1-1-2015

Plsi: A Computational Software Pipeline For Pathway Level Disease Subtype Identification

Michele Donato

Wayne State University,

PLSI: A COMPUTATIONAL SOFTWARE PIPELINE FOR PATHWAY LEVEL DISEASE SUBTYPE IDENTIFICATION.

by

MICHELE DONATO

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the

requirements for the degree of

MASTER OF SCIENCE

2015

MAJOR: COMPUTER SCIENCE

Approved By:

_____

Advisor Date

_____

_____

_____

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Sorin Drăghici, the current and past members of the Intelligent Systems and Bioinformatics Laboratory, and the many collaborators I had the honor to work with during my studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: BACKGROUND AND INTRODUCTION

The process of obtaining a comprehensive list of genes, proteins, and metabolites that are different between two phenotypes is a today routine for a multitude of researchers in life sciences. And yet, even though such *high-throughput comparisons* have become relatively easy to perform, the biggest challenge remains: transforming the raw data in a deep understanding of the biological phenomena that determine the observed phenotype. At the same time, we have started to understand that evolution of many diseases such as cancer, are the results of the interplay between the disease itself and the immune system of the host. It is now well accepted that cancer is not a single disease but a "complex collection of distinct genetic diseases united by common hallmarks" [88]. The heterogeneity of diseases such as breast cancer is well recognized [79] and gene expression profiling has been used to identify at least four major subtypes: luminal A, luminal B, HER2+ and basal-like [80, 92]. In the past decade, important clinical advances in cancer treatments are attributed to molecularly targeted treatments aiming at specific genes such as estrogen receptor alpha (ER-α), HER2, EGFR, etc. Targeted treatments result in greater efficacy and fewer debilitating or dose limiting side effects [88]. This clearly proves that **it is important to identify and appropriately treat each individual disease subtype**. However, our current understanding of disease subtypes appears to be very limited. Despite targeted treatment advances, targeted therapies often fail for some patients. For breast cancer, while 20% of tumors overexpress the HER2 oncogene, one-third of these fail to show response to HER2-targeted therapies right from the outset. Research and clinical studies present a similar story for anti-estrogen treatment of ER-α-positive breast cancer and androgen ablation of androgen receptor positive prostate cancer [17, 44] Not all patients show an initial response, and from those who do, a signifi-

cant number will develop resistance. The fact that a substantial fraction of patients with a given subtype of cancer respond very differently to the same treatment, either immediately or later on, means that either: i) **the known subtypes are not truly homogeneous in terms or mechanisms of action and must be further refined**, or ii) that **subgroups of patients may have different mechanisms of defense against the same tumor type**.

Another aspect is related to the choice of optimal treatments. For example, cytotoxic chemotherapy remains the standard adjuvant therapy for lung cancer and it is not routinely recommended as part of the initial course of treatment for individuals with early stage disease [2, 106]. However, the high recurrence rates for stage I non-small cell lung cancer (NSCLC) raises consideration that a subset of patients may benefit from adjuvant therapy. Indeed, recent multinational clinical trials show that adjuvant chemotherapy can significantly improve the survival of patients with advanced early-stage (Stage IB-II) disease [8]. It follows that the capability to prognosticate outcomes – e.g., which tumors are likely to recur after surgical resection – would allow for better disease management where only patients who will benefit are treated and others who will not do not receive unnecessary over-treatments.

Many attempts to achieve this based on gene expression signatures have been undertaken but yielded only modest success so far (no FDA approved gene expression test exists yet).

The goal of PLSI is to go beyond the existing gene expression approaches for disease sub-typing, by exploiting the most recent approaches for the analysis of biological pathways, allowing "mechanism level" sub-typing .

The hypothesis is that a given disease subtype can be triggered by a number of different events, through different genes, but may involve common mechanism(s). As signals propagate along a pathway, the genes that are differentially expressed (DE)

change over time, while the pathway involved remains the same. Hence, we expect that a pathway signature, i.e. the pathways that distinguish between subtypes, will be: i) more easily detectable, ii) more stable, and iii) more useful than a gene signature. Therefore, PLSI allows the use of *pathway profiles* to discover and characterize disease subtypes and patient subgroups.

CHAPTER 2: DISEASE SUB-TYPING APPROACHES

The rapidly increasingly easy collection of whole-genome expression data resulted in thousands of publicly available data sets through databases like ArrayExpress [9, 85] and the Gene Expression Omnibus (GEO) [4, 28]. For example, ArrayExpress contains more than 40,000 RNA screening data sets, out of which more than 15,000 are human data sets. The data sets in these databases are easily retrievable through their websites or through APIs in different programming environments. Another aspect of the fact that expression data is relatively easy and cheap to obtain is related to the *dimensions* of the data sets. When such analyses were novel and expensive, the number of samples was limited to numbers rarely exceeding the dozen, whereas today it is not surprising to find data sets with tens or hundreds of samples[1].

One examples of such studies is The Cancer Genome Atlas (TCGA) [98]. The TCGA initiative is a coordinated effort of analysis and collection of cancer related data sets. Research groups that belong to the TCGA consortium are asked to collect samples, perform the screening using standard analysis protocols, and upload the results on the TCGA website, where they are made available on the TCGA Data Portal. Currently more than 11,000 samples are available for download from the data portal. Table 2.1 shows the distribution of samples for different cancer diseases.

Samples analyzed in the TCGA initiative are not screened only for gene expression, but for a multitude of other kinds of data, including DNA methylation, genomic variants, miRNA expression, etc. With such number of samples available at once, researchers can test new hypotheses related to the heterogeneity of the conditions observed, allowing meaningful analysis of complex diseases. One of the most widely used approach for the detection of disease subtypes is arguably hierarchical

[1]Issues related to the collection of human samples nowadays surpass the issues in screening the samples.

Available cancer types	number of cases
Acute Myeloid Leukemia [LAML]	200
Adrenocortical carcinoma [ACC]	80
Bladder Urothelial Carcinoma [BLCA]	412
Brain Lower Grade Glioma [LGG]	516
Breast invasive carcinoma [BRCA]	1098
Cervical squamous cell carcinoma and endocervical adenocarcinoma [CESC]	308
Cholangiocarcinoma [CHOL]	36
Colon adenocarcinoma [COAD]	461
Esophageal carcinoma [ESCA]	185
Glioblastoma multiforme [GBM]	528
Head and Neck squamous cell carcinoma [HNSC]	528
Kidney Chromophobe [KICH]	66
Kidney renal clear cell carcinoma [KIRC]	536
Kidney renal papillary cell carcinoma [KIRP]	291
Liver hepatocellular carcinoma [LIHC]	377
Lung adenocarcinoma [LUAD]	521
Lung squamous cell carcinoma [LUSC]	504
Lymphoid Neoplasm Diffuse Large B-cell Lymphoma [DLBC]	48
Mesothelioma [MESO]	87
Ovarian serous cystadenocarcinoma [OV]	586
Pancreatic adenocarcinoma [PAAD]	185
Pheochromocytoma and Paraganglioma [PCPG]	179
Prostate adenocarcinoma [PRAD]	498
Rectum adenocarcinoma [READ]	171
Sarcoma [SARC]	261
Skin Cutaneous Melanoma [SKCM]	470
Stomach adenocarcinoma [STAD]	443
Testicular Germ Cell Tumors [TGCT]	150
Thymoma [THYM]	124
Thyroid carcinoma [THCA]	507
Uterine Carcinosarcoma [UCS]	57
Uterine Corpus Endometrial Carcinoma [UCEC]	548
Uveal Melanoma [UVM]	80
total cases	**11041**

Table 2.1: Available TCGA data sets

clustering, with thousands of studies where gene expression profiles of samples are clustered, and clusters then are associated with clinical variables to find meaningful groups of samples. In [61] the authors cluster gene expression profiles of prostate tumor samples, performing feature selection to establish which genes best described the resulting clusters. [92], [103], and [93] cluster breast cancer samples with different outcomes in survival, while [59] cluster gene expression profiles of drugs and diseases together in order to connect drugs and diseases based on their gene expression profile.

Another widely used clustering method for disease subtyping is k-means clustering [42]. Similarly to hierarchical clustering, this method has been widely applied on gene expression data to discover subtypes of diseases. [78] and [87] apply k-means to discover subtypes of glioma, [101] use it to discover subtypes of ovarian cancer linked to clinical outcome of the patients, while [63] use the clusters found with k-means to identify the gene signature of a very aggressive subtype of breast cancer, opening the way for targeted therapies.

More advanced methods are also applied (e.g. spectral clustering, self organizing maps), and the abundance of works that use one of the numerous available clustering methods for detecting disease subtypes shows the extent of the problem of disease subtyping.

Ultimately, by clustering samples of a certain disease to discover subtypes, researchers aim to find the *signatures* of such subtypes, signatures that can be used either for prognosis of new patients, or for discovering mechanisms that are specific for a subtype, allowing the development of targeted treatments that can reduce side effects and increase effectiveness.

One issue with the current approaches to disease subtyping and subsequent signature discovery, is that in the vast majority of the cases the features used for detecting the disease subtypes are *genes*, and the values assigned to those features come from intrinsically noisy measurement methods. While mRNA microarrays are an excellent method for screening the entire genome of a specimen, this technology presents high levels of noise in the measurements, yielding results that are difficult to reproduce, as shown in several studies [14, 21, 34]. Another problem related to microarray experiments is the fact that they represent a *snapshot* of the gene activity at a certain moment, adding another factor that undermines reproducibility and reliability of the results. The issue with gene measurements is that, as signals

among genes propagate through a given pathway, the specific subset of genes that are differentially expressed change continuously, on various time scales. However, the pathways of signal propagation that are impacted in a specific process may remain the same. By focusing on pathways, rather than single genes, we reduce the noise introduced by single gene measurements, resulting in more reliable signatures

CHAPTER 3: PATHWAY ANALYSIS

3.1 Pathway analysis

The first approaches available for the analysis of pathways were over-representation (ORA) (e.g. hypergeometric [26, 97]) and functional class scoring (FCS) (e.g. GSEA [72, 94]). These methods are limited when used for pathway analysis because they completely disregard the topology of the pathway, which captures the way the genes interact with each other - the very reason of existence of signaling and metabolic pathways. Recently an impact analysis approach was developed, able to incorporate gene interaction knowledge into the analysis of signaling pathways [23] (briefly described in section 3.1.2 below). This impact analysis was the first approach that extended the classical analysis by incorporating important biological factors like (i) the magnitude of expression change for each gene, (ii) their position on the pathway, as well as (iii) the type of gene interactions on the pathway. Since the introduction of the impact analysis, more than 20 topology-based methods for pathway analysis have been proposed [23, 24, 27, 32, 35, 36, 38, 39, 40, 41, 47, 48, 49, 68, 70, 83, 90, 96, 104, 108, 110]. The majority of them use variations of centrality measures (e.g., node degree, node betweenness, etc.) to score genes according to their position in the pathway and the number of neighboring genes. In addition, methods like ScorePAGE [83] and PWEA [47] also use gene expression similarity measures (e.g., correlation coefficients) between genes on the same pathway to identify tight clusters of highly correlated genes. Methods such as PARADIGM [104], PathOlogist [32], TAPPA [38], BPA [49] consider the expression of genes (i.e., nodes) in the pathway as random variables and use the interactions to define conditional dependency. Independent of the model used to incorporate pathway topology, all these methods focus on identifying the significantly impacted pathways in a *single given experimental condition*. In this work,

however, we take a step further and focus on the task of **discovering and characterizing disease subtypes and patients subgroups** using significantly impacted pathway signatures.

3.1.1 Over-representation Analysis (ORA)

Before the concept of pathways was introduced to describe how complicated gene signaling processes take place, there were gene annotations, such as those provided by the Gene Ontology (GO) [3], describing what was known about individual genes. One approach to the interpretation of an experiment is to use a hypergeometric test to calculate the probability of observing the actual number of differentially expressed (DE) genes belonging to a GO category just by chance [26, 54]. Since in most applications the interest stands in the GO categories that are enriched in DE genes, this approach has become known as *over-representation analysis* (ORA).

More recently, pathway databases such as KEGG [52], BioCarta [7], and Reactome [51], became available, describing metabolic pathways and gene signaling networks. The new type information offered by these sources offered the potential for a a whole new level of analysis methods, more effective and more refined than simple GO analysis. However, when such pathway databases started to became available, the methods originally developed for GO analysis were immediately used to analyze pathways. The extrapolation was very simple: consider a pathway as merely the set of the genes that are involved in it (discarding the interactions), and perform exactly the same analysis used for GO annotations.

The hypergeometric model is one of the most commonly used methods for performing ORA. This model computes a p-value that represents the probability of obtaining a number of DE genes in a pathway more extreme than the one observed,

taking into account the total number of DE genes an the total number of genes screened. Assuming that N genes are screened, that K genes are found to be DE, that K_P genes are found to be DE in pathway P, and that pathway P has size N_P genes in total, the probability of obtaining exactly K_p DE genes can be computed as in Eq. 3.1.

$$P(X = K_P|N, N_P, K) = \frac{\binom{N_P}{K_P} \cdot \binom{N-N_P}{K-K_P}}{\binom{N}{K}} \tag{3.1}$$

The probability of obtaining a number of genes equal or higher than the observed value K_P can be obtained with Eq. 3.2.

$$P(X \geq K_P) = 1 - \sum_{i=0}^{K_P-1} \frac{\binom{N_p}{i} \cdot \binom{N-N_p}{K-i}}{\binom{N}{K}} \tag{3.2}$$

The hypergeometric p-value computed for each pathway is used to rank them, and it is interpreted as the *amount of involvement* of each pathway in the phenomenon that generated the specific list of DE genes.

Currently, ORA is one of the most widely used method for pathway analysis, as seen in a number of surveys of pathway analysis methods [53, 56, 71].

3.1.2 Impact Analysis

The impact analysis [23] was the first pathway analysis approach that departed from the approaches that looked at pathways as mere sets of genes. This approach, able to capture the phenomena related to the complex interactions and signaling described by the pathway topology, takes into account the interactions among genes, the magnitude of the change in gene expression, and includes the classical over-representation analysis (in terms of the proportion of DE genes in a pathway).

In the impact analysis, an *impact factor* (IF) is computed as follows for each pathway:

$$IF(P_i) = log\left(\frac{1}{p_i}\right) + \frac{\sum_{g \in P_i} |PF(g)|}{\overline{|\Delta E|} \cdot N_{de}(P_i)} \tag{3.3}$$

The first term is related to the classical probability related to the proportion of DE/NDE genes in each pathway. This term captures the information provided by the more traditional, and widely used, classical statistical approaches. We can compute it with either an over-representation approach (e.g., z-test [20], contingency tables [75, 77], etc.), a FCS approach (e.g., GSEA [72, 94]) or other more recent approaches [11, 84, 100]. The p_i value corresponds to the probability of obtaining a proportion of DE/NDE genes in a pathway higher than what expected by chance , when the null hypothesis is true. In the original work presenting ht impact analysis, the authors used Fisher's exact test [26, 97]. This approach computes a probability p_i of observing a number of DE genes , N_{de} or greater, given a set of M genes tested to be used as a reference, and a total of N DE genes, just by chance.

The topology of each pathway is captured by the sum of perturbation factors (PF) in Eq. 3.3. The value of this sum depends on i) *the specific genes* that are differentially expressed (in terms of the magnitude of the expression change), ii) the position of each gene in the pathway, and iii) the interactions described by the pathway (i.e., its topology), in terms of efficiency of signal propagation and type of interaction. The denominator is a "pathway normalization factor" that takes into account the number of DE genes in the pathway and the average differential gene expression over the pathway.

By summing up the gene *perturbation factors* (PF) for all genes on a pathway, this term represents an aggregate measure for the entire pathway from the perspective

of signal propagation through the pathway. For each gene g, the perturbation factor represents the effect that genes upstream of g exercise on it, through the interactions described by the pathway, and it can be calculated as follows:

$$PF(g) = \Delta E(g) + \sum_{u \in US_g} \beta_{ug} \cdot \frac{PF(u)}{N_{ds}(u)} \tag{3.4}$$

In this expression, the first term represents the observed magnitude of the effect of the phenotype on the gene. The term $\Delta E(g)$ represents the measured expression change of gene g. One among many methods available for determining differential activity in phenotype comparison experiments can be used to obtain this value [16, 25, 82, 109]. A common choice, when analyzing a phenotype comparison experiment, is to average the expression levels in each phenotype, and provide the log-transformed ratio of the averages. We refer to this as the *log-fold change* or more simply just *fold change*. The second term consists in the sum of the perturbation factors of all the genes u that affect gene g, and it represents the effect of the part of the pathway that it is upstream of g. This term is normalized by the number of downstream genes of each such gene $N_{ds}(u)$, so that if a particular gene u has many downstream genes, its effect is *diluted*. Lastly, the upstream effect is weighted by a factor β_{ug}, which reflects the efficiency of the perturbation propagation. These values have to be determined before the analysis, either through prior knowledge or directly from the data. Lastly, the effect on the gene due to the genes that are upstream of it is captured by the term US_g.

The null hypothesis assumes that the list of DE genes only contains random genes. If the null hypothesis is true, the impact factor depends only on the number of DE genes found in the given pathway. As the size of the pathway increases, therefore, the impact factor grows accordingly. This is the reason why the second term of the

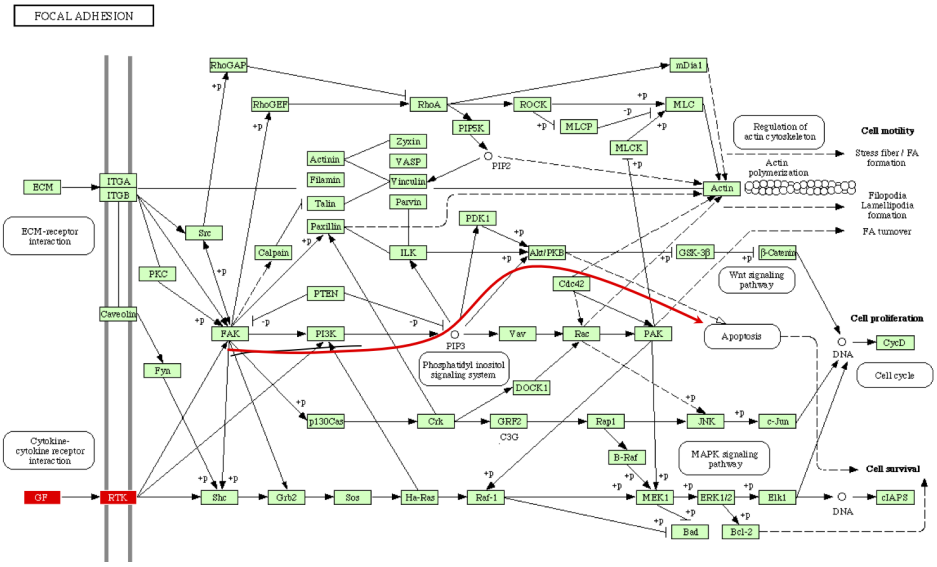

Figure 3.1: KEGG representation of Focal Adhesion. The red nodes represent differentially expressed genes, while the red arrow represents an example of the propagation of the perturbation that generates from the DE genes. The experiment analyzed the differences between lung cancer samples and normal samples.

impact factor is normalized by the number of DE genes that fall on the pathway. In essence, the PFs are calculated in a manner that implies the propagation of the perturbation on the pathway, following the interactions described by it (see Fig. 3.1 showing the propagation of the perturbations on a sample pathway).

The perturbation factors are computed in order to satisfy a *steady state* of the system described by all the equations used to compute the perturbation factors for all genes in each pathway. Although this could create problems with non-solvable systems (non-invertible matrices) this can easily be solved by computing pseudo-inverse matrices of such systems.

This method, however, still requires the *a priori* selection of DE genes based on p-value, in order to compute an enrichment p-value. Analyzing only the list of DE genes might represents an artificial truncation of the information available, as well as an unnecessary reliance on an upstream gene selection method, which may be far

from optimal. It has been shown that the choice of threshold in selecting important genes highly affects any analysis based on a list of DE genes [76]. Also, the statistical model used to test whether a particular gene is DE as well as the the method used for multiple correction comparison can further influence the set of DE genes obtained for a given patient. Furthermore, individual gene expression levels are intrinsically very noisy and subject to fluctuations, so testing the same person at a later date may yield a rather different set of DE genes.

In order to address these issues, a number of pathway analysis methods have been developed that do not rely on the selection of DE genes, using the entire set of measured genes to perform the analysis.

One of these methods is the latest implementation of the impact analysis [105], described in Section 3.1.2. In this approach, the model described by equation 3.4 above is modified as follows:

$$PF(g) = \alpha_g \cdot \Delta E(g) + \sum_{u \in US_g} \beta_{ug} \cdot \frac{PF(u)}{N_{ds}(u)} \tag{3.5}$$

This model considers all genes with their measured expression change $\Delta E(g)$ (log ratio with respect to the mean of the controls) but weights them with a factor α_g. Two alternatives for these weights include:

$$\alpha_g = -\log \frac{p_g}{p_{max}} \; and \; \alpha_g = 1 - \frac{p_g}{p_{max}} \tag{3.6}$$

This approach still uses gene p-values but these are not used to reject or not reject a hypothesis but rather as ranking information, so we can afford to be more lenient with various assumptions. If we consider a situation in which the least and most significant p-values after the correction for multiple comparisons are $p_{max} = 1$ and $p_{min} = 10^{-3}$ for instance, the expression to the left in equation 3.6 will provide weights α_g in the

range from zero ($p = p_{max}$) to 3 (for $p = 10^{-3}$ and log in base 10), while the expression to the right will yield weights in the range from zero to 0.999. In [105] we show that a meaningful pathway analysis can indeed be done without first selecting DE genes.

3.2 Sample level pathway analysis

Existing pathway analysis methods where designed to study **changes between conditions and are not able to identify significant pathways at the sample/patient level**. Usually, the measured expression changes are used in a summarized way, in the form of a list of DE genes between two groups of samples. Even though such approaches are useful in discovering the general mechanism of a disease, the specific response of a patient can be significantly different. By grouping together all disease samples, the existing methods are not sensitive to this specific response. Moreover, **these methods assume that the groups of samples are homogeneous** (i.e., all the samples in a group share the same characteristics). As discussed above, this is a gross oversimplification of the clinical reality.

In PlDis, we identify pathways signatures, rather than gene signatures. The classical hunt for gene signatures yielded partial success. On the one hand, the success of therapies targeted at specific genes such as HER2 shows that sub-typing the disease at the molecular level is the right strategy. On the other hand, the fact that within currently used disease subtypes the response to therapy varies so greatly between individuals shows that our current molecular classifications are not sufficiently accurate. The logical step forward is to perform such molecular sub-typing of disease and patient groups at a system level, using pathways, rather than at individual gene level.

3.2.1 Approach

The classical approach used to analyze the results of a high-throughput experiment such as DNA microarrays or RNA-seq starts with a matrix of genes and samples. In this matrix, each row corresponds to a gene and each column corresponds to a sample. Samples are usually belonging to two different phenotypes. An example of such matrix can be seen in Table 3.2.

	Control samples				Disease samples			
	sample 1	sample 2	$\cdots$	sample n	sample n+1	sample n+2	$\cdots$	sample n+m
gene 1	$exp_{1,1}$	$exp_{1,2}$	$\cdots$	$exp_{1,n}$	$exp_{1,n+1}$	$exp_{1,n+2}$	$\cdots$	$exp_{1,n+m}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
gene k	$exp_{k,1}$	$exp_{k,2}$	$\cdots$	$exp_{k,n}$	$exp_{k,n+1}$	$exp_{k,n+2}$	$\cdots$	$exp_{k,n+m}$

Table 3.2: Gene-sample matrix for a typical high-throughput data based experiment. In this matrix each row corresponds to a gene, and each column to a sample. Two groups of samples are compared, coming from two different phenotypes (e.g. disease versus control, treated versus untreated, etc). The cell i, j contains the expression value for gene i measured in sample j, in this matrix represented by the value $exp_{i,j}$.

In many cases, the goal of the analysis includes a comparison between two phenotypes such as disease and appropriately matched controls (see Fig. 3.2). Hence, the columns of the matrix are divided into two subsets, corresponding to the two phenotypes. The classical approach considers each gene (row) at a time, and uses a classical statistical testing technique to compare the behavior of this gene between the two groups. Such technique can range from the simplest (e.g. two-sample t-test) to the more sophisticated (moderated t-tests [91], linear models [69], general estimating equations (GEE)[65], etc.). The research hypothesis is that the gene has significantly different distributions in the two phenotypes, the null hypothesis is that the two distribution do not differ significantly. The approach calculates p-values for each gene, then corrects for multiple comparisons with an appropriate method, such as FDR [5, 6]. Genes with a p-value smaller than a certain threshold are considered

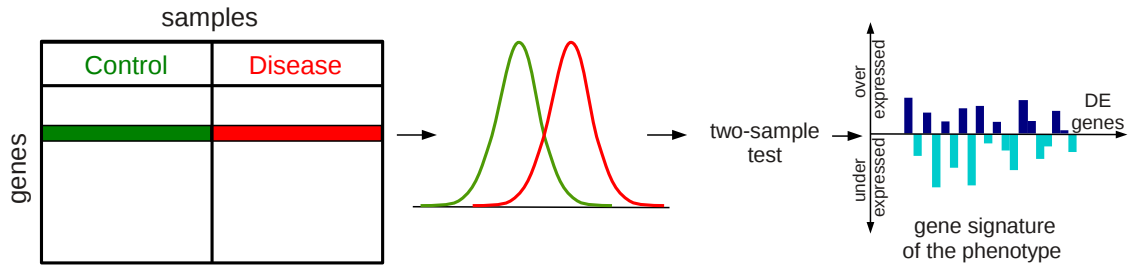

Figure 3.2: The classical approach for obtaining a molecular signature of a phenotype. A classical statistical test (e.g. two-sample t-test, linear models, GEE) is performed for each gene. After a correction for multiple comparisons, the genes with significant p-values are considered differentially expressed (DE) and constitute a *gene signature*.

as *differentially expressed* (DE) between the two groups in a statistically meaningful way. This set of DE genes can be seen as the *gene signature of the phenotype*. Subsequently, pathway analysis approaches such as the impact analysis [23, 96] use these sets of DE genes to identify the pathways that are significantly impacted in the given condition. Albeit this approach is statistically sound, solid, and widely used, it does not provide any information regarding individual samples. The goal of *sample level pathway analysis* is to identify **the pathways that are significantly different in an individual patient with respect to the control individuals**. In order to achieve this, we use two approaches. Both approaches aim to identify significant pathways in a given patient. The first approach still uses the concept of DE genes, while the second approach departs from this concept and use the expression level of all genes.

In the first approach, we start with the same gene-sample matrix. However, now for every gene (row), we consider the distribution of the values in the controls, and we compare the value of that gene for each disease sample with the distribution of controls. The null hypothesis is that the patient value comes from the same distribution of the controls, while the research hypothesis is that this value does not come

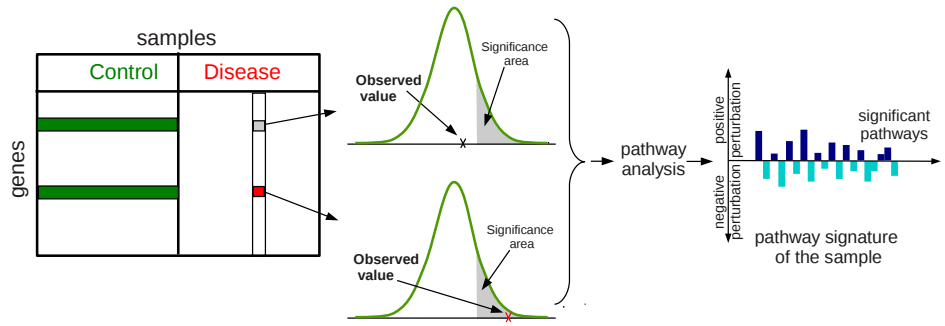

Figure 3.3: The proposed approach to obtain a sample pathway profile. A statistic such as effect size (e.g. Z-score or measured value divided by mean of controls) is calculated for each gene in a given sample. A significance test is performed on the individual effect sizes, yielding a set of genes that are DE in each sample with respect to the distribution of that gene in the control population. These sample-specific DE genes are then submitted to pathway analysis (e.g. impact analysis [23]) which identifies the set of significantly impacted pathways that will form the pathway profile of the sample.

from same distribution (see Fig. 3.3). In PLSI we used the simple Z distribution, but an arbitrary level of sophistication can be used. This provides a p-value for each gene in the given disease sample. After the correction for multiple comparison, we will have a set of genes *in this specific sample* whose expression levels are *unlikely to come from the distribution of the values of the same genes in the control group*. These will be used as DE genes in the pathway analysis, which in turn will provide the set of pathways that are significantly impacted **in the given sample**.

The approach above may not work well if the number of samples in the control group is not large enough, and/or if the distribution of the expression values in the control group is not normal. Therefore, we will use the version of the approach described in Equation 3.5, which does not require selection of DE genes, without using p-value for ranking genes.

Independently from the approach used, the result is a *pathway-sample matrix* of the format shown in Table 3.3.

	Control samples				Disease samples			
	sample 1	sample 2	$\cdots$	sample n	sample n+1	sample n+2	$\cdots$	sample n+m
pathway 1	$slpa_{1,1}$	$slpa_{1,2}$	$\cdots$	$slpa_{1,n}$	$slpa_{1,n+1}$	$slpa_{1,n+2}$	$\cdots$	$slpa_{1,n+m}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
pathway k	$slpa_{k,1}$	$slpa_{k,2}$	$\cdots$	$slpa_{k,n}$	$slpa_{k,n+1}$	$slpa_{k,n+2}$	$\cdots$	$slpa_{k,n+m}$

Table 3.3: Pathway-sample matrix resulting from the sample level pathway analysis. In this matrix each row corresponds to a pathway, and each column to a sample. The cell i, j contains the value for the activity of pathway i in the particular phenomenon analyzed, in the specific sample j, henceforth referred to as Sample Level Pathway Activity (slpa), in this matrix represented by the value $slpa_{i,j}$.

The key differences between Table 3.3 and Table 3.2 are: i) now the rows represent pathways, and the values represent the *sample level pathway activity*, i.e. the activity of a specific pathway in a specific sample, and ii) the control samples are not in the matrix anymore, as they have been used to compute differential gene expression of each disease sample.

This matrix is used for the detection of subtypes instead of the gene-sample matrix. Our approach is to detect both the subtypes and the pathways that are associated with the subtype, which represent the mechanisms of action of a specific subtype. This is achieved by using state of the art methods for partitioning the data, as well as extracting the meaningful features (i.e. pathways).

CHAPTER 4: PATHWAY LEVEL DISEASE SUBTYPING

4.1 PLSI

In this section we will assume that two phenotypes, condition and control, are compared. Two sample dataset obtained from the Gene Expression Omnibus (GEO) are provided with PLSI. The first data set contains the genome-wide expression levels on 156 samples, divided in 91 lung cancer samples and 65 adjacent normal lung tissue sample (GEO identifier GSE19188 [45]). The second data set investigates the effects of cigarette smoking and its association with lung adenocarcinoma. The samples are grouped by metadata factors including smoking (smoker, former smoker, never smoked), sex, and stage (I-IV) of the tumor (GEO identifier GSE10072 [60]). The starting point of the PLSI pipeline is the matrix of gene measurements described in Table 3.2. A sample dataset can be loaded after loading the PLSI package, with the commands (in this case for the GSE19188 data set):

```
> ## package loading
> library(PLSI)
> data(gse19188)
```

The `gse19188` data object contains two objects: `gse19188.exprData` containing the expression levels, and `gse19188.design` containing the design of the experiment. Part of the matrix containing the expression levels is shown in Table 4.4

The design of this example of the first six samples is shown in Table 4.5

From this point, the user can select one of two options for computing the effect size at sample level. The function `sleffect` can be used for this purpose. The parameter `sltype` determines the type of sample level effect returned by the function. At this moment two options are possible for the value of this parameter. The first

```
       GSM475656   GSM475657  GSM475658   GSM475659   GSM475660
1      0.1885843 -0.10727474 -0.3684598 -0.12642158 -0.02708948
10     0.1180333  0.22886640 -0.1087965  0.19290213 -0.06819548
100   -0.4145817 -0.61328920 -0.4421180 -0.03193571 -0.98947728
1000   2.6670965 -0.78480325 -0.3589639 -0.59348202 -0.45769893
10000  0.4970104 -0.00436277  0.3021071  0.21718295 -0.08975685
```

Table 4.4: The expression levels of the first five genes of the first five samples in the dataset GSE19188 provided with the PLSI package.

	tumor	healthy
GSM475656	1	0
GSM475657	0	1
GSM475658	0	1
GSM475659	0	1
GSM475660	0	1
GSM475661	1	0

Table 4.5: Design for the first 6 samples of the GSE19188 dataset provided with the PLSI package.

is the z-score (argument `z`), defined as the expression level of each condition sample, divided by the mean of the control samples, and divided by the standard deviation of control samples. The second option for sample level effect is the fold change (argument `fc`), defined as the expression level of each condition sample divided by mean of the control samples. The default action is to return the $log2$ transformation of this value.

```
> gse19188sleffect <-  sleffect(eData = gse19188.exprData,
+     expDesign = gse19188.design,
+     refLabel = "healthy", sltype = 'z')
```

When the `sltype` parameter is set as 'z' the `sleffect` function returns both z-score and p-value of the expression of each sample. Part of the complete matrix is shown in Table 4.6. Only the z-scores and the p-values of the first three genes in the first two samples are shown in the table.

	GSM475656	GSM475661	GSM475656.P.Value	GSM475661.P.Value
hsa:1	1.38	0.43	0.17	0.67
hsa:10	0.55	-0.79	0.58	0.43
hsa:100	-0.19	1.28	0.85	0.20

Table 4.6: Results of the `sleffect` function when the `sltype` parameter is set as 'z', for the first three genes of the first two samples.

Once the sample level effects are computed at gene level, the user needs to choose the type of analysis to perform. PLSI provides functions for performing two types of analysis. The first, basic analysis is the over-representation analysis (ORA) through Fisher's exact test, as described in Section 3.1.1. The function to perform ORA is `slFisher`. This kind of analysis requires a selection of DE genes, which, in `slFisher`, is made through the parameter `de.threshold`.

```
> oraRes <- slFisher(gse19188sleffect,
+          contrasts = colnames(gse19188sleffect),
+          de.threshold = 0.05)
```

An example of the sample level pathway activity matrix coming from the `slfisher` function is shown in Table 4.7.

	GSM475656	GSM475661	GSM475662	GSM475664	GSM475668
path:hsa03008	0.15	0.00	0.01	0.15	0.98
path:hsa03013	0.75	0.00	0.00	0.01	0.91
path:hsa03015	0.27	0.02	0.16	0.28	0.39
path:hsa03018	0.93	0.08	0.05	0.32	0.36

Table 4.7: Results of the `slFisher` function. Only the p-value is reported.

The second type of pathway analysis is the impact analysis described in Section 3.1.2, through the function `slpe` (Sample Level Pathway Express). The following example shows the use of the impact analysis without selection of DE genes.

```
> peRes <- slpe(effectSize <- gse19188sleffect,
+     contrasts <- colnames(gse19188sleffect),
+     de.threshold = NULL,
+     nboot=2000)
```

The sample level results of the `pe` function is a list where each element contains a number of computed statistics for each sample. These statistics are computed, in each element of the list, for all the pathways in KEGG. Therefore, the function `peTables` is needed to extract the desired computed statistic from the list. The following command extracts the value of normalized total perturbation.

```
> peTRes <- peTables(peRes, out="totalPertNorm")
```

After computing the sample level pathway information, the user can proceed in the identification of the subtypes. PLSI provides a number of accessory functions to perform subtype discovery. The function `slPCAT` performs PCA transformation of the results matrix, scaling it, and returning the number of PCs that explain a certain percentage of variance, specified by the parameter `varThr`.

```
> pcaORA <- slPCAT(oraRes, varThr=0.95)
```

The format of the pcaORA matrix is the same as the original sample level matrix.

```
                PC1        PC2       PC3        PC4       PC5
GSM475656  8.4342980 -5.1317496 -5.280836  1.4784779 -2.958403
GSM475661  5.6300122  9.4074006  2.488379 -1.5055306 -2.514222
GSM475662 -0.8635042  5.0071116  1.298642  0.4525431  2.915951
GSM475664 -2.3315332  2.6352725  2.576171  5.1484617  1.980153
GSM475668  1.9009694  0.7809436  1.523771 -2.8403260  5.922033
```

At this point the user has the choice to apply a number of techniques to the data in order to identify the subtypes and the mechanisms distinguishing subtypes. PLSI provides wrapper functions for this purpose.

The first wrapper function is the `ccWrapper` function. This function applies the *CrossClustering* method [99] to the data. CC is a novel clustering method that has two advantages with respect to traditional clustering methods: first, CC is able to automatically identify the number of clusters in the data. Second, CC is able to identify *outliers* in the data, i.e. elements that do not fit properly in any cluster.

```
> clres <- ccWrapper(pcaORA, kw=c(2,10), kcmax=50)
```

The result of `ccWrapper` is a list containing the list of clusters, and the cluster memberships for each elements as a numeric vector indicating which cluster each element belongs to. Clusters are given an index starting from 0, where the cluster with index 0 represents the outliers and the clusters with index greater than zero represent proper clusters.

```
GSM475656 GSM475661 GSM475662 GSM475664 GSM475668
        2         2         2         0         2
```

Following is an example of plotting the results of the `ccWrapper` function. Gray elements represent outliers. The `plotcres` function allows the user to interact with the graph by clicking the figure and dragging it, rotating it around. A snapshot of the figure is shown in Figure 4.1.

```
> plotcres(pcaORA, dimensions = 3,
+         memberships= clres$plotcols)
```

The second clustering method included in PLSI is k-means clustering. However, k-means presents two important issues. First, the number of clusters has to be

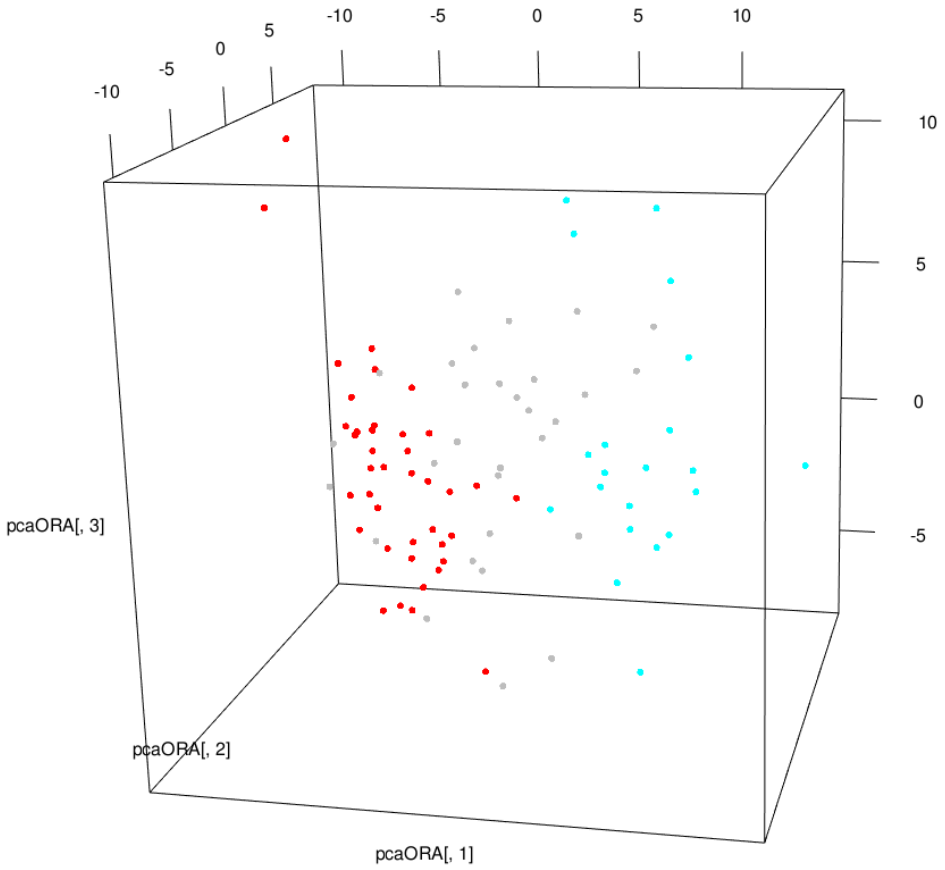

Figure 4.1: Plot of the first three principal components of the sample level pathway profiles of the dataset GSE19188. The data have been clustered with the *CrossClustering* method. The method indicated two clusters. Blue dots represent elements belonging to the first cluster, red dots represent elements belonging to the second cluster, while grey dots represent samples that were identified as outliers.

provided by the user. This is an issue since, in applications like disease subtyping, the number of subtypes is usually unknown. Second, k-means is non deterministic and the results depend on the initial choice of centers. In order to solve these two issues PLSI implements the suggestions found in [67]. The approach is as follows. First, a certain range of K (the number of clusters) is chosen by the user. The idea is to select, in that range, the value of K that minimizes an objective function. One possible option for such function could be the value of K that minimizes the residual sum of square (RSS). Unfortunately, the RSS is monotonically decreasing in K, and it reaches its minimum when $K = N$, where N is the number of elements to be clustered, i.e. when each element is center and only element of its own cluster. A possible solution to this would be to find the point where the decreasing curve of RSS presents an elbow, i.e. when the successive values of RSS decreases less. PLSI provides the function `obfElbow` for detecting the optimum K based on the value of K at the elbow. Another possible option, which is the one implemented in PLSI, is to create another objective function introducing a penalty for each new cluster. In [67], this penalty represents the *complexity of the model*, and in the context of clustering it is reasonable to set it as a function of the number of clusters. The *Akaike Information Criterion* (AIC) [1] can be used to define an objective function, as follows:

$$K = \arg\min_{K}(RSS_{min}(K) + 2MK) \tag{4.1}$$

In Equation 4.1, M represents the number of features available.

The second issue with k-means is the non-deterministic component of the algorithm. In PLSI this is solved by performing k-means many times and choosing the configuration of the centers that minimizes the total RSS, with the function `kmean-`

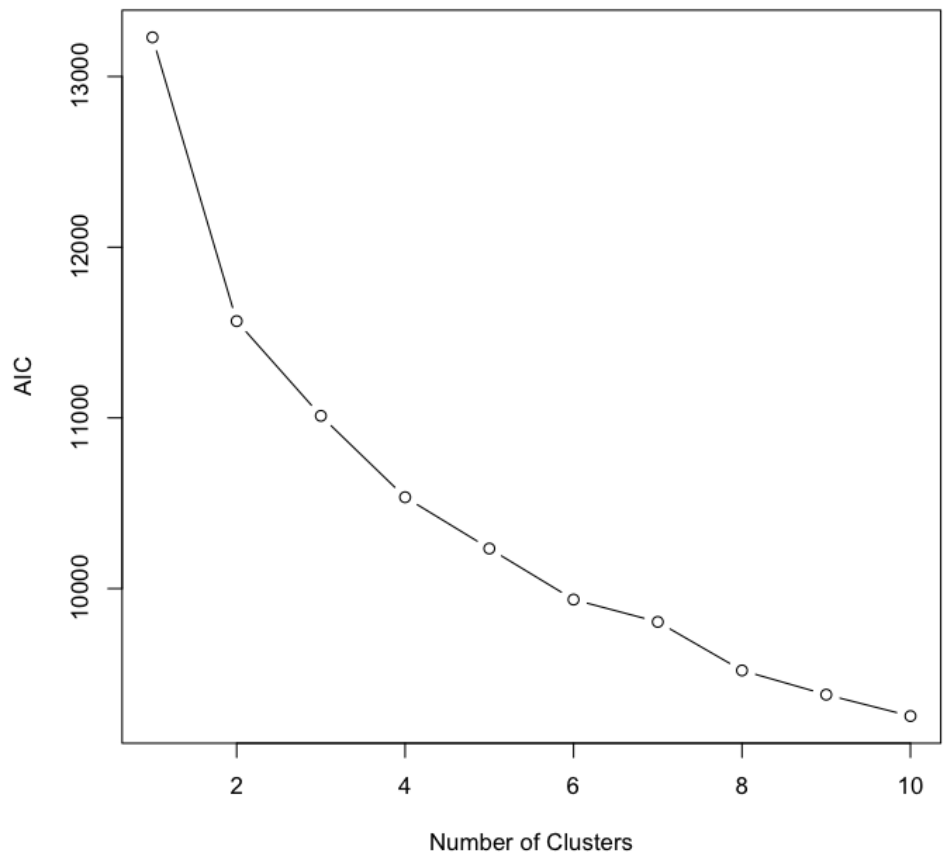

Figure 4.2: AIC values in presence of high RSS. When clustering the GSE19188 dataset the average value for RSS is $10,500$ when K ranges from 1 to 10, whereas the second component of the AIC ranges from 250 to 1250. In this situation the AIC behaves exactly like the RSS.

`sAIC`. If the AIC is used, the AIC value is computed for that center configuration. One issue with the formulation of the AIC in Equation 4.1 is that the RSS may be dominating the function. This is indeed what happens in the GSE19188 dataset. The average RSS is approximatively $10,500$, while $2MK$ ranges from 250 to 1250. This kind of situation makes the AIC behave exactly like the objective function that includes the RSS only, as it is shown in Figure 4.2

Clearly, increasing the range of K, for example up to 50, could fix this issue. This is indeed the case in this example, where a value of $K = 30$ shows a change

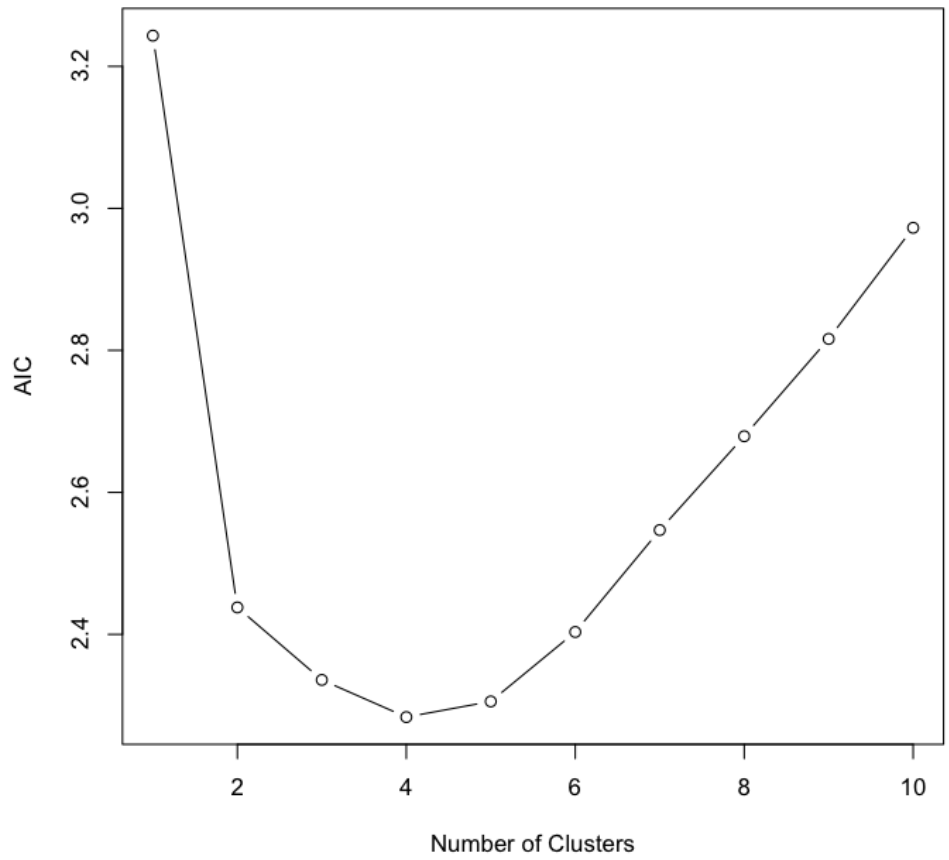

Figure 4.3: Scaled AIC objective function. The two components of the AIC are scaled and shifted so their minimum value is equal to 0, making them comparable.

in the direction of the AIC. However, such a high number of clusters might result in clusters that are not meaningful from the biological perspective.

Hence, PLSI allows for scaling of the two components of the objective function. The parameter `scaledAIC` controls this functionality. If set to `TRUE` the RSS and the values $2MK$ are first scaled, then shifted so that the minimum value of each one is 0, and then summed to obtain the vector of AICs. This scaled AIC results in a curve that indicates an optimum at $K = 4$ clusters, as shown in Figure 4.3.

Hierarchical clustering with p-values. The third subtype discovery method provided by PLSI is an application of hierarchical clustering. As hierarchical clustering by itself does not provide any indication about the number of clusters, for each cluster a p-value is computed. Such p-value represents the confidence on how much the cluster is supported by the data, i.e. high p-values represent clusters that are not likely to be obtained by chance. The function `pvclust`[95] performs the hierarchical clustering, and computes two p-values: the AU (Approximately Unbiased) and BP (Bootstrap Probability). The BP p-value of a cluster is obtained by bootstrapping the data [29] and counting how many times the cluster appears in the bootstrap iterations. The AU p-value is based on multiscale bootstrap resampling [89], where multiple sample sizes are chosen for the bootstrap samples, in order to eliminate bias in the p-values as discussed in [30] and [31].

Parameters of this function are the distance to be used, the clustering method, and the number of permutations to be performed in the bootstrap procedure.

```
> pvclustres <- pvclust(t(pcaORA), method.dist='euclidean',
+           method.hclust="ward", nboot=1000)
```

Results can be plotted and clusters with a p-value higher than a certain threshold can be highlighted with the function `pvrect`, as shown in Figure 4.4. In this figure we chose to highlight the AU p-values, using the following code.

```
> plot(pvclustres)
> pvrect(pvclustres, alpha=0.8, pv='au')
```

Finally, the clusters can be retrieved by calling the `pvpick` function. Only the first two clusters are shown.

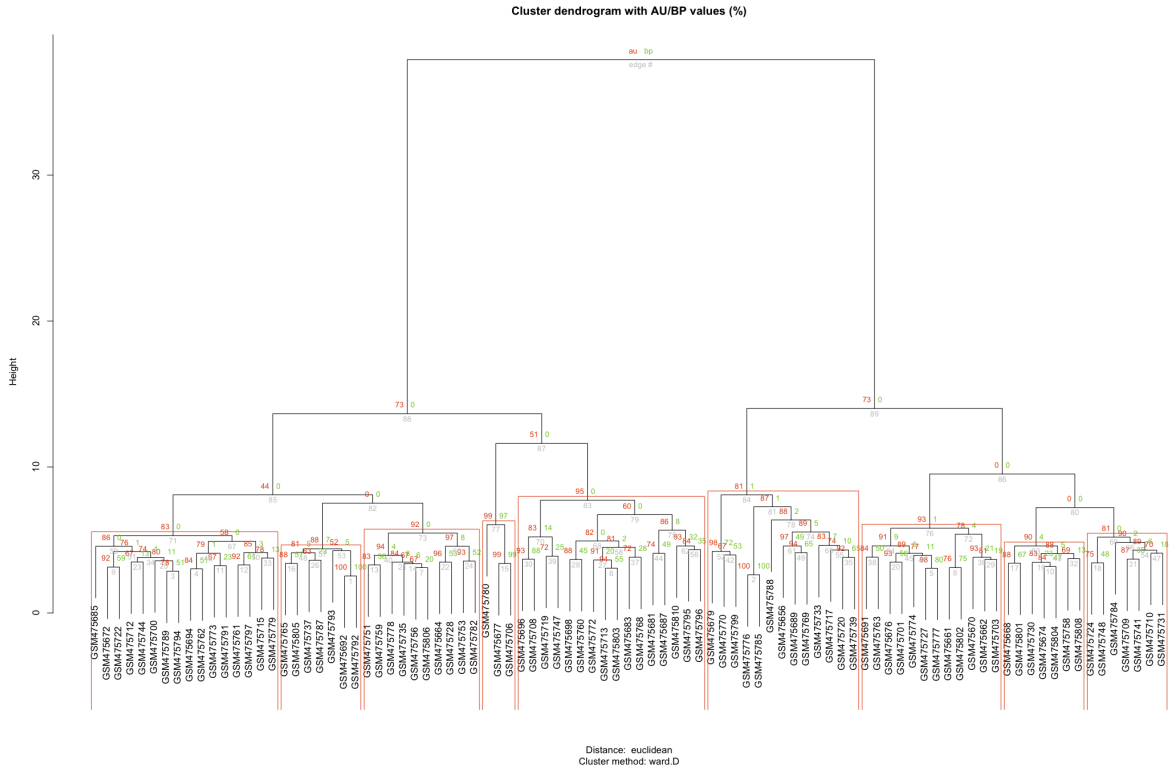

Figure 4.4: Results of the pvclust method. Red rectangles highlight the clusters with a p-value higher than 0.8. The p-value used here for the highlight is the AU p-value.

```
> sigClusts <- pvpick(pvclustres,alpha=0.8,pv="au")$clusters
> sigClusts[1:2]

[[1]]
[1] "GSM475692" "GSM475737" "GSM475765" "GSM475787" "GSM475792" "GSM475793"
[7] "GSM475805"

[[2]]
[1] "GSM475668" "GSM475674" "GSM475730" "GSM475758" "GSM475801" "GSM475804"
[7] "GSM475808"
```

4.1.1 Flexible procedures for clustering

The last methods for the identification of the disease subtypes provided by PLSI are the methods described in [43]. These methods allow for the assessment of cluster stability, allowing an accurate determination of the number of subtypes. Once the number is determined, the subtypes are determined by either k-means or hierarchical clustering.

PLSI provides two functions for assessing clustering stability: `fpcJitter` and `fpcBoot`. The concept of the functions is the following: performing clustering on the data assuming a certain number k of clusters, obtaining a partitioning of the original data P_{orig}. Then, the data is transformed. in `fpcJitter` noise is added to the data, while in `fpcBoot` the data is resampled. Clustering is performed again, obtaining a partitioning $P_{modifieddata}$. A similarity is computed between P_{orig} and $P_{modifieddata}$ by comparing the clusters belonging to each of the partitionings: the Jaccard similarity [50] is computed between each cluster in P_{orig} and each cluster in $P_{modifieddata}$, and then averaged over the most similar unique k pairs. This average represents the

similarity between two partitionings. This process is repeated a number of times, and then the similarity values are averaged again. This last average represents the stability of the initial partitioning. The idea is that if the data can be separated into k clusters, then the clusters obtained will be robust to slight changes to the data, whereas for different values the partitionings will not be stable. For `fpcJitter`, the noise is estimated from the data.

Each of the functions requires a range of values for k, and the stability is assessed for each value. The functions allow to perform parallel evaluation of the stability of the different values of k, determined by the value of the `parallel` parameter. Two different clustering methods are provided, k-means and hierarchical clustering.

```
> cdata=fpcJitter(pcaORA, nboot=100, kmrange=2:10, parallel=TRUE)
```

The two functions return the most stable cluster and the averages of the jaccard similarities obtained for the various values of k in the range provided by the parameter `kmrange`.

The `plotcres` function can be used to visualize the results. The `dimensions` parameter lets the user chose if the resulting plot should be two-dimensional or three-dimensional. The three-dimensional plot can be rotated with mouse input.

```
> plotcres(pcaORA, dimensions = 3,
+          memberships = cdata$clusterResult$partition)
```

Plotting the values of the Jaccard averages gives an idea of the behavior of the stability over different values of k. An example can be seen in Figure 4.5.

```
> plot(1:9, fpcresult$`jaccard averages`, type='o',
+           xlab='number of clusters', ylab = 'stability', col='red')
```

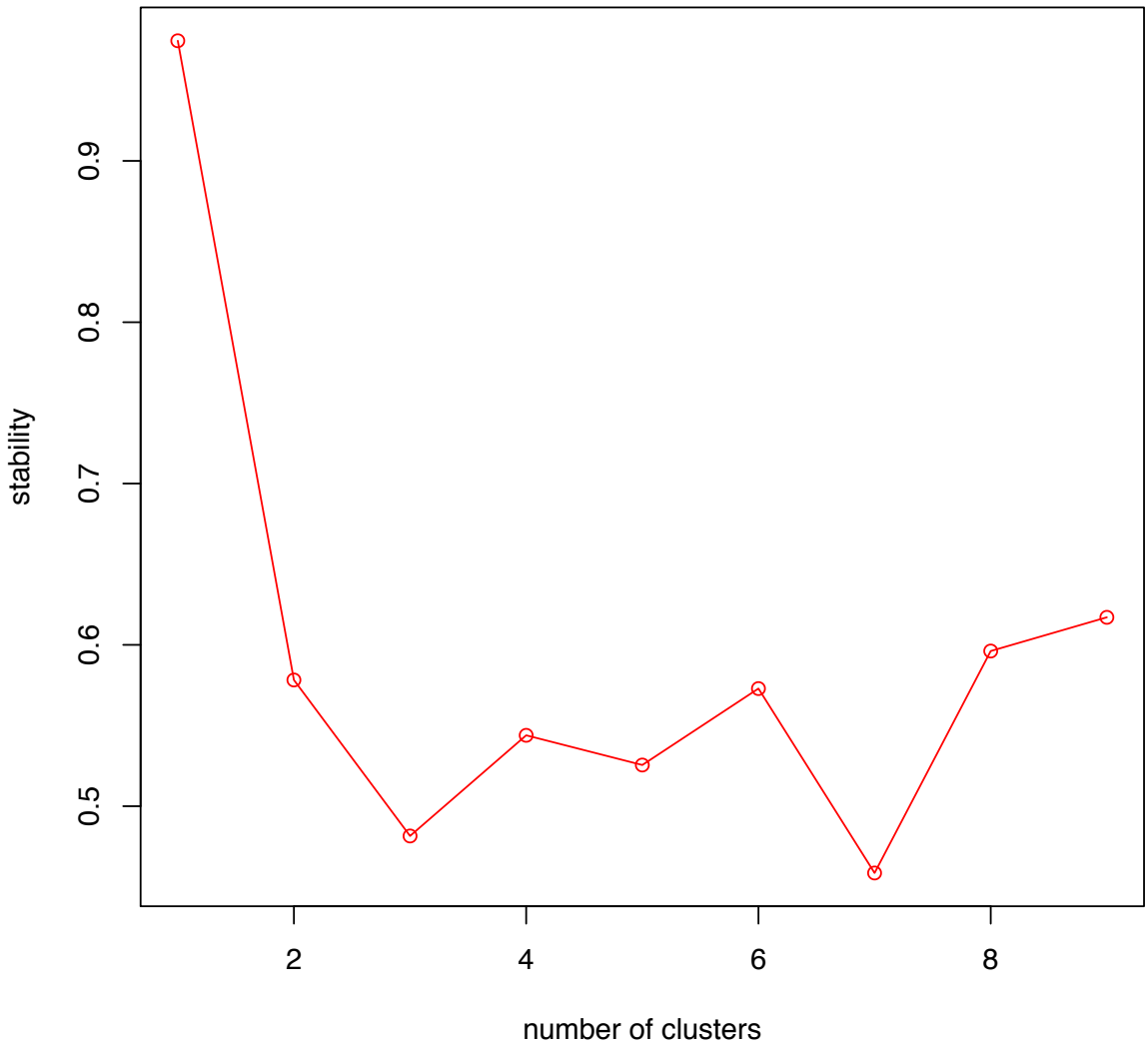

Figure 4.5: Stability of the clusters when the number of clusters varies. The stability for a specific clustering with k clusters is computed as average Jaccard similarity between the clustering of the original data and the clusterings of data with added noise.

4.1.2 Pathway level signature of subtypes

All the methods described above are able to find disease subtypes based on sample level pathway information, but do not give any information about the characteristics of the subtype. In order to fill this gap we need to identify *which pathways* are responsible for the observed stratification of samples. PLSI provides wrapper methods for feature selection as described by [58], using a naive Bayes classifier and a recursive feature elimination based on random forests [10] as implemented in the Caret R package [37].

Wrapper methods for feature selection

Feature selection based on wrapper methods follows a simple concept: including the classification problem at hand in the selection of important features. This is opposed to *filter approaches*, which instead select features independently of the classification problem. In gene expression analysis, for example, it is common to select genes based on the result of statistical tests (e.g. t-test, ANOVA, moderated t-test). Wrapper methods, instead, try to classify the data for all the possible subsets of the features, and selecting the subset with the best performance. Our problem is slightly different from a classification problem. We start with an unsupervised approach (the clustering methods described above), where we do not know the subtypes in the data, and we try to extract the features after we determine the subtypes. Therefore, we need to choose a classifier to use with wrapper methods. In PLSI we provide the Naive Bayes classifier and recursive feature elimination based on random forests.

Naive Bayes classifier The function `featureWrapperNB` implements the wrapper method with naive Bayes. We use it here with the results obtained with the Cross-Clustering method (in the object `clres`). The first action that must be performed is

dropping the outliers, since CrossClustering returns that information. As described in the previous sections, the cluster membership in the CrossClustering results is in the `clusterMemberships` field. We need to eliminate the outliers since they represent *noise* elements. Then, the cluster information is extracted and passed to the `featureWrapper` function. Additionally, the `split` parameter can be set in the function, defining the fraction of data that will be considered training. The default value for this parameter splits the data as follows: $\frac{2}{3}$ of the data goes into the training set, while $\frac{1}{3}$ goes into the test set.

```
> outlFilter <- clres$clusterMemberships != 0
> wrapData <- oraRes[outlFilter, ]
> meaningfulClusters <- clres$clusterMemberships[outlFilter]
> selectedFeatures <- featureWrapperNB(oraRes,
+         clusterResult,
+         split = 2/3,
+         method="best")
```

The `featureWrapperNB` function provides a number of search alternatives: *backward search, forward search*, and *best first search. Forward search* and *backward search* are two greedy search methodologies. The first starts from an empty set of features, evaluates all of them, then chooses the best one. Then, it evaluates all the combination between the chosen one and one of the other features. Again, the best pair is chosen, and the algorithm stops when no addition of new features improves the evaluation. The second strategy starts from all the features, and removes one feature at the time, evaluating the resulting set every time, and stopping when removing one feature does not improve the evaluation result. The *best first search* algorithm is similar to forward search, except that it does not stop at the first occurrence

of new feature that does not improve the evaluation, but it backtracks and tries a new solution from the ones already evaluated. In PLSI this algorithm backtracks a maximum of $N/10$ times, where N is the number of features.

Recursive feature elimination via random forests. The second approach for feature selection with wrapper methods is the recursive feature elimination via random forests procedure. The random forest procedure creates a number of decision trees, each one with a subset of the original features. Such subsets of features are obtained by bootstrapping the initial set of features. The set of decision trees is the *random forest*. Given an input, each one of the trees in the random forest casts a *vote* about the class of the input, and the majority of the votes decides the class. This approach has a number of advantages over well known approaches for classification, in that it does not overfit, it can be used for multi-class problems, and it is robust with regards to a large amount of noise variables in the training set [46]. Another advantage of random forests is that it assigns a measure of *importance* to each feature, proportional to the decrease in classification accuracy when the values of such variable are permuted randomly. The *recursive feature elimination* extension to random forests iteratively eliminates a fraction of the features, by ordering them in order of this importance value, and dropping the worst ones. In [18], the authors drop the 20% worst performing features. In the context of feature selection, though, it is preferable to choose a certain size of the features we want to keep, rather than the ones we want to eliminate. The recursive feature elimination provided by the `Caret` package implements the following algorithm: first, the training is performed, and the importance of all the features is determined. Then, a number k of *feature subset sizes* is chosen. We will refer to the $i-th$ subset size as S_i. These represent the sizes of the feature subsets that we want to test. For example, we could be interested in testing the performance of subsets of

size 10, 50, and 100. For each of these subsets, keep the S_i most important features, where the importance is assessed with random forest, and train again the classifier with those features. The algorithm is incorporated in a 10-fold cross validation, i.e. the train set itself is split in 10 parts, and the algorithm is applied to the subsets obtained by removing, in turn, one of the folds.

In PLSI the function that performs this selection is `featureWrapperRFE`. The parameters for random forests and recursive feature elimination are set as follows: the number of features are sampled from the logarithmic function that goes from 2 to approximatively 20% of the total number of variables.

```
> selectedFeatures <- featureWrapperRFE(wrapData, meaningfulClusters)
```

In this case the number of features was 149 (the total number of pathways). The object returned by the `featureWrapperRFE` contains the features ordered by their importance, as well as the accuracy obtained with the various subsets.

```
> selectedFeaturesRFE

Recursive feature selection

Outer resampling method: Cross-Validated (10 fold, repeated 10 times)

Resampling performance over subset size:

 Variables Accuracy  Kappa AccuracySD KappaSD Selected
         2   0.9563 0.9085    0.07240 0.15057
         3   0.9670 0.9318    0.06035 0.12389
         4   0.9660 0.9294    0.05790 0.12033
```

```
  7   0.9751 0.9480    0.05560 0.11614
 12   0.9731 0.9439    0.05644 0.11771
 20   0.9783 0.9545    0.05254 0.10986
149   0.9877 0.9756    0.03709 0.07375       *

The top 5 variables (out of 149):
   path:hsa05150, path:hsa05416, path:hsa04650, path:hsa05332, path:hsa05140
```

In this case, although the best performance is obtained by using all the features, selecting the top 7 features results in 97% accuracy, with a small standard deviation. The results, in terms of accuracy, are shown in Figure 4.6.

4.1.3 Biclustering

So far the methods employed perform subtype discovery in one direction only. This means that the subtypes found include all the pathways, and that the pathways that discriminate subtypes are not specific for each subtype. In other words, once the subtypes are identified, we need to apply a feature selection method to detect which pathways discriminate among *all* the subtypes. But what happens if only a few pathways are behaving similarly in a subset of patients, and another set of pathways is behaving similarly in *another* subset of patients? This would represent a case in which, for example, a subtype of lung cancer could involve some mechanisms in some of the patients, but another subtype could involve completely different mechanisms in other patients. One could in theory apply clustering twice, once on the sample level and once on pathway level. This approach, however, is limited by the fact that, in whichever direction we perform the clustering, *all the features of that direction* are used at the same time. This could present issues related to some features being noise

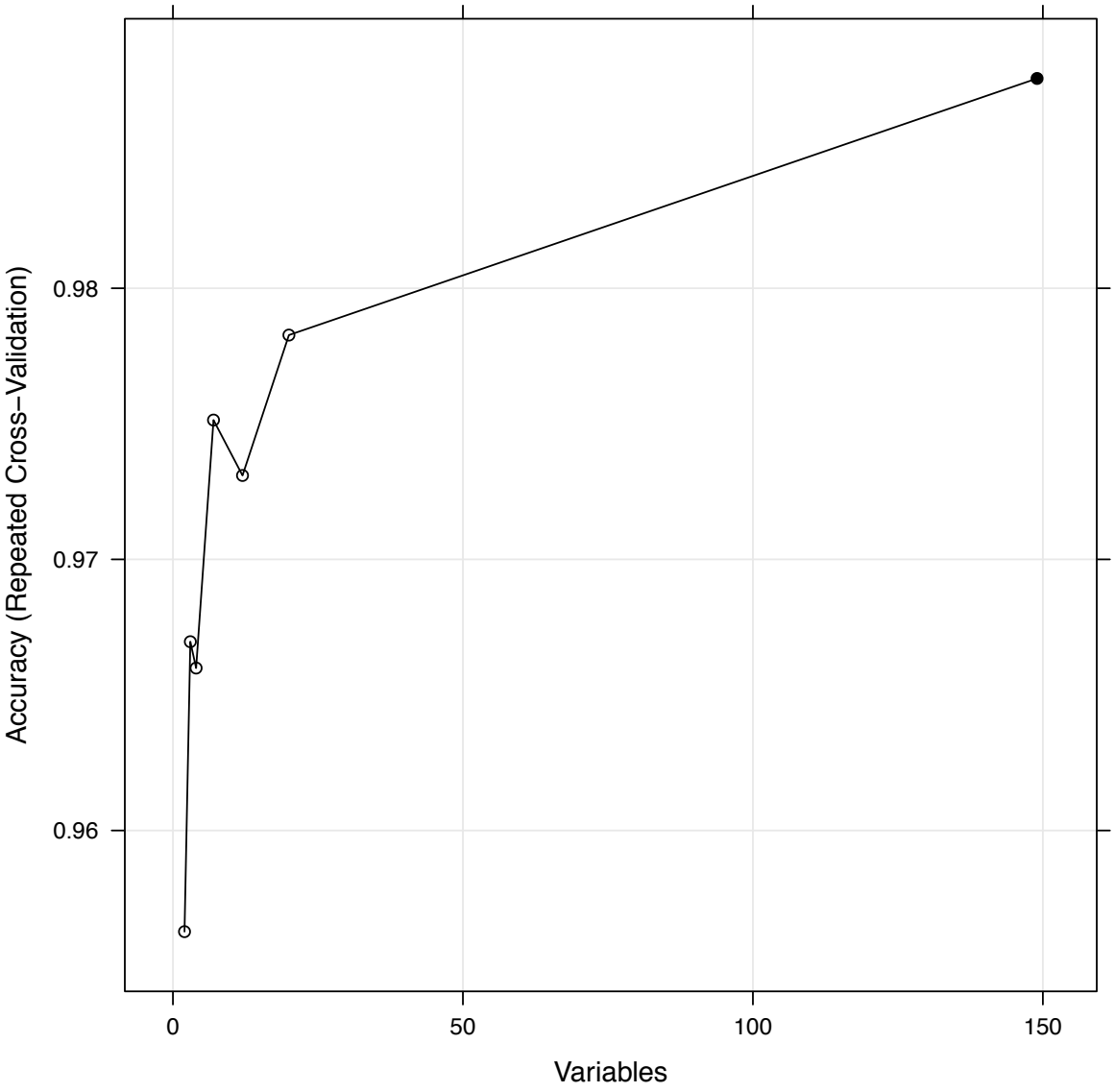

Figure 4.6: Accuracy of sets of 2, 3, 4, 7, 12, 20, and 149 features (pathways) on the GSE19188 dataset. By using the 7 best features we reach an accuracy of 97.5%. Increasing the number of features results in marginally better accuracy.

in that specific direction. In the lung cancer example we could have a few pathways being involved in some subtype, where the other pathways are noise, potentially masking the effect of the relevant pathways. Methods have been develop to deal with similar situations *at gene level*. Biclustering is one of such techniques, first developed by [15]. This method starts from the same gene expression-sample matrix E described in Table 3.2. Then, the method searches for sub-matrices of E that satisfy certain conditions on the within-submatrix variance. These sub-matrices are the biclusters. In the context of pathways, a bicluster represents a subset of pathways that are correlated under a subset of samples. Biclustering has been used in several applications, from the analysis of microarray data, as shown in the review in [66], to the identification of protein interactions [64]. Again, the same issues relative to the use of gene information apply. Gene behavior is not constant over time, and the snapshot-type measurements that are characteristics of microarray experiment change unpredictably, making it difficult to identify a phenomenon by looking at genes alone. This is why in PLSI we use biclustering in a completely innovative way to find biclusters of pathways and patients that share similar pathway perturbations. The concept is the same as described in the previous sections: pathway measurements are going to represent whole mechanisms and to be more stable than genes. In PLSI we look for biclusters in the pathway impact-sample matrix obtained with the *sample level pathway analysis* methods provided. The result is similar to the results obtained in Section 4.1.2: a pathway signature containing the pathways that are impacted coherently in a subtype. The difference between this result and the previous ones is that in this case pathway signatures are *unique for each subtype*, whereas in the previous approaches we were only able to find a set of pathways that behaved *differently* among subtypes. An overview of this approach is shown in Figure 4.7. This image shows the pathway impact-sample matrix, where cells of the matrix represent

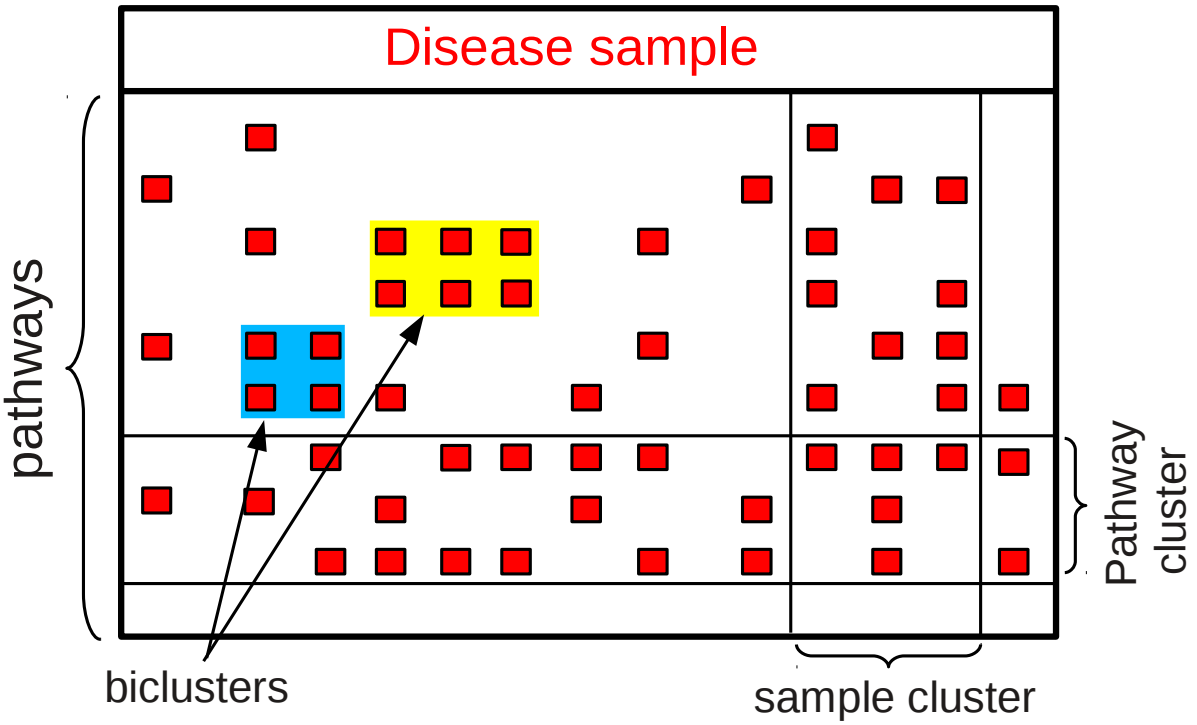

Figure 4.7: The proposed approach to identify subtypes of disease and subgroups of patients: biclustering allows PLSI to find patients that are similar over only a subgroup of pathways.

the impact of a specific pathway in a specific sample. On the bottom right of the figure we can see the results of classical cluster analysis. If we cluster rows of the matrix (pathways) we obtain similarly impacted pathways across all the patients, disregarding the subtype information. Vice versa, if we cluster columns of the matrix (disease samples) we obtain samples that behave similarly across all the pathways, disregarding that pathways may behave differently in different groups of samples. The yellow and blue boxes represent biclusters, identifying groups of pathways that behave similarly in a subset of samples.

In PLSI biclustering is performed by calling the function `biclust` from the homonymous package. This function provides a number of biclustering methods:

the [15] method, which searches for bicluster with variance lower than a predetermined threshold, the Bimax algorithm [81], a divide-and-conquer approach for biclustering that decomposes the starting matrix into three sub matrices based on their content, and then processes them recursively, the Plaid Biclustering method [62, 102], which models the expression level of a single gene in a single sample as a linear combination of the effects of a series of additive *layers* representing the various biclusters, the Xmotif and Questmotif algorithms [73], which randomly choses samples as *seeds* and uses an iterative approach to find biclusters (called *motifs* in the original work) agreeing with the sample *at least* for a big enough proportion of the total samples, the Spectral biclustering algorithm described in [57], which finds biclusters by using Singular Value Decomposition of the input matrix, in conjunction with normalization of the data.

In our tests we obtained the best results by discretizing the input matrix

```
> bcmat <- discretize(oraRes, nof=100)
> ccres <- biclust(bcmat, method=BCCC(), alpha=1)
> ccres

An object of class Biclust


call:
        biclust(x = bcmat, method = BCCC(), alpha = 1)


Number of Clusters found:  17


First  5  Cluster sizes:
                  BC 1 BC 2 BC 3 BC 4 BC 5
Number of Rows:      11    9    9    8    8
```

```
Number of Columns:   10    5    4    6    4
```

The function `simpleBC` allows us to retrieve information about the samples-pathways in a specific bicluster.

```
> simpleBC(bcmat, ccres, clusterNo=3)

$bcMat
          path:hsa04020 path:hsa04060 path:hsa04080 path:hsa04740
GSM475664             1             1             1             1
GSM475713             4             1             1             1
GSM475715             4             4             1             1
GSM475722             4             1             1             1
GSM475728             2             1             1             1
GSM475744             4             1             1             1
GSM475753             2             1             1             1
GSM475787             2             1             1             1
GSM475789             8             7             4             2

$rows
[1]  4 30 31 35 38 46 50 73 75

$cols
[1]  9 10 14 70
```

This bicluster, for example, contains nine samples, out of which six belong to the Squamous Cell Carcinoma (SCC) subtype (Fisher exact test p-value = 0.015). The pathways belonging to this bicluster are *Cytokine-cytokine receptor interaction*,

Calcium signaling pathway, *Neuroactive ligand-receptor interaction*, and the *Olfactory transduction pathway*. The *Cytokine-cytokine receptor interaction* [86] and *Calcium signaling pathway* [107] have been found to be altered in patients with esophageal SCC, while the *Olfactory transduction pathway* has been found to be down-regulated in a number of SCC cell lines [33]. The *Neuroactive ligand-receptor interaction* pathway has been related to chromosomal alteration in patients with esophageal SCC [13].

Finally, a complete list of biclusters can be found with the `bcList` function. This function returns a list of biclusters. Each element of the list contains two characters vectors, `samples` and `features` describing the bicluster. Only the first three biclusters are here reported.

```
> bl <- bcList(bcmat, ccres)
> bl[1:3]

$BiCluster_1
$BiCluster_1$samples
 [1] "GSM475694" "GSM475706" "GSM475751" "GSM475759" "GSM475761" "GSM475762"
 [7] "GSM475791" "GSM475794" "GSM475797" "GSM475803" "GSM475806"


$BiCluster_1$features
 [1] "path:hsa04060" "path:hsa04062" "path:hsa04080" "path:hsa04620"
 [5] "path:hsa04650" "path:hsa04672" "path:hsa04740" "path:hsa05145"
 [9] "path:hsa05152" "path:hsa05164"



$BiCluster_2
$BiCluster_2$samples
```

```
[1] "GSM475662" "GSM475670" "GSM475692" "GSM475698" "GSM475712" "GSM475756"
[7] "GSM475760" "GSM475779" "GSM475782"

$BiCluster_2$features
[1] "path:hsa04060" "path:hsa04080" "path:hsa04740" "path:hsa04742"
[5] "path:hsa05152"



$BiCluster_3
$BiCluster_3$samples
[1] "GSM475664" "GSM475713" "GSM475715" "GSM475722" "GSM475728" "GSM475744"
[7] "GSM475753" "GSM475787" "GSM475789"

$BiCluster_3$features
[1] "path:hsa04020" "path:hsa04060" "path:hsa04080" "path:hsa04740"
```

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

It is now well accepted that many diseases such as cancer are not a single, monolithic disease, but rather a collection of distinct diseases that show common features. Notable examples are breast cancer, which consists in at least four major subtypes (luminal A, luminal B, HER2 positive, and basal-like), Acute Myeloid Leukemia, Alzheimer, etc. Knowing exactly the subtype of a certain disease is crucial, since treatments targeted for a specific subtype are more likely to address the mechanisms of action of that subtype, and therefore more likely to succeed, and the identification of patients that are affected by a subtype makes it sure that the proper treatment is administered to the right patient. Although our knowledge of disease subtypes improved during the years, the approaches to discovery are still limited to the analysis of gene expression profiles.

Here we describe an approach that goes beyond simple gene expression, and performs sub-type discovery at *pathway level*, looking at mechanisms rather than single genes. This approach consists in two steps. First, the sample level expression profile is computed, describing how gene activity changes from sample to sample. Then, a sample level pathway profile is computed by using pathway analysis approaches. Lastly, the discovery is performed on pathway profiles, therefore looking at the behavior of complete mechanisms rather than single genes.

REFERENCES

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, 199–213. Springer, 1998.

[2] R. Arriagada, B. Bergman, T. L. Chevalier, J.-P. Pignon, and J. Vansteenkiste. Cisplatin-based adjuvant chemotherapy in patients with completely resected non-small-cell lung cancer. *The New England Journal of Medicine*, 350(4):351, 2004.

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

[4] T. Barrett, D. B. Troup, S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. A. Marshall, K. H. Phillippy, P. M. Sherman, R. N. Muertter, M. Holko, O. Ayanbule, A. Yefanov, and A. Soboleva. NCBI GEO: archive for functional genomics data sets – 10 years on. *Nucleic Acids Research*, 39(suppl 1):D1005–D1010, 2011.

[5] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of The Royal Statistical Society B*, 57(1):289–300, 1995.

[6] Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, 29(4):1165–1188, August 2001.

[7] BioCarta. BioCarta - Charting Pathways of Life. http://www.biocarta.com.

[8] C. Booth and F. Shepherd. Adjuvant chemotherapy for resected non-small cell lung cancer. *Journal of Thoracic Oncology*, 1(2):180–187, 2006.

[9] A. Brazma, H. Parkinson, U. Sarkans, M. Shojatalab, J. Vilo, N. Abeygunawardena, E. Holloway, M. Kapushesky, P. Kemmeren, G. G. Lara, A. Oezcimen, P. Rocca-Serra, and S.-A. Sansone. ArrayExpress–a public repository for microarray gene expression data at the EBI. *Nucleic Acids Research*, 31(1):68–71, 2003.

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

[11] T. Breslin, M. Krogh, C. Peterson, and C. Troein. Signal transduction pathway profiling of individual tumor samples. *BMC Bioinformatics*, 6:163, 2005.

[12] R. D. Canales, Y. Luo, J. C. Willey, B. Austermiller, C. C. Barbacioru, C. Boysen, K. Hunkapiller, R. V. Jensen, C. R. Knight, K. Y. Lee, Y. Ma, B. Maqsodi, A. Papallo, E. H. Peters, K. Poulter, P. L. Ruppel, R. R. Samaha, L. Shi, W. Yang, L. Zhang, and F. M. Goodsaid. Evaluation of DNA microarray results with quantitative gene expression platforms. *Nature Biotechnology*, 24(9):1115–22, September 2006.

[13] I. Chattopadhyay, A. Singh, R. Phukan, J. Purkayastha, A. Kataki, J. Mahanta, S. Saxena, and S. Kapur. Genome-wide analysis of chromosomal alterations in patients with esophageal squamous cell carcinoma exposed to tobacco and betel quid from high-risk area in india. *Mutation Research/Genetic Toxicology and Environmental Mutagenesis*, 696(2):130–138, 2010.

[14] J. J. Chen, H.-M. Hsueh, R. R. Delongchamp, C.-J. Lin, and C.-A. Tsai. Reproducibility of microarray data: a further analysis of microarray quality control (MAQC) data. *BMC Bioinformatics*, 8(1):412, 2007.

[15] Y. Cheng and G. M. Church. Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, 93–103. AAAI Press, 2000.

[16] G. A. Churchill. Fundamentals of experimental design for cDNA microarrays. *Nat. Genet.*, 32(Suppl. S):490–495, Dec. 2002.

[17] R. Clarke, M. C. Liu, K. B. Bouker, Z. Gu, R. Y. Lee, Y. Zhu, T. C. Skaar, B. Gomez, K. O'Brien, Y. Wang, and L. A. Hilakivi-Clarke. Antiestrogen resistance in breast cancer and the role of estrogen receptor signaling. *Oncogene*, 22(47):7316–7339, 2003.

[18] R. Díaz-Uriarte and S. A. De Andres. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7(1):3, 2006.

[19] M. Donato and S. Draghici. Crosstalk: a package for the analysis and correction of crosstalk effect in the analysis of signaling pathways. *Bioconductor*, 2015.

[20] S. W. Doniger, N. Salomonis, K. D. Dahlquist, K. Vranizan, S. C. Lawlor, and B. R. Conklin. MAPPFinder: using Gene Ontology and GenMAPP to create a global gene expression profile from microarray data. *Genome biology*, 4(1):R7, 2003.

[21] S. Drăghici, P. Khatri, A. C. Eklund, and Z. Szallasi. Reliability and reproducibility issues in DNA microarray measurements. *Trends in Genetics*, 22(2):101–109, 2006.

[22] S. Drăghici, P. Khatri, A. C. Eklund, and Z. Szallasi. Reliability and reproducibility issues in DNA microarray measurements. *Trends in Genetics*, 22(2):101–109, 2006.

[23] S. Drăghici, P. Khatri, A. L. Tarca, K. Amin, A. Done, C. Voichiţa, C. Georgescu, and R. Romero. A systems biology approach for pathway level analysis. *Genome Research*, 17(10):1537–1545, 2007.

[24] I. Drozdov, C. Ouzounis, A. Shah, and S. Tsoka. Functional Genomics Assistant (FUGA): a toolbox for the analysis of complex biological networks. *BMC Research Notes*, 4(1):462, 2011.

[25] S. Drăghici. Statistical intelligence: effective analysis of high-density microarray data. *Drug Discovery Today*, 7(11):S55–S63, 2002.

[26] S. Drăghici, P. Khatri, R. P. Martins, G. C. Ostermeier, and S. A. Krawetz. Global functional profiling of gene expression. *Genomics*, 81(2):98–104, 2003.

[27] B. Dutta, A. Wallqvist, and J. Reifman. PathNet: A tool for pathway analysis using topological information. *Source Code for Biology and Medicine*, 7(1):10, 2012.

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

[29] B. Efron. Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979.

[30] B. Efron, E. Halloran, and S. Holmes. Bootstrap confidence levels for phylogenetic trees. *Proceedings of the National Academy of Sciences*, 93(23):13429–13429, 1996.

[31] B. Efron and R. Tibshirani. The problem of regions. *Annals of Statistics*, 1687–1718, 1998.

[32] S. Efroni, C. F. Schaefer, and K. H. Buetow. Identification of Key Processes Underlying Cancer Phenotypes Using Biologic Pathway Analysis. *PLoS One*, 2(5):e425, 2007.

[33] N. F. Erdem, E. R. Carlson, and D. A. Gerard. Characterization of gene expression profiles of 3 different human oral squamous cell carcinoma cell lines with different invasion and metastatic capacities. *Journal of Oral and Maxillofacial Surgery*, 66(5):918–927, 2008.

[34] X. Fan, L. Shi, H. Fang, S. Harris, R. Perkins, and W. Tong. Investigation of reproducibility of differentially expressed genes in DNA microarrays through statistical simulation. *BMC Proceedings*, 3(Suppl 2):S4, 2009.

[35] Z. Fang, W. Tian, and H. Ji. A network-based gene-weighting approach for pathway analysis. *Cell Research*, 22(3):565–580, 2011.

[36] F. Farfán, J. Ma, M. A. Sartor, G. Michailidis, and H. V. Jagadish. THINK Back: knowledge-based interpretation of high throughput data. *BMC Bioinformatics*, 13(Suppl 2):S4, 2012.

[37] M. K. C. from Jed Wing, S. Weston, A. Williams, C. Keefer, A. Engelhardt, T. Cooper, Z. Mayer, B. Kenkel, the R Core Team, M. Benesty, R. Lescarbeau, A. Ziem, and L. Scrucca. *caret: Classification and Regression Training*, 2015. R package version 6.0-41.

[38] S. Gao and X. Wang. TAPPA: topological analysis of pathway phenotype association. *Bioinformatics*, 23(22):3100–3102, 2007.

[39] E. Glaab, A. Baudot, N. Krasnogor, R. Schneider, and A. Valencia. EnrichNet: network-based gene set enrichment analysis. *Bioinformatics*, 28(18):i451–i457, 2012.

[40] E. Glaab, A. Baudot, N. Krasnogor, and A. Valencia. TopoGSA: network topological gene set analysis. *Bioinformatics*, 26(9):1271–1272, 2010.

[41] Z. Gu, J. Liu, K. Cao, J. Zhang, and J. Wang. Centrality-based pathway enrichment: a systematic approach for finding significant pathways dominated by key genes. *BMC Systems Biology*, 6(1):56, 2012.

[42] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A k-means clustering algorithm. *Applied statistics*, 100–108, 1979.

[43] C. Hennig. Cluster-wise assessment of cluster stability. *Computational Statistics & Data Analysis*, 52(1):258–271, 2007.

[44] C. J. Hoimes and W. K. Kelly. Redefining hormone resistance in prostate cancer. *Therapeutic Advances in Medical Oncology*, 2(2):107–123, 2010.

[45] J. Hou, J. Aerts, B. Den Hamer, W. Van Ijcken, M. Den Bakker, P. Riegman, C. van der Leest, P. van der Spek, J. A. Foekens, H. C. Hoogsteden, F. Grosveld, and S. Philipsen. Gene expression-based classification of non-small cell lung carcinomas and survival prediction. *PLoS One*, 5(4):e10312, 2010.

[46] J. Hua, Z. Xiong, J. Lowey, E. Suh, and E. R. Dougherty. Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, 21(8):1509–1515, 2005.

[47] J.-H. Hung, T. W. Whitfield, T.-H. Yang, Z. Hu, Z. Weng, and C. DeLisi. Identification of functional modules that correlate with phenotypic difference: the influence of network topology. *Genome Biology*, 11(2):R23, 2010.

[48] M. A.-H. Ibrahim, S. Jassim, M. A. Cawthorne, and K. Langlands. A Topology-Based Score for Pathway Enrichment. *Journal of Computational Biology*, 19(5):563–573, 2012.

[49] S. Isci, C. Ozturk, J. Jones, and H. H. Otu. Pathway analysis of high-throughput biological data within a Bayesian network framework. *Bioinformatics*, 27(12):1667–1674, 2011.

[50] P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912.

[51] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. Gopinath, G. Wu, L. Matthews, S. Lewis, E. Birney, and L. Stein. REACTOME: a knowledgebase of biological pathways. *Nucleic Acids Research*, 33(Database issue):D428–432, 2005.

[52] M. Kanehisa, S. Goto, S. Kawashima, Y. Okunom, and M. Hattori. The KEGG resource for deciphering the genome. *Nucleic Acids Research*, 32(Database Issue):277–280, Jan 2004.

[53] P. Khatri and S. Drăghici. A comparison of existing tools for ontological analysis of gene expression data. In *Encyclopedia of Genetics, Genomics, Proteomics and Bioinformatics*, chapter 4. Wiley Online Library, 2005. 4.5:54.

[54] P. Khatri, S. Drăghici, G. C. Ostermeier, and S. A. Krawetz. Profiling gene expression using Onto-Express. *Genomics*, 79(2):266–270, 2002.

[55] P. Khatri, S. Drăghici, A. L. Tarca, S. S. Hassan, and R. Romero. A system biology approach for the steady-state analysis of gene signaling networks. In *CIARP'07 Proceedings of the Congress on pattern recognition 12th Iberoamerican conference on Progress in pattern recognition, image analysis and applications*, 32–41, Valparaiso, Chile, 13-16 Nov. 2007. ACM.

[56] P. Khatri, M. Sirota, and A. J. Butte. Ten years of pathway analysis: current approaches and outstanding challenges. *PLoS Computational Biology*, 8(2):e1002375, 2012.

[57] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome research*, 13(4):703–716, 2003.

[58] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.

[59] J. Lamb, E. D. Crawford, D. Peck, J. W. Modell, I. C. Blat, M. J. Wrobel, J. Lerner, J.-P. Brunet, A. Subramanian, K. N. Ross, et al. The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *science*, 313(5795):1929–1935, 2006.

[60] M. T. Landi, T. Dracheva, M. Rotunno, J. D. Figueroa, H. Liu, A. Dasgupta, F. E. Mann, J. Fukuoka, M. Hames, A. W. Bergen, et al. Gene expression signature of cigarette smoking and its role in lung adenocarcinoma development and survival. *PloS one*, 3(2):e1651, 2008.

[61] J. Lapointe, C. Li, J. P. Higgins, M. van de Rijn, E. Bair, K. Montgomery, M. Ferrari, L. Egevad, W. Rayford, U. Bergerheim, P. Ekman, A. M. DeMarzo, R. Tibshirani, D. Botstein, P. O. Brown, J. D. Brooks, and J. R. Pollack. Gene expression profiling identifies clinically relevant subtypes of prostate cancer. *Proceedings of the National Academy of Sciences of the United States of America*, 101(3):811–816, 2004.

[62] L. Lazzeroni and A. Owen. Plaid models for gene expression data. *Statistica Sinica*, 12(1):61–86, 2002.

[63] B. D. Lehmann, J. A. Bauer, X. Chen, M. E. Sanders, A. B. Chakravarthy, Y. Shyr, and J. A. Pietenpol. Identification of human triple-negative breast cancer subtypes and preclinical models for selection of targeted therapies. *The Journal of clinical investigation*, 121(7):2750, 2011.

[64] J. Li, K. Sim, G. Liu, and L. Wong. Maximal quasi-bicliques with balanced noise tolerance: Concepts and co-clustering applications. In *Proc. SIAM Int. Conf. on Data Mining SDM'08*, 72–83, Apr. 2008.

[65] K.-Y. Liang and S. L. Zeger. Longitudinal data analysis using generalized linear models. *Biometrika*, 13–22, 1986.

[66] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[67] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[68] M. S. Massa, M. Chiogna, and C. Romualdi. Gene set analysis exploiting the topology of a pathway. *BMC Systems Biology*, 4(1):121, 2010.

[69] P. McCullagh and J. A. Nelder. *Generalized linear models*, volume 37. CRC press, 1989.

[70] J. Mieczkowski, K. Swiatek-Machado, and B. Kaminska. Identification of Pathway Deregulation–Gene Expression Based Analysis of Consistent Signal Transduction. *PLoS ONE*, 7(7):e41541, 2012.

[71] C. Mitrea, Z. Taghavi, B. Bokanizad, S. Hanoudi, R. Tagett, M. Donato, C. Voichiţa, and S. Drăghici. Methods and approaches in the topology-based analysis of biological pathways. *Frontiers in Physiology*, 4:278, 2013.

[72] V. K. Mootha, C. M. Lindgren, K.-F. Eriksson, A. Subramanian, S. Sihag, J. Lehar, P. Puigserver, E. Carlsson, M. Ridderstråle, E. Laurila, N. Houstis, M. J. Daly, N. Patterson, J. P. Mesirov, T. R. Golub, P. Tamayo, B. Spiegelman, E. S. Lander, J. N. Hirschhorn, D. Altshuler, and L. C. Groop. PGC-11α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genetics*, 34(3):267–273, Jul 2003.

[73] T. Murali and S. Kasif. Extracting conserved gene expression motifs from gene expression data. In *Pacific Symposium on Biocomputing*, volume 8, 77–88, 2003.

[74] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, 1998.

[75] D. Pan, N. Sun, K.-H. Cheung, Z. Guan, L. Ma, M. Holford, X. Deng, and H. Zhao. PathMAPA: a tool for displaying gene expression and performing statistical tests on metabolic pathways at multiple levels for Arbidopsis. *BMC Bioinformatics*, 4(1):56, Nov 2003.

[76] K.-H. Pan, C.-J. Lih, and S. N. Cohen. Effects of threshold choice on biological conclusions reached during analysis of gene expression by DNA microarrays.

Proceedings of the National Academy of Sciences of the United States of America, 102(25):8961–8965, 2005.

[77] R. Pandey, R. K. Guru, and D. W. Mount. Pathway Miner: extracting gene association networks from molecular pathways for predicting the biological significance of gene expression microarray data. *Bioinformatics*, 20(13):2156–2158, Sep 2004.

[78] H. S. Phillips, S. Kharbanda, R. Chen, W. F. Forrest, R. H. Soriano, T. D. Wu, A. Misra, J. M. Nigro, H. Colman, L. Soroceanu, et al. Molecular subclasses of high-grade glioma predict prognosis, delineate a pattern of disease progression, and resemble stages in neurogenesis. *Cancer cell*, 9(3):157–173, 2006.

[79] K. Polyak. Heterogeneity in breast cancer. *The Journal of Clinical Investigation*, 121(10):3786–3788, 2011.

[80] K. Polyak and P. K. Vogt. Progress in breast cancer research. *Proceedings of the National Academy of Sciences*, 109(8):2715–2717, 2012.

[81] A. Prelić, S. Bleuler, P. Zimmermann, A. Wille, P. Bühlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129, 2006.

[82] J. Quackenbush. Computational analysis of microarray data. *Nature Reviews Genetics*, 2:418–427, 2001.

[83] J. Rahnenführer, F. S. Domingues, J. Maydt, and T. Lengauer. Calculating the Statistical Significance of Changes in Pathway Activity From Gene Expression Data. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 2004.

[84] P. N. Robinson, A. Wollstein, U. Bohme, and B. Beattie. Ontologizing gene-expression microarray data: characterizing clusters with Gene Ontology. *Bioinformatics*, 20(6):979–81, 2004.

[85] G. Rustici, N. Kolesnikov, M. Brandizi, T. Burdett, M. Dylag, I. Emam, A. Farne, E. Hastings, J. Ison, M. Keays, N. Kurbatova, J. Malone, R. Mani, A. Mupo, R. P. Pereira, E. Pilicheva, J. Rung, A. Sharma, Y. A. Tang, T. Ternent, A. Tikhonov, D. Welter, E. Williams, A. Brazma, H. Parkinson, and U. Sarkans. ArrayExpress update–trends in database growth and links to data analysis tools. *Nucleic Acids Research*, 41(D1):D987–D990, 2013.

[86] A. Saleh, R. Zain, H. Hussaini, F. Ng, V. Tanavde, S. Hamid, A. Chow, G. Lim, M. Abraham, S. Teo, et al. Transcriptional profiling of oral squamous cell carcinoma using formalin-fixed paraffin-embedded samples. *Oral oncology*, 46(5):379–386, 2010.

[87] R. Shai, T. Shi, T. J. Kremen, S. Horvath, L. M. Liau, T. F. Cloughesy, P. S. Mischel, and S. F. Nelson. Gene expression profiling identifies molecular subtypes of gliomas. *Oncogene*, 22(31):4918–4923, 2003.

[88] S. V. Sharma and J. Settleman. Oncogene addiction: setting the stage for molecularly targeted cancer therapy. *Genes & Development*, 21(24):3214–3231, 2007.

[89] H. Shimodaira et al. Approximately unbiased tests of regions using multistep-multiscale bootstrap resampling. *The Annals of Statistics*, 32(6):2616–2641, 2004.

[90] A. Shojaie and G. Michailidis. Analysis of Gene Sets Based on the Underlying Regulatory Network. *Journal of Computational Biology*, 16(3):407–426, 2009.

[91] G. K. Smyth. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical applications in genetics and molecular biology*, 3(1):Article3, 2004.

[92] T. Sørlie, C. M. Perou, R. Tibshirani, T. Aasf, S. Geislerg, H. Johnsenb, T. Hastiee, M. B. Eisen, M. van de Rijn, S. S. Jeffreyj, T. Thorsenk, H. Quistl,

J. C. Matesec, P. O. Brownm, D. Botsteinc, P. E. Lønningg, and A.-L. Børresen-Daleb. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences, USA*, 98(19):10869–10874, 2001.

[93] C. Sotiriou, S.-Y. Neo, L. M. McShane, E. L. Korn, P. M. Long, A. Jazaeri, P. Martiat, S. B. Fox, A. L. Harris, and E. T. Liu. Breast cancer classification and prognosis based on gene expression profiles from a population-based study. *Proceedings of the National Academy of Sciences*, 100(18):10393–10398, 2003.

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

[95] R. Suzuki and H. Shimodaira. Pvclust: an r package for assessing the uncertainty in hierarchical clustering. *Bioinformatics*, 22(12):1540–1542, 2006.

[96] A. L. Tarca, S. Drăghici, P. Khatri, S. S. Hassan, P. Mittal, J.-S. Kim, C. J. Kim, J. P. Kusanovic, and R. Romero. A novel signaling pathway impact analysis (SPIA). *Bioinformatics*, 25(1):75–82, 2009.

[97] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, 1999.

[98] TCGA Research Network. The Cancer Genome Atlas. http://cancergenome.nih.gov/.

[99] P. Tellaroli, M. Bazzi, M. Donato, A. R. Brazzale, and S. Draghici. Cross-clustering: a partial clustering algorithm with automatic estimation of the number of clusters. *Submitted to Bioinformatics*, 2015.

[100] L. Tian, S. A. Greenberg, S. W. Kong, J. Altschuler, I. S. Kohane, and P. J. Park. Discovering statistically significant pathways in expression profiling studies. *Proceeding of The National Academy of Sciences of the USA*, 102(38):13544–13549, 2005.

[101] R. W. Tothill, A. V. Tinker, J. George, R. Brown, S. B. Fox, S. Lade, D. S. Johnson, M. K. Trivett, D. Etemadmoghadam, B. Locandro, et al. Novel molecular subtypes of serous and endometrioid ovarian cancer linked to clinical outcome. *Clinical Cancer Research*, 14(16):5198–5208, 2008.

[102] H. Turner, T. Bailey, and W. Krzanowski. Improved biclustering of microarray data demonstrated through systematic performance tests. *Computational statistics & data analysis*, 48(2):235–254, 2005.

[103] M. J. Van De Vijver, Y. D. He, L. J. van't Veer, H. Dai, A. A. Hart, D. W. Voskuil, G. J. Schreiber, J. L. Peterse, C. Roberts, M. J. Marton, et al. A gene-expression signature as a predictor of survival in breast cancer. *New England Journal of Medicine*, 347(25):1999–2009, 2002.

[104] C. J. Vaske, S. C. Benz, J. Z. Sanborn, D. Earl, C. Szeto, J. Zhu, D. Haussler, and J. M. Stuart. Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using PARADIGM. *Bioinformatics*, 26(12):i237–i245, 2010.

[105] C. Voichiţa, M. Donato, and S. Drăghici. Incorporating gene significance in the impact analysis of signaling pathways. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 1, 126–131, Boca Raton, FL, USA, 12-15 Dec. 2012. IEEE.

[106] T. Winton, R. Livingston, D. Johnson, J. Rigas, M. Johnston, C. Butts, Y. Cormier, G. Goss, R. Inculet, E. Vallieres, W. Fry, D. Bethune, J. Ayoub, K. Ding, L. Seymour, B. Graham, M.-S. Tsao, D. Gandara, K. Kesler,

T. Demmy, and F. Shepherd. Vinorelbine plus cisplatin vs. observation in resected non–small-cell lung cancer. *New England Journal of Medicine*, 352(25):2589–2597, 2005.

[107] Y. Wu, A. J. Palad, W. J. Wasilenko, P. F. Blackmore, W. A. Pincus, G. L. Schechter, J. R. Spoonster, E. C. Kohn, and K. D. Somers. Inhibition of head and neck squamous cell carcinoma growth and invasion by the calcium influx inhibitor carboxyamido-triazole. *Clinical cancer research*, 3(11):1915–1921, 1997.

[108] J. Xia and D. S. Wishart. MetPA: a web-based metabolomics tool for pathway analysis and visualization. *Bioinformatics*, 26(18):2342–2344, 2010.

[109] Y. H. Yang and T. Speed. Design issues for cDNA microarray experiments. *Nature Reviews Genetics*, 3(8):579–588, 2002.

[110] Y. Zhao, M.-H. Chen, B. Pei, D. Rowe, D.-G. Shin, W. Xie, F. Yu, and L. Kuo. A Bayesian Approach to Pathway Analysis by Integrating Gene–Gene Functional Directions and Microarray Data. *Statistics in Biosciences*, 4(1):105–131, 2012.

ABSTRACT

PLSI: A COMPUTATIONAL SOFTWARE PIPELINE FOR PATHWAY LEVEL DISEASE SUBTYPE IDENTIFICATION

by

MICHELE DONATO

December 2015

Advisor: Dr. Sorin Draghici

Major: Computer Science

Degree: Master of Science

It is accepted that many complex diseases consist in collections of distinct genetic diseases. Clinical advances in treatments are attributed to molecular treatments aimed at specific genes resulting in greater efficacy and fewer debilitating side effects. This proves that it is important to identify and appropriately treat each individual disease subtype. Our current understanding of subtypes is limited: despite targeted treatment advances, targeted therapies often fail for some patients. The main limitation of current methods for subtype identification is that they focus on *gene expression*, and they are subject to its intrinsic noise. Signaling pathways describe *biological processes* that are carried out by networks of genes interacting with each other. We developed PLSI, a software that allows to identify the specific pathways impacted in individual patients, subgroups of patients, or a given subtype of disease. The expected impact includes a better understanding of disease and resistance to treatment.

AUTOBIOGRAPHICAL STATEMENT

Michele Donato was born in Carrara, Italy. He received a Master's degree in Computer Engineering at the University of Pisa in 2006. He joined Wayne State University in the Winter of 2008 to pursue graduate studies in Computer Science, with his advisor Sorin Draghici. His research interests include analysis and interpretation of biological networks.