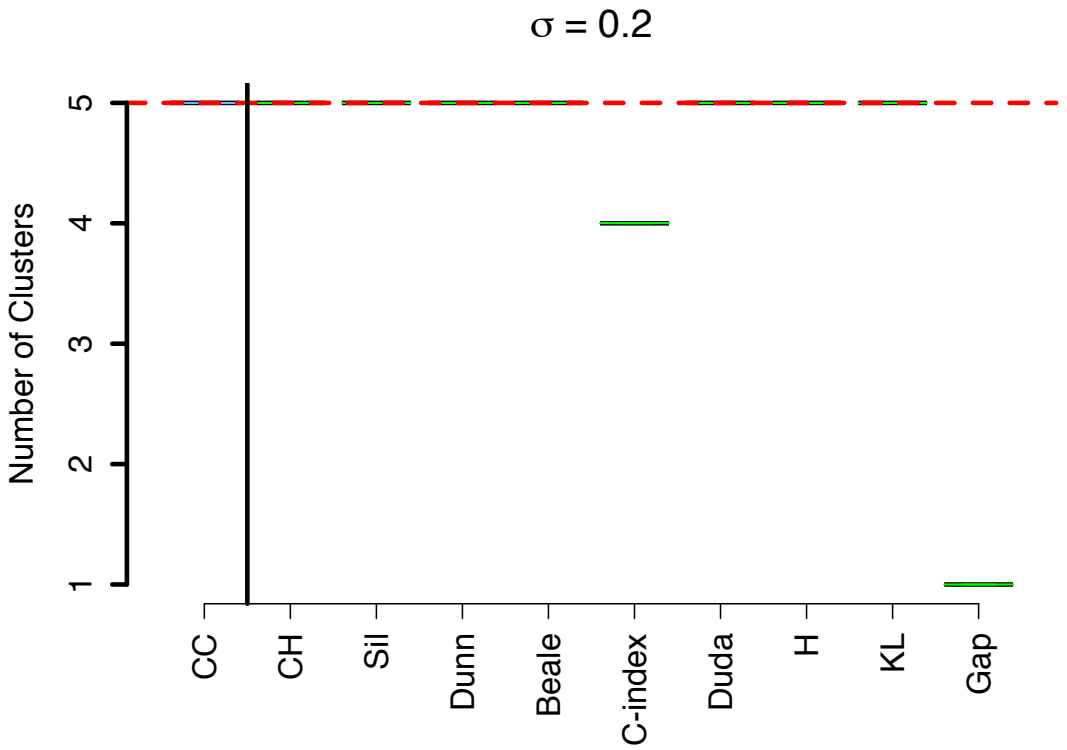

Figure A.10: Boxplots reporting the number of clusters detected on simulated data with $\sigma = 0.2$ by the CC and Ward's minimum variance algorithms using different methods, respectively: CH, average Silhouette Width, Dunn index, Beale, C-index, Duda index, H, KL, and Gap. The horizontal red line represents the actual number of clusters, $K = 5$. Here CC proved to be one of the best performing methods, always being able to detect the actual number of clusters in 100 simulations. C-index and Gap consistently missed the actual number of clusters.

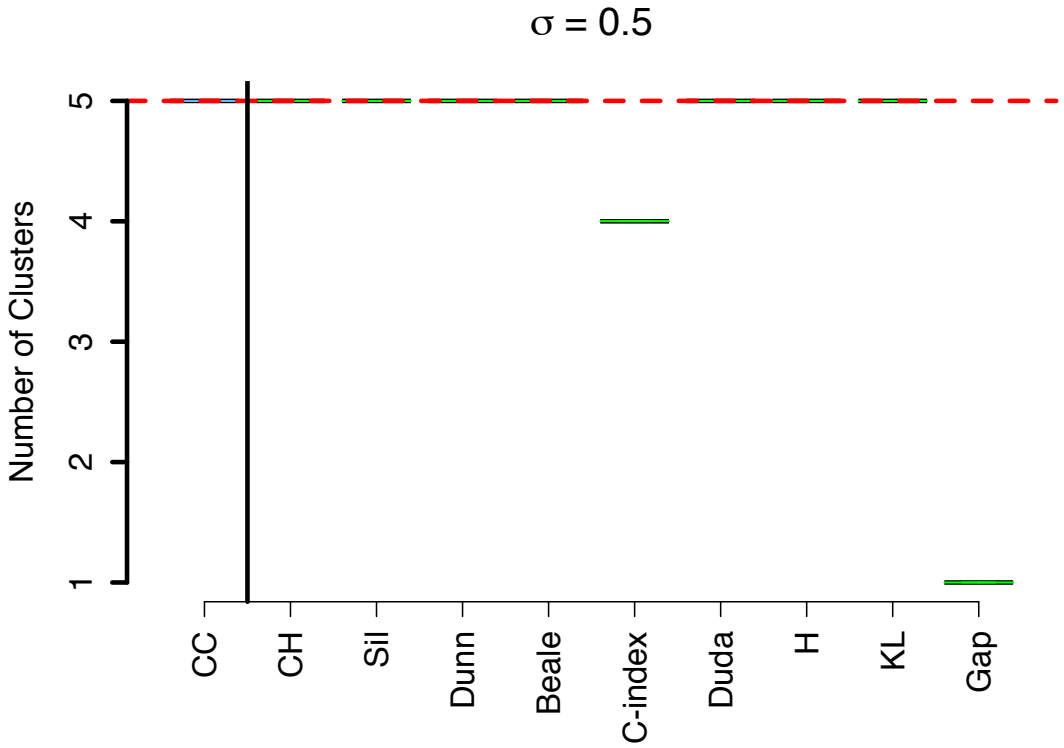

Figure A.11: Boxplots reporting the number of clusters detected on simulated data with $\sigma = 0.5$ by the CC and Ward's minimum variance algorithms using different methods, respectively: CH, average Silhouette Width, Dunn index, Beale, C-index, Duda index, H, KL, and Gap. The horizontal red line represents the actual number of clusters, $K = 5$. Here CC proved to be one of the best performing methods, always being able to detect the actual number of clusters in 100 simulations. C-index and Gap consistently missed the actual number of clusters.

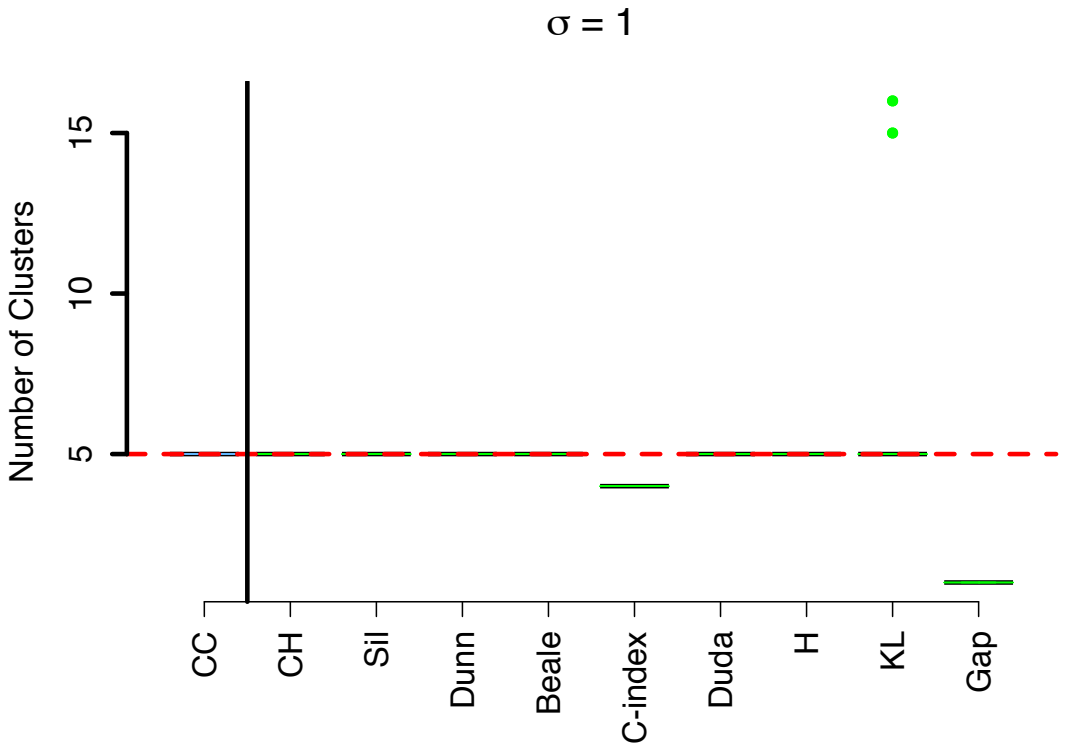

Figure A.12: Boxplots reporting the number of clusters detected on simulated data with $\sigma = 1$ by the CC and Ward's minimum variance algorithms using different methods, respectively: CH, average Silhouette Width, Dunn index, Beale, C-index, Duda index, H, KL, and Gap. The horizontal red line represents the actual number of clusters, $K = 5$. Here CC proved to be one of the best performing methods, always being able to detect the actual number of clusters in 100 simulations. C-index and Gap consistently missed the actual number of clusters.

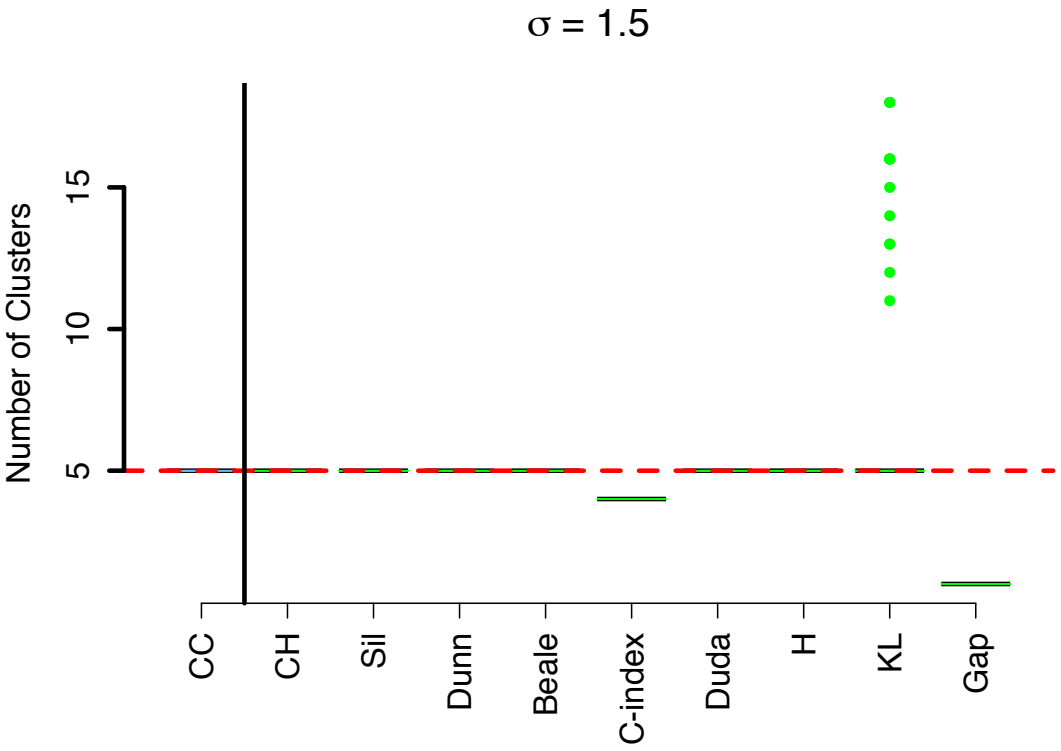

Figure A.13: Boxplots reporting the number of clusters detected on simulated data with $\sigma = 1.5$ by the CC and Ward's minimum variance algorithms using different methods, respectively: CH, average Silhouette Width, Dunn index, Beale, C-index, Duda index, H, KL, and Gap. The horizontal red line represents the actual number of clusters, $K = 5$. Here CC proved to be one of the best performing methods, always being able to detect the actual number of clusters in 100 simulations. C-index and Gap consistently missed the actual number of clusters. At the increase of σ even KL starts performing badly.

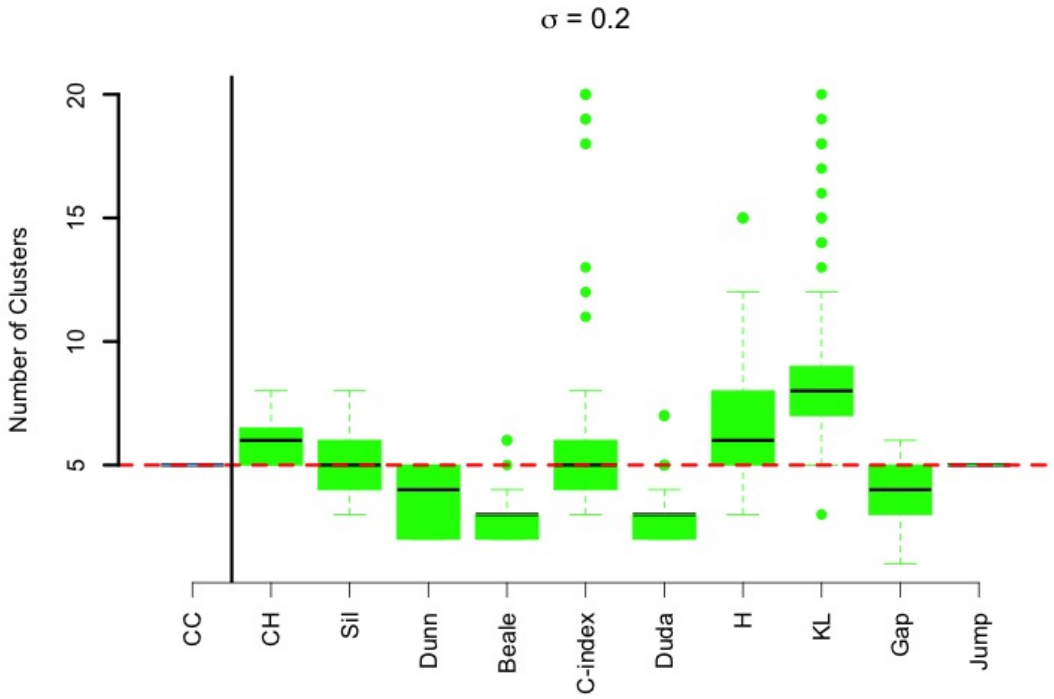

Figure A.14: Boxplots reporting the number of clusters detected on simulated data with $\sigma = 0.2$ by the CC and K-means using different algorithms, respectively: CH, average Silhouette Width, Dunn index, Beale, C-index, Duda index, H, KL, Gap, and Jump. The horizontal red line represents the actual number of clusters, $K = 5$. Here CC and Jump methods proved to be the best performing methods, showing the smaller variability of results in 100 replications.

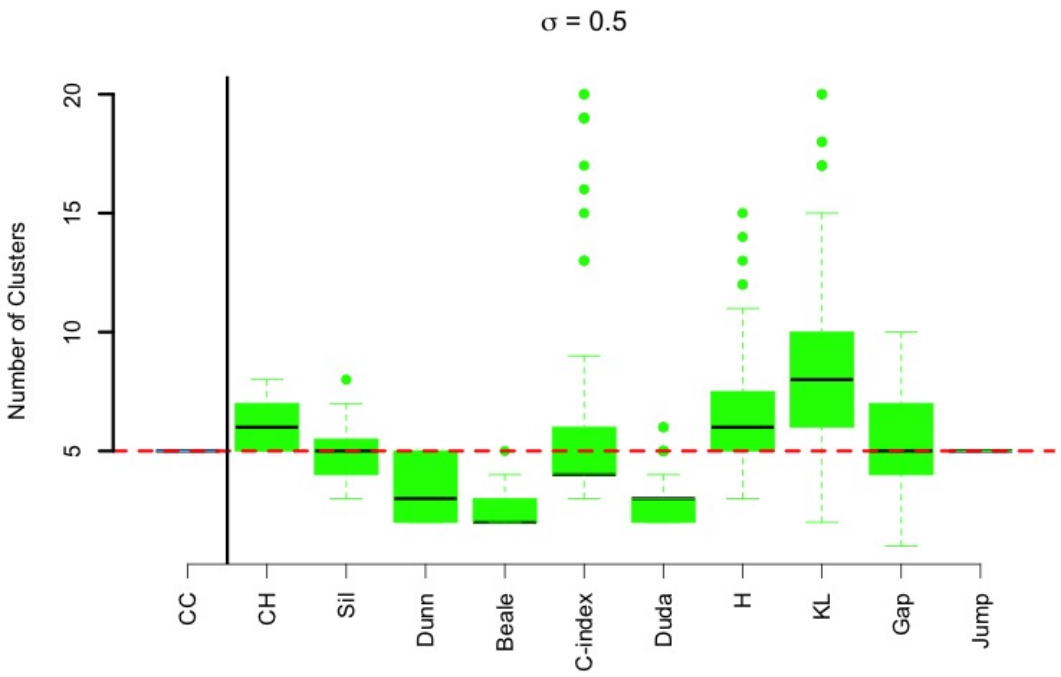

Figure A.15: Boxplots reporting the number of clusters detected on simulated data with $\sigma = 0.5$ by the CC and K-means algorithms using different methods, respectively: CH, average Silhouette Width, Dunn index, Beale, C-index, Duda index, H, KL, Gap, and Jump. The horizontal red line represents the actual number of clusters, $K = 5$. Here CC and Jump methods proved to be the best performing methods, , always being able to detect the actual number of clusters in 100 replications.

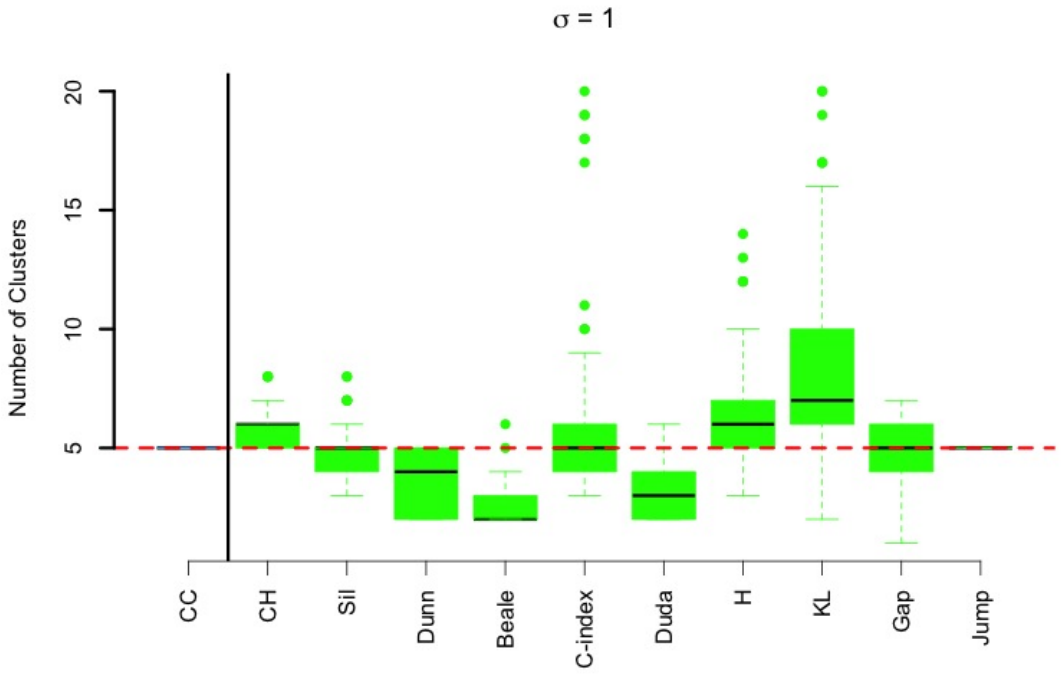

Figure A.16: Boxplots reporting the number of clusters detected on simulated data with $\sigma = 1$ by the CC and K-means algorithms using different methods, respectively: CH, average Silhouette Width, Dunn index, Beale, C-index, Duda index, H, KL, Gap, and Jump. The horizontal red line represents the actual number of clusters, $K = 5$. Here CC and Jump methods proved to be the best performing methods, , always being able to detect the actual number of clusters in 100 replications.

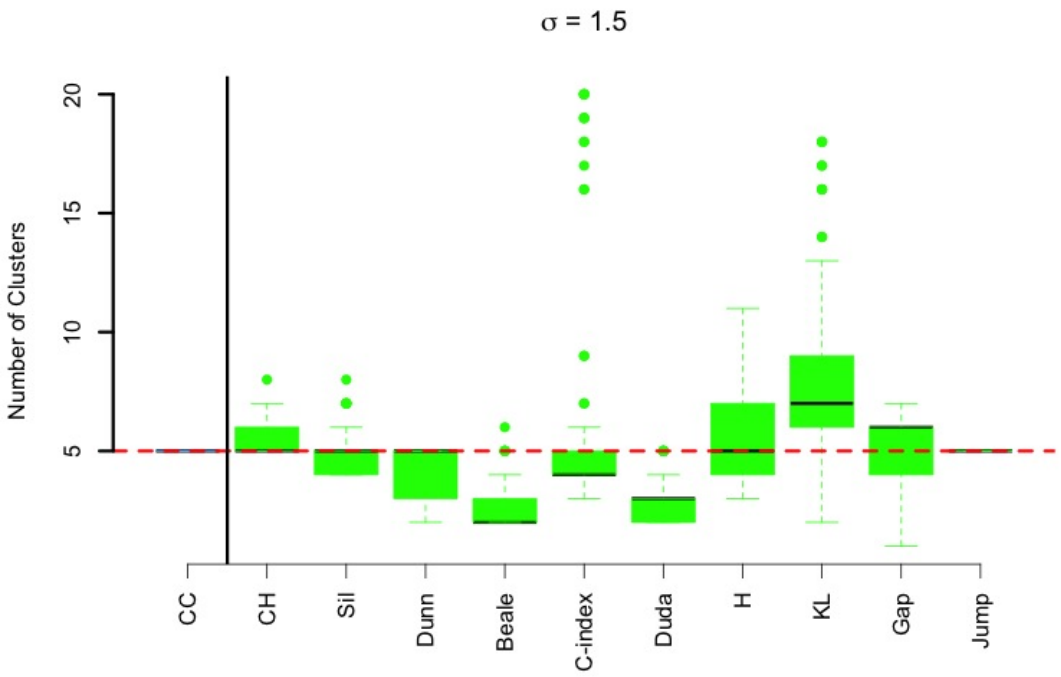

Figure A.17: Boxplots reporting the number of clusters detected on simulated data with $\sigma = 1.5$ by the CC and K-means algorithms using different methods, respectively: CH, average Silhouette Width, Dunn index, Beale, C-index, Duda index, H, KL, Gap, and Jump. The horizontal red line represents the actual number of clusters, $K = 5$. Here CC and Jump methods proved to be the best performing methods, , always being able to detect the actual number of clusters in 100 replications.

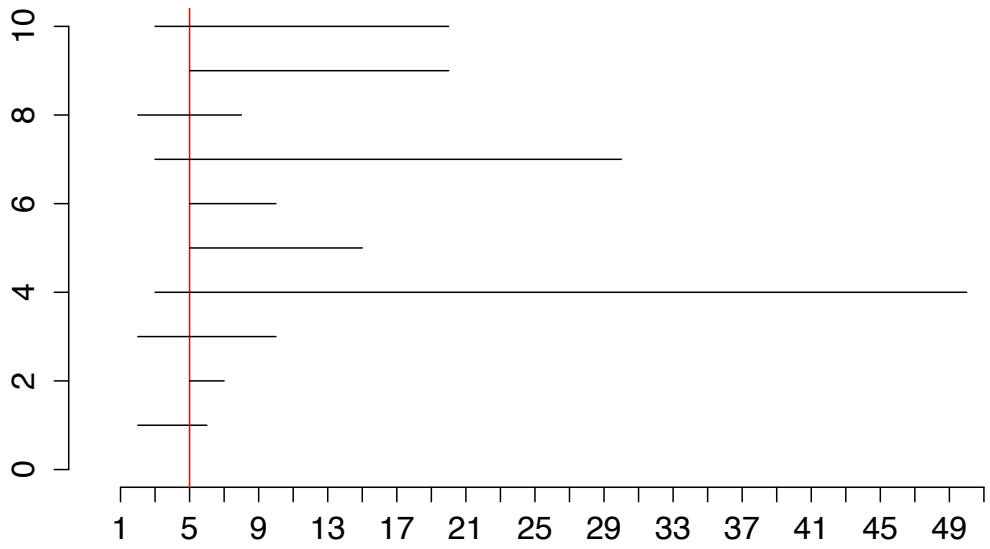

Figure A.18: Representation of the boundaries of the ten different intervals I^W used for choosing the number of clusters in the Ward's minimum variance method inside the CC algorithm. The red line represents the correct number of clusters, 5, in simulated data after the removal of outliers. The length of the line represents the width of the interval.

Index	$\sigma = 0.2$				$\sigma = 0.5$				$\sigma = 1$				$\sigma = 1.5$			
	K	W	CL	CC	K	W	CL	CC	K	W	CL	CC	K	W	CL	CC
-				**100**				**100**				**100**				**100**
CH	30	100	100		28	100	100		44	100	100		55	100	100	
Sil	32	100	100		32	100	100		46	100	100		57	100	100	
Dunn	30	100	100		28	100	100		44	100	100		55	100	100	
Beale	1	100	0		1	100	6		1	100	29		6	100	47	
C-index	18	0	0		17	0	0		24	0	1		19	0	4	
Duda	4	100	0		5	100	6		8	100	29		8	100	50	
H	21	100	100		22	100	100		24	100	100		23	100	100	
KL	6	100	100		11	100	95		14	96	90		19	87	49	
Gap	21	0	37		10	0	31		11	0	30		8	0	27	
Jump	100	-	-		100	-	-		100	-	-		100	-	-	
Average	26.30	77.78	59.67	100	25.4	77.78	59.78	100	31.60	77.33	64.33	100	35	76.33	64.11	100
Median	21	100	100	100	19.5	100	95	100	24	100	90	100	21	100	50	100

Table A.1: Percentages of success (identification of 5 clusters) with K-means (K), Ward's minimum variance method (W), CL and CC for $\sigma = 0.2, 0.5, 1, 1.5$ in correspondence with different indexes. The Jump method is only defined for K-means. This table shows how CC was always able to detect the actual number of clusters in 100 replications, while other methods exhibit some variability.

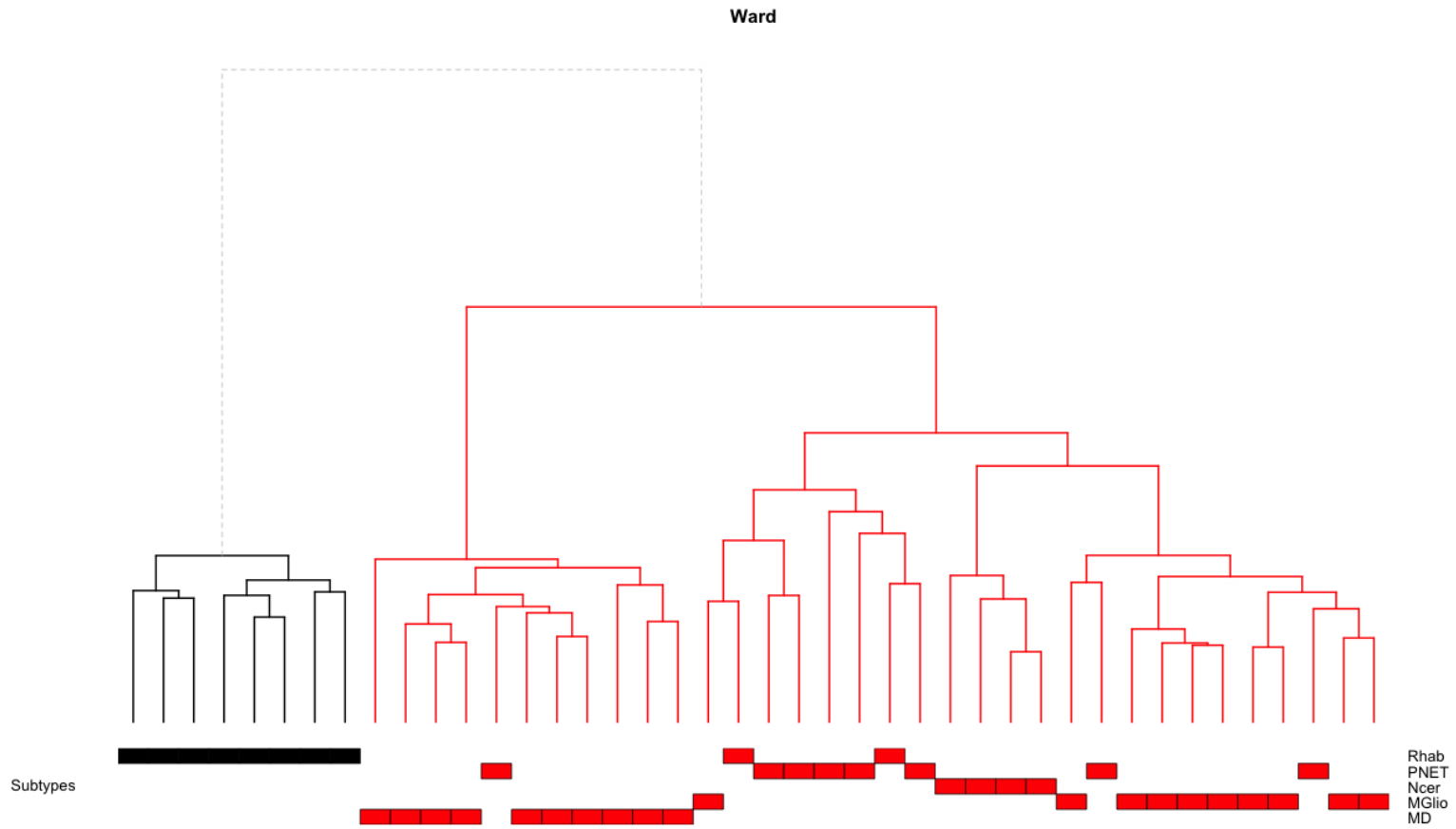

Figure A.19: Dendrogram resulting from clustering with Ward's algorithm 42 brain tumors samples. Leafs are colored depending on the results of the clustering. Boxes in the bottom represent the real subtypes.

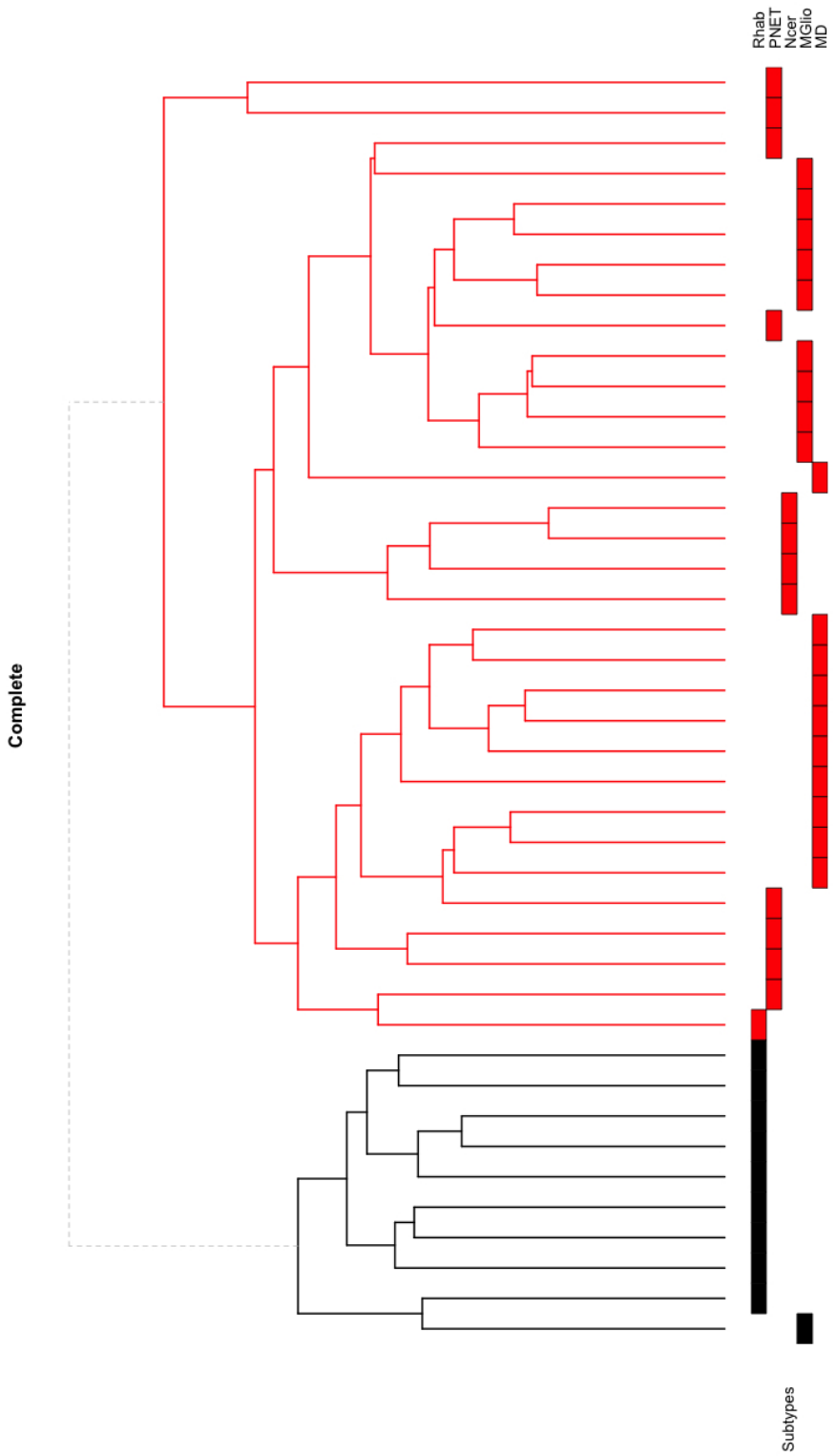

Figure A.20: Dendrogram resulting from clustering with Complete-linkage (CL) 42 brain tumors samples. Leafs are colored depending on the results of the clustering. Boxes in the bottom represent the real subtypes.

Comb. #	$x$dim	$y$dim	Tot.
1	1	2	2
2	1	3	3
3	1	4	4
4	1	5	5
5	1	6	6
6	1	7	7
7	1	8	8
8	1	9	9
9	1	10	10
10	1	11	11
11	1	12	12
12	1	13	13
13	1	14	14
14	1	15	15
15	1	16	16
16	1	17	17
17	1	18	18
18	1	19	19
19	1	20	20
20	2	1	2
21	2	2	4
22	2	3	6
23	2	4	8
24	2	5	10
25	2	6	12
26	2	7	14
27	2	8	16
28	2	9	18
29	2	10	20
30	3	1	3
31	3	2	6
32	3	3	9
33	3	4	12
34	3	5	15
35	3	6	18

Comb. #	$x$dim	$y$dim	Tot.
36	3	7	21
37	4	1	4
38	4	2	8
39	4	3	12
40	4	4	16
41	4	5	20
42	5	1	5
43	5	2	10
44	5	3	15
45	5	4	20
46	6	1	6
47	6	2	12
48	6	3	18
49	6	4	24
50	7	1	7
51	7	2	14
52	7	3	21
53	8	1	8
54	8	2	16
55	8	3	24
56	9	1	9
57	9	2	18
58	9	3	27
59	10	1	1
60	10	2	20
61	11	1	11
62	12	1	12
63	13	1	13
64	14	1	14
65	15	1	15
66	16	1	16
67	17	1	17
68	18	1	18
69	19	1	19
70	20	1	20

Table A.2: 70 combinations of x and y dimensions for the SOM's grid in order to decide which one to use on the brain tumors dataset. The last column of both tables reports the total number of resulting available cells.

	MD	MGlio	Ncer	PNET	Rhab	
Clu1	10	10	4	8	2	34
Clu2	0	0	0	0	8	8
	10	10	4	8	10	42

Table A.3: Contingency table reporting both the real classification in subtypes and the clusters identified by the Ward's minimum variance method on the brain tumors dataset. Values in red are those which resulted statistically significant after the over-representation analysis.

	MD	MGlio	Ncer	PNET	Rhab	
Clu1	10	9	2	1	10	32
Clu2	0	1	2	7	0	10
	10	10	4	8	10	42

Table A.4: Contingency table reporting both the real classification in subtypes and the clusters identified by the CL method on the brain tumors dataset. Values in red are those which resulted statistically significant after the over-representation analysis.

	MD	MGlio	Ncer	PNET	Rhab	
Clu1	10	9	1	1	9	30
Clu2	0	0	1	0	1	2
Clu3	0	1	2	7	0	10
	10	10	4	8	10	42

Table A.5: Contingency table reporting both the real classification in subtypes and the clusters identified by the K-means method on the brain tumors dataset. Values in red are those which resulted statistically significant after the over-representation analysis.

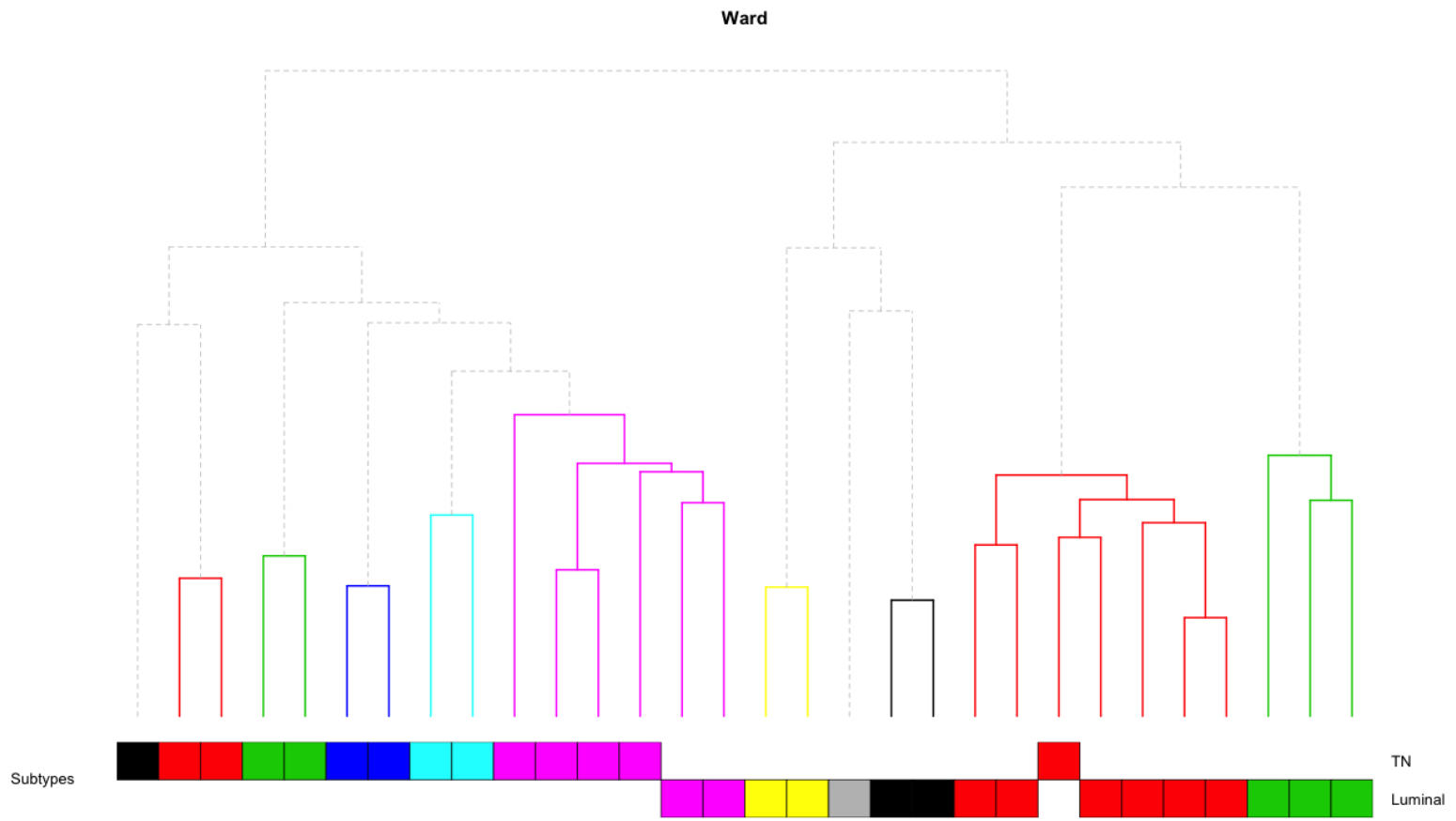

Figure A.21: Dendrogram resulting from clustering with Ward's algorithm 30 breast cancer tumor samples with Euclidean distance. Leafs are colored depending on the results of the clustering. Boxes in the bottom represent the real subtypes.

	MD	MGlio	Ncer	PNET	Rhab	
Clu1	0	0	0	1	0	1
Clu2	10	10	4	7	10	41
	10	10	4	8	10	42

Table A.6: Contingency table reporting both the real classification in subtypes and the clusters identified by the SOM algorithm for the brain tumors dataset. No subtypes are over-represented by any cluster.

	MD	MGlio	Ncer	PNET	Rhab	
Outliers	1	0	0	0	0	1
Clu1	0	9	0	2	0	11
Clu2	9	0	0	1	0	10
Clu3	0	0	0	0	8	8
Clu4	0	0	4	0	0	4
Clu5	0	1	0	0	1	2
Clu6	0	0	0	1	1	2
Clu7	0	0	0	2	0	2
Clu8	0	0	0	1	0	1
Clu9	0	0	0	1	0	1
	10	10	4	8	10	42

Table A.7: Contingency table reporting both the real classification in subtypes and the clusters identified by CC for the brain tumors dataset. Values in red are those which resulted statistically significant after the over-representation analysis. Although CC identified more than five clusters, four of them almost perfectly represented MD, MGlio, Ncer, and Rhab subtypes. The PNET subtype was represented by six of CC clusters, but it is well known for having an high heterogeneity.

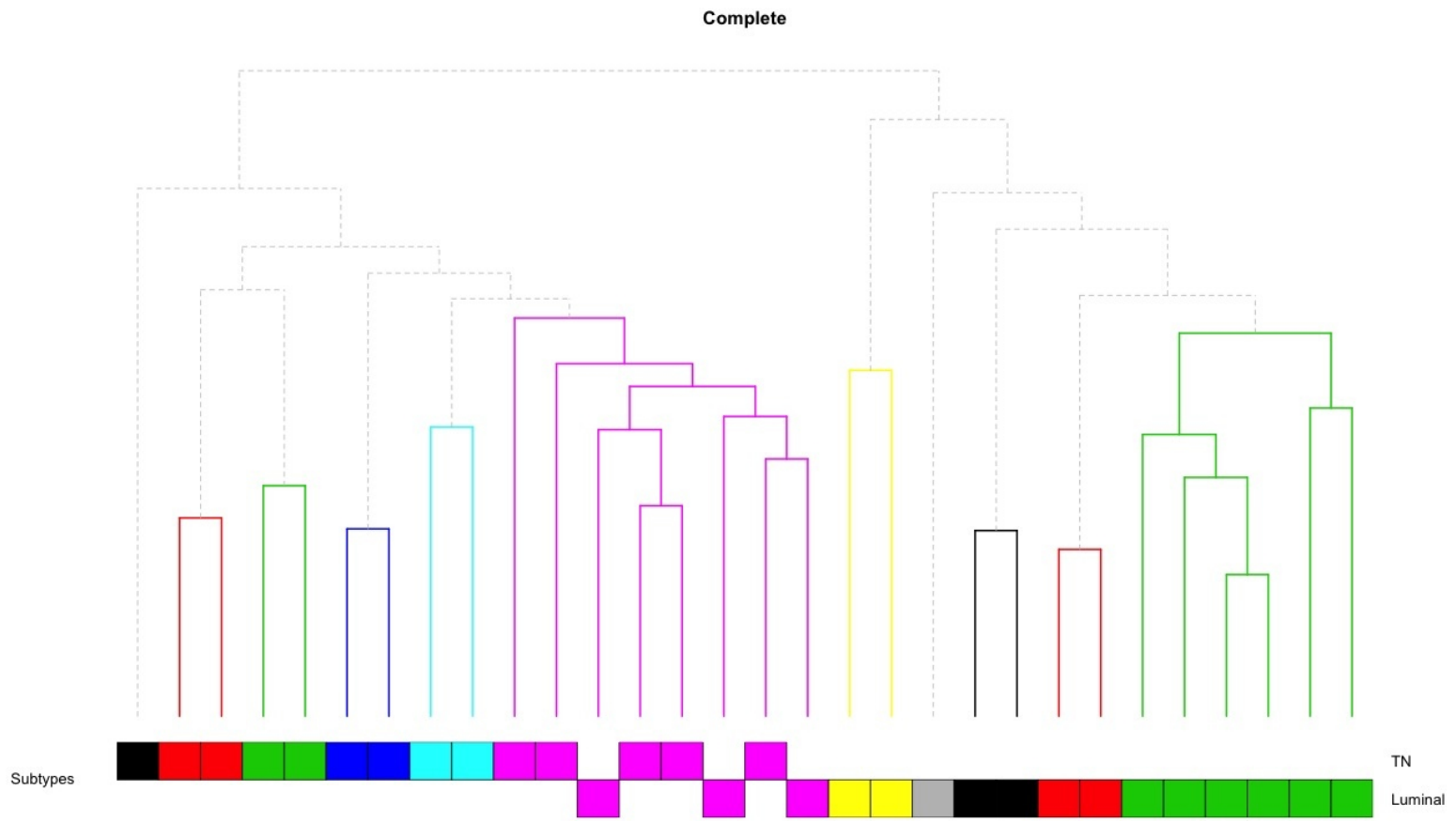

Figure A.22: Dendrogram resulting from clustering with Complete-linkage 30 breast cancer tumor samples with Euclidean distance. Leafs are colored depending on the results of the clustering. Boxes in the bottom represent the real subtypes.

	Luminal	TN	
Clu1	6	1	7
Clu2	2	4	6
Clu3	1	0	1
Clu4	2	0	2
Clu5	2	0	2
Clu6	0	2	2
Clu7	0	2	2
Clu8	0	2	2
Clu9	0	2	2
Clu10	0	1	1
Clu11	3	0	3
	16	14	30

Table A.8: Contingency table reporting both the real classification in subtypes and the clusters identified by the Ward's minimum variance method for the breast cancer dataset. No subtypes are over-represented by any cluster.

	Luminal	TN	
Clu1	3	5	8
Clu2	4	2	6
Clu3	1	0	1
Clu4	2	0	2
Clu5	1	1	2
Clu6	2	0	2
Clu7	0	2	2
Clu8	0	2	2
Clu9	2	0	2
Clu10	1	0	1
Clu11	0	2	2
	16	14	30

Table A.9: Contingency table reporting both the real classification in subtypes and the clusters identified by the Complete-linkage method for the breast cancer dataset. No subtypes are over-represented by any cluster.

	Luminal	TN	
1	2	0	2
2	1	0	1
4	0	2	2
6	0	1	1
7	1	0	1
8	0	2	2
9	2	1	3
10	2	0	2
11	1	0	1
12	0	1	1
13	0	3	3
14	0	2	2
15	0	2	2
16	1	0	1
17	2	0	2
18	4	0	4
	16	14	30

Table A.10: Contingency table reporting both the real classification in subtypes and the clusters identified by SOM for the breast cancer dataset. No subtypes are over-represented by any cluster.

	Luminal	TN	
Clu1	1	0	1
Clu2	3	1	4
Clu3	0	2	2
Clu4	0	2	2
Clu5	1	1	2
Clu6	1	0	1
Clu7	4	5	9
Clu8	2	0	2
Clu9	2	0	2
Clu10	2	0	2
Clu11	0	3	3
	16	14	30

Table A.11: Contingency table reporting both the real classification in subtypes and the clusters identified by K-means for the breast cancer dataset. No subtypes are over-represented by any cluster.

	Luminal	TN	
Outliers	1	1	2
Clu1	2	13	15
Clu2	13	0	13
	16	14	30

Table A.12: Contingency table reporting both the real classification in subtypes and the clusters identified by Cross-clustering for the breast cancer dataset. Values in red are those which resulted statistically significant after the over-representation analysis.

A.2 Chapter 5

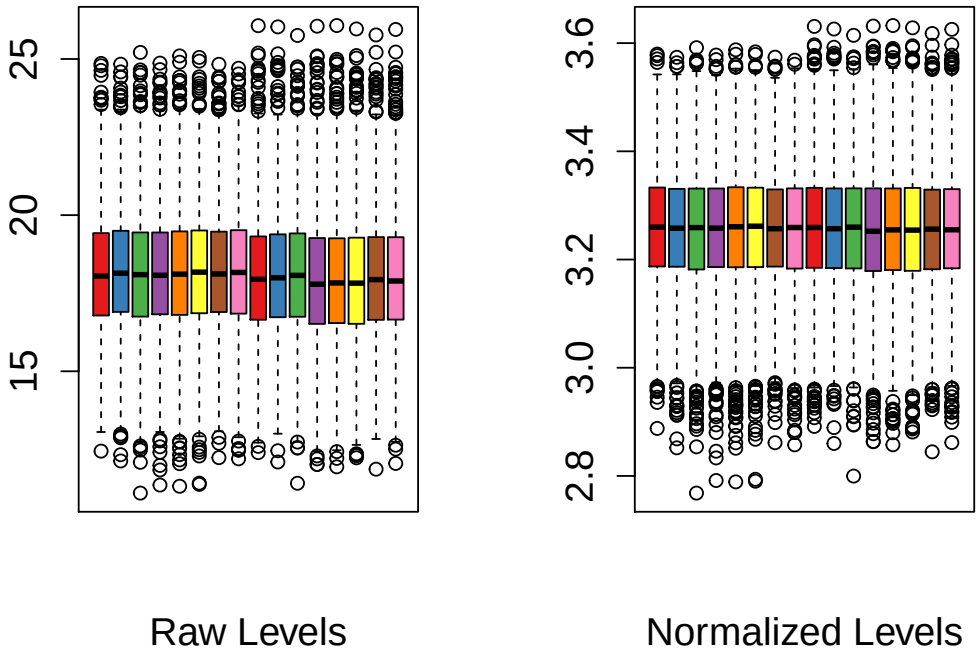

Figure A.23: Boxplots for every experimental condition of both raw and normalized data on logarithmic scale.

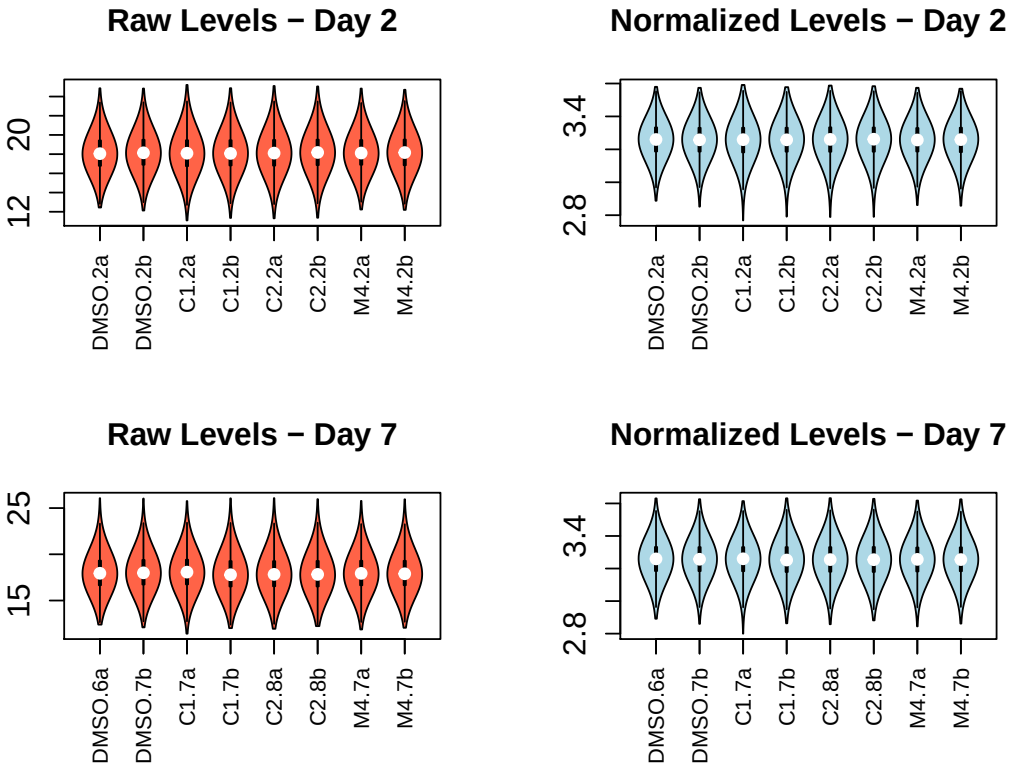

Figure A.24: Violin plots for every experimental condition of both raw and normalized data on logarithmic scale. Dots represents median values.

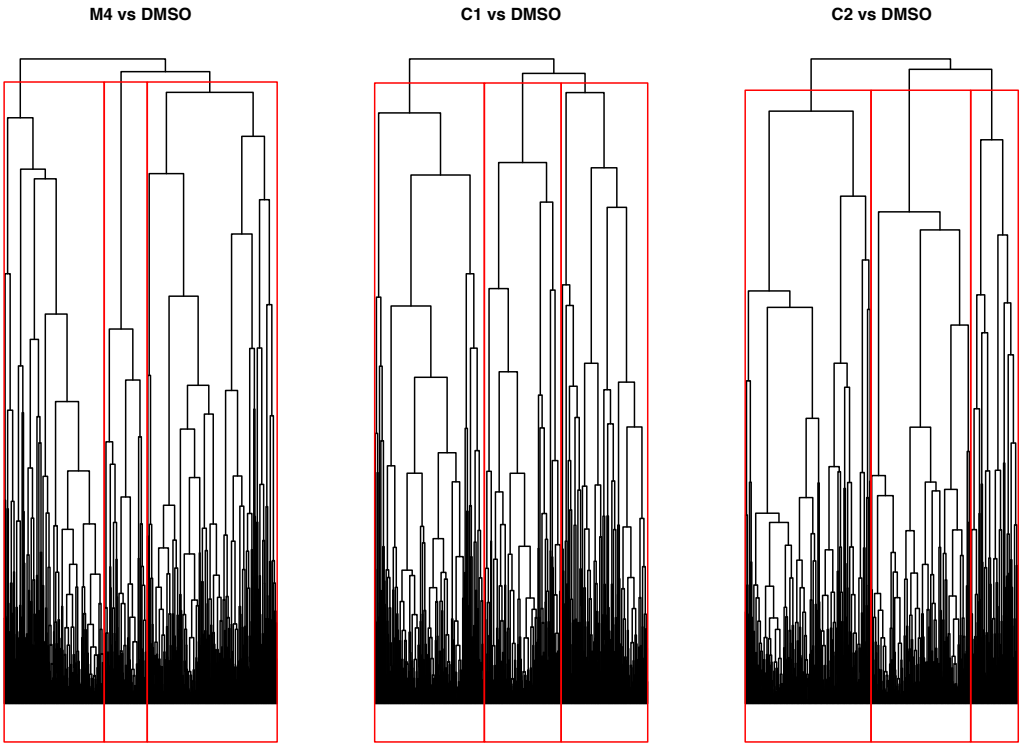

Figure A.25: Dendrograms of the cluster analysis using Pearson's correlation as distance and Complete-linkage as clustering algorithm.

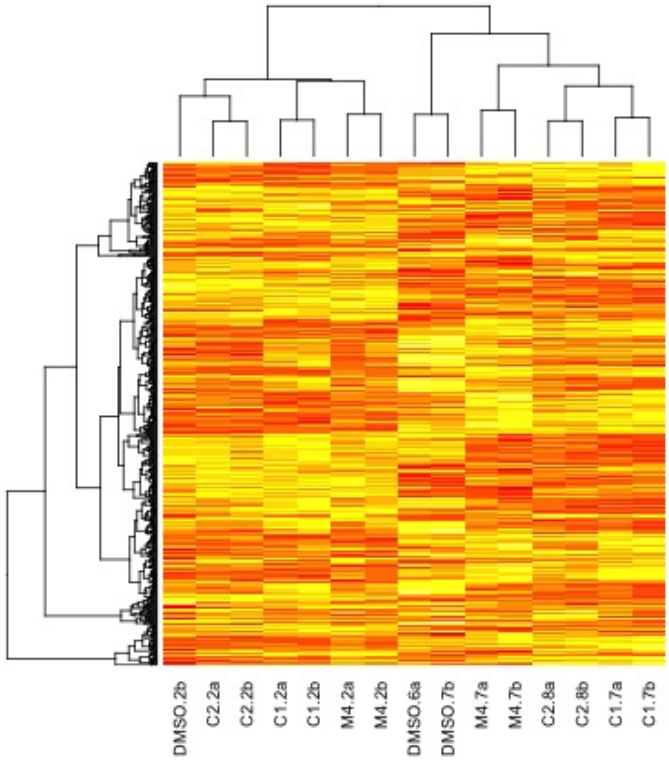

Figure A.26: Heatmap of the 559 significant proteins in the comparison M4 vs DMSO, using correlation as distance measure and Complete-linkage as clustering algorithm.

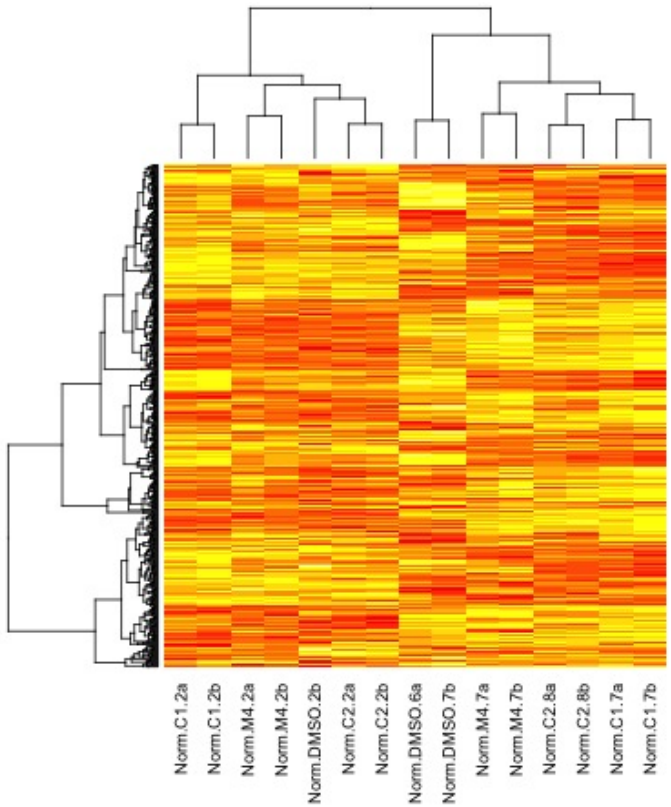

Figure A.27: Heatmap of the 555 significant proteins in the comparison C1 vs DMSO, using correlation as distance measure and Complete-linkage as clustering algorithm.

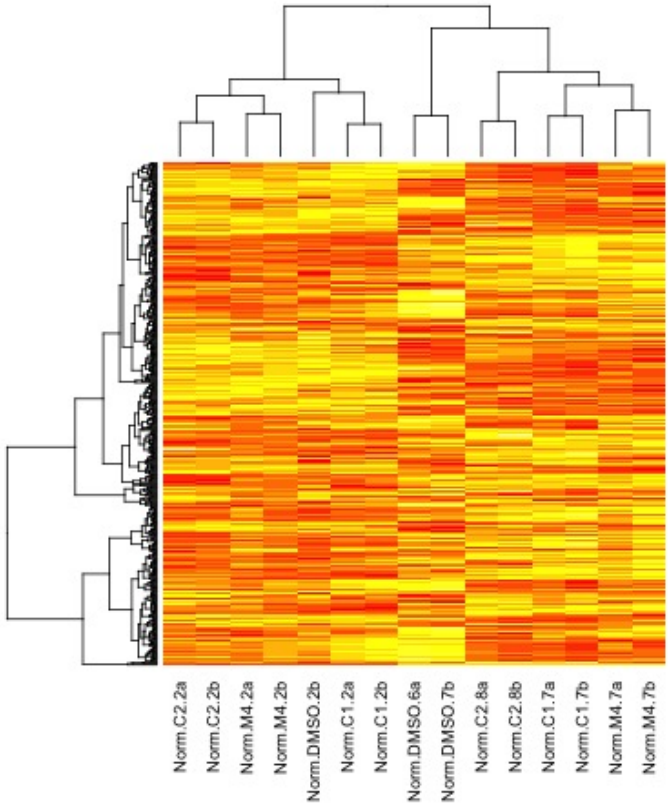

Figure A.28: Heatmap of the 499 significant proteins in the comparison C2 vs DMSO, using correlation as distance measure and Complete-linkage as clustering algorithm.

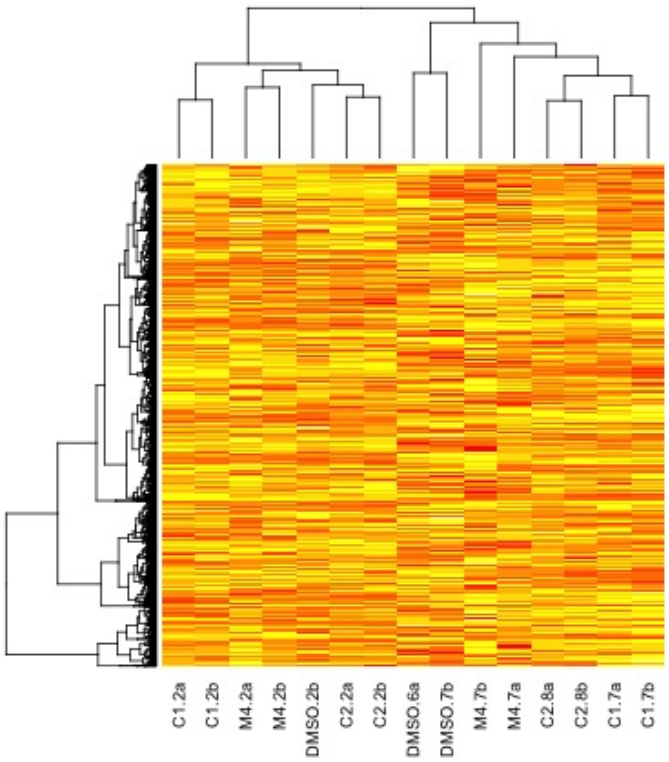

Figure A.29: Heatmap of the $1,997$ proteins excluded from the analysis.

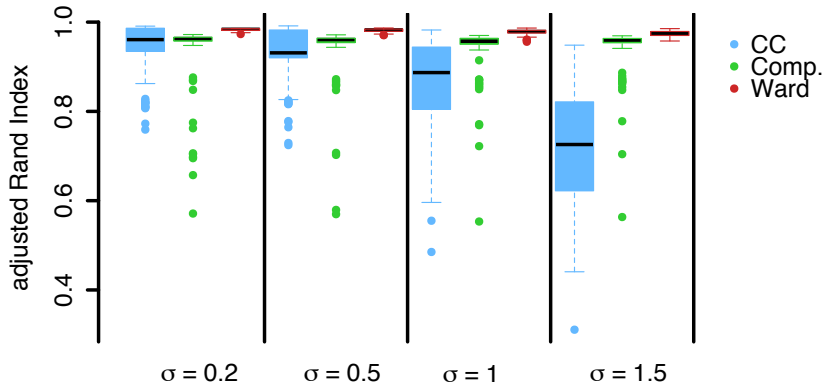

Figure A.30: Boxplots of adjusted Rand Index (ARI) resulting from Cross-clustering (CC) on B-spline coefficients, Complete-linkage (CL) and Ward's algorithm with $\sigma = 0.2, 0.5, 1, 1.5$ and Chebychev distance for simulated data. The ARI measures the agreement between two partitions (the higher, the better). This figure proves that CC does not perform better than reference methods, as its distribution of ARIs over 100 replications is always lower than its competitors.

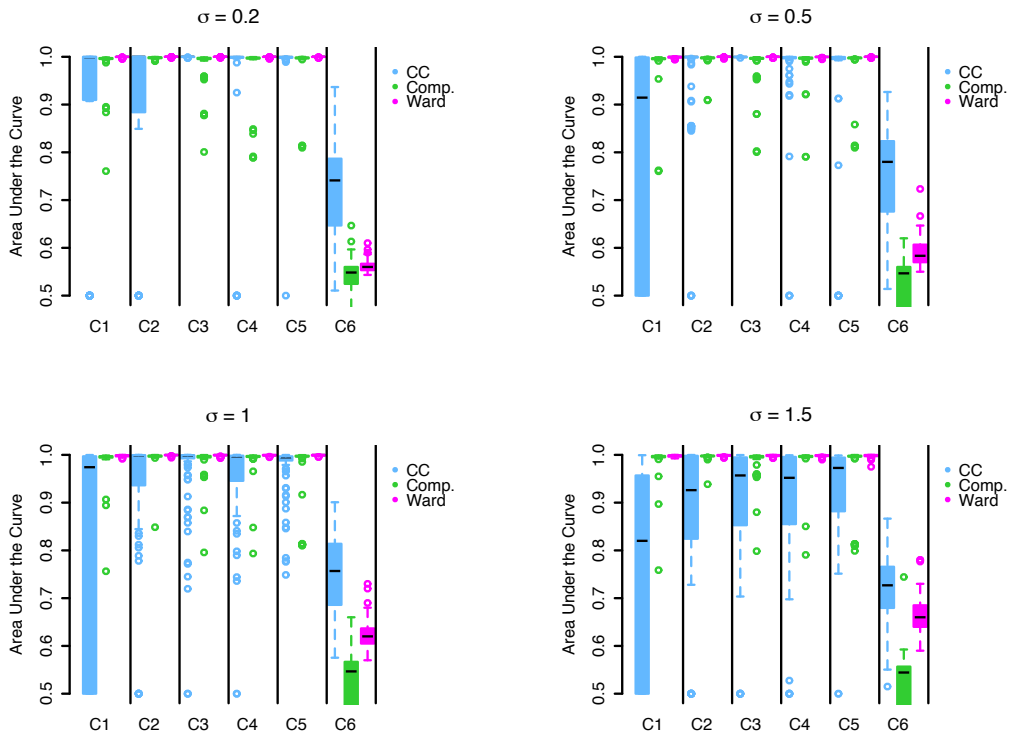

Figure A.31: Boxplots of the Area Under the Curve (AUC) resulting from Cross-clustering (CC), Complete-linkage (CL) and Ward's algorithm with $\sigma = 0.2, 0.5, 1, 1.5$ and Chebychev distance in each of the six clusters of the simulated data. The last cluster is the outliers' one. The AUC is a measure of sensitivity and specificity (the higher, the better). This figure proves that CC does not perform better than the reference methods, as the distribution of AUCs over 100 replications is generally lower than other distributions.

A.3 Chapter 6

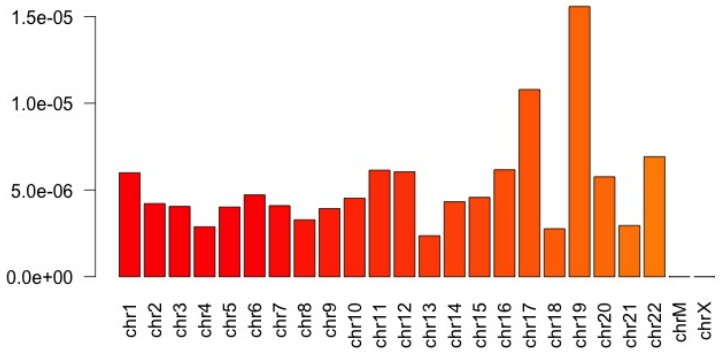

Figure A.32: Relative frequencies of the number of peaks in each chromosome compared to the length of the chromosome.

	HPSC/Myelo/Eritro	HPSC/Myelo/Eritro %
chr1	662	0.44
chr2	467	0.46
chr3	352	0.44
chr4	239	0.43
chr5	351	0.48
chr6	392	0.49
chr7	317	0.49
chr8	224	0.47
chr9	233	0.42
chr10	258	0.42
chr11	375	0.45
chr12	352	0.44
chr13	130	0.48
chr14	200	0.43
chr15	223	0.48
chr16	215	0.39
chr17	347	0.40
chr18	96	0.44
chr19	380	0.41
chr20	161	0.44
chr21	57	0.40
chr22	144	0.41
chrM	47	0.47
chrX	220	0.44

Table A.13: Number and percentage of peaks for each chromosome detected as significantly different in the comparison between the three cell lines with a χ^2 test.

	H/M	H/M %	H/E	H/E %	M/E	M/E %
chr1	482	0.43	235	0.21	566	0.49
chr2	353	0.45	168	0.21	393	0.49
chr3	237	0.38	122	0.20	304	0.49
chr4	177	0.42	82	0.20	198	0.46
chr5	257	0.45	125	0.22	284	0.51
chr6	275	0.44	180	0.28	302	0.47
chr7	237	0.45	108	0.21	252	0.48
chr8	166	0.47	89	0.24	180	0.50
chr9	178	0.43	83	0.20	192	0.46
chr10	186	0.42	91	0.21	211	0.45
chr11	307	0.49	141	0.24	289	0.47
chr12	256	0.42	125	0.21	287	0.48
chr13	108	0.54	43	0.22	107	0.53
chr14	151	0.41	61	0.17	163	0.46
chr15	169	0.47	59	0.18	178	0.49
chr16	155	0.36	64	0.15	174	0.40
chr17	252	0.40	124	0.20	285	0.44
chr18	66	0.40	20	0.13	90	0.55
chr19	271	0.38	142	0.20	302	0.42
chr20	112	0.40	46	0.17	137	0.49
chr21	48	0.44	21	0.20	44	0.40
chr22	94	0.38	65	0.25	115	0.43
chrM	37	0.56	3	0.06	41	0.65
chrX	161	0.43	84	0.22	180	0.48

Table A.14: Numbers and percentages of peaks for each chromosome considered significantly different using a test of equal proportions.

	HPSCvsMyelo	HPSCvsErythro	MyeloVSErythro
1	1244	1254	1293
2	841	875	903
3	685	660	711
4	470	474	491
5	616	601	642
6	699	692	721
7	564	567	583
8	396	402	407
9	454	459	482
10	512	514	540
11	698	686	724
12	678	660	688
13	221	219	232
14	397	409	411
15	407	385	415
16	458	446	486
17	719	699	747
18	176	182	174
19	788	765	798
20	322	310	324
21	123	115	118
22	280	281	307
23	85	55	85
24	407	422	435

Table A.15: Number of peaks detected as significantly different in each comparison for each chromosome with the Audic and Claverie test after FDR correction for multiple comparisons ($p < 0.05$).

	HPSCvsMyelo	HPSCvsErythro	MyelovsErythro
1	391	201	481
2	288	144	341
3	198	96	272
4	156	72	174
5	219	107	247
6	238	158	267
7	202	93	223
8	145	72	158
9	148	65	169
10	163	77	187
11	268	119	251
12	215	100	242
13	89	37	95
14	133	53	151
15	149	48	162
16	129	50	151
17	210	102	241
18	51	16	71
19	236	119	258
20	97	42	123
21	42	16	35
22	77	48	94
23	36	0	36
24	134	71	161

Table A.16: Number of peaks detected as significantly different expressed in each comparison for each chromosome both with the Audic and Claverie test after FDR correction for multiple comparisons ($p < 0.05$) on raw data and with the test of equal proportions using normalized data.

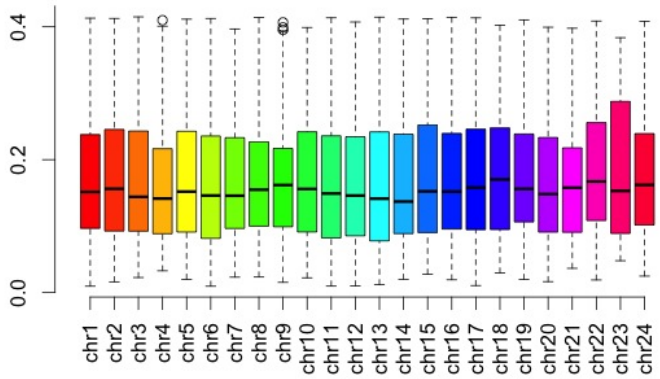

Figure A.33: Boxplots of the detected effect sizes for each chromosome in the comparison across the three cellular lines, using a χ^2_2 test with $\alpha = 0.05$ and $1 - \beta = 0.8$.

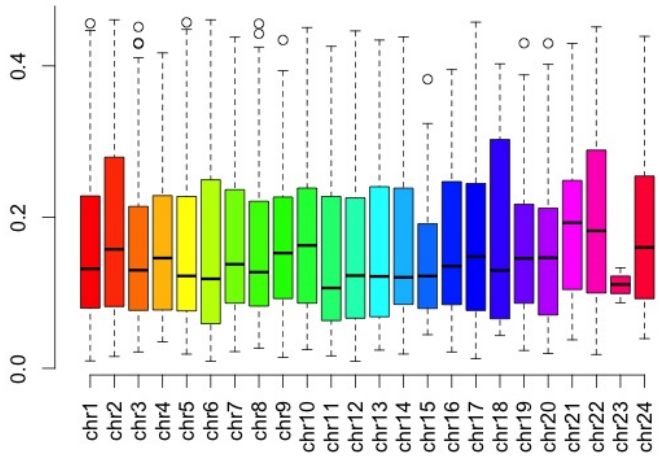

Figure A.34: Boxplots of the detected effect sizes for each chromosome in the comparison between HPSC and Erythroblasts, using the test of proportions with $\alpha = 0.05$ and $1 - \beta = 0.8$.

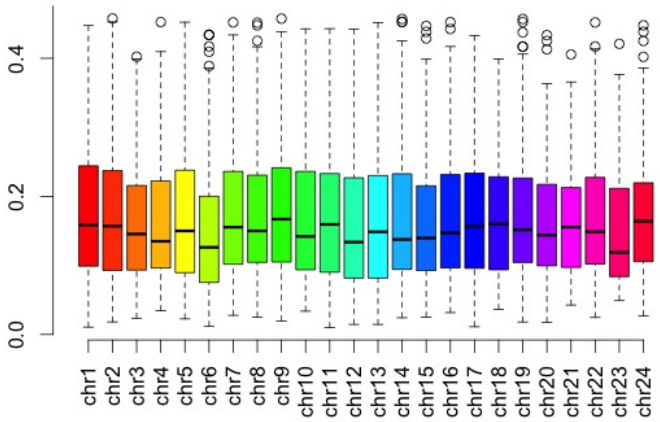

Figure A.35: Boxplots of the detected effect sizes for each chromosome in the comparison between HPSC and Myeloblasts, using the test of proportions with $\alpha = 0.05$ and $1 - \beta = 0.8$.

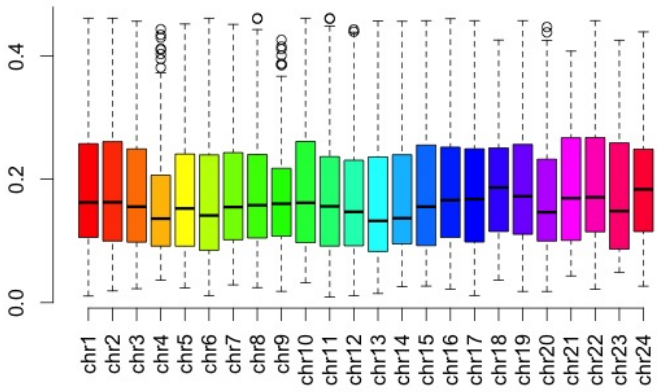

Figure A.36: Boxplots of the detected effect sizes for each chromosome in the comparison between Erythroblasts and Myeloblasts, using the test of proportions with $\alpha = 0.05$ and $1 - \beta = 0.8$.

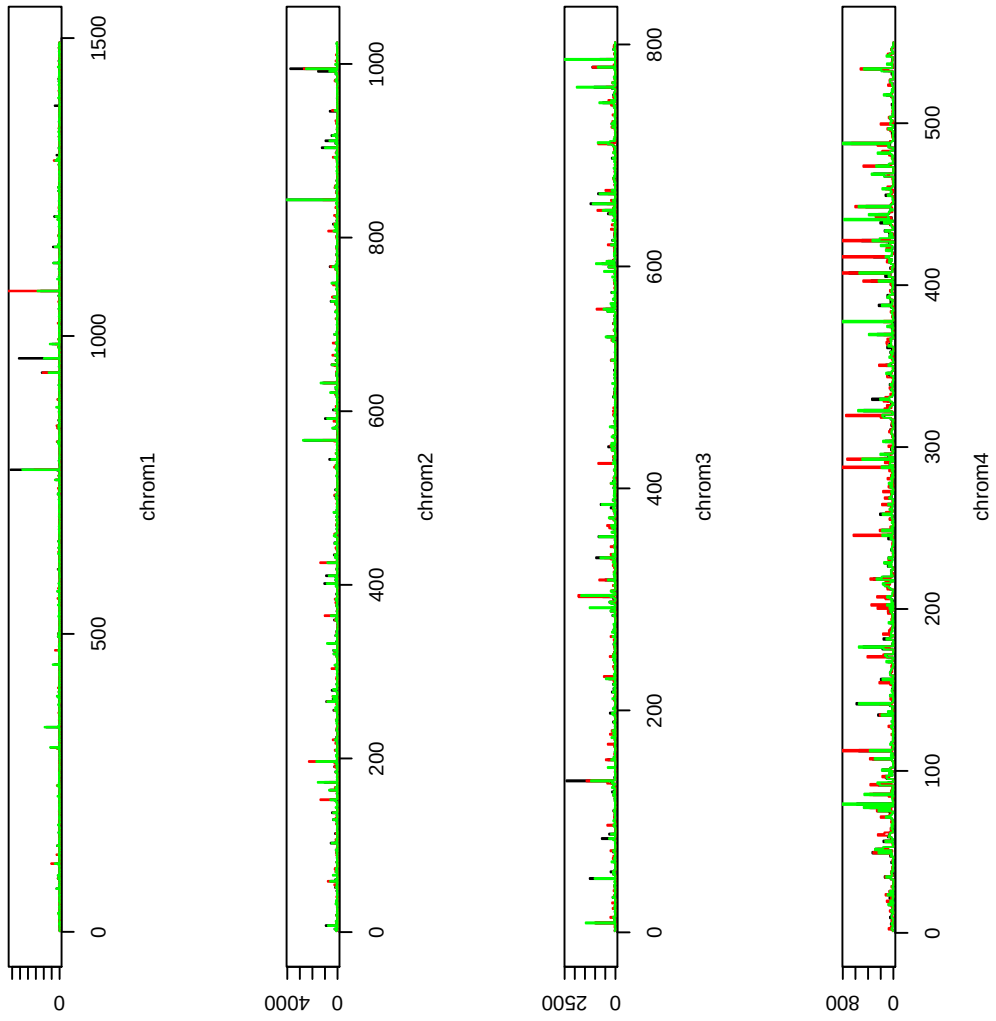

Figure A.37: Graphical representation of peaks in four chromosomes in the three cellular lines. The black line corresponds to the HPSC cell line, the red one to the myeloblasts and the green one to erythroblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

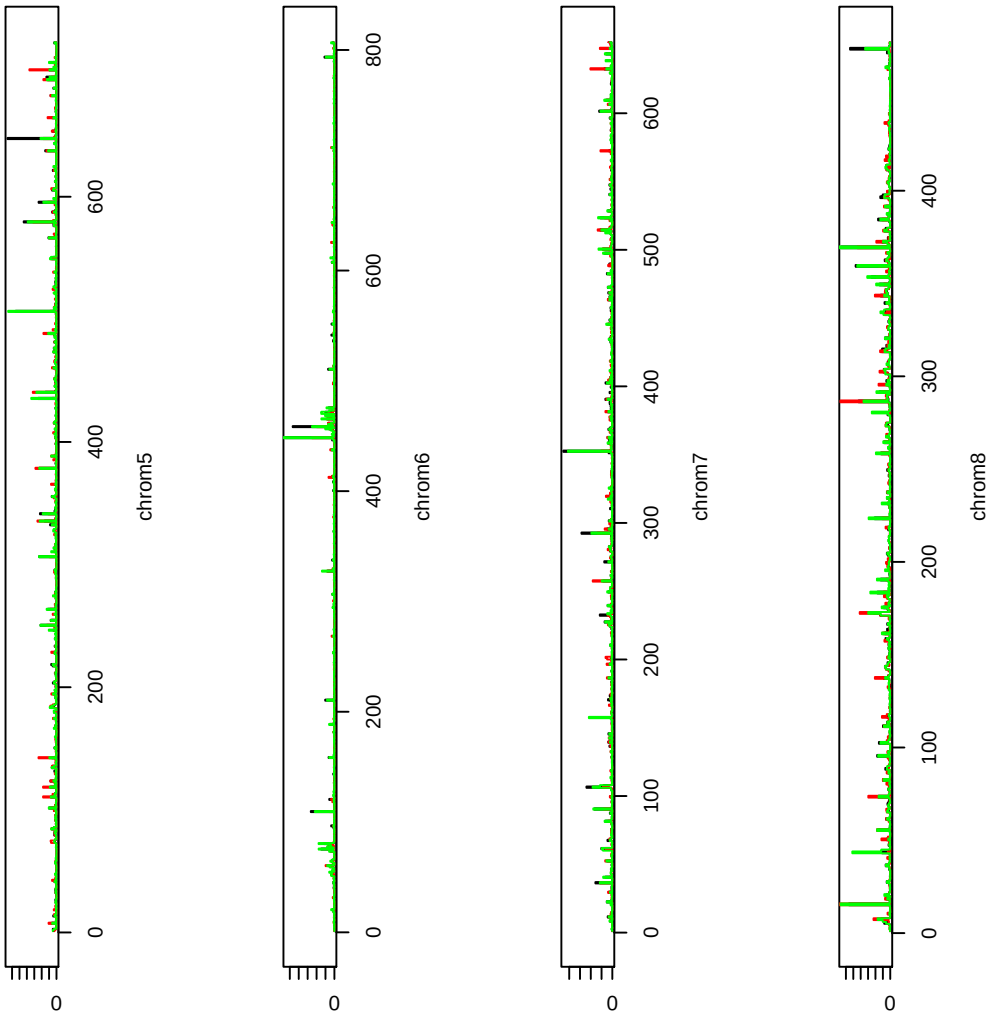

Figure A.38: Graphical representation of peaks in four chromosomes in the three cellular lines. The black line corresponds to the HPSC cell line, the red one to the myeloblasts and the green one to erythroblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

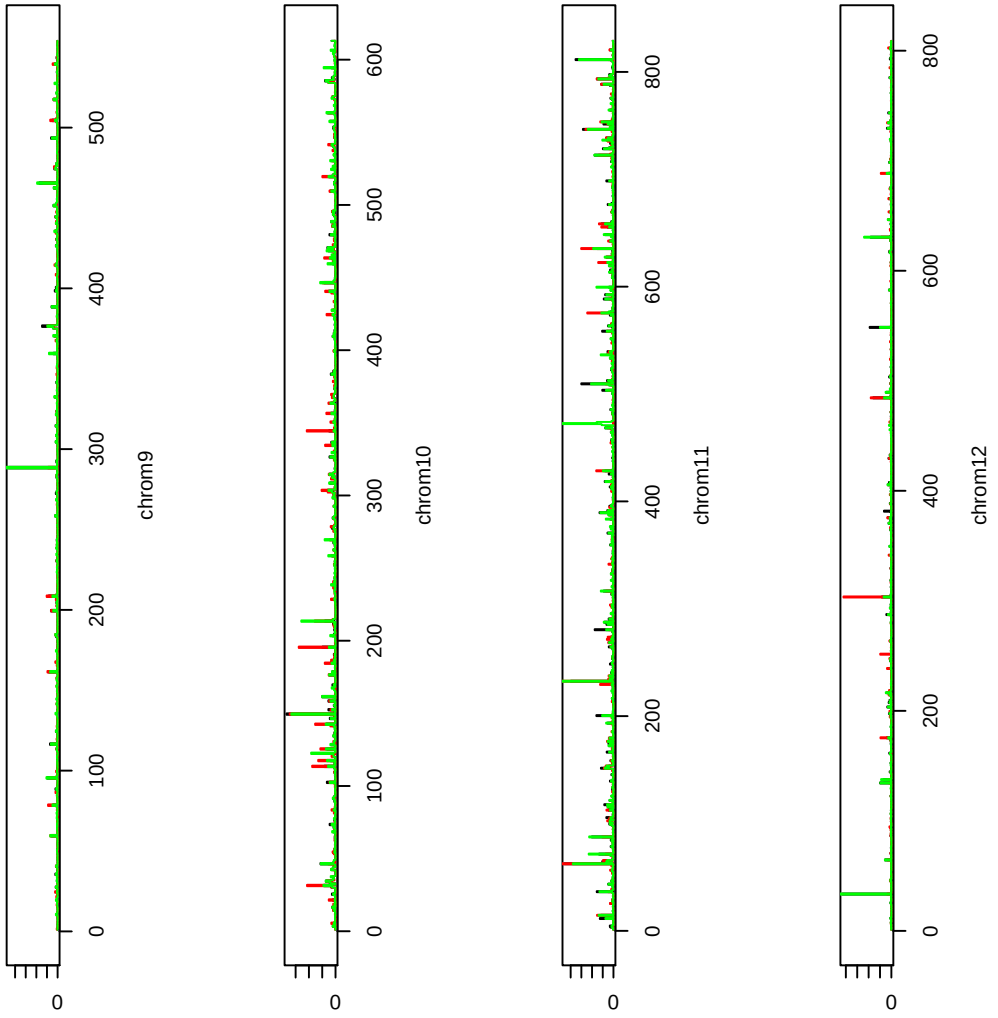

Figure A.39: Graphical representation of peaks in four chromosomes in the three cellular lines. The black line corresponds to the HPSC cell line, the red one to the myeloblasts and the green one to erythroblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

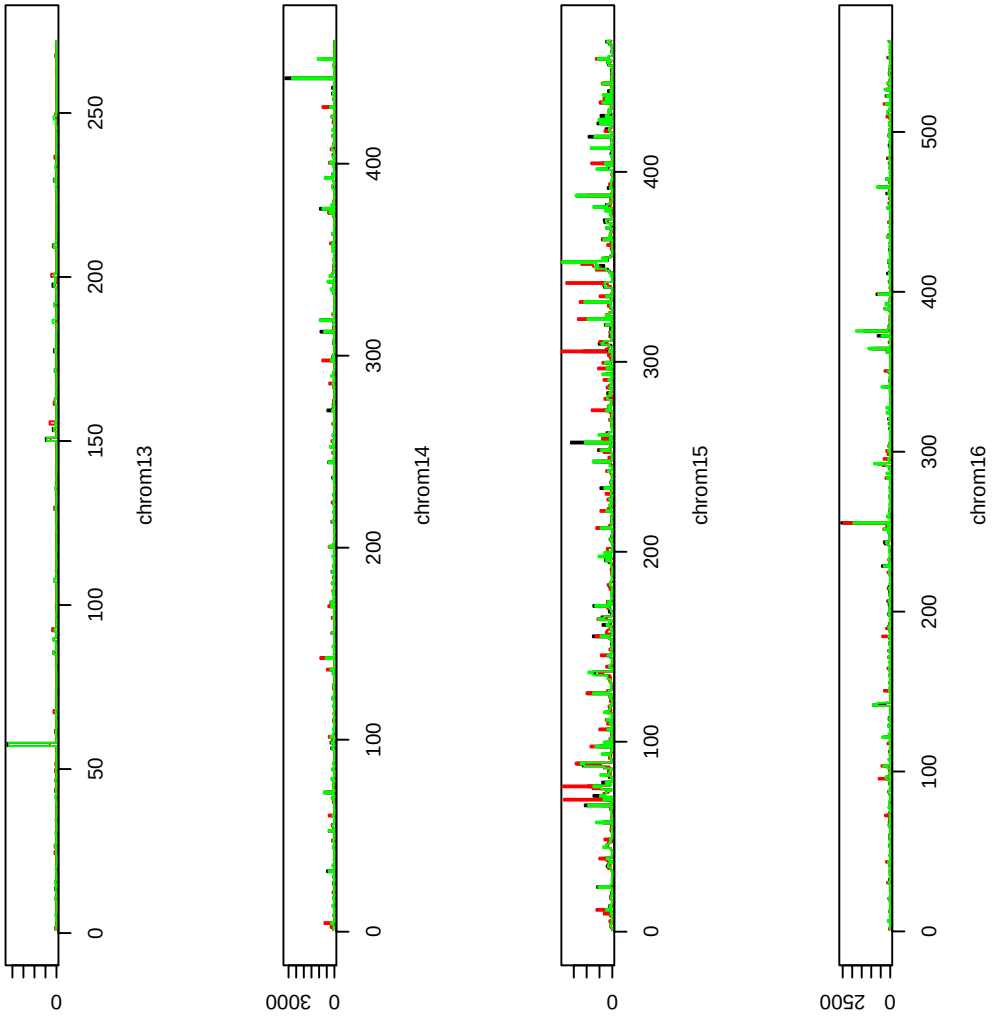

Figure A.40: Graphical representation of peaks in four chromosomes in the three cellular lines. The black line corresponds to the HPSC cell line, the red one to the myeloblasts and the green one to erythroblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

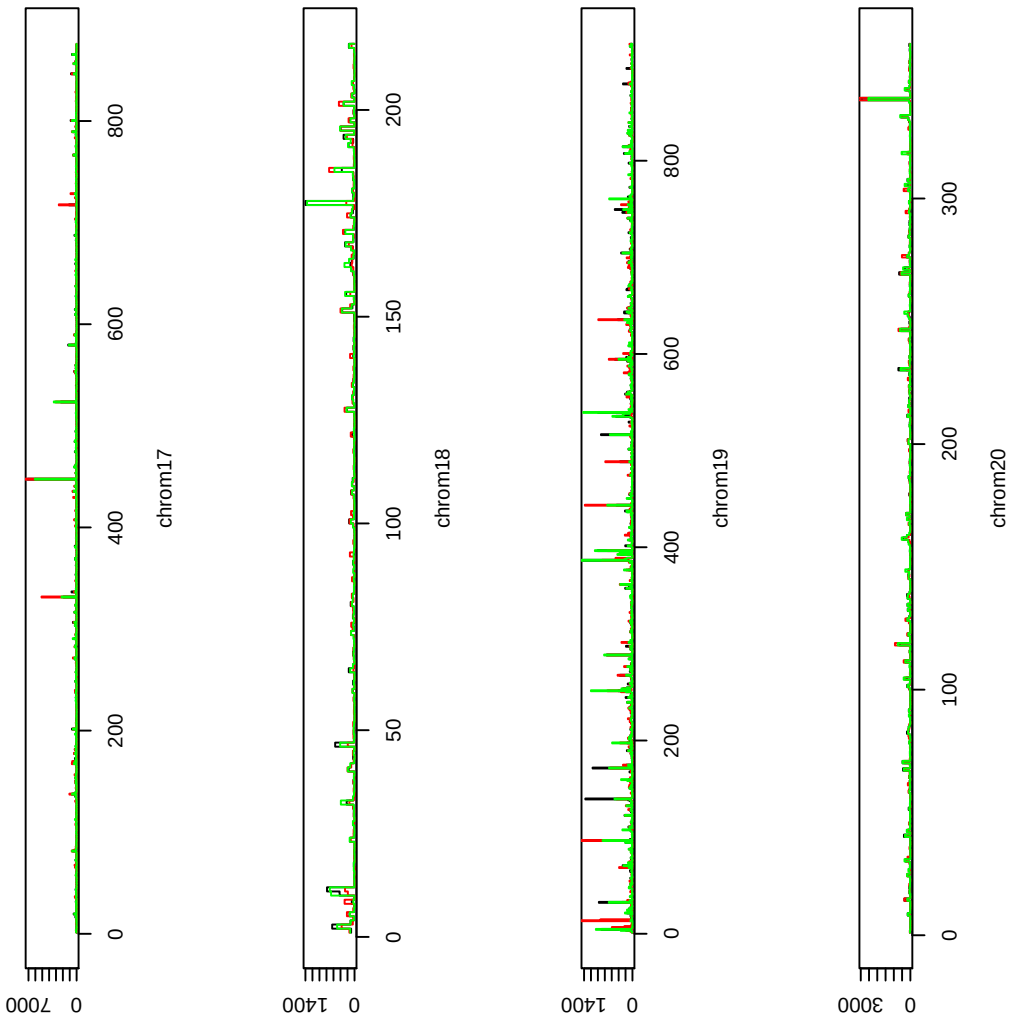

Figure A.41: Graphical representation of peaks in four chromosomes in the three cellular lines. The black line corresponds to the HPSC cell line, the red one to the myeloblasts and the green one to erythroblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

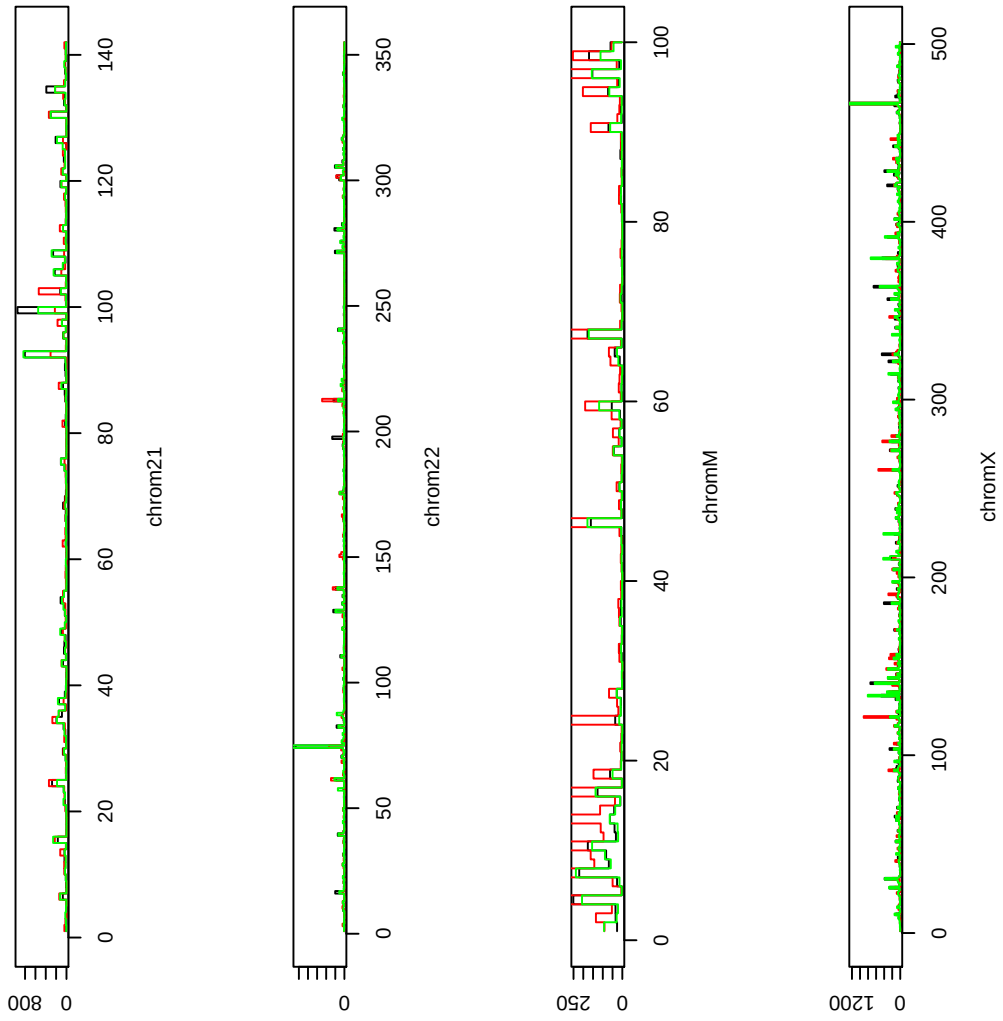

Figure A.42: Graphical representation of peaks in four chromosomes in the three cellular lines. The black line corresponds to the HPSC cell line, the red one to the myeloblasts and the green one to erythroblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

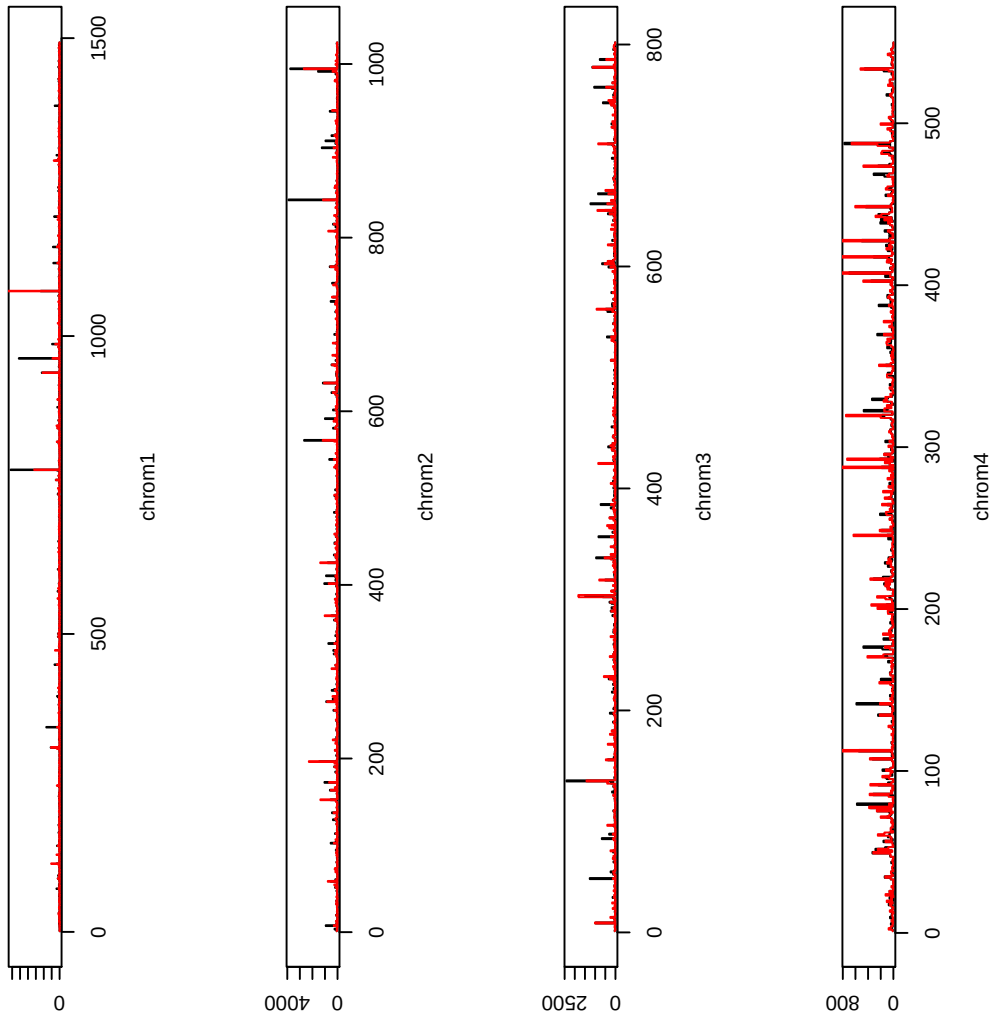

Figure A.43: Graphical representation of peaks in four chromosomes in two cellular lines. The black line corresponds to the HPSC cell line and the red one to the myeloblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

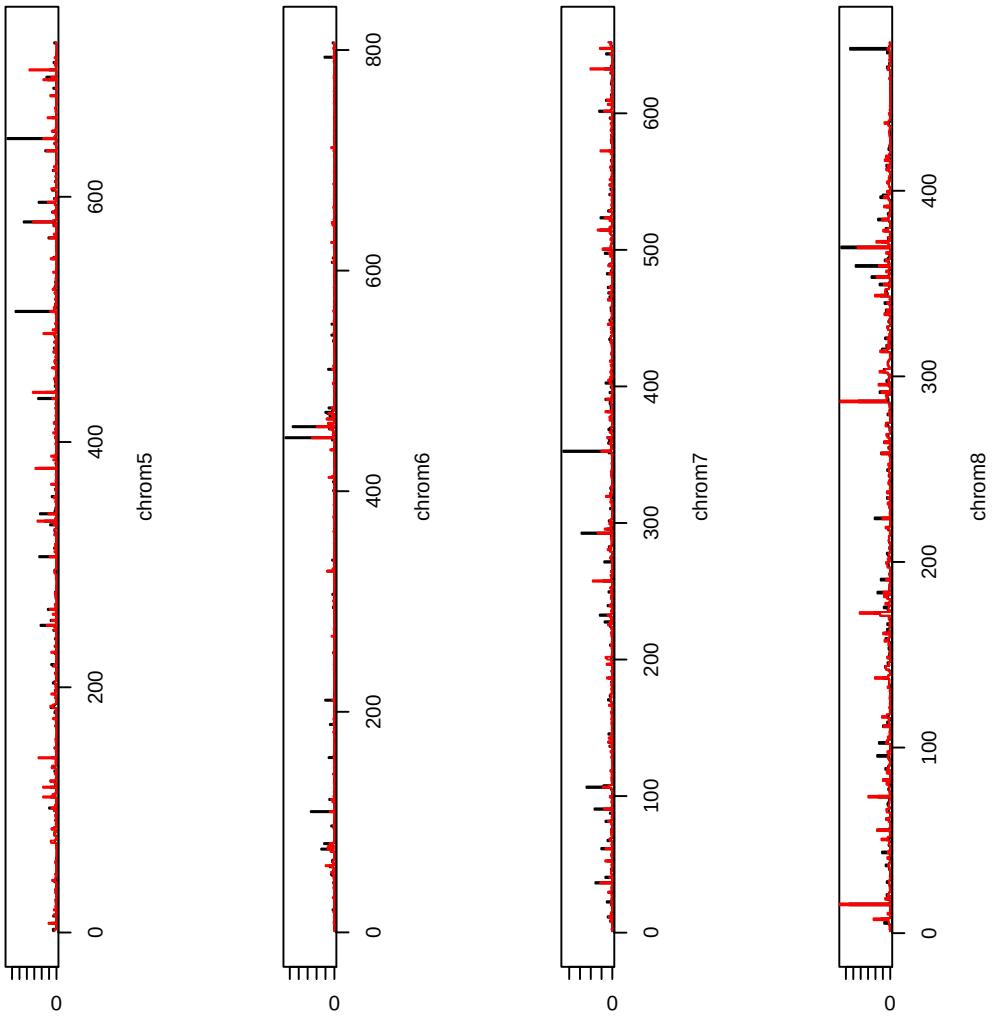

Figure A.44: Graphical representation of peaks in four chromosomes in two cellular lines. The black line corresponds to the HPSC cell line and the red one to the myeloblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

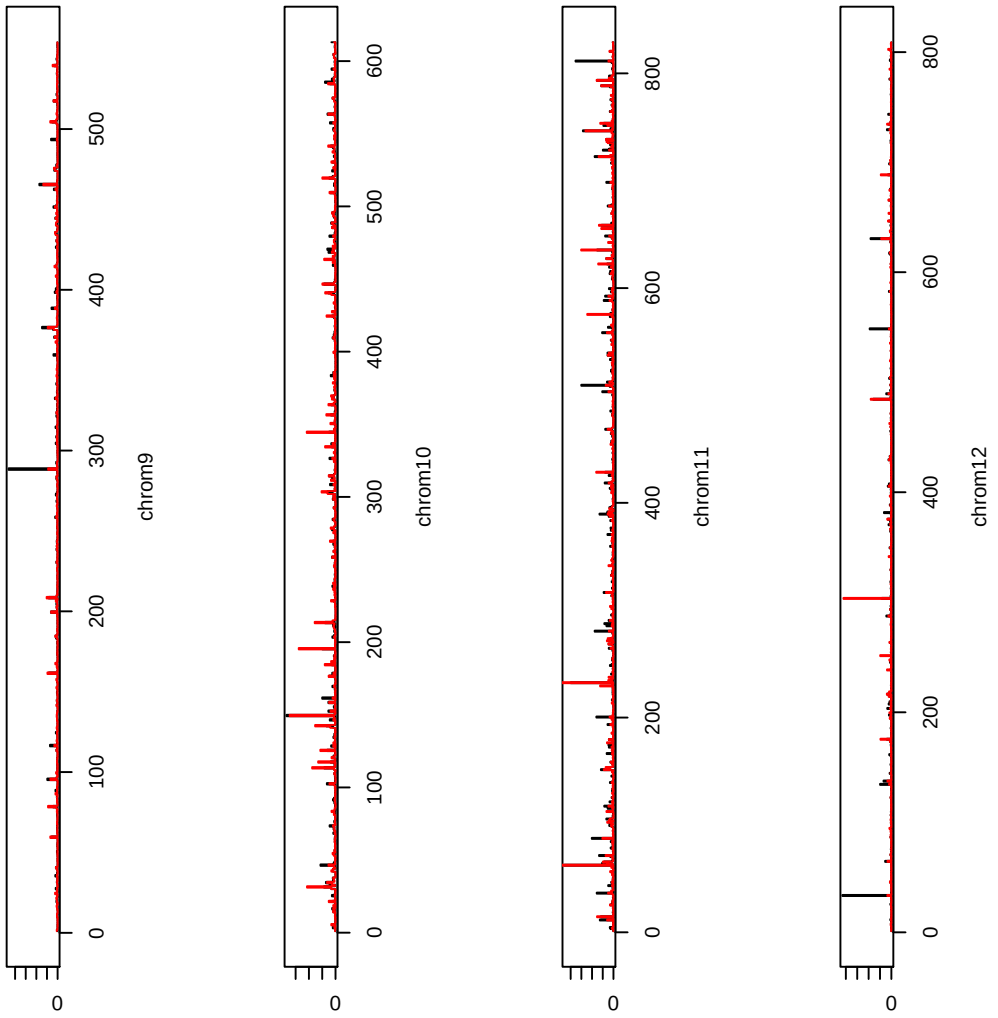

Figure A.45: Graphical representation of peaks in four chromosomes in two cellular lines. The black line corresponds to the HPSC cell line and the red one to the myeloblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

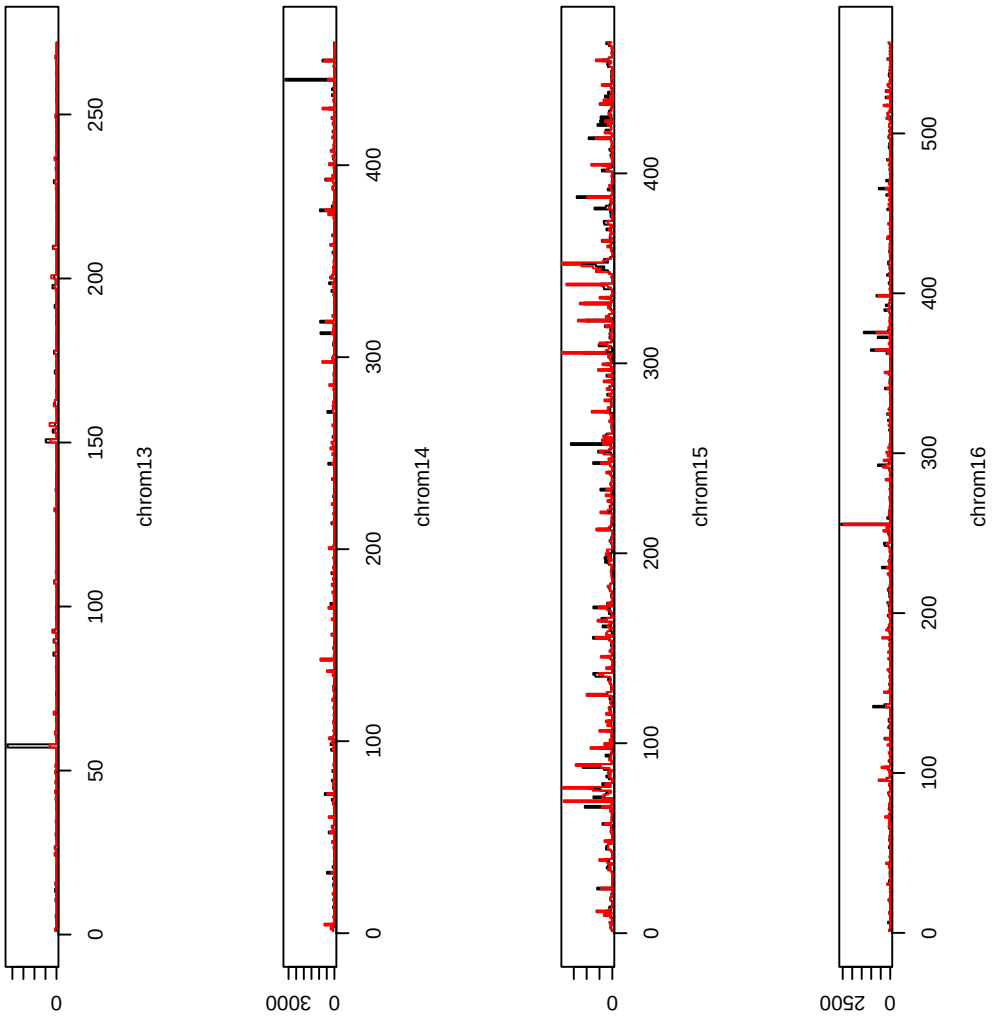

Figure A.46: Graphical representation of peaks in four chromosomes in two cellular lines. The black line corresponds to the HPSC cell line and the red one to the myeloblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

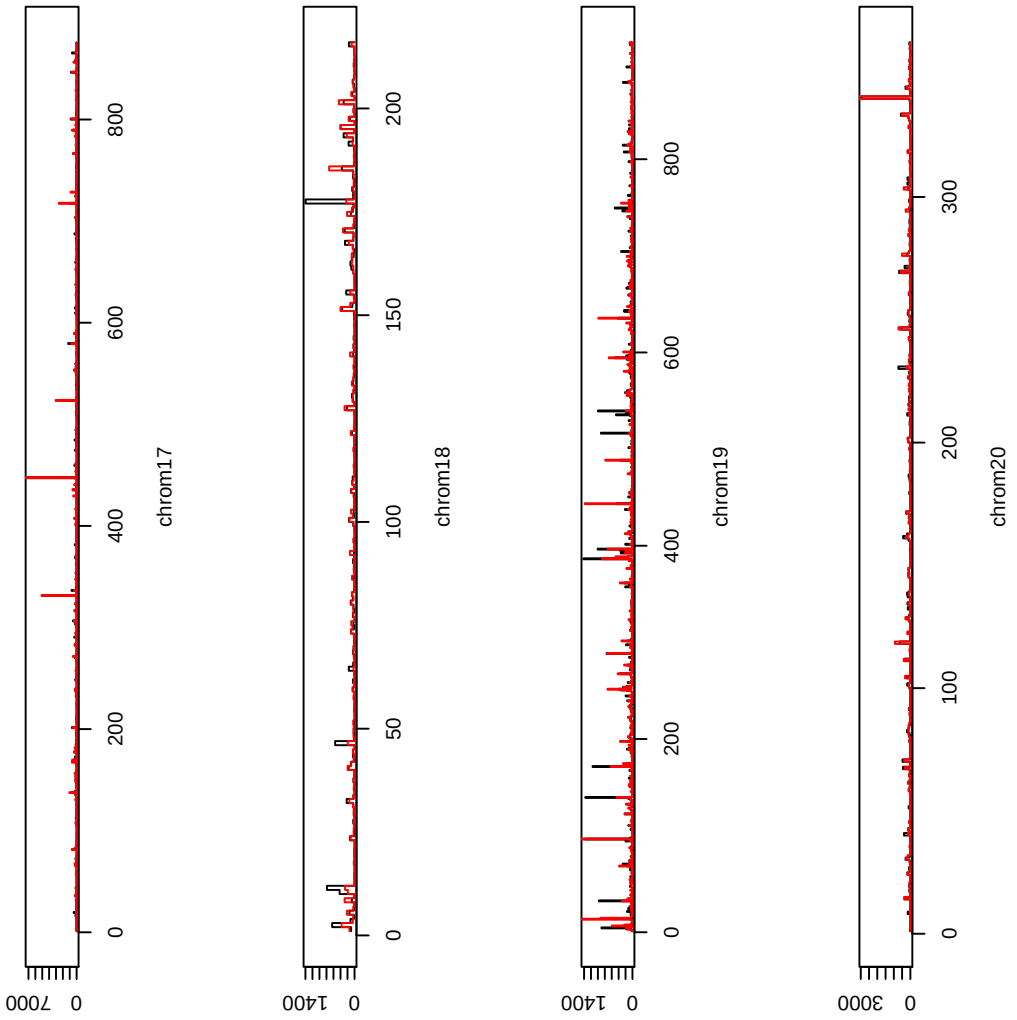

Figure A.47: Graphical representation of peaks in four chromosomes in two cellular lines. The black line corresponds to the HPSC cell line and the red one to the myeloblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

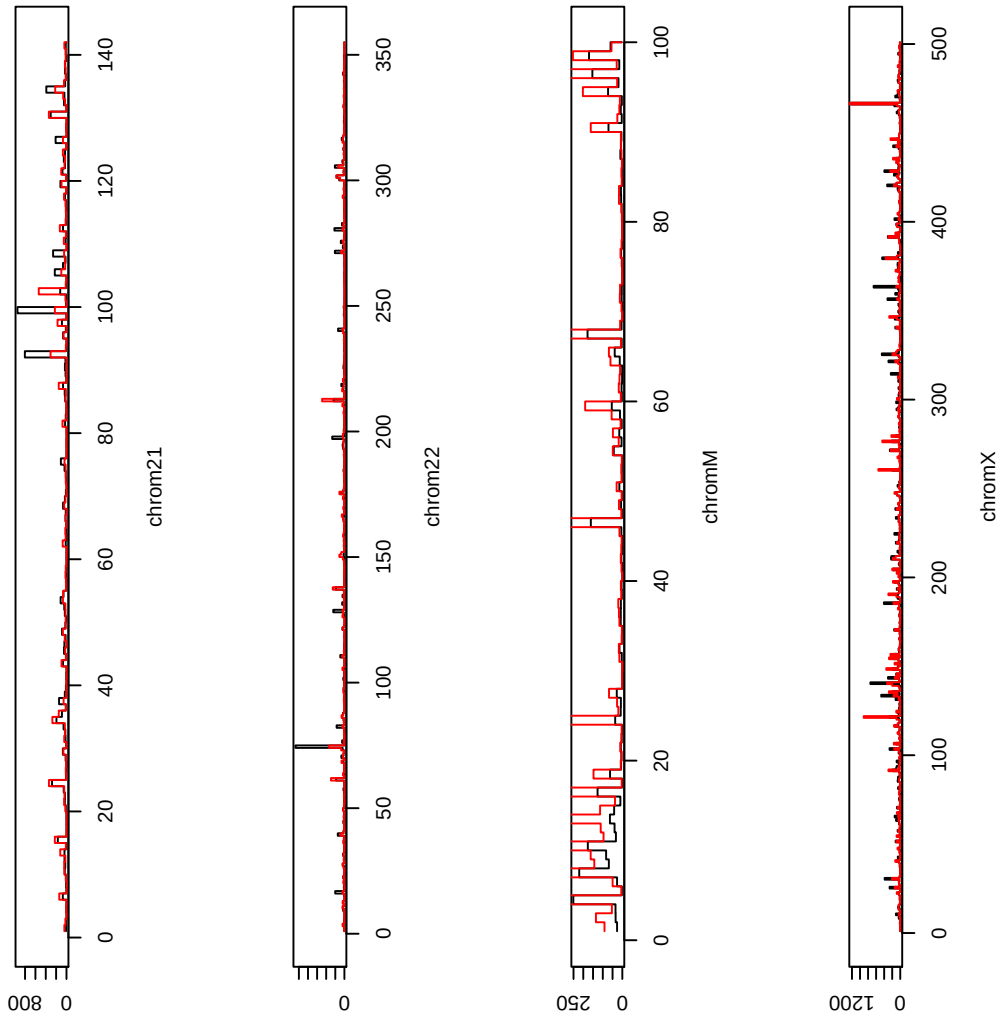

Figure A.48: Graphical representation of peaks in four chromosomes in two cellular lines. The black line corresponds to the HPSC cell line and the red one to the myeloblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

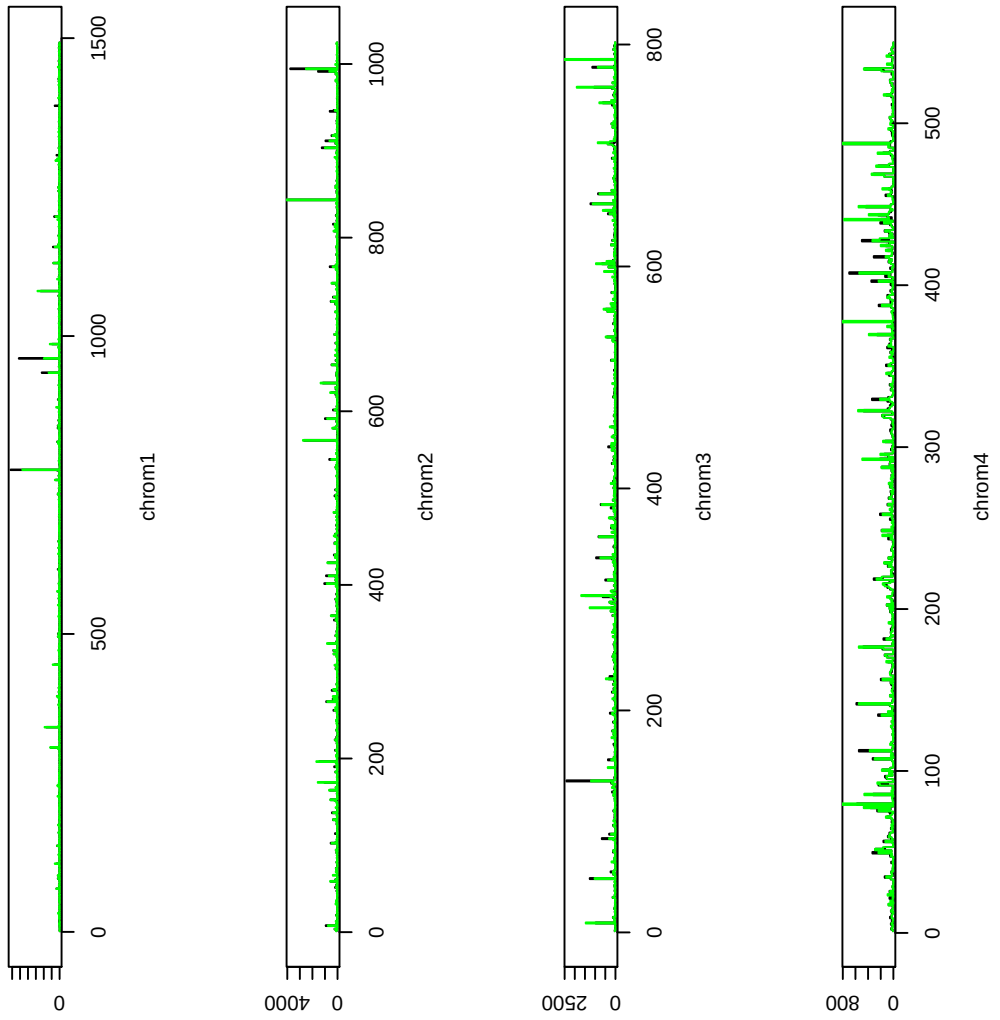

Figure A.49: Graphical representation of peaks in four chromosomes in two cellular lines. The black line corresponds to the HPSC cell line and the green one to the erythroblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

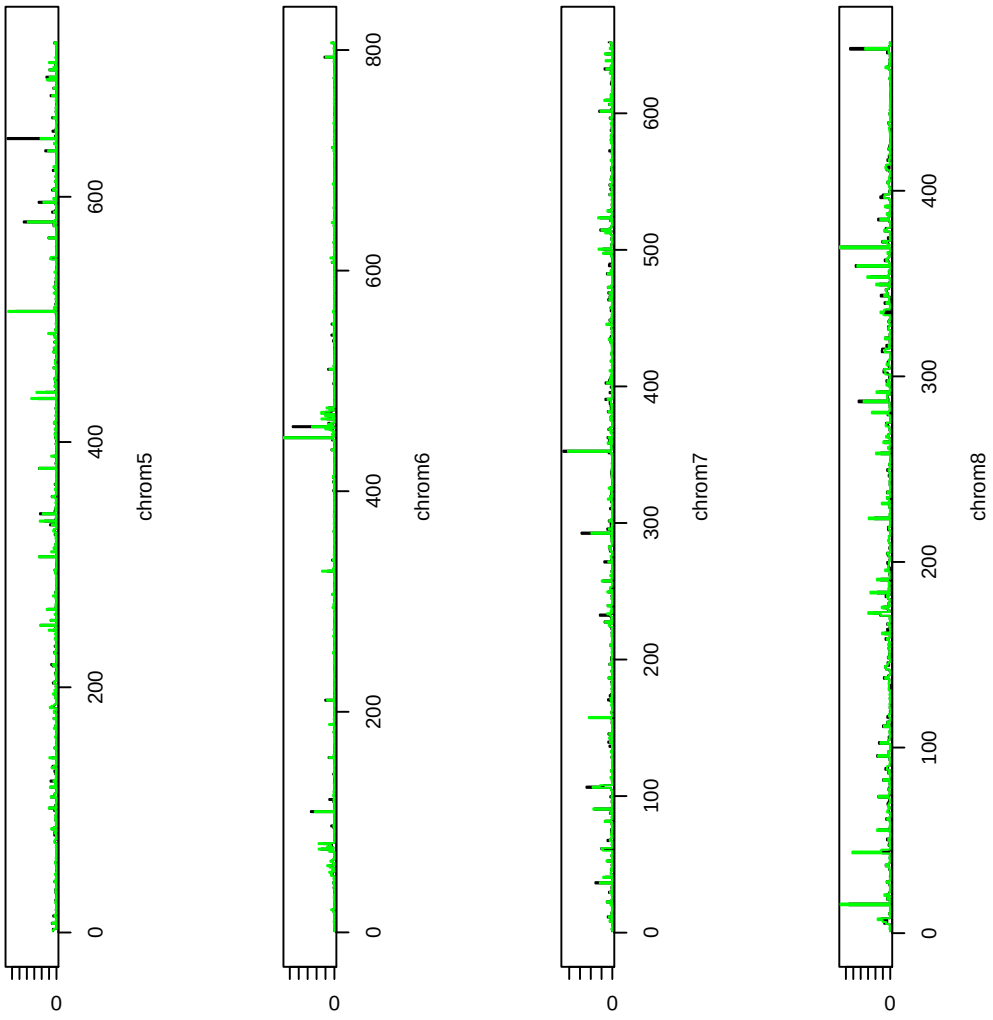

Figure A.50: Graphical representation of peaks in four chromosomes in two cellular lines. The black line corresponds to the HPSC cell line and the green one to the erythroblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

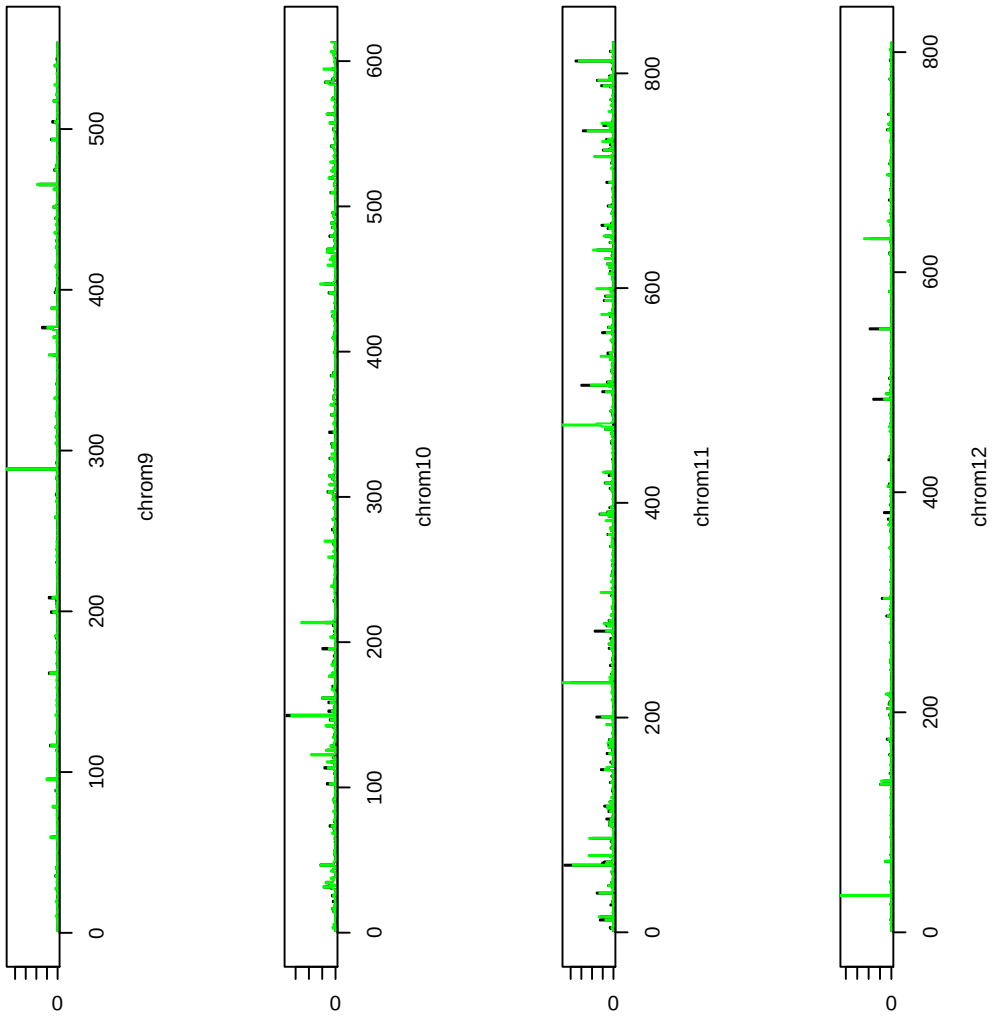

Figure A.51: Graphical representation of peaks in four chromosomes in two cellular lines. The black line corresponds to the HPSC cell line and the green one to the erythroblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

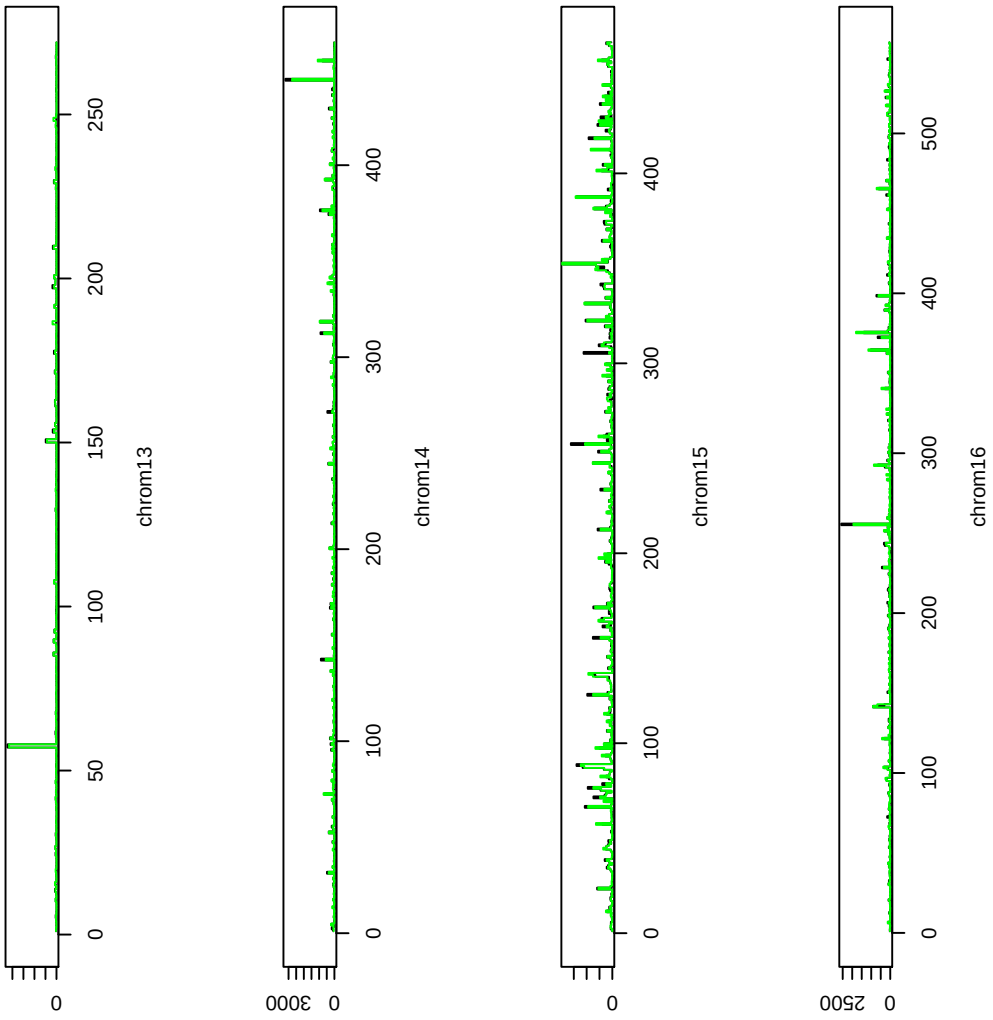

Figure A.52: Graphical representation of peaks in four chromosomes in two cellular lines. The black line corresponds to the HPSC cell line and the green one to the erythroblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

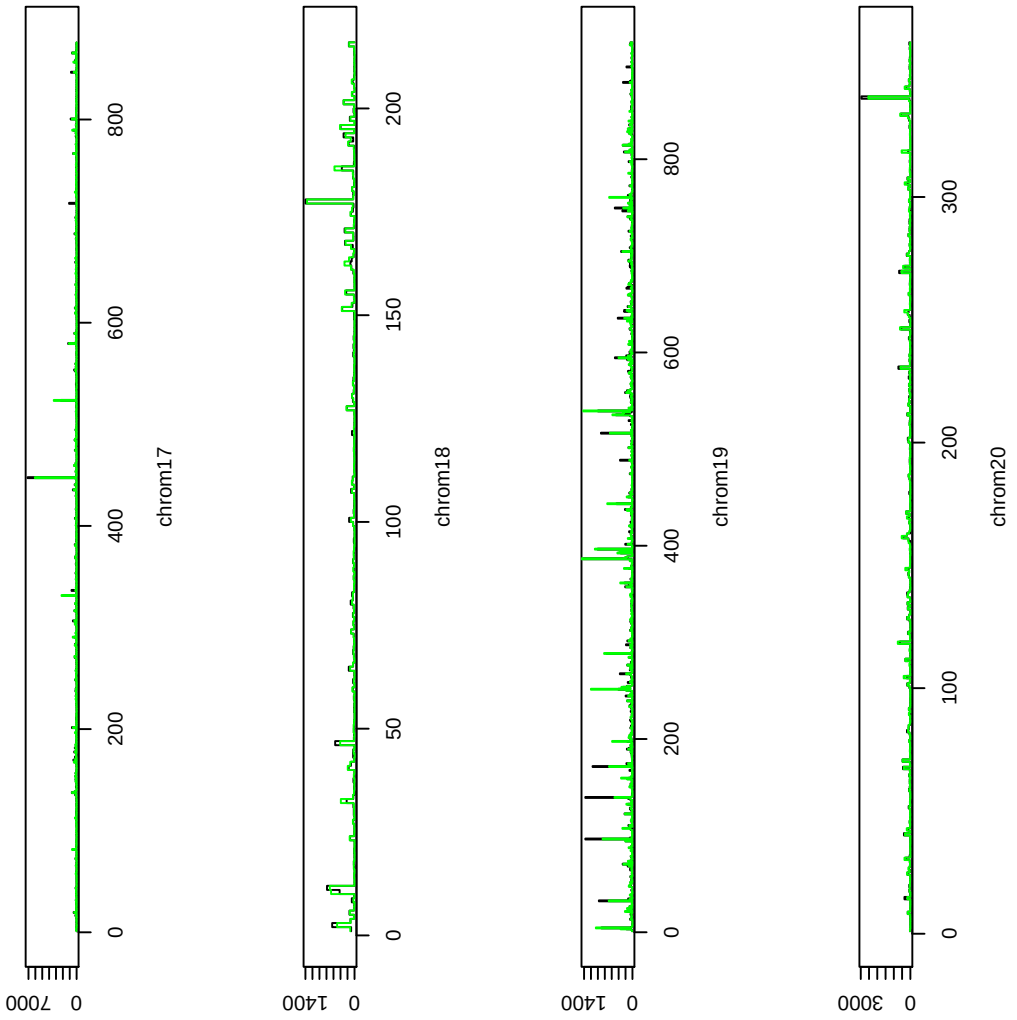

Figure A.53: Graphical representation of peaks in four chromosomes in two cellular lines. The black line corresponds to the HPSC cell line and the green one to the erythroblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

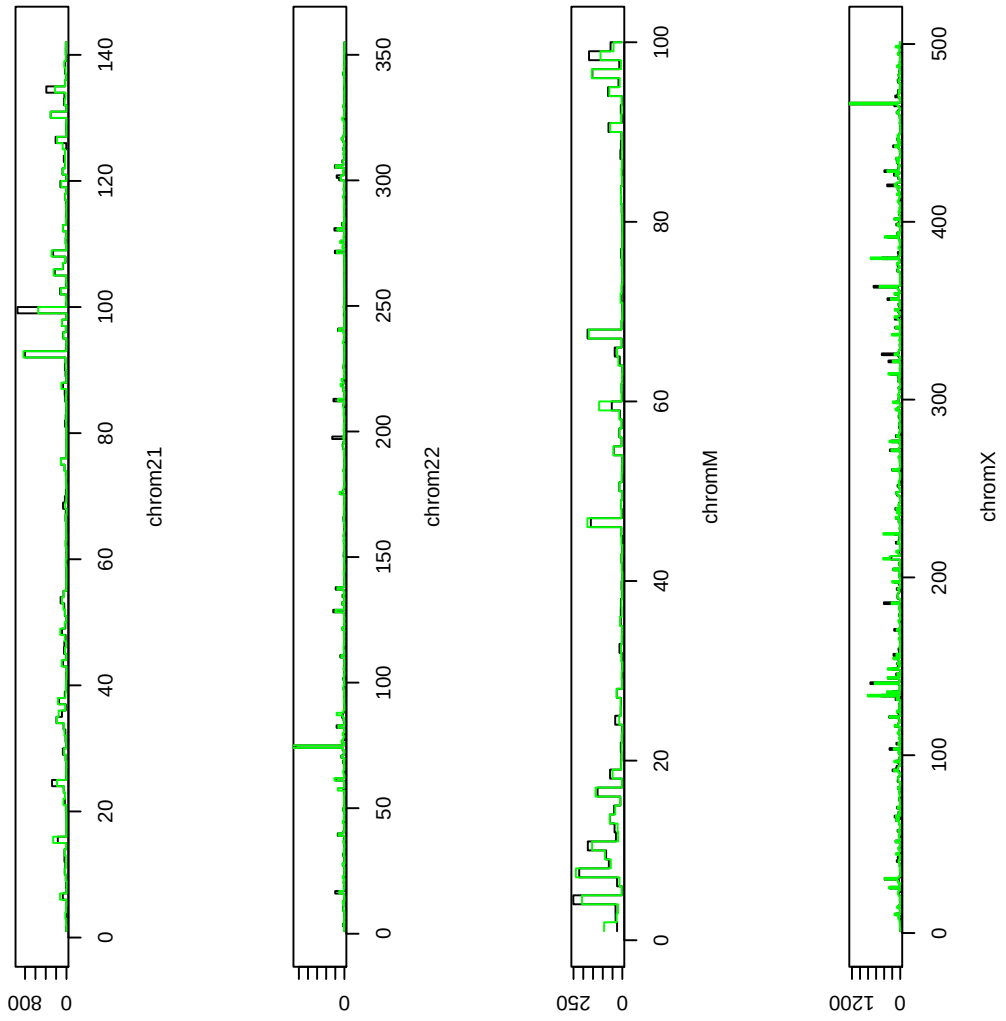

Figure A.54: Graphical representation of peaks in four chromosomes in two cellular lines. The black line corresponds to the HPSC cell line and the green one to the erythroblasts. On the x-axis the number of peaks is reported, on the y-axis the height of the peaks is reported.

Appendix B

Code

```
# The following function identifies in which of the clusters of a list each gene is.

geneinlista <- function(mygene, lista)        {

  which( sapply( lista, function( gene, elem ) gene %in% elem, gene = mygene ) )

                                                }

# The following function compute the number of elements classified.

max.proportion.function <- function( k, beta.clu.ward, beta.clu.complete,
                                     dist, return.list = F ) {
  k.w <- k[ 1 ]
  k.c <- k[ 2 ]
  tree.ward <- cutree( beta.clu.ward, k = k.w )
  tree.complete <- cutree( beta.clu.complete, k = k.c )
  N <- sum( tree.ward*0 + 1)
  A <- table( tree.ward, tree.complete )
  A.star <- diag( 0, k.w )
  if( return.list == T ) beta.list <- list( )
  i <- 1
  for( i in 1 : k.w )        {

    A.star[ i, i ] <- max( A )
    A.max <- which( A == max( A ), arr.ind = TRUE )[ 1, ]
    r.max <- A.max[ 1 ]
```

```
    c.max <- A.max[ 2 ]
    if( return.list == T )
      beta.list[ [ i ] ] <- ( 1:N )[ ( tree.ward == r.max ) &
                                ( tree.complete == c.max ) ]
    A[ r.max, ] <- 0
    A[ ,c.max ] <- 0

                                  }

  while ( ( A.star[ nrow( A.star ), ncol( A.star ) ] == 0 ) ) {

  # in case of an empty cluster, it eliminates it

    A.star <- A.star[ -( nrow( A.star ) ), -( ncol( A.star ) ) ]

      if( return.list == T )
      beta.list = beta.list[ -length( beta.list ) ]

      if( is.null( dim( A.star ) ) )
      break

      if( return.list == T )
      return( list( "beta.list" = beta.list, "A.star" = A.star ) )

  sum( A.star )
                                                                              }


# This is the core function.

CrossClustering <- function( d, k.w.min, k.w.max, k.c.max, out = TRUE ) {
  require( cluster )
  n <- ( 1 + sqrt( 1 + 8*length( d ) ) ) / 2
  beta.clu.ward <- hclust( d, method = "ward" )
  beta.clu.complete <- hclust( d, method = "complete" )
  grid <- as.matrix( expand.grid( k.w.min:k.w.max, k.w.min:k.c.max ) )

  if ( out == T )
            grid <- grid[ grid[ ,2 ] > grid[ ,1 ], ]
  else  grid <- grid[ grid[ ,2 ] >= grid[ ,1 ], ]

  grid <- cbind( grid,0 )
```

```
  colnames( grid ) <- c( "Ward", "Complete", "N. classified" )
  n.clu <- NULL

  for( i in 1:dim( grid )[ 1 ] )        {

    n.clu[ i ] <- max.proportion.function( grid[ i, ],
                      beta.clu.ward = beta.clu.ward,
                      beta.clu.complete = beta.clu.complete )

                                          }

  grid[ ,3 ] <- n.clu
  grid.star <- which( grid == max( grid[ ,3 ] ), arr.ind = TRUE )[ , 1 ]
  k.star <- rbind( grid[ grid.star, 1:2 ] )
  if( is.null( dim( k.star ) ) ) {

                cluster.list <- max.proportion.function( k.star,
                  beta.clu.ward = beta.clu.ward,
                  beta.clu.complete = beta.clu.complete,
                  return.list = T )

                clustz <- sapply( 1:n, geneinlista, cluster.list$beta.list )

                                } else {

                cluster.list <- apply( k.star, 1, max.proportion.function,
                  beta.clu.ward = beta.clu.ward,
                  beta.clu.complete = beta.clu.complete,
                  return.list = T )

                clustz <- sapply( cluster.list, function( lasim )
                  sapply( 1:n, geneinlista, lista = lasim$beta.list ) )

                                        }

clustz[ clustz == "integer( 0 )" ] <- 0

if( is.null( dim( clustz ) ) ) {
               clustz <- matrix( clustz, ncol = 1 )
                               }
```

```
Sil <- list( )

for ( c in 1:ncol( clustz ) ) {
              Sil[c] <- mean( silhouette( as.numeric( clustz[ ,c ] ),
                        dist = d )[ ,3 ] )
                               }

if( is.null( dim( k.star ) ) ) {
              k.star.star <- k.star[ which.max( Sil ) ]

                              } else {

              k.star.star <- k.star[ which.max( Sil ), ]

                                      }

  Cluster.list <- cluster.list[ [ which.max( Sil ) ] ]$beta.list
  n.clustered <-length( unlist( Cluster.list ) )
  return( list( "Optimal.cluster" = length( cluster.list[ [ which.max( Sil ) ] ]
              $beta.list ),
              "Cluster.list" = Cluster.list,
              "Silhouette" = max( unlist( Sil ) ),
              "n.total" = n,
              "n.clustered" = n.clustered ) )

}
```

Bibliography

AACH, J. & CHURCH, G. M. (2001). Aligning gene expression time series with time warping algorithms. *Bioinformatics* **17**, 495–508.

ABRAHAM, C., CORNILLON, P.-A., MATZNER-LØBER, E. & MOLINARI, N. (2003). Unsupervised curve clustering using b-splines. *Scandinavian journal of statistics* **30**, 581–595.

AKAIKE, H. (1998). Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*. Springer, pp. 199–213.

ALBERTS, B., BRAY, D., HOPKIN, K., JOHNSON, A., LEWIS, J., RAFF, M., ROBERTS, K. & WALTER, P. (2013). *Essential cell biology*. Garland Science.

ALTMAN, D. G. & BLAND, J. M. (1983). Measurement in medicine: the analysis of method comparison studies. *The statistician* , 307–317.

ARBEITMAN, M. N., FURLONG, E. E., IMAM, F., JOHNSON, E., NULL, B. H., BAKER, B. S., KRASNOW, M. A., SCOTT, M. P., DAVIS, R. W. & WHITE, K. P. (2002). Gene expression during the life cycle of drosophila melanogaster. *Science* **297**, 2270–2275.

ARNAU, J. & BONO, R. (2001). Autocorrelation and bias in short time series: An alternative estimator. *Quality and Quantity* **35**, 365–387.

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. et al. (2000). Gene ontology: tool for the unification of biology. *Nature genetics* **25**, 25–29.

AUDIC, S. & CLAVERIE, J.-M. (1997). The significance of digital gene expression profiles. *Genome research* **7**, 986–995.

BALWIERZ, P. J., CARNINCI, P., DAUB, C. O., KAWAI, J., HAYASHIZAKI, Y., VAN BELLE, W., BEISEL, C. & VAN NIMWEGEN, E. (2009). Methods for analyzing deep sequencing expression data: constructing the human and mouse promoterome with deepcage data. *Genome biol* **10**, R79.

BAR-JOSEPH, ZIV, G. G., GIFFORD, K., JAAKKOLA, T. S. & SIMON, I. (2003). Continuous

representations of time series gene expression data. *Journal of Computational Biology* .

BARENCO, M., TOMESCU, D., BREWER, D., CALLARD, R., STARK, J. & HUBANK, M. (2006). Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome biology* **7**, R25.

BARRETT, T., SUZEK, T. O., TROUP, D. B., WILHITE, S. E., NGAU, W. C., LEDOUX, P., RUDNEV, D., LASH, A. E., FUJIBUCHI, W. & EDGAR, R. (2005). NCBI GEO: mining millions of expression profiles–database and tools. *Nucleic Acids Res* **33**, D562–6. 1362-4962.

BEALE, E. (1969). *Euclidean cluster analysis*. Scientific Control Systems Limited.

BENJAMINI, Y. & HOCHBERG, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of The Royal Statistical Society B* **57**, 289–300.

BORGATTA, M., HERNANDEZ, C., DECOSTERD, L. A., CHÈVRE, N. & WARIDEL, P. (2014). Shotgun ecotoxicoproteomics of daphnia pulex: biochemical effects of the anticancer drug tamoxifen. *Journal of proteome research* .

CALIŃSKI, T. & HARABASZ, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods* **3**, 1–27.

CARNINCI, P. (2006). Tagging mammalian transcription complexity. *TRENDS in Genetics* **22**, 501–510.

CARNINCI, P. (2009). *Cap-Analysis Gene Expression (CAGE): The Science of Decoding Genes Transcription*. Pan Stanford Publishing.

CARNINCI, P., KVAM, C., KITAMURA, A., OHSUMI, T., OKAZAKI, Y., ITOH, M., KAMIYA, M., SHIBATA, K., SASAKI, N., IZAWA, M. et al. (1996). High-efficiency full-length cdna cloning by biotinylated cap trapper. *Genomics* **37**, 327–336.

CARNINCI, P., SANDELIN, A., LENHARD, B., KATAYAMA, S., SHIMOKAWA, K., PONJAVIC, J., SEMPLE, C. A., TAYLOR, M. S., ENGSTRÖM, P. G., FRITH, M. C. et al. (2006). Genome-wide analysis of mammalian promoter architecture and evolution. *Nature genetics* **38**, 626–635.

CHANG, F., QIU, W., ZAMAR, R. H., LAZARUS, R. & WANG, X. (2010). clues: An R package for nonparametric clustering based on local shrinking. *Journal of Statistical Software* **33**, 1–16.

CHO, R. J., CAMPBELL, M. J., WINZELER, E. A., STEINMETZ, L., CONWAY, A., WODICKA, L., WOLFSBERG, T. G., GABRIELIAN, A. E., LANDSMAN, D., LOCKHART, D. J. et al. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular cell* **2**,

65–73.

CHU, S., DERISI, J., EISEN, M., MULHOLLAND, J., BOTSTEIN, D., BROWN, P. O. & HERSKOWITZ, I. (1998). The transcriptional program of sporulation in budding yeast. *Science* **282**, 699–705.

CIAMPI, A., CAMPBELL, H., DYACHENKO, A., RICH, B., MCCUSKER, J. & COLE, M. (2012). Model-based clustering of longitudinal data: Application to modeling disease course and gene expression trajectories. *Communications in Statistics-Simulation and Computation* **41**, 992–1005.

CLAUSET, A., SHALIZI, C. R. & NEWMAN, M. E. (2009). Power-law distributions in empirical data. *SIAM review* **51**, 661–703.

COFFEY, N., HINDE, J. & HOLIAN, E. (2014). Clustering longitudinal profiles using p-splines and mixed effects models applied to time-course gene expression data. *Computational Statistics & Data Analysis* **71**, 14–29.

COHEN, J. (2013). *Statistical power analysis for the behavioral sciences*. Routledge Academic.

CONESA, A., NUEDA, M. J., FERRER, A. & TALÓN, M. (2006). masigpro: a method to identify significantly differential expression profiles in time-course microarray experiments. *Bioinformatics* **22**, 1096–1102.

CONSORTIUM, G. O. et al. (2001). Creating the gene ontology resource: design and implementation. *Genome research* **11**, 1425–1433.

COOKE, E. J., SAVAGE, R. S., KIRK, P. D., DARKINS, R. & WILD, D. L. (2011). Bayesian hierarchical clustering for microarray time series data with replicates and outlier measurements. *BMC bioinformatics* **12**, 399.

COSTA, I. G., DE AT DE CARVALHO, F. & DE SOUTO, M. C. P. (2002). A symbolic approach to gene expression time series analysis. In *Neural Networks, 2002. SBRN 2002. Proceedings. VII Brazilian Symposium on*. IEEE.

COSTA, I. G., DE CARVALHO, F. D. A. & DE SOUTO, M. C. (2004). Comparative analysis of clustering methods for gene expression time course data. *Genetics and Molecular Biology* **27**, 623–631.

COX, D. & BARNDORFF-NIELSEN, O. (1994). *Inference and asymptotics*, vol. 52. CRC Press.

CRICK, F. et al. (1970). Central dogma of molecular biology. *Nature* **227**, 561–563.

DE HOON, M. J., IMOTO, S. & MIYANO, S. (2002). Statistical analysis of a small set of time-ordered gene expression data using linear splines. *Bioinformatics* **18**, 1477–1485.

DEMPSTER, A. P., LAIRD, N. M. & RUBIN, D. B. (1977). Maximum likelihood from incom-

plete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* , 1–38.

DRAGHICI, S. (2012). *Statistics and Data Analysis for Microarrays Using R and Bioconductor*. CRC Press.

DRĂGHICI, S. (2011). *Statistics and Data Analysis for Microarrays using R and Bioconductor*. Chapman and Hall/CRC Press.

DUBES, R. C. (1987). How many clusters are best? - an experiment. *Pattern Recognition* **20**, 645–663.

DUDA, R. O. & HART, P. E. (1973). *Pattern recognition and scene analysis*. Wiley.

DUNN, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics* **4**, 95–104.

D'URSO, P. & MAHARAJ, E. A. (2012). Wavelets-based clustering of multivariate time series. *Fuzzy Sets and Systems* **193**, 33–61.

EISEN, M. B., SPELLMAN, P. T., BROWN, P. O. & BOTSTEIN, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences* **95**, 14863–14868.

ERNST, J., NAU, G. J. & BAR-JOSEPH, Z. (2005). Clustering short time series gene expression data. *Bioinformatics* **21**, i159–i168.

ESTER, M., KRIEGEL, H.-P., SANDER, J. & XU, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, vol. 96.

EVERITT, B., LANDAU, S. & LEESE, M. (2001). Cluster analysis. 2001. *Arnold, London* .

FISHER, B., BRYANT, J., DIGNAM, J. J., WICKERHAM, D. L., MAMOUNAS, E. P., FISHER, E. R., MARGOLESE, R. G., NESBITT, L., PAIK, S., PISANSKY, T. M. et al. (2002). Tamoxifen, radiation therapy, or both for prevention of ipsilateral breast tumor recurrence after lumpectomy in women with invasive breast cancers of one centimeter or less. *Journal of clinical oncology* **20**, 4141–4149.

FISHER, B., COSTANTINO, J. P., WICKERHAM, D. L., CECCHINI, R. S., CRONIN, W. M., ROBIDOUX, A., BEVERS, T. B., KAVANAH, M. T., ATKINS, J. N., MARGOLESE, R. G. et al. (2005). Tamoxifen for the prevention of breast cancer: current status of the national surgical adjuvant breast and bowel project p-1 study. *Journal of the National Cancer Institute* **97**, 1652–1662.

FORGY, E. W. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics* **21**, 768–769.

FRIEDMAN, J. H. & SILVERMAN, B. W. (1989). Flexible parsimonious smoothing and additive modeling. *Technometrics* **31**, 3–21.

FRIEDMAN, N., LINIAL, M., NACHMAN, I. & PE'ER, D. (2000). Using bayesian networks to analyze expression data. *Journal of computational biology* **7**, 601–620.

FRITH, M. C., VALEN, E., KROGH, A., HAYASHIZAKI, Y., CARNINCI, P. & SANDELIN, A. (2008). A code for transcription initiation in mammalian genomes. *Genome research* **18**, 1–12.

GASCH, A. P., SPELLMAN, P. T., KAO, C. M., CARMEL-HAREL, O., EISEN, M. B., STORZ, G., BOTSTEIN, D. & BROWN, P. O. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Molecular biology of the cell* **11**, 4241–4257.

GOETZ, M. P., KNOX, S. K., SUMAN, V. J., RAE, J. M., SAFGREN, S. L., AMES, M. M., VISSCHER, D. W., REYNOLDS, C., COUCH, F. J., LINGLE, W. L. et al. (2007). The impact of cytochrome p450 2d6 metabolism in women receiving adjuvant tamoxifen. *Breast cancer research and treatment* **101**, 113–121.

GORDON, A. D. (1996). Null models in cluster validation. In *From data to knowledge*. Springer, pp. 32–44.

GYGI, S. P., CORTHALS, G. L., ZHANG, Y., ROCHON, Y. & AEBERSOLD, R. (2000). Evaluation of two-dimensional gel electrophoresis-based proteome analysis technology. *Proceedings of the National Academy of Sciences* **97**, 9390–9395.

GYGI, S. P., RIST, B., GERBER, S. A., TURECEK, F., GELB, M. H. & AEBERSOLD, R. (1999). Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. *Nature biotechnology* **17**, 994–999.

HABERLE, V., HABERLE, M. V., RSAMTOOLS, I., GENOMICRANGES, I., PREPROCESSING, S. & IMPORTFUNCTIONS, C. A. R. G. R. (2013). Package 'cager' .

HARTIGAN, J. A. (1975). Clustering algorithms .

HARTIGAN, J. A. & WONG, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Applied statistics* , 100–108.

HEARD, N. A., HOLMES, C. C. & STEPHENS, D. A. (2006). A quantitative study of gene regulation involved in the immune response of anopheline mosquitoes: An application of bayesian hierarchical clustering of curves. *Journal of the American Statistical Association* **101**, 18–29.

HEARD, N. A., HOLMES, C. C., STEPHENS, D. A., HAND, D. J. & DIMOPOULOS, G. (2005). Bayesian coclustering of anopheles gene expression time series: study of immune defense re-

sponse to multiple experimental challenges. *Proceedings of the National Academy of Sciences of the United States of America* **102**, 16939–16944.

HELLER, K. A. & GHAHRAMANI, Z. (2005). Bayesian hierarchical clustering. In *Proceedings of the 22nd international conference on Machine learning*. ACM.

HENNIG, C. (2014). *fpc: Flexible procedures for clustering*. R package version 2.1-7.

HENSMAN, J., LAWRENCE, N. D. & RATTRAY, M. (2013). Hierarchical bayesian modelling of gene expression time series across irregularly sampled replicates and clusters. *BMC bioinformatics* **14**, 252.

HEYER, L. J., KRUGLYAK, S. & YOOSEPH, S. (1999). Exploring expression data: identification and analysis of coexpressed genes. *Genome research* **9**, 1106–1115.

HÖPPNER, F. & KLAWONN, F. (2009). Compensation of translational displacement in time series clustering using cross correlation. In *Advances in Intelligent Data Analysis VIII*. Springer, pp. 71–82.

HOWLADER, N., NOONE, A., KRAPCHO, M., GARSHELL, J., MILLER, D., ALTEKRUSE, S., KOSARY, C., YU, M., RUHL, J., TATALOVICH, Z. et al. (2013). Seer cancer statistics review, 1975–2011.

HUBER, W., VON HEYDEBRECK, A., SÜLTMANN, H., POUSTKA, A. & VINGRON, M. (2002). Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics* **18**, S96–S104.

HUBERT, L. & ARABIE, P. (1985). Comparing partitions. *Journal of classification* **2**, 193–218.

HUBERT, L. & SCHULTZ, J. (1976). Quadratic assignment as a general data analysis strategy. *British Journal of Mathematical and Statistical Psychology* **29**, 190–241.

HUBERT, L. J. & LEVIN, J. R. (1976). A general statistical framework for assessing categorical clustering in free recall. *Psychological Bulletin* **83**, 1072.

INTERNATIONAL AGENCY FOR RESEARCH ON CANCER AND OTHERS (2014). Globocan 2012: estimated cancer incidence, mortality and prevalence worldwide in 2012. *World Health Organization. http://globocan. iarc. fr/Pages/fact_sheets_cancer. aspx. Accessed on* **9**.

INTERNATIONAL BREAST CANCER STUDY GROUP AND OTHERS (2006). Tamoxifen after adjuvant chemotherapy for premenopausal women with lymph node-positive breast cancer: International breast cancer study group trial 13-93. *Journal of Clinical Oncology* **24**, 1332–1341.

IVANOVA, N. B., DIMOS, J. T., SCHANIEL, C., HACKNEY, J. A., MOORE, K. A. & LEMISCHKA, I. R. (2002). A stem cell molecular signature. *Science* **298**, 601–604.

JACCARD, P. (1912). The distribution of the flora in the alpine zone. *New phytologist* **11**, 37–50.

JAIN, N., THATTE, J., BRACIALE, T., LEY, K., O'CONNELL, M. & LEE, J. K. (2003). Local-pooled-error test for identifying differentially expressed genes with a small number of replicated microarrays. *Bioinformatics* **19**, 1945–1951.

KEOGH, E. & LIN, J. (2005). Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and information systems* **8**, 154–177.

KIDDLE, S. J., WINDRAM, O. P., MCHATTIE, S., MEAD, A., BEYNON, J., BUCHANAN-WOLLASTON, V., DENBY, K. J. & MUKHERJEE, S. (2010). Temporal clustering by affinity propagation reveals transcriptional modules in arabidopsis thaliana. *Bioinformatics* **26**, 355–362.

KIM, J. & KIM, J. H. (2007). Difference-based clustering of short time-course microarray data with replicates. *BMC bioinformatics* **8**, 253.

KIM, S. K., LUND, J., KIRALY, M., DUKE, K., JIANG, M., STUART, J. M., EIZINGER, A., WYLIE, B. N. & DAVIDSON, G. S. (2001). A gene expression map for caenorhabditis elegans. *Science* **293**, 2087–2092.

KISANGA, E. R., MELLGREN, G. & LIEN, E. A. (2005). Excretion of hydroxylated metabolites of tamoxifen in human bile and urine. *Anticancer research* **25**, 4487–4492.

KOHONEN, T. (1995). Springer series in information sciences. *Self-organizing maps* **30**.

KRZANOWSKI, W. J. & LAI, Y. (1988). A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics* , 23–34.

LASSMANN, T., FRINGS, O. & SONNHAMMER, E. L. (2009). Kalign2: high-performance multiple alignment of protein and nucleotide sequences allowing external features. *Nucleic acids research* **37**, 858–865.

LOUIS, D. N., OHGAKI, H., WIESTLER, O. D., CAVENEE, W. K., BURGER, P. C., JOUVET, A., SCHEITHAUER, B. W. & KLEIHUES, P. (2007). The 2007 who classification of tumours of the central nervous system. *Acta neuropathologica* **114**, 97–109.

LU, X., ZHANG, W., QIN, Z. S., KWAST, K. E. & LIU, J. S. (2004). Statistical resynchronization and bayesian detection of periodically expressed genes. *Nucleic acids research* **32**, 447–455.

MA, P., CASTILLO-DAVIS, C. I., ZHONG, W. & LIU, J. S. (2006). A data-driven clustering method for time course gene expression data. *Nucleic Acids Research* **34**, 1261–1269.

MACQUEEN, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and*

probability, vol. 1. California, USA.

MAEDA, N., NISHIYORI, H., NAKAMURA, M., KAWAZU, C., MURATA, M., SANO, H., HAYASHIDA, K., FUKUDA, S., TAGAMI, M., HASEGAWA, A. et al. (2008). Development of a dna barcode tagging method for monitoring dynamic changes in gene expression by using an ultra high-throughput sequencer. *Biotechniques* **45**, 95.

MARTIN BLAND, J. & ALTMAN, D. (1986). Statistical methods for assessing agreement between two methods of clinical measurement. *The lancet* **327**, 307–310.

MILLIGAN, G. W. & COOPER, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika* **50**, 159–179.

MILLIGAN, G. W. & COOPER, M. C. (1986). A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research* **21**, 441–458.

MILLIGAN, G. W. & COOPER, M. C. (1987). Methodology review: Clustering methods. *Applied psychological measurement* **11**, 329–354.

MÖLLER-LEVET, C. S., KLAWONN, F., CHO, K.-H. & WOLKENHAUER, O. (2003). Fuzzy clustering of short time-series and unevenly distributed sampling points. In *Advances in Intelligent Data Analysis V*. Springer, pp. 330–340.

NATARAJAN, K., MEYER, M. R., JACKSON, B. M., SLADE, D., ROBERTS, C., HINNEBUSCH, A. G. & MARTON, M. J. (2001). Transcriptional profiling shows that gcn4p is a master regulator of gene expression during amino acid starvation in yeast. *Molecular and cellular biology* **21**, 4347–4368.

NAU, G. J., RICHMOND, J. F., SCHLESINGER, A., JENNINGS, E. G., LANDER, E. S. & YOUNG, R. A. (2002). Human macrophage activation programs induced by bacterial pathogens. *Proceedings of the National Academy of Sciences* **99**, 1503–1508.

PANDA, S., ANTOCH, M. P., MILLER, B. H., SU, A. I., SCHOOK, A. B., STRAUME, M., SCHULTZ, P. G., KAY, S. A., TAKAHASHI, J. S. & HOGENESCH, J. B. (2002). Coordinated transcription of key pathways in the mouse by the circadian clock. *Cell* **109**, 307–320.

PEDDADA, S. D., LOBENHOFER, E. K., LI, L., AFSHARI, C. A., WEINBERG, C. R. & UMBACH, D. M. (2003). Gene selection and clustering for time-course and dose–response microarray experiments using order-restricted inference. *Bioinformatics* **19**, 834–841.

PHANG, T. L., NEVILLE, M. C., RUDOLPH, M. & HUNTER, L. (2003). Trajectory clustering: a non-parametric method for grouping gene expression time courses, with applications to mammary development. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*. NIH Public Access.

PINHEIRO, J. C. & BATES, D. M. (2000). *Mixed-effects models in S and S-PLUS*. Springer.

POMEROY, S. L., TAMAYO, P., GAASENBEEK, M., STURLA, L. M., ANGELO, M., MCLAUGHLIN, M. E., KIM, J. Y., GOUMNEROVA, L. C., BLACK, P. M., LAU, C. et al. (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature* **415**, 436–442.

PRAMILA, T., MILES, S., GUHATHAKURTA, D., JEMIOLO, D. & BREEDEN, L. L. (2002). Conserved homeodomain proteins interact with mads box protein mcm1 to restrict ecb-dependent transcription to the m/g1 phase of the cell cycle. *Genes & development* **16**, 3034–3045.

QUACKENBUSH, J. (2001). Computational analysis of microarray data. *Nature reviews genetics* **2**, 418–427.

R CORE TEAM (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

RAND, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association* **66**, 846–850.

ROBIN, X., TURCK, N., HAINARD, A., TIBERTI, N., LISACEK, F., SANCHEZ, J.-C. & MÜLLER, M. (2011). proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC Bioinformatics* **12**, 77.

ROUSSEEUW, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* **20**, 53–65.

SAVAGE, R. S., HELLER, K., XU, Y., GHAHRAMANI, Z., TRUMAN, W. M., GRANT, M., DENBY, K. J. & WILD, D. L. (2009). R/bhc: fast bayesian hierarchical clustering for microarray data. *BMC bioinformatics* **10**, 242.

SHARAN, R. & SHAMIR, R. (2000). Click: a clustering algorithm with applications to gene expression analysis. In *Proc Int Conf Intell Syst Mol Biol*, vol. 8.

SHAW, J. R., PFRENDER, M. E., EADS, B. D., KLAPER, R., CALLAGHAN, A., SIBLY, R. M., COLSON, I., JANSEN, B., GILBERT, D. & COLBOURNE, J. K. (2008). Daphnia as an emerging model for toxicological genomics. *Advances in Experimental Biology* **2**, 165–328.

SHIRAKI, T., KONDO, S., KATAYAMA, S., WAKI, K., KASUKAWA, T., KAWAJI, H., KODZIUS, R., WATAHIKI, A., NAKAMURA, M., ARAKAWA, T. et al. (2003). Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage. *Proceedings of the National Academy of Sciences* **100**, 15776–15781.

SLONIM, D. K. (2002). From patterns to pathways: gene expression data analysis comes of

age. *Nature genetics* **32**, 502–508.

SØRENSEN, T. (1948). {A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons}. *Biol. skr.* **5**, 1–34.

SPELLMAN, P. T., SHERLOCK, G., ZHANG, M. Q., IYER, V. R., ANDERS, K., EISEN, M. B., BROWN, P. O., BOTSTEIN, D. & FUTCHER, B. (1998). Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular biology of the cell* **9**, 3273–3297.

STEINLEY, D. (2004). Properties of the Hubert-Arable Adjusted Rand Index. *Psychological methods* **9**, 386.

STORCH, K.-F., LIPAN, O., LEYKIN, I., VISWANATHAN, N., DAVIS, F. C., WONG, W. H. & WEITZ, C. J. (2002). Extensive and divergent circadian gene expression in liver and heart. *Nature* **417**, 78–83.

STRAUME, M. (2004). DNA microarray time series analysis: automated statistical assessment of circadian rhythms in gene expression patterning. *Methods in enzymology* **383**, 149.

SUGAR, C. A. & JAMES, G. M. (2003). Finding the number of clusters in a dataset. *Journal of the American Statistical Association* **98**.

TAMAYO, P., SLONIM, D., MESIROV, J., ZHU, Q., KITAREEWAN, S., DMITROVSKY, E., LANDER, E. S. & GOLUB, T. R. (1999). Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences* **96**, 2907–2912.

TAVAZOIE, S., HUGHES, J. D., CAMPBELL, M. J., CHO, R. J. & CHURCH, G. M. (1999). Systematic determination of genetic network architecture. *Nature genetics* **22**, 281–285.

TIBSHIRANI, R., WALTHER, G. & HASTIE, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**, 411–423.

TOMANCAK, P., BEATON, A., WEISZMANN, R., KWAN, E., SHU, S., LEWIS, S. E., RICHARDS, S., ASHBURNER, M., HARTENSTEIN, V., CELNIKER, S. E. et al. (2002). Systematic determination of patterns of gene expression during Drosophila embryogenesis. *Genome Biol* **3**, 0081–0088.

TROYANSKAYA, O., CANTOR, M., SHERLOCK, G., BROWN, P., HASTIE, T., TIBSHIRANI, R., BOTSTEIN, D. & ALTMAN, R. B. (2001). Missing value estimation methods for dna microarrays. *Bioinformatics* **17**, 520–525.

VOLLEBERGH, M. A., KLIJN, C., SCHOUTEN, P. C., WESSELING, J., ISRAELI, D., YLSTRA, B., WESSELS, L. F., JONKERS, J. & LINN, S. C. (2014). Lack of genomic heterogeneity at high-resolution acgh between primary breast cancers and their paired lymph node metastases. *PloS one* **9**, e103177.

WARD JR, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American statistical association* **58**, 236–244.

WARREN LIAO, T. (2005). Clustering of time series data—a survey. *Pattern recognition* **38**, 1857–1874.

WASSERMAN, L. (2004). *All of statistics: a concise course in statistical inference*. Springer.

WASSERMAN, L. (2006). *All of nonparametric statistics*. Springer.

WEHRENS, R. & BUYDENS, L. (2007). Self- and super-organising maps in r: the kohonen package. *J. Stat. Softw.* **21**.

WHITFIELD, M. L., SHERLOCK, G., SALDANHA, A. J., MURRAY, J. I., BALL, C. A., ALEXANDER, K. E., MATESE, J. C., PEROU, C. M., HURT, M. M., BROWN, P. O. et al. (2002). Identification of genes periodically expressed in the human cell cycle and their expression in tumors. *Molecular biology of the cell* **13**, 1977–2000.

XU, R., WUNSCH, D. et al. (2005). Survey of clustering algorithms. *Neural Networks, IEEE Transactions on* **16**, 645–678.

XU, X. L., OLSON, J. M. & ZHAO, L. P. (2002). A regression-based method to identify differentially expressed genes in microarray time course studies and its application in an inducible huntington's disease transgenic model. *Human Molecular Genetics* **11**, 1977–1985.

ZHANG, B., VERBERKMOES, N. C., LANGSTON, M. A., UBERBACHER, E., HETTICH, R. L. & SAMATOVA, N. F. (2006). Detecting differential and correlated protein expression in label-free shotgun proteomics. *Journal of proteome research* **5**, 2909–2918.

ZHAO, L. P., PRENTICE, R. & BREEDEN, L. (2001). Statistical modeling of large microarray data sets to identify stimulus-response profiles. *Proceedings of the National Academy of Sciences* **98**, 5631–5636.

ZHENG, Y., SUN, D., SHARMA, A. K., CHEN, G., AMIN, S. & LAZARUS, P. (2007). Elimination of antiestrogenic effects of active tamoxifen metabolites by glucuronidation. *Drug Metabolism and Disposition* **35**, 1942–1948.

ZHU, G., SPELLMAN, P. T., VOLPE, T., BROWN, P. O., BOTSTEIN, D., DAVIS, T. N. & FUTCHER, B. (2000). Two yeast forkhead genes regulate the cell cycle and pseudohyphal growth. *Nature* **406**, 90–94.

Paola Tellaroli

CURRICULUM VITAE

Contact Information

University of Padova
Department of Statistics
via Cesare Battisti, 241-243
35121 Padova, Italy
e-mail: tellaroli@stat.unipd.it
Personal page: http://www.stat.unipd.it/fare-ricerca/tellaroli-paola

Current Position

Since January 2012 (expected completion: January 2015)
PhD Student in Statistical Sciences, University of Padova, Italy
Thesis title: Topics in Omics Research
Supervisor: Prof. Alessandra R. Brazzale
Co-supervisors: Prof. Sorin Drăghici, Prof. Silvio Bicciato

Research Interests

- Statistical genetics
- Meta-analysis
- Clustering methods
- Pathway analysis

Education

October 2008 – March 2011

Master's degree (*laurea magistrale***) in Biostatistics**

University of Milano-Bicocca, Faculty of Statistics

Title of dissertation: "Sentinel lymph node biopsy versus standard axillary dissection for breast cancer: a classic and Bayesian Meta Analysis" (in Italian)

Supervisors: Prof. Gianluca Baio, Prof. Vincenzo Bagnardi, Dr Gian Luca De Salvo

Final mark: 110/110 cum laude

October 2005 – July 2008

Bachelor's degree (*laurea triennale***) in Statistics and Social Research**

University of Bologna, Faculty of Statistics

Title of dissertation: "The avoidable death in the province of Bologna in the period 1993 – 2006" (in Italian)

Supervisors: Prof. Giulia Cavrini, Dr Paolo Pandolfi

Final mark: 110/110 cum laude

Visiting periods

October 2013 – September 2014

Department of Computer Science, Wayne State University, Detroit, MI (USA)

Supervisor: Prof. Sorin Drăghici

February 2007 – June 2007

École Nationale de la Statistique et de l'Analyse de l'Information, Rennes (France)

Supervisor: Prof. Daniela Cocchi

Further education

June 23^{rd} – 28^{th}*, 2013*

Computational Statistics for Genome Biology, Bressanone, Italy

Participation in workshops and conferences

July 8^{th} – 12^{th}*, 2013*

28^{th} **International Workshop on Statistical Modelling**, Palermo, Italy

May 16^{th} – 18^{th}*, 2014*

Great Lakes Bioinformatics Conference, Cincinnati, OH, USA

November 18^{th}*, 2014*

Workshop on Clustering methods and their applications, Bolzano, Italy

Work experience

January 2015 – December 2015

Research Fellowship

Department of Cardiac, Thoracic and Vascular Sciences, Padova, Italy

September 2010 – December 2011

Biostatistician (Scholarship holder)

Oncology Institute of Veneto Region, Clinical Trials and Biostatistics Unit, Padova, Italy

November 2007 – April 2008

Statistician (Trainee)

Local Health Unit "AUSL", Bologna, Italy

Computer skills

I obtained ECDL (European Computer Driving License) and I have a good knowledge of:

- Statistical softwares: R, STATA, SAS, WinBUGS
- Microsoft Office tools: Word, Excel, Access, PowerPoint
- Demographic tools: HFA
- LaTeX

Language skills

- Italian: mother tongue
- French: fluent – *I studied and lived 6 months in France. TFI certificate: results 830/990*
- English: fluent – *I followed an Academic English Speaking Course and I lived 1 year in USA.*
- German: basic – *I studied German for 5 years at high school*
- Chinese: A2 level – *Ongoing*

Publications

Working papers

Tellaroli P., Borgatta M., Chèvre N., Hernandez C., Waridel P., Brazzale A. R. (2013). Toxicity of Tamoxifen on Daphnia pulex. *Working Paper Series N.11*

Tellaroli P., Bazzi M., Donato M., Brazzale A. R., Draghici S. (2014). Cross-clustering: a partial clustering algorithm with automatic estimation of the number of clusters. *Submitted*

Donato M., Draghici S., Hassan S., Romero R., **Tellaroli P.**, Tomoiaga A., Westfall P. (2014) A Bayesian hierarchical model to simultaneously shrink and disentangle overlapping pathway effects. *Submitted*

Conference presentations

Bazzi M., Tellaroli P. (2013). Finding profiles in time-course gene expression. In *Proceedings of the 28th International Workshop on Statistical Modelling, vol. 2*, (Editors Muggeo VMR, Capursi V, Boscaino G, Lovison G, pp. 501–506), Palermo, Italy, 8 – 12 July

Teaching experience

April, 2013 – June, 2013

Course name: Statistics
Teaching task: Exercises, 25 hours
Institution: Department of Biology
Instructor: Prof. Tiziano Vargiolu

References

Prof. Alessandra R. Brazzale
University of Padova
Department of Statistical Sciences
via Cesare Battisti, 241-243
e-mail: alessandra.brazzale@unipd.it
web: www.stat.unipd.it/ brazzale

Prof. Sorin Drăghici
Wayne State University
Computer Science Department
5050 Anthony Wayne Dr., Detroit, MI 48202
e-mail: sod@cs.wayne.edu
web: http://engineering.wayne.edu/profile/sorin.draghici