2013 | School voor Informatietechnologie

DOCTORAATSPROEFSCHRIFT

Statistical Methods for Analysis of High Throughput Experiments in Early Drug Development

Proefschrift voorgelegd tot het behalen van de graad van doctor
in de statistiek, te verdedigen door

Khamiakova Tatsiana

Promoter: prof. dr. Ziv Shkedy

D/2013/2451/60

Aan Artjom en Wibren

Acknowledgements

The First of the three Newton's laws of graduation (www.phdcomics.com) says that "A student in procrastination remains in procrastination unless an external force is applied to it". Thank you, Ziv, for applying this external force to finalize my PhD work and write this dissertation and for dealing with me during these four years! I would like to thank the two people who were especially supportive and kind to me in the first stage of my PhD, - Suzy van Sanden and Dan Lin.

Marc en Tomasz, bedankt dat jullie in mij en mijn onderzoek geloofde en om jullie kostbare tijd in de evaluatie meetings. Tomasz, nogmaals bedankt voor het zorgvuldig doorlezen van mijn proefschrift.

I would also like to thank Willem Talloen, who readily took a role of being the PhD mentor from our industrial partner, Janssen Pharmaceutica. I am grateful to Adetayo Kasim for his optimism and positive mood at any stage of my PhD research. I would like to thank Hinrich Göhlmann, Luc Bijnens and Dhammika Amaratunga for the vivid and enriching discussions as well as the organization of the meetings at La Calestienne. Speaking of La Calestienne, I cannot forget mentioning the friendly atmosphere, stimulating environment any researcher can dream of. My appreciation goes to our scientific collaborators from Johannes Kepler University and specifically to Sepp Hochreiter, Ulrich Bodenhofer and Okko - it was a sheer pleasure to work with you (despite the grinding comments of Sepp on the paper of biclustering diagnostics). An de Bondt en Tine Casneuf, hartelijk bedankt voor jullie interesse en ondersteuning in projecten waarmee we samenwerkte! Sebastian Kaiser and Friedrich Leisch, thank you for your inspiration work on biclustering and the development of R software.

I would like to thank my office mates at C107 and B2 and the other colleagues of

CenStat for their company and good mood during coffee and lunch breaks. The Beerse-trip team, which has changed from the four years ago (Setia, Kasim, and Dan) and now (Martin, Nolen, Doreen, Sammy, Leaky, and Theo).
My period at UHasselt would not be complete without a diverse international group of friends from Italy, Czech Republic, the Ukraine, Colombia, Brazil, China, Spain, Germany, the Netherlands and Belgium.
Я не могу себе вообразить учебу в Бельгии без постоянной поддержки со стороны моих друзей во Франции и родных в Беларуси. Мама, твой пример неоднократно вдохновлял меня и заставлял двигаться дальше, даже когда казалось, что я в тупике и выхода не было видно. Папа и мама, Аксюта и Тьерри – вы все сделали свой огромный вклад в то, чтобы я смогла учиться в магистратуре, а потом и завершить работу над диссертацией. Спасибо вам от всей души!
Ik ben heel erg dankbaar mijn schoonfamilie die ondersteuning bidde aan mijn man, Wibren en ons zoontje Artjom. De laatste jaren waren niet gemakkelijk en zonder deze ondersteuning zou het erg lastig zijn voor hun om bij mij de hele tijd te zijn. Bedankt voor de gezellige week-ends in Bennekom!
Mijn lieve schat, Wibren, bedankt om altijd bij mij te zijn wanneer ik je nodig had. Jou liefde en ondersteuning hebben het duizenden keer gemakkelijker gedaan en de zware last van alle "dip" momenten lichter gemaakt.
Ten slotte, bedankt allemaal, wie ik misschien vergeten ben en niet expliciet had benoemd! Thank you so much for just being there for me!

Diepenbeek, October, 2013

Contents

Publications

The material presented in Part I has been based on the following publications and reports:

Khamiakova T., Shkedy Z., Amaratunga D., Talloen W., Göhlmann H., Bijnens L., Kasim A. (2013) Quality Control of Platinum Affymetrix Microarray Dataset by Probe Level Mixed Models. Manuscript under revision at *Mathematical Biosciences.*

Kasim A., **Khamiakova T.**, Shkedy Z. , Gill J., Talloen W.(2013) Using hierarchical mixed effects model for detection of genome-wide differential splicing using exon arrays. *Working paper.*

Khamiakova T., Talloen W., Shkedy Z., Kasim A. (2010) I/NI calls: adjustment for covariates. *Technical report.*

The material presented in Part II has been based on the following publications and reports:

Khamiakova T., Shkedy Z., Hochreiter S., Leisch F., Kaiser S., Amaratunga D., Talloen W., Kasim A. (2013) Diagnostics of Biclustering Solutions in Gene Expression Experiments: Stratification. *Working Paper.*

Hochreiter S., Bodenhofer U., Heusel M., Mayr A., Mitterecker A., Kasim A., **Khamiakova T.**, Van Sanden S., Lin D., Talloen W., Bijnens L., Göhlmann H., Shkedy Z., Clevert D.-A. (2010) Fabia: factor analysis for bicluster acquisition. *Bioinformatics*, **26**, 1520-1527.

The material presented in Part III has been based on the following publication:

Khamiakova T., Shkedy Z., Hochreiter S., Talloen W., Kasim A. (2013) Joint biclustering analysis of miRNA and mRNA data. *Working Paper*

Software development:

Kaiser S., Santamaria R., **Khamiakova T.**, Sill M., Theron R., Quintales L., Leisch F. (2011) *biclust: BiCluster Algorithms.* `http://CRAN.R-project.org/package=biclust`. R package version 1.0.1.

Khamiakova T. (2013) *superbiclust: Generating Robust Biclusters from a Bicluster Set (Ensemble Biclustering).* `http://CRAN.R-project.org/package=superbiclust`. R package version 0.1.

Mengsteab A., Otava M., **Khamiakova T.** (2013) *BcDiag: Diagnostics plots for Bicluster Data.* `http://CRAN.R-project.org/package=bcdiag`. R package version 1.0.2.

Chapter 1

Introduction

Advances in biotechnology and the ability to obtain molecular profiles of biological samples, and in particular, the transcriptomic data, have been transforming the way biomedical research and early drug development are carried out for more than a decade (Clarke *et al.*, 2004; Chengalvala *et al.*, 2007; Hughes *et al.*, 2011). In view of increasing costs of the drug development and nevertheless a large number of drugs which fail the clinical trials either due to the lack of efficacy or side effects, novel technologies can potentially assist the selection of the most promising compounds for later stages in the drug discovery process. In particular, gene expression experiments are known to be of use for the compound target identification and side-effect profiling or *toxicogenomics* (Chengalvala *et al.*, 2007). The target identification sheds a light on mechanism of action of a drug, whereas toxicogenomics can assist in prioritizing compounds for further development. The knowledge of gene expression activity induced by certain treatment can pinpoint genotypes which would either benefit from the treatment or have side effects in the late phases of clinical trials, thus, saving development costs. As pointed out by Xie *et al.* (2012); Ma and Zhao (2012a), cancer, neurological disorders and other complex diseases involve many genes and biological pathways. Therefore, development of multi-targeting therapeutics is needed for successful treatment of complex disorders.

To obtain the knowledge of activated pathways, either on-target, desirable effects or off-target, possible side-effects, large numbers of gene expression experiments are carried out. In addition to the *in-house* data, i.e., data collected within a particular company, pharmaceutical industry may profit from the public data collections, such as Connectivity Map (Lamb, 2007) and Comparative Toxicogenomics Database (Davis *et al.*, 2011). The efficient use of the data collections, either public or commercial, requires high quality data preprocessing and analysis tools. This PhD dissertation is dedicated to statistical aspects of data preprocessing and analysis.

The dissertation consists of three parts. Part I focuses on the probe-level analysis of Affymetrix microarray data. Part II deals with biclustering analysis of gene expression data. Part III introduces joint biclustering methods for data integration and exploration.

1.1 Probe-level analysis of Affymetrix microarray data

Development of best practices in gene expression data preprocessing is impossible without deep understanding of the technology generating the data. There are a variety of tools for measuring gene expression, and at the moment, the most popular tools are microarrays. Among the large number of microarray datasets stored in the public repository Gene Expression Omnibus (Edgar *et al.*, 2002), the most widely used platform is Affymetrix (Lockhart *et al.*, 1996). Affymetrix microarrays quantify the gene expression by the hybridization of RNA sequences or *transcripts* to multiple probes, which are 25 base pairs long sequences printed on a microarray slide. The multiple probes which are targeting the same gene define a *probe set*. Affymetrix supplies annotation files, which contain definitions of probe sets. However, due to the fact that information about genomes is being constantly updated there are alternative ways of defining the probe sets in view of the latest genome build. The probe set definition is a very important factor in gene expression analysis, since in most of the analysis workflows expression values of probes in a probe set are summarized.

The research discussed in this dissertation is aimed at investigation of the question how standard workflows involving data summarization compare to the approach when summarization step is not performed. In addition, we cover a number of possible scenarios in the data: presence of factors influencing gene expression, identification of technical artifacts, comparability of technical and biological replicates, data-driven definition of probe sets. For these scenarios we introduce a family of linear mixed-effects models.

In the context of the analysis of probe level data and filtering, the linear mixed-effects model framework is a versatile tool, which decomposes the total variability in the probe level data into different sources of variability. This enables us to diagnose and, among others, correct for batch effects and provide more reliable estimates of standard errors for the effects of interest. Chapter 2 provides an overview of the probe level analysis of Affymetrix microarray data and the statistical aspects which are further addressed within the first part of the dissertation. Chapter 3 discusses filtering of probe sets which have inconsistent variation and develops concepts of probe-level analysis based on the experimental design. Chapter 4 presents diagnostics of a newly introduced Affymetrix benchmark dataset by using the methodology described in Chapter 3. Chapter 5 shows a simulation study to compare linear mixed-effects model framework with the currently used tools for the differential expression analysis. In Chapter 6, we illustrate how the current annotation of the Affymetrix data can be improved by applying the linear mixed-effects models to the case studies.

1.2 Biclustering analysis of gene expression experiments

In the early drug discovery, hundreds of compounds can be profiled in terms of gene expression within one experiment. Due to the decreased price of the microarrays, gene expression plays an increasingly important role for the selection of lead compounds, which are recommended for further investigation within a certain drug development program.

The datasets generated from gene expression profiling experiments have the following characteristics: availability of controls, i.e., baseline gene expression, no or low number of replicates for a given drug, chemical diversity of the selected compounds. In addition, most drug development programs deal with complex diseases, therefore, multiple processes are expected to be activated or suppressed at the cell level. This setting is very heterogeneous, compared to the straightforward 'treatment *versus* control' experiments. This setting motivated development of unsupervised data mining tools, in particular, *biclustering*.

Biclustering is a data mining process, which discovers potential drug-gene expression interactions or gene expression modules activated by certain classes of drugs. The advantage of unsupervised learning is that it does not need prior knowledge about genes groups or pathways, which is incomplete. Moreover, the compound grouping is a data-driven process. Biclustering is an exploratory technique and is designed to deal with high dimensional data. Due to the growing interest in biclustering techniques, we have investigated a number of questions related to the interpretation of biclustering results.

We start the second part of the dissertation with a review of the biclustering problem and the biclustering methods in Chapter 7. Further, we focus on three major statistical aspects of biclustering: (1) evaluation of biclustring methods in terms of robustness to noise and the initialization with random seeds, via a simulation study (Chapter 8), (2) ensemble of biclustering results in order to obtain robust biclusters and the developed R software (Chapter 9), and (3) stratification of biclusters based on co-expression patterns (Chapter 10).

1.3 Joint unsupervised analysis of high throughput data

Gene expression experiments provide important characteristics of compound activity. In addition to gene expression data, other valuable sources of information are collected to better characterize underlying cellular processes or chemical properties of compounds. While the amount of the *paired* experiments, i.e., experiments in which one drug is profiled in terms of miRNA, gene, protein expression and other sources, such as chemical properties, bioassay data is increasing, the joint analysis of the paired data is of primary interest (Ma and Zhao, 2012b).

Biclustering and other dimensionality reduction techniques gained high popularity for the analysis of high dimensional data sets separately, leaving integration for an extra step in analysis workflow. The full potential of biclustering methods to jointly explore paired datasets has not been used though. To fill this gap, in the third part of the PhD dissertation we extend the biclustering setting from one high dimensional dataset to the paired high dimensional data.

The concepts of *joint biclustering* are discussed in Chapter 11. Then, in Chapter 12 we show how gene expression modules are discovered when an *in-house* gene expression dataset is analyzed jointly with Connectivity Map profiles. In addition we describe a semi-supervised biclustering, when pathways of interest are used to initialize the bicluster discovery. Chapter 13 focuses on the discovery of joint miRNA-mRNA modules in paired miRNA and mRNA expression data by a biclustering method, which utilizes anti-correlation properties of miRNA and mRNA expression values.

Chapter 14 deserves special attention as it presents joint analysis of gene expression data and representation of chemical structure of investigated compounds. The brief overview of the way to represent a three-dimensional molecule of a compound as a bit pattern is shown and the discovery of common substructures of the molecules together with the induced pathways is discussed.

Part I

Probe-level analysis of Affymetrix microarray data

Chapter 2

Introduction to probe-level analysis of Affymetrix GeneChip data

Throughout the first part of the dissertation we are going to deal with Affymetrix microarray data. In Section 2.1 the Affymetrix GeneChip® data structure is introduced and the probe-level data is described. Section 2.2 presents research questions dealt within this part of the dissertation. The two summarization methods for the probe-level Affymetrix GeneChip® data are described in Section 2.3. Section 2.4 gives overview of four microarray case studies, to which the methods for probe-level analysis and filtering are applied.

2.1 Probe-level data structure

The most popular microarray platform for capturing gene expression is Affymetrix GeneChip®. It is an oligonucleotide microarray, where nucleotides of the probe sequences are printed on a silicone slide or chip. After the hybridization between labeled transcript and a chip probe occurs, the machine scans the microarray and based on the light intensities, quantification of gene expression is performed. A typical probe for the GeneChip® is a nucleotide sequence of 25 base pairs, chosen to target parts of the known gene sequence. Depending on the organism, expression levels of $11,000$ to $42,000$ genes are measured. In a GeneChip®, expression of a single gene is captured by 11 to 20 probes. The collection of probes intended to measure a given gene is called a probe set. The experimental data are written in CEL files where raw intensities per each probe are stored. The intensities of probes in a probe set are not the same due to a number of factors: a specific sequence (e.g., GC binds better than AT), a location of the probe on the chip, defects in probe manufacturing. On the other hand, intensities for a given probe set can vary due to array-specific effects (technical artifacts, e.g., batch effects) and experimental factors (conditions) such as treatment or time.

2.2 Statistical aspects of microarray probe-level data analysis

A standard approach for the analysis of microarray experiments is a sequential workflow in which probes' intensities are preprocessed (taking out systematic effects by background correction and normalization) and summarized. The major motivation for summarizing the probe-level data is reducing dimensionality and facilitating computational tasks. Figure 2.1 illustrates the difference between probe-level and summarized data, where expression values for each probe in a probe set are plotted together with the summarized gene expression by the two methods, RMA and FARMS, described in Section 2.3. For each array only one value is obtained per gene after summarization.

Many methods have been developed to reduce the bias and systematic error in the data during the preprocessing and summarization stages. However, the more different steps should be combined before data are analyzed, the higher is the chance of increasing the bias in the final step of the analysis. Therefore, there has been a growing interest in analyzing probe-level data of GeneChip® experiments (Chu *et al.*, 2002; Cambon *et al.*, 2007; De Neve *et al.*, 2009; Stevens *et al.*, 2010; Gupta *et al.*, 2010; De Neve *et al.*, 2013) and other types of microarray experiments, such as Illumina and tiling arrays (Forcheh *et al.*, 2012; Clement *et al.*, 2012; De Beuf *et al.*, 2012). For the remainder of this section we discuss several aspects that should be taken into account while analyzing Affymetrix GeneChip® data, namely probe-level filtering of genes, probe-level analysis of differential expression, quality control of the experiment and tackling technical artifacts, experiment-specific annotation of probe sets and differential splicing.

2.2.1 Probe-level filtering in microarray experiments

In a typical microarray experiment, only a fraction of genes are expressed, specific to the experimental factors and the tissue. In fact, various experimental data show that about 10-20% of genes are expressed in a cell (Su *et al.*, 2002). If an experiment

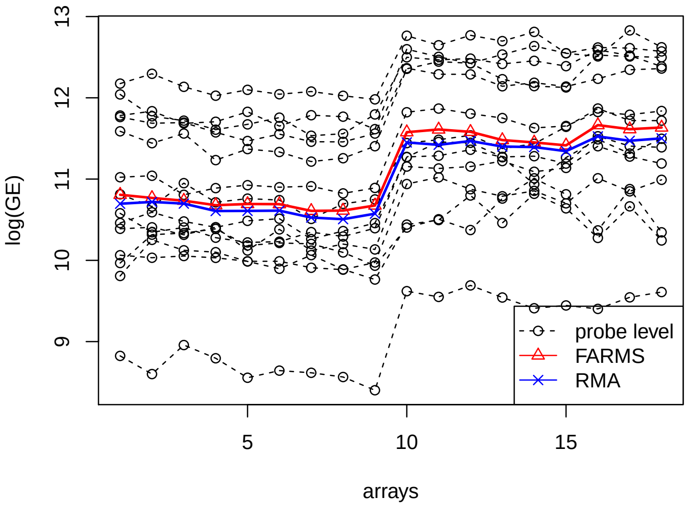


Figure 2.1: Probe-level and summarized gene-expression (GE) profiles. At the probe-level, 14 lines represent probe-specific expression profiles, whereas in FARMS and RMA all probes expression values for a given array are summarized in one value.

is conducted properly, the expressed genes will be captured by designated probe sets, and the rest of the probe sets will contain noise. Summarization and inclusion of the noisy probe sets into the analysis workflow can essentially affect the results. Therefore, a feature selection process, termed *gene filtering*, in which biologically relevant probe sets are chosen for the analysis, becomes a standard step of the data preprocessing. Gene filtering can be done either on the level of summarized data (Bourgon *et al.*, 2010) or on the probe-level data (Talloen *et al.*, 2007; Kasim *et al.*, 2010).

The probe-level data have an advantage, since they carry information about whether probe sets capture gene transcripts or not. Thus, by studying probe-level intensities one can filter out the irrelevant, noisy probe sets from further analysis, improving the power of downstream analysis.

The feature selection based on probe-level analysis is an area of ongoing research (Lu *et al.*, 2011; Forcheh *et al.*, 2012). The essence of any probe-level filtering method is in the ability of probes for a given gene to capture the true expression, which is unobserved. For the Affymetrix platform, if the probes measure the same transcript present in the sample, the probe-level intensities should vary consistently. The consistency of probe expression levels for a given gene can be measured by introducing array effect as an observed factor (Calza *et al.*, 2007) or as a latent factor (Talloen *et al.*, 2007; Kasim *et al.*, 2010; Lu *et al.*, 2011). If a probe set passes the filtering criterion it is called informative; otherwise it is filtered out and declared non-informative. This process is termed a Informative/Non informative call or I/NI call.

The filtering score in the I/NI procedure proposed by Talloen *et al.* (2007) is the conditional variance of a latent factor obtained from Factor Analysis for Robust Microarray Summarization (FARMS) (Hochreiter *et al.*, 2006). Lu *et al.* (2011) proposed a filtering procedure based on principal component analysis, in which the filtering score is the proportion of variability explained by the first principal component. PCA-based and FARMS-based filtering have the same underlying idea as it has been shown in Lu *et al.* (2011).

Calza *et al.* (2007) proposed Filtering Likely Uninformative Sets of Hybridizations (FLUSH) method based on probe-set-specific linear models with fixed effects for arrays and probes. If the probe set has high array-to-array variability compared to the measurement error, it is called informative. The estimation of array-to-array variability is captured by a Chi-square statistic (Calza *et al.*, 2007). Kasim *et al.* (2010) argued that due to the nature of Affymetrix data, the approach of FLUSH (Calza *et al.*, 2007) is not always appropriate since different probes in the same probe set are assumed to be independent. As an alternative, a linear mixed-effects model approach was used in order to check if the probes in a probe set measure the same transcript. Kasim *et al.* (2010) introduced several filtering scores such as intra-cluster correlation, model selection criteria, and variance components significance testing for detecting informative/non-informative calls in the probe-level data. Moreover, it has been shown that different parametrizations of the linear mixed-effects model correspond to the confirmatory factor analysis and the underlying factor analysis model used by the FARMS-based I/NI calls. Both methods exploit the idea of investigating the sources of variability in the data. I/NI calls and linear mixed-effects model filtering perform equally well, as reported by Kasim *et al.* (2010). The comparison of the PCA- and FARMS-based filtering has shown that they performed equally well or PCA-based filtering was slightly better for the small sample size (Lu *et al.*, 2011).

It should be mentioned that the unsupervised filtering methods consider the signal intensity at the probe level and do not look into the underlying structure of the signal. However, in more complex experimental designs the variability in expression values can be influenced by the conditions of interest, technical artifacts, or both. In this case, one would like to deconvolute the overall signal and disentangle the effect of design variable(s) from other sources of variability that can be attributed to a latent variable. Deconvolution of the signal into the variation caused by experimental factors and unexplained residual variation provides more insights and exploits the richness of probe-level microarray data.

Our first research goal in this part of the dissertation is to investigate how the experimental design influences the variability structure of the probe-level data and affects the filtering. In addition, we investigate how the criteria for the unsupervised filtering capture explained and unexplained variability. To perform the signal decomposition and track changes in variation sources we propose to extend the linear mixed-effects model approach of Kasim *et al.* (2010). The linear mixed-effects model framework is flexible and can be readily extended to the complex setting with a number of design covariates as well as with various correlation structures. From here onward we refer to this modeling approach as the extended mixed-effects model approach.

2.2.2 Differential expression analysis in microarray experiments

Since the expression of a gene is captured by several probes on a microarray chip, the set of intensities corresponding to one gene represent a multivariate measurement of gene expression. Taking into account multivariate nature of probe-level data, Chu *et al.* (2002) proposed the linear mixed-effects model that can take into account fixed experimental covariates and random effects of array. Their model had several fixed effects, such as cell line, treatment, probe effects and a random effect of an array. The flexibility of mixed models allows to construct various extensions and to test for various effects in the data. Cambon *et al.* (2007) studied different sources of variability in the probe sets, such as probe location within the transcript, middle base pair of the probe sequence, probe overlap, sequence homology and affinity. They have discovered that affinity, middle base pair and probe location effects account only for a small proportion of the variation in the data at the probe set level.

Stevens *et al.* (2010) have conducted the comparison of probe-level and probe-set models for the small sample size datasets. Stevens *et al.* (2010) have shown on the example of several *spiked-in* datasets that the probe-set models are not consistently superior to the probe-level models.

The second research goal in this part of the dissertation is to compare between the test of differential expression of the probe-level and summarized gene expression data. As it has been mentioned above, analysis based on probe-level intensities uses more information about data structure and is expected to increase the power of differential expression testing. In Chapter 5 we present a simulation study in which the two analysis methods are compared and evaluated for different signal-to-noise ratio and varied sample sizes.

2.2.3 Experiment-specific probe set annotation

The probe-level analysis and filtering methods largely rely on the correctness of a probe set corresponding to a gene. Kasim *et al.* (2010) have shown that this assumption can be violated due to a biological phenomenon known as differential splicing. With more advanced knowledge of a genome annotation for various organisms, the statically defined Affymetrix probe sets can be updated, forming alternative probe sets of a variable size, ranging from 3 up to more than a hundred probes per probe set. The alternative way of refining probe sets annotation is to use experimental data. Kasim *et al.* (2010) proposed a mixture model to detect an alternative grouping within a probe set. The third research goal in this part of the dissertation, modeling and decomposition of the variability sources in the data, is presented in Chapter 6 in which we illustrate how the linear mixed-effects model can be used in order to improve the I/NI calls and in particular how the mixed model is formulated in order to find a subset of the probes within a probe set, which we term *the informative core of the probe set* that capture the transcript or its part of a given gene. Furthermore, in Chapter 3 we illustrate how patterns in the data which are not expected to appear in a specific experiment can be detected using the extended mixed model.

2.3 Summarization methods

In this section we describe two most widely applied summarization methods of the Affymetrix microarray data.

2.3.1 Factor analysis for robust microarray summarization (FARMS)

FARMS was introduced by Hochreiter *et al.* (2006). To obtain a summary measure for a probe set, the factor analysis model (2.1) is used. Since the actual gene expression is not observed, it is modeled by means of a latent factor. Each probe in a probe set is related to the latent factor by means of factor loadings. The FARMS model is formulated as

$$Y_{ij} = \lambda_j \cdot Z_i + \varepsilon_{ij}, \tag{2.1}$$

where Y_{ij} is a logarithm of the perfect match (PM) expression measure of probe j on array i, λ_j is the factor loading of probe j, Z_i is a latent factor, and ε_{ij} is the residual error.

The summarized value is obtained by averaging all the factor loadings λ_j and taking a scalar product of average or median of factor loadings and factor scores per gene.

2.3.2 Robust microarray summarization (RMA)

RMA is the most widely used summarization method, introduced by Irizarry *et al.* (2003). The summarized values for each gene are obtained from the background-adjusted, normalized, and log-transformed PM intensities (Y_{ij}) which are expected to follow an additive model given by

$$Y_{ij} = \mu_i + \alpha_j + \varepsilon_{ij}. \tag{2.2}$$

Here, α_j denotes a probe effect, μ_i is the effect for array i, and ε_{ij} is the residual error. The estimation procedure is robust to outlying probes as the unknown parameters are estimated by means by means of the median polish method.

2.4 Case studies for probe-level analysis

Several datasets are used in order to illustrate the methodology presented in this part of the dissertation.

2.4.1 Affymetrix benchmark datasets

A popularity of microarray experiments raised a need for creating the benchmark datasets, i.e., golden standards to check the quality of the platform as well as to validate the results of analytic tools. The importance of benchmark datasets cannot be overestimated since they are used to select the optimal way of preprocessing and analyzing particular data types (Cope *et al.*, 2004; Sasidharan Nair and Vihinen, 2012). However, due to the limited sensitivity of the Affymetrix microarrays or probe sequences' design, the datasets are not always reflecting the original experimental design. For instance, the Affycomp benchmark data had originally 42 spiked-in transcripts, but due to cross-hybridization 22 additional probe sets have captured these transcripts, resulting in 64 probe sets measuring spiked-in genes (Irizarry *et al.*, 2004). In the experiment mentioned in Cope *et al.* (2004), two extra probe sets were identified to be measuring the spiked-in transcripts. Inclusion of the extra probe sets to the original set of probe sets targeting spiked-in transcripts improved the performance of many predictive methods, including sensitivity and specificity (McGee and Chen, 2006).

Lately, a new benchmark dataset called Platinum Data has been introduced (Zhu *et al.*, 2010). The dataset was constructed in the way that would mimic a microarray experiment. Compared to the previous spiked-in datasets, there was a larger number of known transcripts, spiked at a range of different concentrations. The difference between two groups ranged from no to 4-fold difference, including up- and downregulated transcripts. Platinum spiked-in dataset is a very promising benchmark dataset since it has a balanced number of down-regulated and up-regulated transcripts as well as transcripts spiked in at the same concentration, mimicking the real-world gene expression data. Recently, it has been used to verify several filtering methods as well as the identification of differentially expressed genes (Stevens and Nicholas, 2012; Subramaniam and Hsiao, 2012; Lu *et al.*, 2011). Lu *et al.* (2011) showed that identification of spiked-in transcripts decreased with the difference in log-fold changes. In addition, Subramaniam and Hsiao (2012) showed that the identification of differentially expressed transcripts depended on the fold changes at which they were spiked in.

2.4.1.1 The Platinum spike-in dataset

The data consists of 18 Affymetrix Drosophila Genome 2.0 microarrays (Zhu *et al.*, 2010), divided into two groups (A and B). There are $13,777$ empty probe sets in the dataset which are not supposed to capture any spiked-in transcripts. The $2,189$ RNAs are spiked in so that the number of up- and down-regulated transcripts between the two conditions is balanced. Moreover, $3,426$ transcripts were spiked in at the same concentrations for both conditions. In each group, there were three samples synthesized and for each sample three technical replicates were used for the microarray analysis. The study design has two levels of clustering: the arrays nested within samples. The authors suggested that all arrays have to be treated as independent units. The independence can be checked by estimating the strength of correlation within each biological sample. We assume that the data have been background-corrected and normalized according to Irizarry *et al.* (2003). In this dataset, Affymetrix CDF

files for probe set definition were used.
This dataset will be used in Chapter 4 in order to illustrate the extended linear mixed-effects model and its application as a quality control tool on the probe-level data. Furthermore, it will be also used in Chapter 6 to demonstrate how annotation of probe sets can be improved based on extended linear mixed-effects model approach.

2.4.1.2 Affycomp spiked-in dataset (Human Genome U133)

The spiked-in dataset was created by Affymetrix company for studying performance of various statistical methods for differential expression (Irizarry *et al.*, 2004). Data were collected from 42 microarrays. Fourteen groups, each composed of three arrays, measured 14 concentrations of 42 spiked transcripts ranging from 0.125 pM to 512 pM. Thirty of the spikes are isolated from a human cell line, four spikes are bacterial controls, and eight spikes are artificially engineered sequences believed to be unique in the human genome. Each microarray contains $22,300$ probe sets with $22,158$ probe sets measuring background RNA mixture. After the dataset was published, additional 22 probe sets were detected as capturing spiked-in transcripts, resulting in 64 spiked probe sets. For the analysis, original Affymetrix CDF files were used.
This dataset will be used in Chapter 3 in order to illustrate the extended linear mixed-effects model and the deconvolution of a probe-level signal into various sources of variation.

2.4.2 Gene expression studies

2.4.2.1 Breast cancer data with known batch effect

The Breast cancer dataset (Pawitan *et al.*, 2005) contains gene expression measurements of 159 patients. Before the start of the project, it was known that adjuvant breast cancer therapy significantly improves survival of the patients, but overtreatment and undertreatment were major problems. The project aimed at using gene expression profiling to identify patients whose tumors had a low malignant potential, making adjuvant therapy unnecessary and potentially harmful, and to identify patients in need of more effective adjuvant therapies.
The gene expression was profiled by Affymetrix HGU133a arrays. The arrays have been processed in several batches in a period longer than one year. Similar to the work of McCall *et al.* (2009), the arrays were grouped into six batches based on the dates of generation. The data are publicly available at the NCBI GEO database, accession GSE1456 (Pawitan *et al.*, 2005). The original Affymetrix CDF files were

used to obtain probe-level data. This dataset will be used in Chapter 3 in order to illustrate how the extended linear mixed-effects model can detect probe sets with batch effects.

2.4.2.2 Sialin knock-out experiment

This dataset has been introduced in the work of Raghavan *et al.* (2007). Defects in the metabolism of sialic acid are known to be responsible for the so-called sialic acid storage diseases. These are autosomal recessive neurodegenerative disorders that may present as a severe infantile form (ISSD or infantile sialic acid storage disease) or a slowly progressive adult form (Salla disease). Both forms of sialic acid storage disease are caused by mutations in Slc17A5, which encodes the protein sialin. To perform the experiment, RNA samples from total brain were derived from newborn and 18-day-old mice for each of two groups: Slc17A5 knockout ('KO') and wild type ('WT'). There were six biological samples in each group. Microarray experiments were performed on the RNA samples using Affymetrix Mouse430_2 GeneChips. For the current analysis, alternative CDF ENTREZ annotation v.15 was used resulting in $17,370$ re-defined probe sets.

This dataset will be used in Chapter 3 in order to illustrate how the extended linear mixed-effects model can detect probe sets with treatment effect and how the presence of a grouping factor influences the I/NI filtering. In addition, it will be also used in Chapter 6 to demonstrate how annotation of probe sets can be improved.

Chapter 3

Model based filtering and analysis of probe-level data

This chapter focuses on a number of aspects of probe-level filtering and analysis. Section 3.1 gives a description of linear mixed-effects modeling framework and incorporation of various fixed and random effects according to the experimental design. Section 3.2 shows results after applying probe-level modeling to the case studies, including batch-effect correction at the probe level.

3.1 Linear mixed-effects model for probe-level filtering

3.1.1 Basic model for probe-level filtering

The basic filtering linear mixed-effects model (LMM) for informative and non-informative calls (I/NI calls) is described in Kasim *et al.* (2010). Similar to FLUSH, FARMS-based and PCA-based filtering models, the LMM is a probe set-specific model. Let Y_{ij} be the intensity measurement of jth probe on ith array, then

$$Y_{ij} = \mu_j + b_i + \varepsilon_{ij},\ i = 1, \ldots, n,\ j = 1, \ldots, l, \tag{3.1}$$

where $b_i \sim N(0, \sigma_b^2)$ is an array-specific random effect, and $\varepsilon_{ij} \sim N(0, \sigma_\varepsilon^2)$ are residual errors. Parameter estimates can be obtained by maximum likelihood or restricted maximum likelihood (Pinheiro and Bates, 2000). As pointed out by Kasim *et al.*

(2010), the parameter estimates for σ_b^2 and σ_ε^2 can be used to infer about the strength of array effect and compare it to the residual variance. The measure that can be used in this case is the *intra-cluster correlation* (ICC), $\rho_1 = \sigma_b^2/(\sigma_b^2 + \sigma_\varepsilon^2)$, (Verbeke and Molenberghs, 2002). A probe set is declared informative if $\rho_1 > \tau$ where τ is a pre-specified threshold for filtering (Talloen *et al.*, 2007).

The current setting assumes that any relevant variation or signal is captured by array-to-array variability. It does not, however, exploit the information given by experimental design and does not indicate which is the major source of variation. To incorporate the additional sources of information the existing model needs to be extended.

3.1.1.1 LMM with adjustment for experimental factors

Consider the experimental design with several conditions such as treatment, dose, time etc. The linear mixed-effects model specified in (3.1) can be re-formulated for Y_{ij} as

$$Y_{ij} = \mu_j + \beta_1 X_{1ij} + \ldots + \beta_p X_{pij} + b_i + \varepsilon_{ij}, i = 1, \ldots, n, j = 1, \ldots, l, \tag{3.2}$$

where $\beta_s, s = 1, \ldots, p$ is a vector of fixed-effects coefficients, $\boldsymbol{X}_{sij}$ is a $1 \times p$ vector of regressors, $b_i \sim N(0, \sigma_b^2)$ is array-specific random effect, and $\varepsilon_{ij} \sim N(0, \sigma_\varepsilon^2)$ are residual errors. In this case, the array-to-array variability is used to capture the residual correlation of the probes within a probe set after correcting for the variability induced by the design variables. If the array-to-array variability remains significant after adjustment for the fixed effects, there is an extra unexplained source of variability in the data. Hence, the adjusted array-to-array variability compared to basic array-to-array variability indicates the presence of extra effect, while the latter gives the indication of presence of any effect, either explainable or unexplainable. Therefore, the decomposition of total variation by the model (3.2) into fixed and random sources would stratify probe sets into four categories, summarized in Table 3.1. Type 1 probe sets do not contain any biological signal, since the covariate effect is not present and the array-to-array variability caused by some other factor is not significant. This type of probe sets would be considered non-informative by model (3.1). The other three types would be called informative by the basic model, since the unadjusted array-to-array variability is high. The probe sets of Type 2 do not have a significant covariate effect, however, the non-zero adjusted array-to-array variability indicates the presence of an unexplained factor. Type 3 probe sets would have a significant covariate effect and no extra variability, carrying information solely about the experimental factors. Type 4 probe sets have a significant variability due to the covariate effect(s) and have indication that there is an additional variability source, which is not explained by the original experimental design.

Table 3.1: Four major types of probe sets related to the significance of fixed and random effects.

		Random effect of array	
		Non-significant (NS)	Significant (S)
Design variable	NS	Type 1	Type 2
	S	Type 3	Type 4

Schematic scatterplots for two probes in probe sets of different types are shown in Figure 3.1.

Applying the probe-level model (3.2) to the microarray experiment, one can discard truly non-informative probe sets (Type 1) from further analysis. For the quality control of the experiment, an essential number of Type 2 and Type 4 probe sets would indicate that some important factors have not been considered by the design or technical artifacts are present.

3.1.1.2 A LMM with nested designs

Let us consider two levels of experimental design in which arrays are processed in several batches. In this setting, arrays are likely to be more similar within one batch and this needs to be taken into account by the model. Similarly, considering a number of technical replicates for one sample would imply extra level of correlation between the arrays coming from the same sample. To capture this source of variation we extend the LMM model (3.2) by adding a nested random effect for each array:

$$Y_{ijk} = \mu_j + \beta_1 X_{1ij} + \ldots + \beta_p X_{pij} + b_i + u_{(i)k} + \varepsilon_{ijk},\ i = 1 \ldots n, j = 1 \ldots l, k = 1 \ldots N_b, \tag{3.3}$$

where $b_i \sim N(0, \sigma_b^2)$ are level-1 random effects (arrays), $u_{(i)k} \sim N(0, \sigma_u^2)$ are level-2 random effects (batches or technical replicates), and $\varepsilon_{ijk} \sim N(, \sigma_\varepsilon^2)$ are residual errors. It is assumed that level-2 random effects are independent of level-1 random effects and of residual errors. The definition of the ICC in this model is $\rho_2 = \sigma_b^2/(\sigma_b^2 + \sigma_u^2 + \sigma_\varepsilon^2)$. The ICC ρ_2 indicates whether probes are correlated after adjusting for fixed effects and whether extra level of clustering is present. Additional ICC can be calculated to measure correlation of arrays within batches: $\rho_3 = \sigma_u^2/(\sigma_b^2 + \sigma_u^2 + \sigma_\varepsilon^2)$.

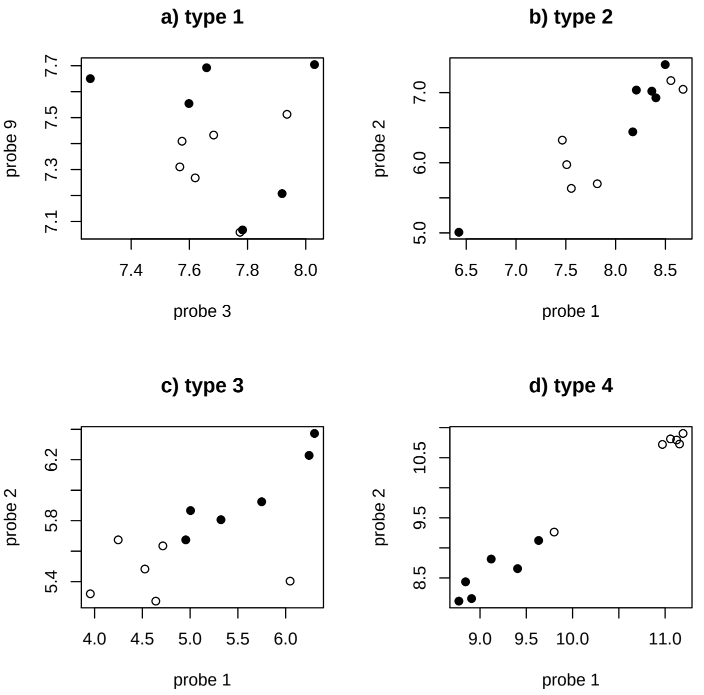


Figure 3.1: Two selected probes from probe sets of four different types according to the stratification in terminology of the LMM. Illustration is based on the probe sets from the Sialin dataset, where black circles correspond to the 'WT' group and the open circles correspond to the 'KO' group.

To obtain stratification for the probe sets in a multi-level setting, the information criteria AIC or BIC and the likelihood ratio tests can be used. Note that the stratification of probe sets in Table 3.1 has a different interpretation. For example, while the most treatment-specific probe sets would be of Type 3 in the treatment effect setting described in previous section, the batch affected probe sets of type 3 or 4 would require the batch effect correction before drawing conclusions about biological effects. Type 2 probe sets would be the most informative in an experiment where batch effects are present.

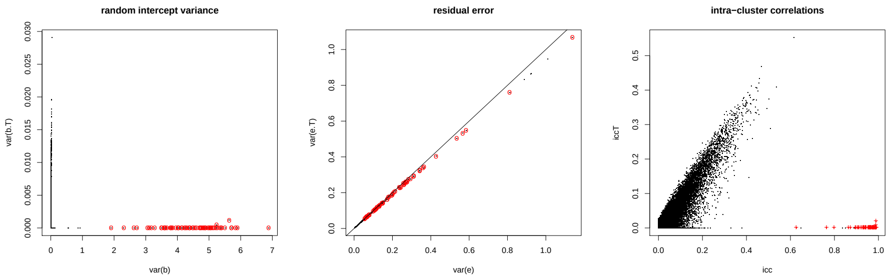


Figure 3.2: Affycomp dataset. Parameter estimates for the variance components and the intra-cluster correlations obtained from model (3.1) and (3.2). The values for probe sets capturing the spiked-in transcripts are in red.

3.2 Results

3.2.1 Analysis of `Affycomp` spiked-in data

The Affycomp dataset has experimental grouping as a covariate of interest. Therefore, we use one level model with the spiked group as a design variable and the model (3.2) takes the following form:

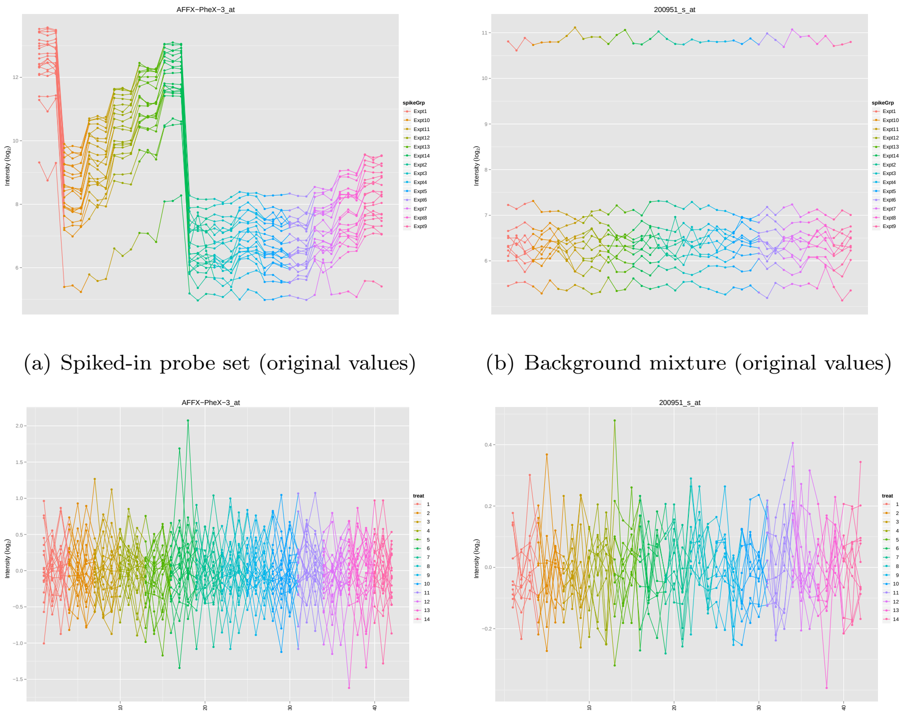
$$Y_{ijg} = \mu_j + \alpha_g + b_i + \varepsilon_{ijg},\ i = 1\dots 3,\ j = 1\dots l,\ g = 1\dots 14, \qquad (3.4)$$

where μ_j is probe specific mean, α_g is a fixed effect of a concentration group g, b_i is a random effect of the ith array and ε_{ijg} is a normally distributed residual error term. Figure 3.2 shows the variance components and intra-cluster correlations estimated with and without an adjustment for the covariate effect. The residual variance remains unchanged, which means that the array-to-array variability absorbs the treatment effect. Adjustment for the treatment effect results, as expected, in the decreased ICC values. All spiked-in probe sets (in red) have high ICC in the basic model due to the difference between concentration groups.

In the Affycomp spiked-in data, two types of probe sets are observed. The background mixtures are of the Type 1 and can be called non-informative and the spiked-in genes are informative with respect to the covariate (grouping of arrays according to the concentrations). Figures 3.3(a),(b) present an example of a spiked-in probe set (Type 2) and a background mixture probe set (Type 1). We note that for the spiked-in probe set the correlation between probes is due to the differences in concentrations. Adjusting for this difference and removing the concentration effect results in the data similar to the background mixture, where no extra array-to-array variation is observed

(Figure 3.3 (c) and (d)).

The stratification of all probe sets in the two types (type 1 and type 2) shows that



(a) Spiked-in probe set (original values)

(b) Background mixture (original values)

(c) Spiked-in probe set (adjusted values)

(d) Background mixture (adjusted values)

Figure 3.3: Affycomp dataset. Probe profiles: 3.3(a), 3.3(c): probe set for the spiked in transcript (the concentrations are grouped by experiment number), 3.3(b), 3.3(d) probe sets corresponding to a background gene. 3.3(a), 3.3(b): original values, 3.3(c), 3.3(d) the treatment and probe effect corrected values.

the Affycomp dataset comes from fully controlled experiment with known sources of variation, which might not be the case in the real life datasets.

3.2.2 Analysis of the Sialin data

The covariate of interest in the Sialin data is the genotype of mice (KO or WT), which can be directly incorporated in the model. Thus, the basic model (3.1) can be extended as follows:

$$Y_{ijk} = \mu_j + \alpha_k + b_i + \varepsilon_{ijk}, i = 1 \ldots 6, j = 1 \ldots l, \tag{3.5}$$

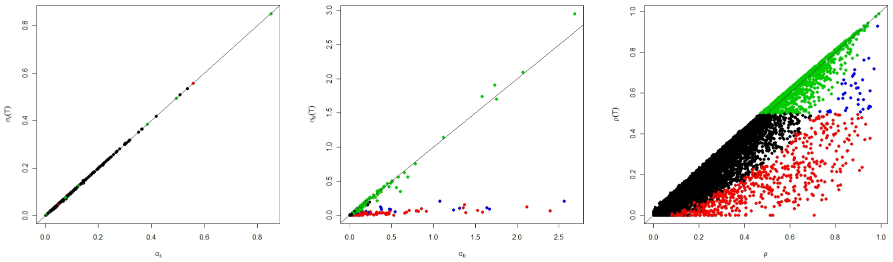


Figure 3.4: Sialin data. The changes in the variance components and intra-cluster variation based on the estimates from the model with adjustment for the treatment (LMM-2) and without (LMM-1). Color corresponds to the Type of probe set: Type 1 - black, Type 2 - red, Type 3 - green and Type 4 - blue.

where α_k is the fixed effect of a group $k = 1, 2$. The specification of random effects remains the same as in the model (3.4).

Figure 3.4 displays the changes in variability for model with and without adjusting for the covariate. It can readily be observed that in the presence of the covariate in the model the residual variance is similar to the residual variance from the model without adjusting for the covariate (with a slight decrease for some probe sets). The variances of random intercepts (array effects) on the other hand, for most probe sets decrease, affecting the values of intra-cluster correlations, which decrease as well.

The stratification of probe sets is presented in Table 3.2. Six-hundred and ten probe sets of Type 3 would be of key interest for the research since these are the condition-specific probe sets. Fifty-seven probe sets of Type 4, as well as, $1,017$ probe sets of Type 2 indicate that there is an extra latent source of variation. In an experiment without any extra sources of variation, Type 2 and 4 probe sets would less likely be present. Type 1 and Type 2 are of less interest since they do not carry information related to the condition of interest.

The plots of 4 different types of probe sets are displayed in Figure 3.5.

3.2.3 Analysis of data with batch effects from the GSE1456 experiment

The presence of batch effects is common in high throughput experiments due to the different processing times, lab protocols, or technicians carrying out the experiments. However, the batch effects are often overlooked in the analysis (Leek *et al.*, 2010). Batch effects can be even more pronounced when they confound the effects of main

Table 3.2: Types of probe sets according to the significance of difference between KO and WT group and the significance of σ_b^2 in Sialin dataset. ”S” denotes significant effect, ”NS” denotes non-significant effect.

		σ_b^2	
		NS	S
WT vs. KO	NS	15685	1017
	S	610	57

experimental factors, such as treated vs. non-treated or affected vs. healthy patients. Therefore, batch-effect correction should be addressed in the analysis, either on the probe level or during the summarization process.

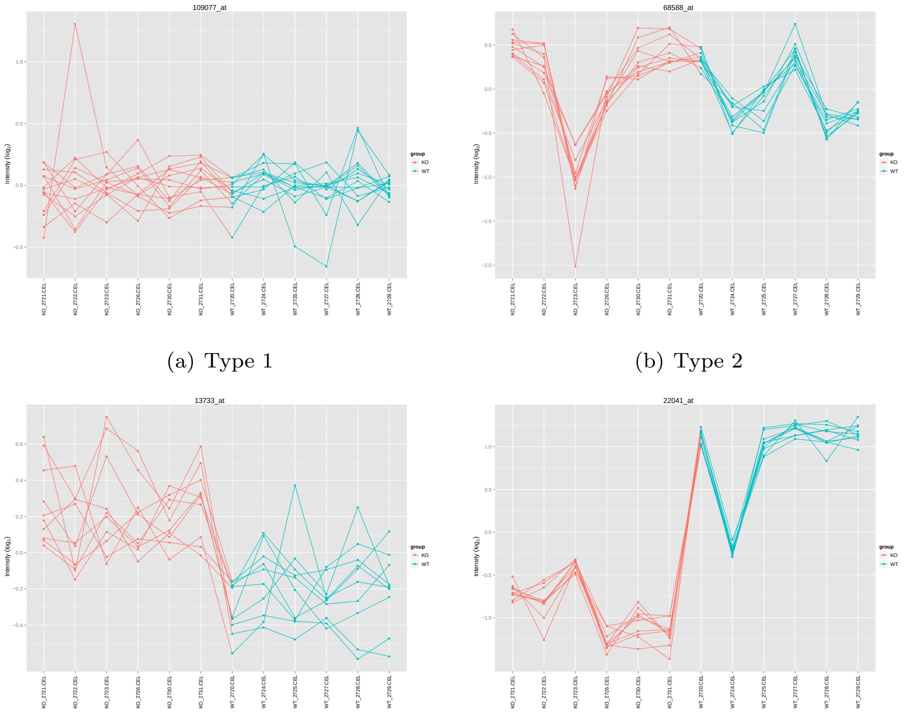
McCall *et al.* (2010) have developed a method called frozen RMA (fRMA) to address batch-effect correction prior to summarization and to downweight probes in a probe set affected by batches. In essence, the method of McCall *et al.* (2010) extends RMA model (2.2) by adding a random effect of probe-batch interaction to the model. The estimation of the array effect, probe effect, and probe-batch interaction is carried out by the weighted least squares procedure, and the variances for random effects are estimated based on the large pool of data sets analyzed by the same platform.

Alternatively to the fRMA, linear mixed-effects model, described in Section 3.1.1.2, can be used to take into account batch effects and correct for them at the probe level. Let us consider the GSE1456 dataset, which was reported by McCall *et al.* (2010) to have six batches of processed arrays. We model probe-level expression by a linear mixed-effects model given by

$$Y_{ijk} = \mu_j + b_i + u_{(i)k} + \varepsilon_{ijk},\ i = 1 \dots N_k,\ j = 1, \dots, l,\ k = 1, \dots, 6. \tag{3.6}$$

Here, k is the index for batch and N_k is a number of samples per batch k ($N_1 = 19$, $N_2 = 40$, $N_3 = 19$, $N_4 = 18$, $N_5 = 37$, $N_6 = 26$). Unlike the fRMA, the LMM model (3.6) models $u_{(i)k}$, the nested batch-array random effect, and the probe-specific intensity μ_j is a fixed effect. The estimation of variance components and fixed effects is carried out by REML procedure. The model allows us to test for the significance of batch effect using likelihood ratio test and to check whether there remains any residual array-to-array variability after the correction for batch effects.

In the GSE1456 experiment, the number of probe sets affected by batch effects is not known upfront. Thus, we will stratify the probe sets according to the presence of



(a) Type 1

(b) Type 2

(c) Type 3

(d) Type 4

Figure 3.5: The probe expression profiles for 4 different types of probe sets in the Sialin Data: (a) a probe set of type 1 with no significant KO vs. WT difference, no significant array-to-array variability; (b) a probe set of type 2 with significant array-to-array variability; (c) a probe set of type 3 with significant KO vs. WT difference; (d) a probe set with significant array-to-array variability and KO vs. WT difference.

batch effects and the biological effects. Afterwards we will select the probe sets which carry information related to the biological effects. In this case, Type 1 and Type 3 probe sets (non-informative) have variation due to the batch effects, and Type 2 and Type 4 (informative) have extra biological variability that makes them focus of further analysis.

Figure 3.6 shows the parameter estimates for the variance components. It reveals that there are probe sets that have relatively high intra-cluster correlation without and with adjustment for batch effects. A second variability pattern that can be observed in Figure 3.6 are probe sets for which the intra-cluster correlation decreases after the adjustment for the batch effects. This type of probe sets would be wrongly

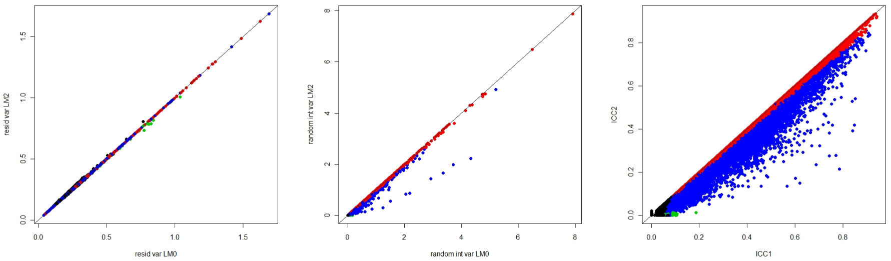


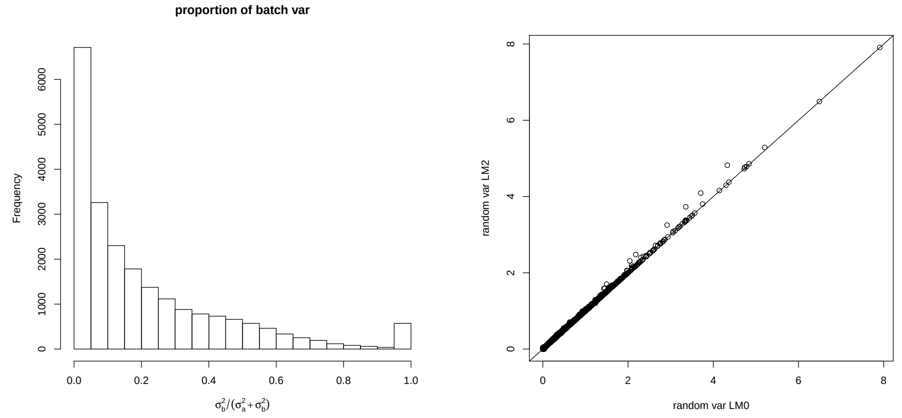
Figure 3.6: The GSE1456 experiment. The changes in the variance components and intra-cluster variation based on the estimates from the model with adjustment for the batch effect (y-axis) and without (x-axis). Color corresponds to the Type of probe set: Type 1 - black, Type 2 - red, Type 3 - green and Type 4 - blue.

called informative by the INI-calls procedure, while array-to-array variability in these probe sets is caused by the batch effects. Figure 3.7(a) shows the proportion of array-to-array variation estimated by σ_b from the model with two-level clustering in the total variability (i.e., $\sigma_b + \sigma_u + \sigma_\varepsilon$). The batch effects are less prominent than array-to-array variability for a large number of probe sets. Figure 3.7(b) shows the parameter estimates for the variance components obtained from the two models. The variability of residual error remains almost unchanged after adjusting for the batch effects, which suggests that the 1-level array-to-array variability absorbs the batch-to-batch variability.

The four types of probe sets stratified by the AIC are shown in Figure 3.8: the probe sets of Type 4 are probes with both levels of clustering (Figure 3.8(d), the probe sets of Type 1 do not have biological or variability between batches (Figure 3.8(a)), the probe sets of Type 3 have array effects (Figure 3.8(c)) and the probe sets of Type 2 have batch effects (Figure 3.8(b)).

The distribution of probe sets among the four types is summarized in Table 3.3. In Table 3.3 the stratification is performed according to the AIC, i.e., for the Type 1 probe sets the AIC is the lowest for the basic LMM, whereas for the Type 4 probe sets the AIC is the lowest for the model with two levels of random effects.

Figure 3.9(a) shows histograms of ICC for batches in the data. Each panel shows a subset of values for the model preferred by the AIC. The ICCs for probe sets supporting the basic model are in general low. The probe sets supporting the model with random effect of array only have relatively low values of ICC, even though they are slightly higher than in the basic model. The large values of ICC are observed in



(a)

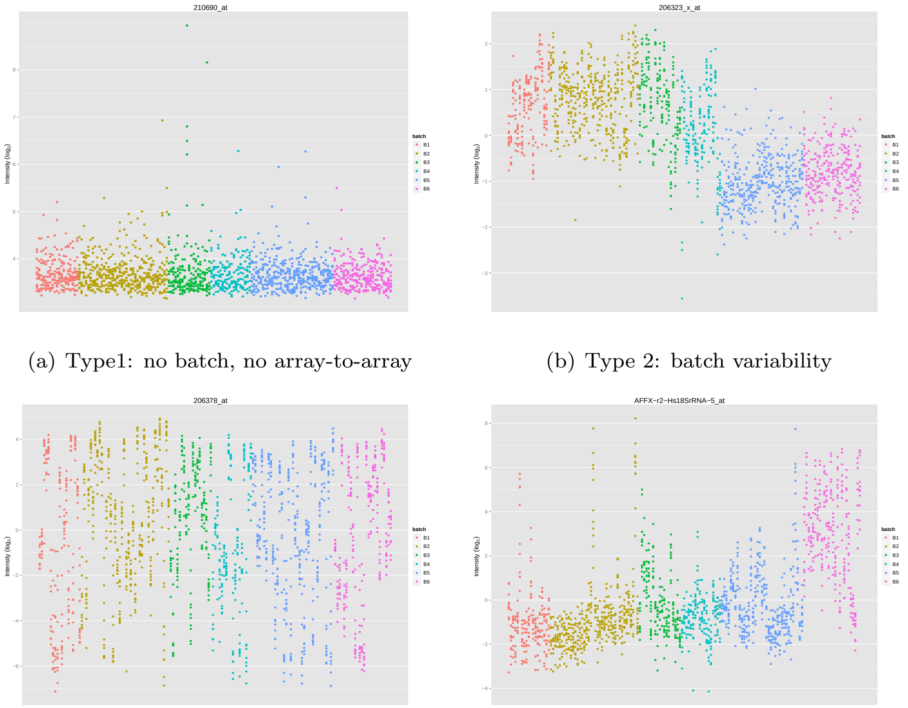
(b)

Figure 3.7: The variance of random effects: (a) the histogram of proportion of batch effect in the total random variation; (b) changes in variance of residuals in the basic model after adjustment for batch effect.

Table 3.3: Distribution of types of probe sets in GSE1456 dataset. The AIC was used for model selection.

		Batch effects	
		No	Yes
Array	No	3299	125
effect	Yes	6804	12055

the probe sets with two levels of clustering. Similarly to the estimation of the batch effects, we investigate the distribution of the ICCs for arrays from the model without batch effect in the data to see how specific is the AIC in this setting (Figure 3.9(b)). The enrichment of both within-array and within-batch correlations is observed in models favored by lower values of the AIC.



(a) Type1: no batch, no array-to-array

(b) Type 2: batch variability

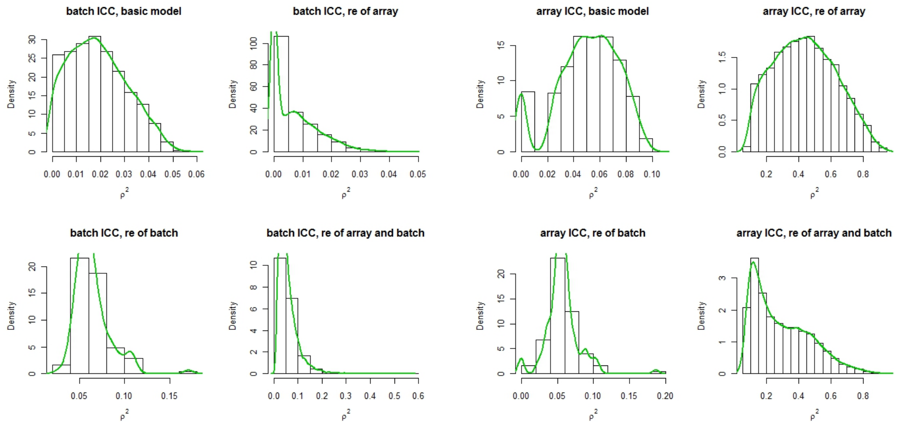
(c) Type3: array variability

(d) Type4: both array and batch variability

Figure 3.8: The probe profile plots for 4 different types of probe sets in GSE1456 data.

3.3 Discussion

The analysis of the sources of variation in the probe-level data coming from Affymetrix microarray experiments allows researcher to extract more information from the data. The array-to-array variability is one of the most important sources, which can be used to check the quality of a given probe set on an array. In previous studies (Kasim *et al.*, 2010; Talloen *et al.*, 2007) the ratio between array-to-array variability and the measurement error was used as a filtering criterion. Both the FARMS-based and the LMM-based I/NI calls suggested that if the set of probes measures the true intensity of a transcript then the ICC for probes within a probe set is expected to be high or, alternatively, the conditional variance of the data given the latent factor is low. In the work of Kasim *et al.* (2010) the ICC was used as one of the ways to quantify the proportion of between-array variation in the total variation in the data. Lu *et al.* (2011) have used the proportion of variation explained by the first principal



(a) The intra-batch correlation (b) The intra-array correlation

Figure 3.9: The distribution of the ICCs for batches and arrays across the four different types of probe sets in the GSE1456 dataset.

component (PC1) and selected those probe sets which have relatively high amount of variability explained by the PC1.

In the aforementioned methods other sources of variation were ignored. Both the variation induced by difference in factor levels or phenotypic variables and the variation induced by a latent factor contribute to the overall array-to-array variability. The basic filtering approaches cannot disentangle these sources. However, mixed-effects model framework discussed in this chapter can use the richness of the probe-level data to the fullest and deconvolute the signal into the signal induced by the design or phenotypic covariates from the signal due to unobserved or unmeasured variables. Therefore, we have investigated the effect of design variables and technical covariates (batch effects) on the signal, captured by array-to-array variability. In essence, we have added the design covariates as a part of the model and studied the changes in the sources of variation. The changes in the variation sources have implications for filtering criteria, such as the ICCs, since ICCs are affected by the changes in the array-to-array variability and the residual variance. In the case of fully controlled experiment we would expect that the array-to-array variability accounts for the difference between different levels of factor. Once the difference between the levels of a design variable is adjusted for in the probe-level model, the array-to-array variability will decrease. Thus, the adjusted array-to-array variability reflects the presence of an unobserved factor in the data, which can correspond to a biological effect or a technical artefact. The adjustment of the array-to-array variability can be used for the quality control of the experiment.

For the Affycomp dataset it has been shown that the adjusted array-to-array variability was low for the spiked-in transcripts. It implies that the Affycomp experiment is a fully controlled experiment without any extra unobserved factors and the signal is caused by the differences in transcript abundance between spiked groups. For this dataset both filtering methods give good results: the latent variable approach as well as the extended linear mixed-effects model would call probe sets measuring spiked-in transcripts informative.

In real-life experiments the situation is more complex, though. In the Sialin knock-out experiment, the decrease in the ICCs was not observed for a number of probe sets due to the high adjusted array-to-array variability. In fact, the investigation of the sources of variation allowed us to detect four different patterns in the probe-level data:

1. Type 1 probe sets that had neither array-to-array variability nor knock-out effect and could be compared to the background mixtures of the spiked-in experiment. These probe sets had low ICCs in both models, with or without adjustment for the group difference;

2. Type 2 probe sets that had array-to-array variability but no difference between the wild type and knock-out groups. These probe sets had high ICCs in both models, since the adjustment for the difference between the groups did not influence the array-to-array variability;

3. Type 3 probe sets that had a significant difference between the two groups and no extra array-to-array variability. These probe sets had high ICCs in the model without treatment and low ICC after the adjustment for the treatment. This is a typical example of informative probe sets and can be compared to the spiked-in genes, which carried information about the difference between the experimental conditions;

4. Type 4 probe sets that had difference between the groups and extra array-to-array variability. This type of probe sets had high ICCs in both models, however, the values of the ICCs decreased after adjustment for the difference in expression between groups. This type of probe sets is not specific for the experiment, since they also carry information about some uncontrolled factor.

Except for the design variables, data may be prone to batch effects. This information can be incorporated at the probe-level analysis of the data as well. We have illustrated this type of analysis on the example of the Breast Cancer study. The batch effects were noticed by McCall *et al.* (2010) due to the gaps in the time of microarrays processing. This information was used to construct the two-level linear mixed-effects model and to check how the batch effects influence probe-level filtering. After assessing the presence of the array-to-array variability and the batch-to-batch variability, similarly to the situation with the knock-out experiment, four types of probe sets were identified:

1. Probe sets which had low adjusted and unadjusted array-to-array variability and had low ICCs in both 1-level and 2-level models. These are so-called 'truly non-informative' probe sets and would be filtered out in a justified manner by the I/NI calls.

2. Probe sets with array-to-array variability caused by the difference in expression levels across the batches. These probe sets had high ICCs in the 1-level model, which decreased in the 2-level model. These are the probe sets that are called informative by the basic model, being informative due to the batch effects only.

3. Probe sets which did not show the difference between batches, but had a substantial adjusted array-to-array variability. The ICCs were high in both models and this types of probe sets can be the most interesting for further investigation.

4. Probe sets that had both array-to-array variability and the batch effects. These probe sets had high ICCs in both models, with a decrease in the 2-level model. This type of probe sets can be used in further analysis, however, the expression levels should be adjusted by subtracting estimated batch effects from the original intensities.

For the Breast cancer data, the filtering based on the array-to-array variability has been shown to be inappropriate in a number of cases when variation of probe intensities is due to the differences in batches. Once these differences were accounted for by taking the second level of nesting in the model, the biological variability between arrays decreased. In addition, there are probes which have both batch effect and array-to-array variability. In this case, the batch effect can be corrected for and the probe sets can be used for further analysis.

It should be mentioned that two different approaches to the filtering should be considered. When data can be modeled by one-level LMM, Type 1 probe sets are truly non-informative and the remaining types can be considered for further investigation of fixed and hidden factors. When two-level LMM is applied, Type 1 and Type 3 probe sets are non-informative and Type 2 and Type 4 should be considered for further research by biologists.

The analysis presented in this chapter was focused on model based filtering procedures and several case studies were used to illustrate the use of a LMM as a filtering model. In the next chapter we will discuss the use of the LMM for quality control of a benchmark data set and illustrate how the model can be used in order to identify unexpected or unaccounted effects in the data.

Chapter 4

Quality control of the Platinum Spike dataset by probe-level mixed-effects models

This chapter presents the probe-level analysis of the Platinum Spike dataset and illustrates how the parameter estimates obtained from the LMM can be used for the quality control of the probe-level data. In Section 4.1, a LMM for the experimental set-up of the Platinum spike data is formulated and the tests for significance of variance components are explained in detail. Section 4.2 presents the results of the LMM-based quality control procedure. Graphical displays are used to show probe sets which deviate from the experimental setup. The chapter is concluded by discussion in Section 4.3.

4.1 Probe-level linear mixed-effects models for the Platinum Spike dataset

The probe-level linear mixed-effects models discussed in this section are designed to capture and to quantify different sources of variability in the Platinum Spike dataset. Our starting point, in Section 4.1.1, is a single-level linear mixed-effects model. There are two sources of variability, array-to-array variability and the measurement error, in the model; we ignore any group difference. We use the ratio between the array-to-array variability to the measurement error in order to identify the dominant source of variability for each probe set.

4.1.1 LMM with one-level random effects

For the first stage of the quality control, we fit a basic probe-level linear mixed-effects model (3.1) and afterwards extend it to the model M_1, adjusting for the difference between two experimental groups. From the model M_1, the adjusted ICCs are computed. Comparing unadjusted and adjusted ICCs for the probe sets capturing spiked-in transcripts, we expect to see a large decrease in the values, implying, that, in fact, the ICCs are influenced by the differences in concentrations of spiked-in transipts. The probe sets, which have relatively large unadjusted ICCs compared to the adjusted ICCs, are considered to be the best fit to the experiment design, i.e., to carry the difference in concentrations between the two experimental groups. Our quality control tool highlights the probe sets in each experimental category, which have low adjusted ICCs, and checks the probe-level expression patterns.
Let α_s be the spiked-group specific effect. The linear mixed-effects model (3.1) can then be re-formulated as

$$Y_{ijs} = \mu_j + \alpha_s + b_{1i} + \varepsilon_{ijs},\ i = 1, \ldots, 9,\ j = 1, \ldots, p,\ s = 1, 2, \tag{4.1}$$

where $b_{1i} \sim N(0, \sigma^2_{b_1})$, and $\varepsilon_{ijs} \sim N(0, \sigma^2_\varepsilon)$. In this case, the array-to-array variability is used to capture the residual correlation of the probes within a probe set after correcting for the variability induced by the experimental groups (A or B). This variability is summarized by the adjusted ICC, $\rho_1 = \sigma^2_{b_1}/(\sigma^2_{b_1} + \sigma^2_\varepsilon)$. Note that the residual variability σ^2_ε is considered to be unchanged after the adjustment for the group difference. Hence, it is assumed that that the variability due to the missing covariates in (3.1) is captured by the variance component associated with the array-to-array variability σ^2_b, which is confirmed by the visual inspection of the changes in variances. Figure 4.1 shows that the variances of residual errors remain unchanged, whereas the array-to-arrays variance decreases after the group difference was taken into account by the model.

In the case of the Platinum Spike dataset, the empty probe sets, which do not have target transcripts in the samples, should neither have difference in expression levels, nor high adjusted ICC values. The probe sets corresponding to the transcripts spiked in at different concentrations would have significant differences in the expression values between the two groups and low adjusted ICCs. The probe sets measuring the

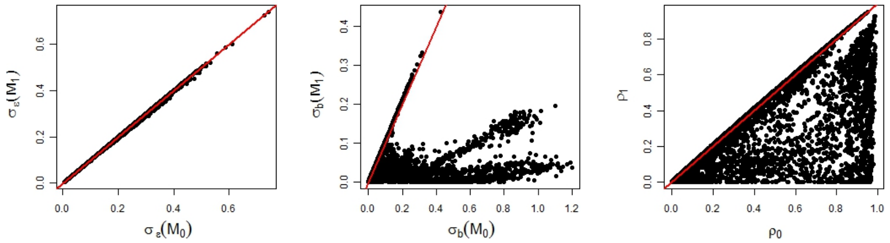


Figure 4.1: Platinum dataset. The changes in the variance components and intra-cluster variation based on the estimates from the model with adjustment for the group effect (y-axis) and without (x-axis).

transcripts spiked in at the same concentrations are to have no difference in gene expression levels but possibly high ICCs explained by the consistency of measurements across biological samples. Since the experiment is fully controlled, no probe sets would be expected to have both high adjusted ICCs and differences in expression levels between the two groups.

To capture quantitative changes in the adjusted and unadjusted ICCs and provide visualization, we use

$$\delta = \rho_0 - \rho_1.$$

Thus, δ is the difference in an array ICC after adjustment for the difference in expression levels between the two spike-in groups. The top N probe sets according to the rank of δ would correspond to top N differentially expressed transcripts, i.e. the transcripts spiked in at different concentrations.

4.1.2 Detecting the effect of biological samples by using a multi-level LMM

In the last stage, the single-level model M_1 (4.1) is extended to a two-level model M_2 in order to check if the genes satisfy the independence assumption for the technical replicates of one biological sample. The Platinum Spike probe-level dataset has one fixed factor and two cluster levels: arrays and biological samples. The model which is considered here is similar to the probe-level model of (Chu *et al.*, 2002) with addition of an extra level of clustering.

For a given probe set, let Y_{ijks} be the jth probe intensity on array i coming from the biological sample k in a spiked group s. The proposed model for the analysis of each probe set is a 2-level linear mixed-effects model given by

$$Y_{ijks} = \mu_j + \alpha_s + b_{2i} + u_{i(k)} + \varepsilon_{ijks},\ i = 1 \dots 3,\ j = 1 \dots p,\ k = 1 \dots 3, s = 1, 2. \quad (4.2)$$

Here $b_{2i} \sim N(0, \sigma^2_{b_2})$ are the level-1 random effects (arrays), $u_{i(k)} \sim N(0, \sigma^2_u)$ are the level-2 random effects (technical replicates from biological sample), and $\varepsilon_{ik} \sim N(0, \sigma^2_\varepsilon)$ are residual errors. The array ICC, which captures the association of the probe intensities measured on the same array isequal to $\rho_2 = \sigma^2_{b_2}/(\sigma^2_{b_2} + \sigma^2_u + \sigma^2_\varepsilon)$. Zhu *et al.* (2010) mentioned that the nesting effect of arrays within biological samples is not necessary and all technical replicates can be considered as independent samples. If this assumption holds, then the array-to-array variability or the measurement error are expected to be the dominant sources of variability in the probe set, while σ^2_u is expected to be relatively small and, therefore, $\rho_2(Bio) = \sigma^2_u/(\sigma^2_{b_2} + \sigma^2_u + \sigma^2_\varepsilon) = 0$.

4.1.2.1 Inference about the biological sample effect

So far, the models discussed in this section were used to explore and to model the different sources of variability in the data. In this section we proceed one step further. Models M_1 and M_2 specified earlier allow us to test the hypothesis that the variability due to biological replicates is not significant. Formally, we wish to test the hypothesis

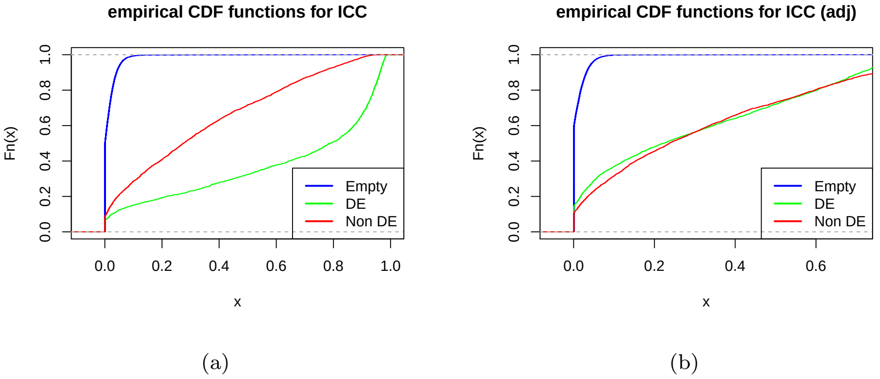
$$H_0 : \sigma^2_u = 0 \ \ \textit{versus} \ \ H_1 : \sigma^2_u > 0.$$

We note that the case in which the null hypothesis cannot be rejected implies that $\rho_2(Bio) = 0$ (*versus* the alternative $\rho_2(Bio) > 0$). The likelihood ratio test (Verbeke and Molenberghs, 2002) was used in order to test the hypotheses above. As we mentioned, if the assumption in (Zhu *et al.*, 2010) holds we do not expect to reject the hull hypothesis. Since the inference consists of testing 18,952 hypotheses, multiplicity adjustment is done by using the false discovery rate (FDR) control method (Benjamini and Hochberg, 1995).

4.2 Quality control of the Platinum Spike dataset

4.2.1 Distributional properties of ICC, adjusted ICC, and δ in the Platinum Spike Dataset

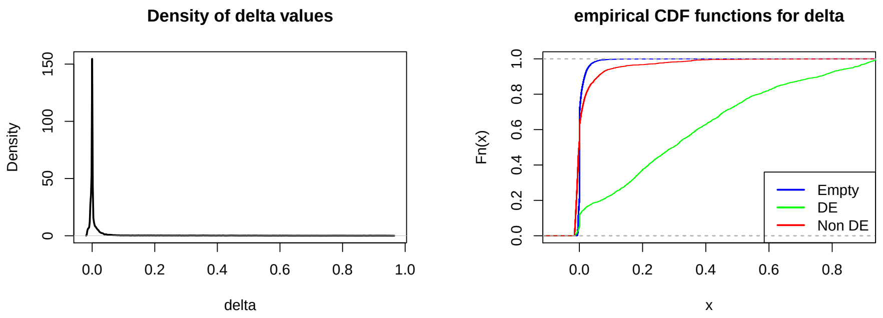
In order to illustrate properties of the ICC in the Platinum Spike dataset, the cumulative distribution function (CDF) of the ICC for empty probe sets and probe sets corresponding to spiked-in transcripts were plotted in Figures 4.2 and 4.3. The general trend in the CDF plots is reflecting the experimental design, i.e., cumulation of low ICC values for the empty probe sets, medium range of values for the probe sets measuring transcripts spiked in at the same concentration, and high values for the probe sets measuring the transcripts spiked in at different concentrations. The difference in unadjusted and adjusted ICCs, δ, has a skewed distribution (Figure 4.3(a)), with almost 90% of the values below 0.1. The empirical CDF functions for δ in each category of probe sets are shown in Figure 4.3(b).



(a) (b)

Figure 4.2: Empirical CDF functions according to the probe set category: (a) unadjusted ICC; (b) adjusted ICC.

Next, in Figure 4.4 we plot the ICCs from M_0 and M_1 for the probe sets according to the fold change of targeted transcripts. Several unexpected patterns can be observed. Firstly, a large number of probe sets do not have changes in ICC values after adjustment for the group difference, which is expected for empty probe sets and probe sets for transcripts spiked in at $FC = 1$, however, it is unlikely to observe this pattern in probe sets measuring transcripts spiked in at different concentration. There is a number of non-zero ICCs for empty probe sets and reversely, a number of very small ICC for probe sets capturing transcripts spiked in at different concentrations. Similar observations are coming from the estimated log fold change, $\hat{\alpha}$, where several empty probe sets have significant group difference and a number of probe sets corresponding to the differentially spiked-in transcripts have no evidence of differential co-expression regardless of the relative concentrations they were spiked in (Figure 4.5).



(a)

(b)

Figure 4.3: (a) Density of δ and (b) the empirical CDF functions of δ according to the probe set category.

4.2.2 ICC-based diagnostic procedure

The graphical summary of the changes in adjusted and unadjusted ICCs is given in Figure 4.6, where δ and estimated log fold changes $\hat{\alpha}$ for each probe set are shown. Assuming that the experiment is fully controlled, the probe sets measuring differentially spiked-in transcripts should have a clear non-zero $\hat{\alpha}$ as well as large decrease in adjusted correlations, i.e., large δ, so that the corresponding green dots will be in the 'tails' of this volcano plot. Since empty probe sets and probe sets measuring non-differentially spiked-in transcripts do not carry information related to the samples from two experimental groups, red and blue dots should be a condensed cloud at the bottom of the plot, where both $\hat{\alpha}$ and δ are low. According to the experimental setup, we would not expect scatter of blue or red dots in the 'green' area. Moreover, we would not expect to see green dots, representing differentially expressed transcripts, with low δ since it would mean that there is an extra unaccounted factor, influencing variation between arrays.

The probe profile plots of a typical empty probe set in the Platinum Spike dataset and a probe set capturing a non-differentially expressed spiked-in transcript from the cloud at the bottom of Figure 4.6 are shown in Figure 4.7. Figure 4.7(a) shows expression pattern of different probes in a probe set, where no changes in expression values are observed from the two groups of samples (A and B) and the lines do not appear parallel. Moreover, the expression values are varying between 2 and 9 without any specific pattern. The median value for log_2 of expression level of probes from

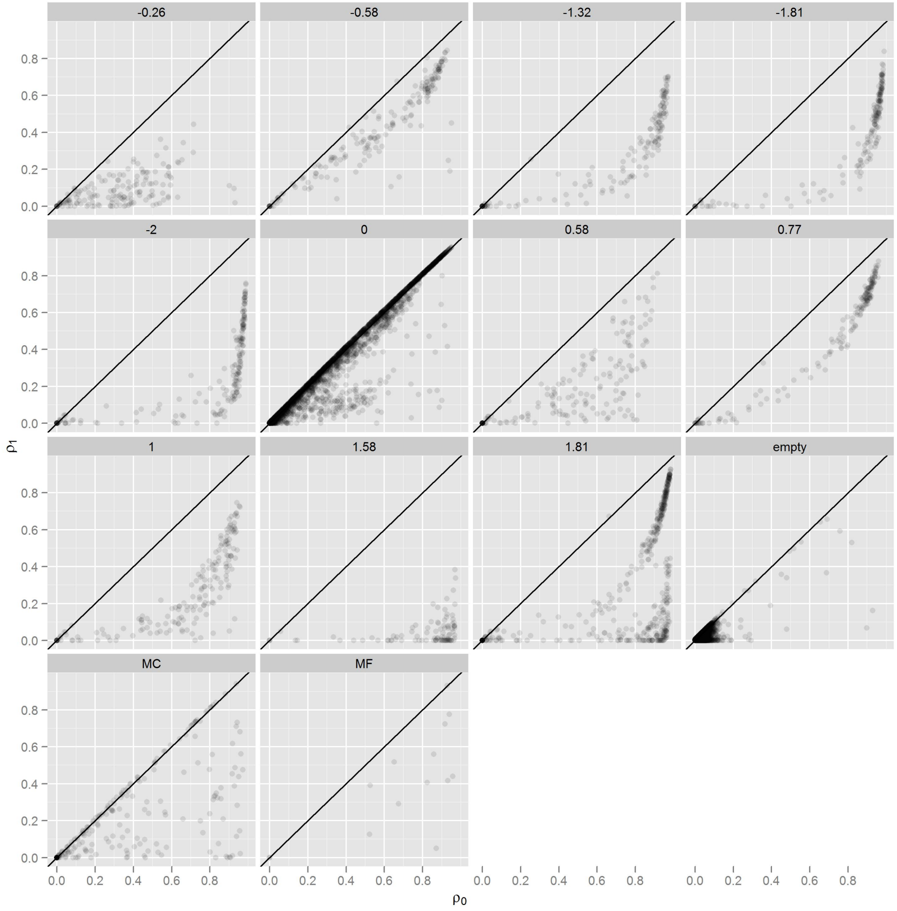


Figure 4.4: Intra-class correlations for the probe sets in the Platinum Spike dataset. x-axis represents ICC from the model when the grouping factor is not included, y-axis shows ICC from the model with group difference. Each panel represents a separate group of probe sets assigned to the transcripts in the experiment. The first five panels show ICCs for probe sets assigned to downregulated spiked transcripts, followed by the probe sets measuring spiked-in transcripts at the same concentration($log_2(FC) = 0$), the five probe sets assigned to upregulated transcripts, empty probe sets, 'MC' probe sets corresponding to multiple transcripts and 'MF' probe sets corresponding to the transcripts with multiple fold changes.

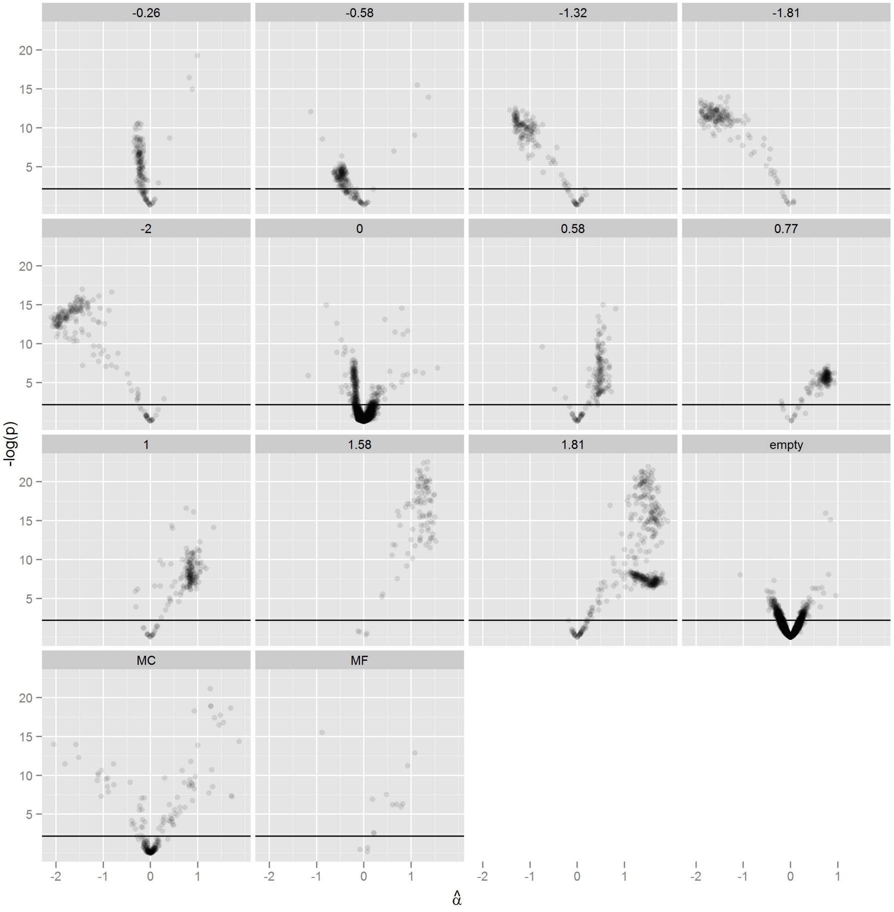


Figure 4.5: Volcano plots for the probe sets in the Platinum Spike dataset. x-axis represents estimated log fold changes, $\hat{\alpha}$, y-axis shows p-values transformed by $-log_{10}$. The reference line is the line above which the p-values are considered significant after adjusting for multiplicity ($p = 0.007$ for FDR=0.05). Each panel represents a separate group of probe sets assigned to the transcripts in the experiment. The first five panels show ICCs for probe sets assigned to downregulated spiked transcripts, followed by the probe sets measuring spiked-in transcripts at the same concentration($log_2(FC) = 0$), the five probe sets assigned to upregulated transcripts, empty probe sets, 'MC' probe sets corresponding to multiple transcripts and 'MF' probe sets corresponding to the transcripts with multiple fold changes.

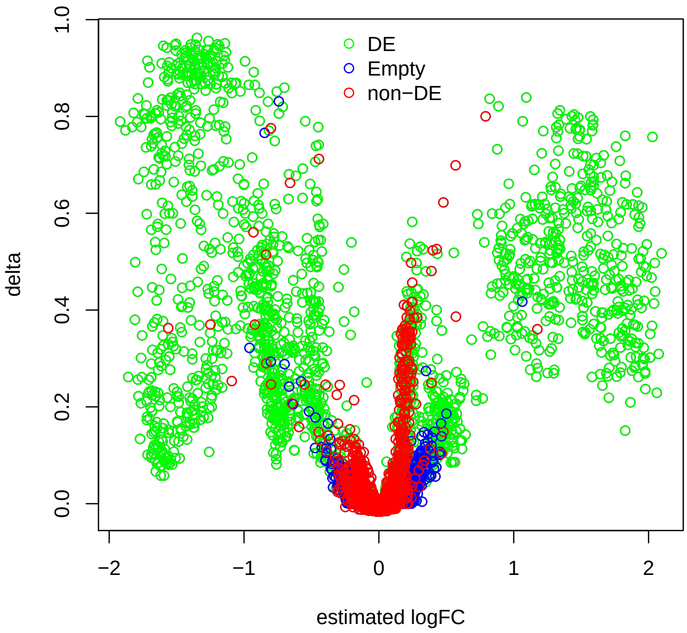
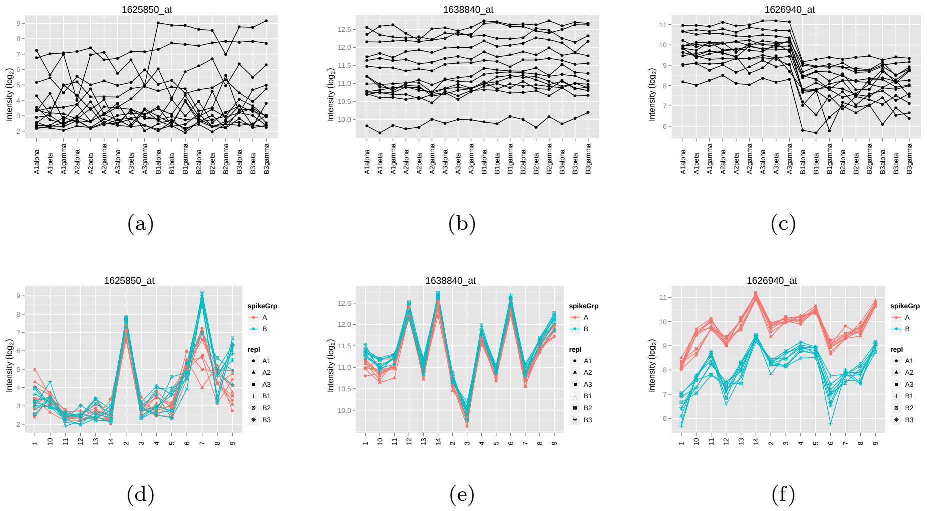


Figure 4.6: Estimated log fold changes of the two groups in the Platinum Spike dataset versus difference in intra-class correlation after adjustment for the group effect. Several probe sets are highlighted, namely, the empty probe sets and probe sets assigned to transcripts spiked at the same concentrations in two groups. These probe sets have the same expression pattern as probe sets assigned to transcripts spiked in at different concentrations.

empty probe sets in the Platinum Spike dataset is 4. As the empty probe sets are not supposed to measure any signal, the median intensity of 4 represents an approximation of the level of the background in the Platinum Spike dataset. On the other hand, the probe profiles in Figure 4.7(b) appear to be more parallel and have higher expression values (between 10 and 12.5) with no difference between the two groups. These expression values are in the typical expression range for probe sets measuring spiked-in transcripts in Platinum Spike dataset. A profile plot for the probe set capturing a transcript spiked in at different concentrations is shown in Figure 4.7(c). In a typical case, the expression values are different between the two groups and their range is above background values.

Having illustrated the probe sets with expression patterns as expected to be observed



(a) (b) (c)

(d) (e) (f)

Figure 4.7: Expression profiles for probes and arrays in a typical empty probe set 1625850_at (a), (d); a typical probe set measuring non-differentially expressed spiked-in transcript 1638840_at (b), (e) and probe set 1626940_at measuring differentially expressed spiked-in transcript (c),(f).

in the experiment, we consider the unexpected patterns in empty probe sets and probe sets measuring non-differentially spiked-in transcripts.

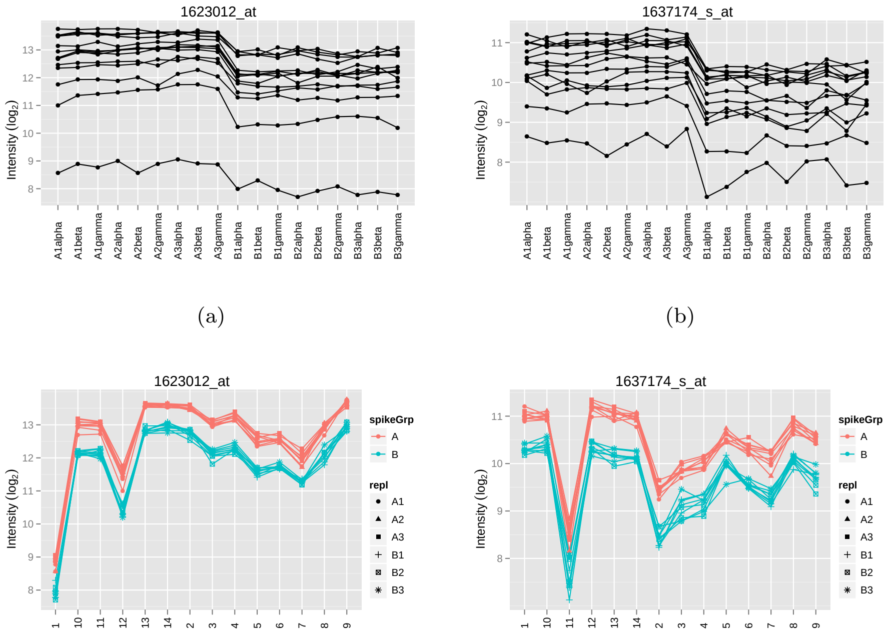
4.2.3 Empty probe sets and non-differentially spiked-in transcripts

There are seven empty probe sets and probe sets measuring non-differentially spiked-in transcripts with $\delta > 0.53$ (Figure 4.6). The corresponding probe sets are included in the list of top 500 probe sets according to δ ranking and having non-zero log-fold change for the two groups, which implies that these probe sets behave similarly to the probe sets measuring transcripts spiked in at different concentrations.

Empty probe sets with the unexpected differential expression are 1623012_at (profile plot is shown in Figure 4.8(a)) and 1633201_at.

Probe sets capturing non-differentially spiked-in transcripts with the unexpected differential expression are 1637174_s_at (profile plot is shown in Figure 4.8(a)), 1637628_at, 1623247_at, 1632358_at, 1632955_at, 1638336_at, 1641208_s_at and 1641606_s_at.

Even though the number of the probe sets with unexpected differential expression pattern is quite low among the top 500 probe sets (the proportion is 0.014), the proportion of the empty probe sets and the probe sets targeting differentially expressed transcripts increases in top 1,500 probe sets to 0.082, and in top 2,000 to 0.13, respectively.



(a)

(b)

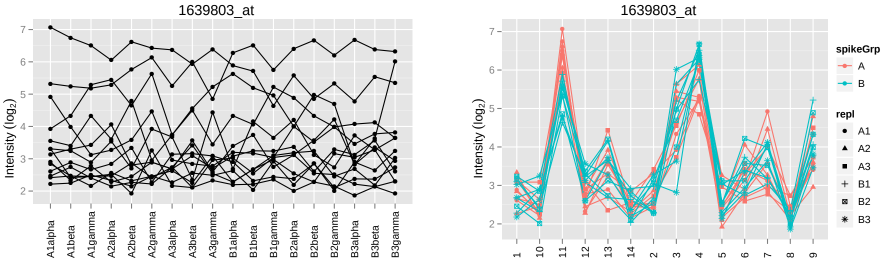
(c)

(d)

Figure 4.8: Probes expression profiles show that these probe sets have a pattern similar to the probe sets capturing transcripts spiked in at different concentrations: (a) an empty probe set, 1623012_at, (b) a probe set for the transcript spiked in at the same concentration in two groups, 1637174_s_at.

4.2.4 Differentially spiked-in transcripts

It is important to re-consider the probe sets measuring differentially spiked-in transcripts which have low δ and low estimated fold change, in other words, which have expression patterns similar to the empty probe sets. We list a subset of the most outlying probe sets in terms of estimated fold changes: 1639803_at (profile plots are shown in Figure 4.9), 1623164_a_at, 1623924_at, 1624390_a_at, 1625707_s_at, 1626428_s_at, 1627983_at, 1628987_at, 1631713_x_at, 1632269_at, 1635639_a_at, 1636160_at, 1638085_at, 1638896_at. Most of the probes in the probe sets have log_2 expression values below 5, which fall in the range of background values in the Platinum Spike data.



(a)

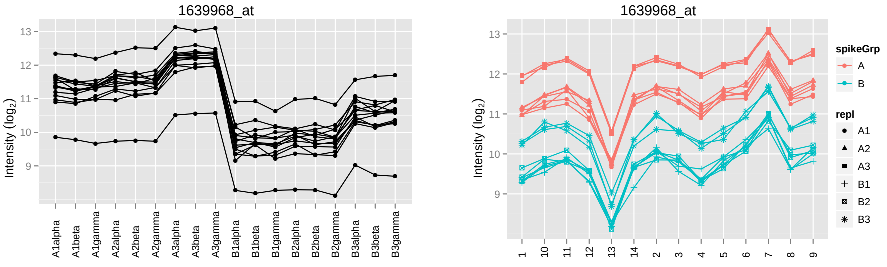
(b)

Figure 4.9: The probe set 1639803_at. Expression profiles for probes (a) and arrays (b) show that this probe set has the expression profile similar to the empty probe set.

4.2.5 Presence of the biological sample effect

The next step is to check why many probe sets had high estimated differences in concentrations and unexpectedly low changes in array ICC. We selected a group of probe sets from the bottom-left corner of Figure 4.6 and checked the values of the array ICC in both models. All these probe sets measuring differentially spiked-in transcripts had high adjusted ICCs, suggesting that there was an important factor omitted from the model, namely, the effect of biological samples. Figure 4.10 shows the expression levels of the probe set 1639968_at and reveals that there is a clear difference in the expression profiles for the replicates coming from one biological sample. The largest difference is observed from the A3 and B3 samples. We consider it to be the experimental artifact due to the subsequent processing of the samples or a possible plate effect. These findings are confirmed by checking the ICC for biological samples from the two-level model. Indeed, the ICC for the biological sample is high ($\rho_2(Bio) > 0.75$), and after adjustment for the biological sample in a two-level model, the array ICC becomes small ($\rho_2 < 0.05$). Note that that the ICC obtained for the one-level model is larger than 0.75, indicating that if the second level is ignored, this variability source is captured by the array random effect and is associated with the measurement error ($\sigma_\varepsilon^2 = 0.013$ for the one- and two-level models).

In order to formally test if the variability due to the biological sample is signif-



(a) (b)

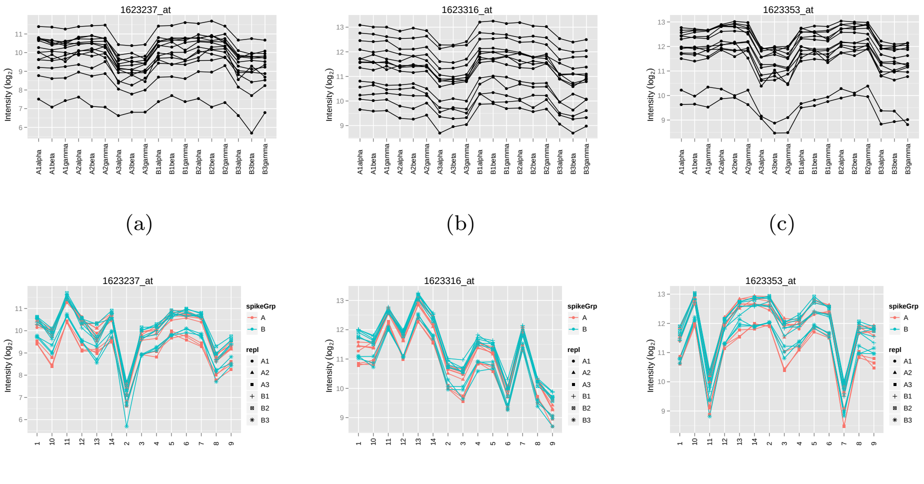
Figure 4.10: Probe set 1639968_at. Expression profiles for probes (a) and arrays (b) show that this probe set has the effect of biological samples A3 and B3 on the expression values (adjusted p-value for LRT < 0.001).

icant, the LRT, discussed in Section 4.1.2.1, was used to test the null hypothesis $\rho_2(Bio) = 0$. After adjustment for multiplicity, $1,793$ probe sets had significant effect of biological sample. The top three probe sets are shown in Figure 4.11. The probe sets 1623237_at, 1623316_at, 1623353_at come from the group of probe sets corresponding to transcripts with the designated $FC = 1$ (i.e., non-differentially spiked in transcripts). Similar to the probe sets in Figure 4.10, the most evident effect comes from the samples A3 and B3.

To illustrate the effect of considering all arrays as independent samples, we extracted standard errors for group difference, the log-fold change, and the corresponding p-values from the models M_1 and M_2. As expected, the mean log-fold changes ($\hat{\alpha}$) are the same in both models. However, the standard error of α is smaller when the random effect of biological replicate is ignored (Figure 4.12). This has a major influence on statistical significance test, since a number of significant differentially expressed genes goes down in the two-level model. As a result, from $2,653$ probe sets with significant difference between groups, only $1,029$ remain significant after taking into account the second-level random effect.

4.3 Discussion

We have carried out the quality control of the probe-level data from the Platinum Spike dataset. We have applied the linear mixed-effects model in order to investigate different factors influenced variation in the expression levels of probes, taking into account probe-specific intensity, variation between biological samples and difference



(a) (b) (c)

(d) (e) (f)

Figure 4.11: The Platinum Spike data. Probes and arrays profile plots for the first top ranked probe sets based on the LRT.

in concentrations between the two spike groups. To capture the amount of variation induced by difference between the two well-defined experimental groups, the unadjusted and adjusted array-ICC were computed and compared to the estimated fold change for each probe set in the data. It is an exploratory tool that allows to check how the experimental concentration of spiked-in transcripts relates to the consistent variability across measurements. We have shown that non-zero adjusted array-ICC indicates the presence of an unobserved or factor not considered in the model. In case of the Platinum Spike dataset, the unexplained high array-ICC was due to the variability between biological replicates. We have observed that technical replicates coming from the same biological sample smaller within-biological-sample variability compared to between-biological-sample variability.

We will summarize the major implications of this work. The ranking of probe sets based on the value of δ, i.e., the difference between adjusted and unadjusted array-ICC, highlights that a number of empty probe sets and probe sets measuring non-differentially spiked-in transcripts captured the signal similar to the probe sets measuring differential spiked in transcripts. At the moment, we cannot determine the actual biological reason behind it. Most likely, it is due to the cross-hybridization of the probes with the transcripts spiked in at different concentrations due to high sequence similarity. On the other hand, not all probe sets measuring differentially spiked-in transcripts revealed the expected expression pattern in the data. The most drastic

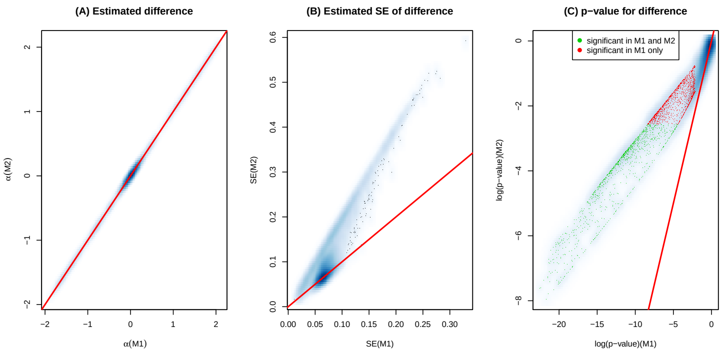


Figure 4.12: Changes in the estimates of log fold changes, associated standard errors and the test statistics for differential expression after adjustment for random effect of biological sample: A. log-fold changes (estimated difference) remain the same. B. The standard errors increase. C. Log-p-values for the differential expression test, where color highlights significant discoveries after adjustment for multiplicity.

reduction in the ability to detect differential expression was observed for the transcripts spike in with low fold changes. It has been noticed before by Lu *et al.* (2011); Subramaniam and Hsiao (2012) during comparative analysis of predictive models. The possible explanation is that small fold changes are unlikely to be detected by the GeneChip. Furthermore, a number of differentially spiked-in transcripts at large fold-changes were not detected either, which was observed in the expression profiles of the probe-level data. It can be explained by the bad performance of the probes within these probe sets caused by a number of reasons, such as probe sequence composition, inefficiency linked to the photolithographic oligonucleotide synthesis, etc. A number of the probe sets which failed to measure spiked-in transcripts should be re-considered. If this type of probe sets is not considered to be empty, the sensitivity of any classification model will not be optimal. Moreover, it may be dangerous to regard to this type of probe sets as false negatives, since there is no evidence in the data for the difference in concentrations between two experimental groups. Therefore, methods that would be able to call probe sets without signal (thus, similar to empty probe sets) as ‘spiked in’ would most likely provide overfitting.

Another important discovery of this quality control procedure was based on the likelihood-ratio test. In contrast to the assumption of Zhu *et al.* (2010), almost 10% of probe sets had a significant effect of biological samples, which implies that arrays cannot be treated independently. Profile plots of the probe sets with the strongest evidence of the biological sample effect confirmed that there was a clear effect of two biological samples. It can be attributed either to the sequential processing of samples in microarray experiment or the plate effect during the synthesis of cRNA. Hence, the arrays in the Platinum Spike data cannot be considered independent and the correction factor should be used prior to any differential expression analysis and inference. As an example, we have shown the effect of ignoring the second level random effects at the probe level on the estimation of the standard error of the treatment and differential expression analysis.

The proposed quality control tool has several limitations. We focused on an exploratory tool to detect the changes in variability caused by the experimental factors. The ranking-based procedure largely depends on the cut-off value, beyond which we cannot distinguish between various classes of transcripts in the data. For the moment, we have considered the strongest outlying signals to revise the list of the probe sets with the signal. However, this list can be further refined by enriching the results with biological interpretation. It should be noted that the LRT for the significance of variance components assumes normally distributed values of probe intensities, which should be validated.

Chapter 5

Analysis of the probe-level Affymetrix data: a simulation study

The probe-level linear mixed-effects model has been extensively studied in a variety of settings before (Stevens *et al.*, 2010). However, there is still little known about its performance compared to the similar analyzes carried out on the summarized data. Therefore, a simulation study was conducted in order to compare analyses based on summarized and probe-level data. Section 5.1 introduces a number of methods for testing the differential expression, which are applied to the case studies in Section 5.2. Motivated by the results of differential expression analysis, a simulation study is carried out and the results are presented in Sections 5.3 and 5.4.

5.1 Differential expression analysis at gene level

Let us consider several methods for testing the differential expression between two groups of samples. Let n_1 and n_2 be the number of samples in group 1 and 2, $\bar{Y}_{g1}$ and $\bar{Y}_{g2}$ be the average expression values for a gene g. In what follows we describe methods for testing the differential expression used in this chapter.

5.1.1 The t-test

When a covariate of interest has only two levels, the t-test can be applied. For each gene g in the dataset, the test statistic $t_g = \left(\bar{Y}_{g1} - \bar{Y}_{g2}\right) / \sqrt{\sigma_g^2}$ is calculated, where σ_g is the standard error of the mean group difference for the gene g. Under null hypothesis of no difference between two groups, t_g statistic follows the t-distribution with $\nu = n_1 + n_2 - 2$ degrees of freedom. For the simple analysis of a microarray experiment, the p-values obtained form the t-test for each gene are adjusted for multiplicity by Benjamini-Hochberg procedure with FDR control at the level 0.05 (Benjamini and Hochberg, 1995).

5.1.2 Linear models for microarray data (limma)

Limma is an empirical Bayes method for testing the differential expression of genes (Smyth, 2004). For testing the differential expression under two conditions, limma assumes that expression of each gene g can be modeled as follows:

$$Y_{gik} = d_{gk} + \varepsilon_{gik},\ g = 1, \ldots, m, i = 1, \ldots, n, k = 1, 2. \tag{5.1}$$

Here, Y_{gik} is a logarithm of summarized expression measurement for gene g on array i for condition k, d_{gk} is the mean gene expression for gene g under condition k, and ε_{ikg} is the measurement error. For testing the differential expression, limma uses a moderated t statistic, t_g^{limma}, computed by

$$t_g^{limma} = \frac{\bar{Y}_{g1} - \bar{Y}_{g2}}{\sqrt{\tilde{s}_g^2}},$$

where $\tilde{s}_g^2 = (\nu_0 s_0^2 + \nu s_g^2)/(\nu_0 + \nu)$ is the adjusted standard error of the difference d_{gk}. The prior parameters for the number of degrees of freedom, ν_0, and the variance, s_0^2, are derived by applying the empirical Bayes procedure to the gene variances $s_g^2, g = 1, \ldots, m$. The estimated test statistic t_g^{limma} is assumed to follow the t distribution with $\nu + \nu_0$ degrees of freedom. The p-values are adjusted for the multiplicity in the same way as described in Section 5.1.1.

For the probe-level data, the limma test is not straightforward to carry on, since the originally developed method made several assumptions: that the number of replicate spots per array is the same for all genes and that the correlation between these spots is comparable. These assumptions are not true in case of Affymetrix probe-level data, where the number of probes per probe set as well as the correlation between probes can vary.

5.1.3 Significance analysis of microarrays (SAM)

SAM is a resampling-based procedure for discovering differentially expressed genes (Tusher *et al.*, 2001). To correct for the genes with small variance, the ordinary t-statistic is adjusted by means of a data-specific fudge factor c. The fudge factor is needed for the moderation of the test statistic and is chosen so that the coefficient of variation of the SAM test statistic is minimized. For a two-group setting, the modified test statistic in SAM is given by

$$t_g^{SAM} = \frac{\bar{Y}_{g1} - \bar{Y}_{g2}}{(s_g + c)}. \tag{5.2}$$

The significance levels of the SAM test statistics are calculated by permutations with a simultaneous control of the FDR. For more details on the computation procedure we refer to Tusher *et al.* (2001) and Lin *et al.* (2008).

The gene level procedure can be readily extended to the probe-level data setting. For the probe-level analysis we adjust the SAM procedure in the following way. The modified test statistic, based on the linear mixed-effects model is given by

$$t_{g,PL}^{SAM} = \frac{d_g}{(s_{g,PL} + c)}, \tag{5.3}$$

where g is the index of a given probe set, d_g is the estimated difference in means between the two groups, $s_{g,PL}$ is the standard error for the group difference from the probe-level LMM, and c is a fudge factor.

5.2 Motivating example: comparison of analyses on probe-level and summarized data

The LMM and SAM were applied to the probe-level data. The fudge factor chosen for the probe-level test statistic was equal to 0.014, therefore the moderated test statistic had comparable values to the ordinary t-statistic. The fudge factor for the summarized data was equal to 0.015 for the RMA expression values and to $4.2 \cdot 10^{-9}$ for the FARMS summarized expression values. Table 5.1 presents the results on the number of genes called significant at $p = 0.05$ after adjusting for multiplicity.

Figure 5.1 graphically illustrates the result. After summarization, the number of differentially expressed genes detected by t-test or limma decreased for the FARMS summarized data. The results of t-test on the RMA data were more similar to the probe-level results, whereas FARMS shared less differentially expressed genes in common.

It must be mentioned that after the RMA summarization the data contained $2,575$ genes with the standard deviation of zero. There are many genes in the FARMS-summarized data with variance close to zero, which lead a large number of differen-

Table 5.1: The Sialin dataset. Number of genes with significant difference in gene expression between the 'KO' and 'WT' groups.

	probe-level	FARMS	RMA
LMM	667	-	-
t-test	-	354	490
Limma	-	484	777
SAM	841	4736	1051

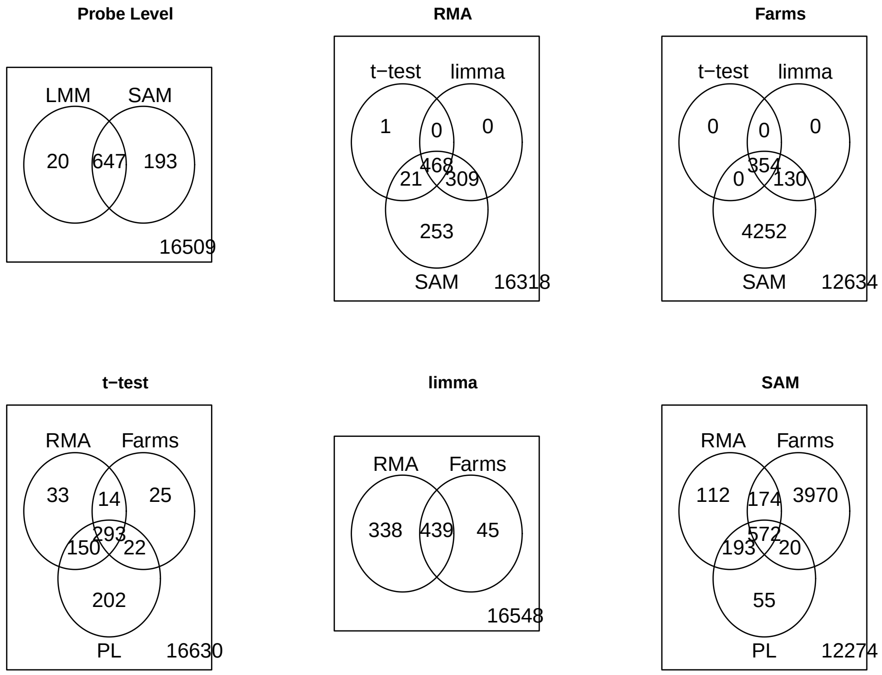


Figure 5.1: Venn diagrams for the overlap in results after testing for treatment effect. PL stands for the probe-level linear mixed-effects model analysis.

tially expressed genes compared to the probe level data. It might be either due to additional false positives in the data, or due to the genes with small variance.

5.3 Simulation study

In order to evaluate the power and FDR of the probe-level analysis and to compare it for various summarization techniques, a simulation study was designed. The data generation model for each probe set is based on the marginal model corresponding to a model with fixed effects of probes, a group difference, and a random effect of array:

$$\boldsymbol{Y}_i \sim N(\boldsymbol{\mu} + \beta_i \cdot \boldsymbol{I}, \sigma_\varepsilon \cdot \boldsymbol{I} + \sigma_b \cdot \mathbf{1}),$$

where $\boldsymbol{\mu}$ is a $l \times 1$ vector of probe means in a probe set, $\boldsymbol{I}$ is the $l \times l$ identity matrix, $\mathbf{1}$ is the $l \times l$ matrix of ones, and β_i is a coefficient for group difference ($\beta_i = 0$ in group 1).

The sample size, probe set size, group difference, and intra-cluster correlation were varied. The number of probe sets was set to $4,000$ and 100 expression matrices were generated. The number of probes per probe set l was set to 11, 20, and 30; for each dataset the number of probes per probe set was kept fixed. The layout of the simulation scenarios is summarized in Table 5.2. The probe-specific effects μ_j and the variances of measurement errors σ_ε were sampled from the Sialin Data. After σ_ε for each probe set were sampled, the variances of random effects σ_b were set to attain the specified ICC value (low, moderate, and high, as specified in Table 5.2). Afterwards, the group difference was selected to be moderate or high according to the probe set category, taking into account the variability in a probe set.

Table 5.2: Structure of the simulated data.

Probe set group	Treatment effect	ICC	No. of genes
A1B1	0	U[0.01, 0.2]	3200
A1B2	U[0.2, 0.5]	U[0.01, 0.2]	100
A1B3	U[0.8 , 1]	U[0.01, 0.2]	100
A2B1	0	U[0.4, 0.6]	100
A2B2	U[0.8, 1]	U[0.4, 0.6]	100
A2B3	U[1.5, 2]	U[0.4, 0.6]	100
A3B1	0	U[0.8, 1]	100
A3B2	U[1.5, 2]	U[0.8, 1]	100
A3B3	U[2.5, 3]	U[0.8, 1]	100

5.4 Simulation results

The results of the simulation study are presented in Table 5.3. There is an obvious dependency of power on the sample size and also on the number of probes in the probe set. However, the FDR is controlled at 0.05 for all methods and combinations of sample size and probe-set size. The simulation confirms that the analysis of the microarray data at the probe-level generated from the normal distribution has slightly higher power for the sample size 12. The graphical summary of Table 5.3 is presented in Figure 5.2.

Table 5.3: Simulation results: the median values and the standard errors for the power and the FDR.

		Power			FDR		
1	β	LMM	FARMS	RMA	LMM	FARMS	RMA
				$n = 12$			
11	$U[0.2, 2]$	0.94 (0.01)	0.92 (0.012)	0.92 (0.012)	0.03 (0.007)	0.03 (0.009)	0.03 (0.008)
	$U[0.8, 3]$	0.99 (0.0048)	0.98 (0.0052)	0.98 (0.0052)			
20	$U[0.2, 2]$	0.93 (0.0088)	0.92 (0.01)	0.92 (0.01)	0.03 (0.0073)	0.03 (0.0078)	0.03 (0.0072)
	$U[0.8, 3]$	0.98 (0.006)	0.98 (0.0061)	0.98 (0.0061)			
30	$U[0.2, 2]$	0.92 (0.01045)	0.91 (0.01041)	0.91 (0.01041)	0.04 (0.0072)	0.03 (0.0079)	0.03 (0.007)
	$U[0.8, 3]$	0.98 (0.0059)	0.98 (0.0062)	0.98 (0.0062)			
				$n = 24$			
11	$U[0.2, 2]$	0.98 (0.0054)	0.98 (0.0052)	0.98 (0.0052)	0.04 (0.0082)	0.04 (0.0081)	0.04 (0.0074)
	$U[0.8, 3]$	0.99 (0.0035)	0.99 (0.0034)	0.99 (0.0034)			
20	$U[0.2, 2]$	0.98 (0.0062)	0.97 (0.0061)	0.97 (0.0061)	0.04 (0.0083)	0.04 (0.0087)	0.04 (0.009)
	$U[0.8, 3]$	1 (0.0025)	0.99 (0.0025)	0.99 (0.0025)			
30	$U[0.2, 2]$	0.97 (0.00625)	0.97 (0.00631)	0.97 (0.00631)	0.04 (0.0088)	0.04 (0.00828)	0.04 (0.0078)
	$U[0.8, 3]$	0.99 (0.00241)	0.99 (0.00237)	0.99 (0.00237)			
				$n = 48$			
11	$U[0.2, 2]$	0.99 (0.0046)	0.99 (0.0046)	0.99 (0.0046)	0.04(0.0089)	0.04(0.0083)	0.04(0.0079)
	$U[0.8, 3]$	1 (0.0027)	1 (0.0028)	1 (0.0028)			
20	$U[0.2, 2]$	0.99 (0.0039)	0.99 (0.0039)	0.99 (0.0039)	0.04 (0.0077)	0.04 (0.0084)	0.04 (0.0084)
	$U[0.8, 3]$	1 (0.0024)	1 (0.00238)	1 (0.00238)			
30	$U[0.2, 2]$	0.99 (0.0049)	0.99 (0.00484)	0.99 (0.00484)	0.04 (0.00728)	0.04 (0.00719)	0.04 (0.00825)
	$U[0.8, 3]$	1 (0.0022)	0.99 (0.0022)	0.99 (0.0022)			

5.5 Discussion

The strength of the probe-level analysis lies in the usage of the information that is contained in probe-level intensities. Although there is a clear reasoning for summarization of the data, such as dimensionality reduction leading to decrease in the computational

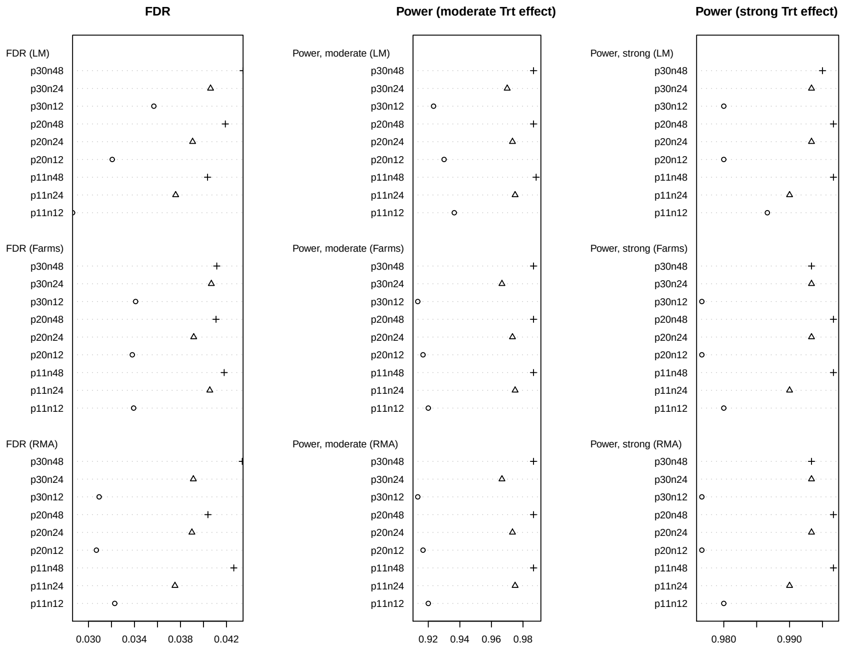


Figure 5.2: *Simulation results: The FDR, power for genes with significant moderate and strong treatment effect. The sample size is denoted by various symbols: circles for $n = 12$, triangles for $n = 24$ and crosses for $n = 48$.*

complexity of the analysis, the structure of the probe-level data contains more information about different sources of variability. The probe-level modeling reflects the original design of microarray experiments, taking into account the random and fixed sources of variation. When the data are summarized in order to obtain one expression value per array per gene, the variation within a probe set is averaged out. It must be mentioned that different summarization techniques try to summarize several observations in the way that would lead to the smallest loss of information but, in general, the loss is unavoidable.

The summarization technique RMA uses a linear model with both probe and array as fixed effects. In contrast, the probe-level LMM models array-to-array variability as a random effect, which is more suitable for the design of the Affymetrix data. It can also be observed that the expression values of the probe sets, which tend to be non-informative, result in small-variance RMA-summarized values. On the other hand, the FARMS summarization takes into account the latent nature of the gene expression and probe intensities. However, the FARMS-summarized values for the non-informative probe sets are set to the same values for all arrays, which suggests the filtering of the probe sets before analyzing the data. Yet, if filtering or summarization steps are not desirable from an analytic point of view, the probe-level models can be used to infer about the effects of important experimental factors without summarization.

Due to the drawbacks of the summarized data, the type I and II error probabilities can be higher in the tests for differential expression compared to the probe-level analysis. This difference was observed by Stevens *et al.* (2010) on the small-size datasets and has been confirmed in our simulation study. The average power of the tests based on the probe-level was slightly higher than of the similar tests conducted on the summarized data (for $n = 12$, $l = 11$), while for all methods FDR was controlled at the nominal level of 0.05. For the rest of the settings with larger sample size and probe set size, the performance of the differential testing procedures was the same for both summarized and probe-level data.

The current simulation setting has several limitations: the simulated data have been sampled from the marginal normal distribution and all probes in a probe set were considered to have the same target transcript. Moreover, the probe sets, which had the same values for all arrays after summarization by FARMS (i.e., the standard deviation was equal to zero), were not filtered out. Therefore, the current simulation study should be extended to take into account filtering effect on power and FDR, different sampling distributions (especially, for the data sets with small sample size) and the violation of assumption of all probes in a probe set targeting the same transcript.

Chapter 6

Data-driven refinement of the probe set annotation

Definition of a probe set, designed to measure the expression of a gene, is platform-dependent and is strongly related to the current genome definition. However, in a given experiment, a number of factors can influence the probe-level intensities, such as improper binding of probes to a target, manufacturing error, or differential splicing. The definition of a probe set for which groups of probes are mapped to the same transcript target has been a source of concern, because it is essential for summarization, analysis, and interpretation of results in a microarray experiment. Based on the Affymetrix probe set definition, it has been established that some probe sets cross-hybridize with two or more target transcripts due to the non-specificity of their probes. In order to solve this problem, Dai *et al.* (2005) proposed alternative chip definition files (referred within this chapter to as CDFs) in which probe sets are redefined to ensure that each probe hits only one genomic location and each probe set contains probes that are mapped to the same target transcript. The alternative annotation is based on the current state of the genome information and is in general static. However, a number of other factors can affect the definition of probe sets except for the genome annotation.

When a number of probes in a probe set fail to pick up the transcript either due to the alternative splicing or due to some technical problems, the whole probe set may be discarded from further analysis. However, it may still carry the important information for the part of a transcript.

In the current chapter we focus on a number of aspects of probe set re-definition based on the extended linear mixed-effects model. The aim of the chapter is to illustrate (1) how labeling of probes in a probe set can be used in a given microarray experiment and (2) how to extend linear mixed-effects model defined in previous chapter to discover probe sets with technical or biological problems on the probe-level. In Section 6.1, we provide the illustration of various scenarios in real-life datasets. In Section 6.2, we discuss some strategies to re-define a probe set. We demonstrate how the I/NI filtering is improved on the experiment-defined probe-set annotation in Section 6.3. The chapter is concluded by a discussion in Section 6.4.

6.1 Motivating examples

Kasim *et al.* (2010) have discussed that the filtering criteria rely on the assumption that all probes in a probe set are measuring the same transcript. In the pilot simulation study it was shown that if a probe set has less than 70% of correlated probes intensities, the probe-level filtering criteria cannot call this probe set informative. However, in an alternative isoform of a gene, only a subsequence is transcribed, and therefore, it is not captured by all probes in a probe set.

To illustrate one of the situations when a probe set actually measures a transcript, but due to alternative splicing it is not selected for the downstream analysis, we use intensity values of a probe set 67790_at from the Sialin experiment plotted in Figure 6.1. Two groups of probes can be identified. The first one with the correlated intensities values and the second one with uncorrelated intensities values. The two groups are confirmed by BLAST to be capturing two different exons of the RAB39B gene. Hypothetically, there could be several possibilities for probes in a probe set. The examples of possible grouping based on two groups of probes are illustrated on a simulated example in Figure 6.2. It is important that two components will be held apart and the probe sets are re-defined. Otherwise problems occur with the summarization and downstream analysis of the data.

6.2 Probes labeling and re-grouping

6.2.1 Classification of probes in probe sets

Kasim *et al.* (2012) proposed a data-driven way of identification of subgroups in a probe set by using a mixture model. In a nutshell, the model includes G classes of

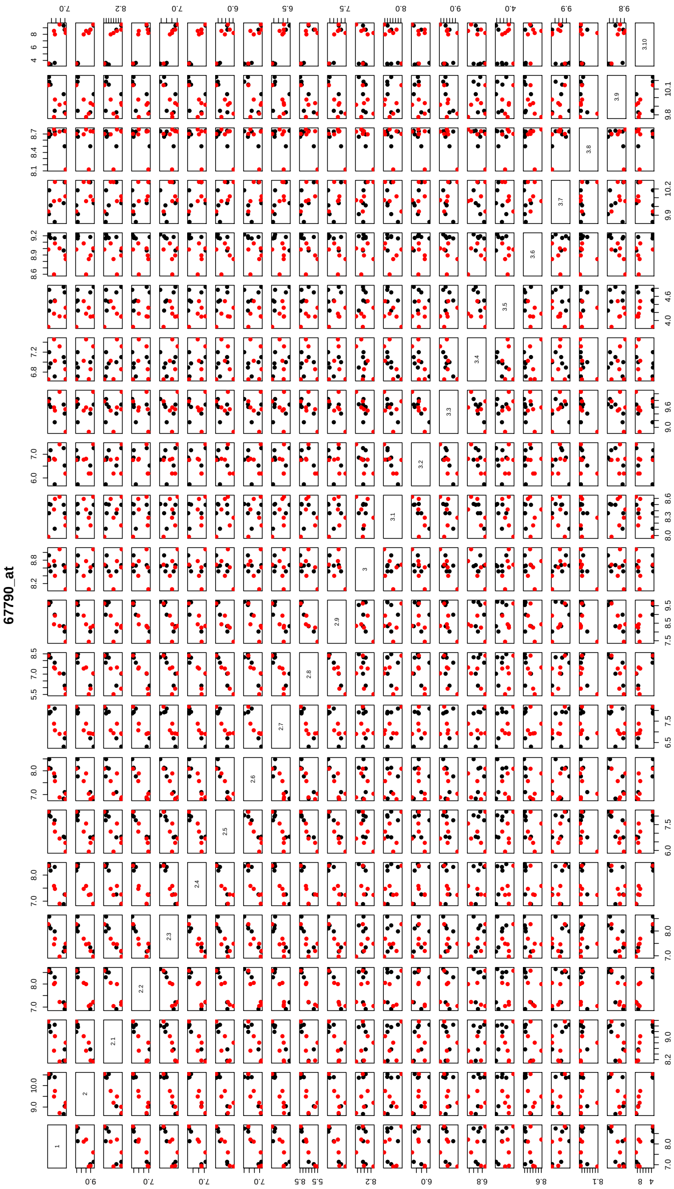
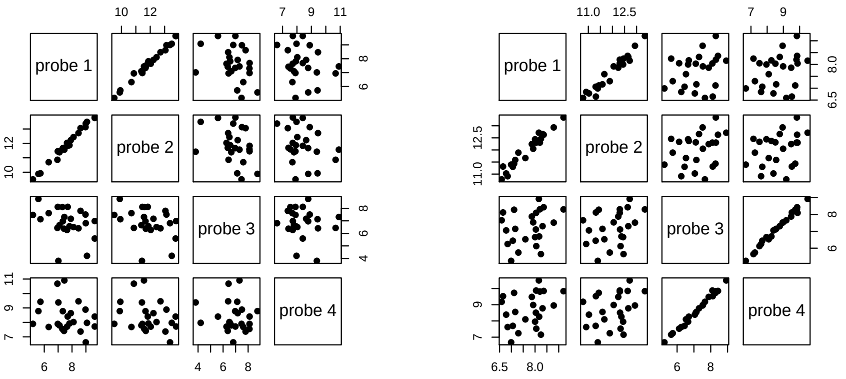


Figure 6.1: A scatterplot of probes intensities in 97790_at probe set shows clearly two components with probes numbered 1-2.9 coming from one exon (the probe intensities have high pairwise correlation) and 3-3.10 coming from another exon of the gene RAB39B (the probe intensities have low pairwise correlation).



(a) 2 components: informative and non-informative

(b) 2 components: both informative

Figure 6.2: Illustrative example. Scatterplots for probe sets with 2 components: (a) two selected probes from component 1 are correlated, whereas two selected probes from component 2 are uncorrelated with each other and with the probes of the first component; (b) two components, wherein probes are correlated, but there is no correlation between the probes from different groups.

probes in a probe set and is formulated as

$$Y_{ij} = \sum_{s=1}^{G} \pi_s N(\mu_j + b_{is}, \sigma_\varepsilon),$$

where π_s, $s = 1, \ldots, G$, is the mixing probability for component s and b_{is} are array-specific random effects for component s.

The parameters of the model are estimated within Bayesian framework by using Gibbs sampler with the normal and Gamma priors for the unknown parameters. In the final step, labels for probes are obtained according to the posterior probability of a probe belonging to the component.

In general, labels can be generated from another classifying method. For instance, when there is a classification model trained to assess the reliability of a probe due to its nucleotide sequence, reliable and unreliable probes can constitute two classes of probes per probe set.

6.2.2 Extension of the basic LMM

Once the labels are obtained, we can extend the LMM, described in detail in Chapter 3, by adjusting the variance-covariance matrix. The random effect model is formulated as

$$Y_{ijs} = \mu_j + b_{is} + \varepsilon_{ijs},\ i = 1 \ldots n_s, j = 1 \ldots l, s = 1 \ldots G, \tag{6.1}$$

where $b_{is} \sim N(0, \sigma^2_{bs})$ is the stratified random effect of an array, σ^2_{bs} is the variance of a component s, and $\varepsilon_{ijs} \sim N(0, \sigma^2_{\varepsilon s})$ is a residual with a component-specific variance to account for heteroscedasticity within each component.

The within-component ICC is given by

$$\rho_s = \frac{\sigma^2_{bs}}{\sigma^2_{bs} + \sigma^2_{\varepsilon s}}. \tag{6.2}$$

At the next step, for each probe set, the basic ICC ρ_0 defined in Chapter 3.1 is compared to the ρ_s and in case when $\rho_s > \rho_0$, we can re-define a probe set as follows: in the setting with two components, where one component has high ρ_{s1} (i.e., following the criterion of Kasim *et al.* (2010), $\rho_s > 0.5$), and the other with low $\rho_{s2} < 0.5$, we can consider excluding component $s2$ from downstream analysis or label as a *non-informative part* of a probe set. Thus, we define a component s of a probe set *informative* if $\rho_s > 0.5$. The probe set is informative if one of its components is informative.

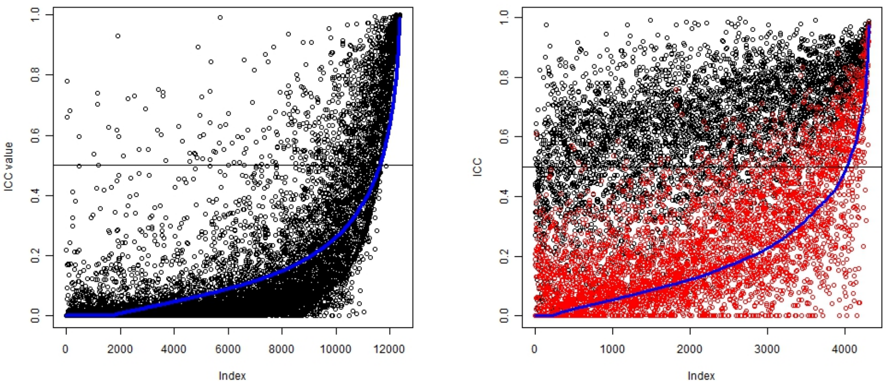
6.3 Results of improved probe set definition

6.3.1 The Sialin data

For the Sialin data, the majority of components had a very low number of probes (either one or two probes per probe set). The probe set components with less than 3 probes are removed from the analysis, since from a biological point of view it is difficult to interpret a component with one or two probes. Moreover, from the computational point of view, the estimates of random effects for one probe is meaningless. Once the probes from components of size one or two are removed from the probe sets, the distribution of probe sets per number of components is as follows: $12,356$ probe sets with one component, $4,302$ probe sets with two components, 660 probe sets with three components, and 52 probe sets with four non-trivial components.

It is remarkable that after some singletons and two-probes components have been removed, the ICC got substantially higher for some probe sets (Figure 6.3(a)). If we focus on 2-component probe sets and obtain ICCs for each component, then we can clearly see an increase in the values of ICCs (Figure 6.3(b)), which implies, that a substantial number of probe sets will pass the filtering criterion.

A similar trend is observed for probe sets with 3 and 4 non-redundant components.



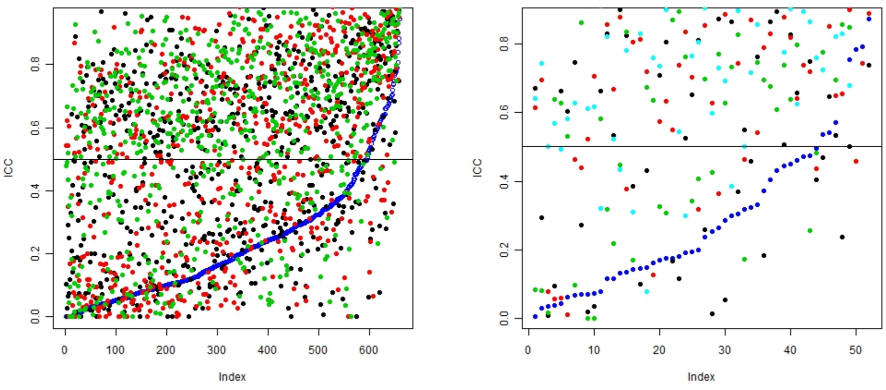
(a) (b)

Figure 6.3: (a) ICC for the probe sets with 1 component ($n = 12356$): blue line corresponds to the ICC from basic model, the dots represent the ICC for component after discarding components with one ore two probes. (b) ICC for the probe sets with 2 components ($n = 4302$): blue line corresponds to the ICC from basic model, the dots colors represent the ICC for the two components.

The changes in ICC for the 3 and 4 component probe sets are shown in Figure 6.4.

The plots of the ICCs highlight several important issues: the informative probe sets with a high ICC from basic model can be further refined by excluding non-informative components and it leads to the increase in ICC values, since the probes with noisy measurements are removed. However, the most interesting class of probe sets are those, which have a low ICC in the basic model and informative components with high ICC. The total number of probe sets with at least one informative component was $5,428$, which is more than five times higher than the number of informative probe sets from the basic model $(1,080)$.

We have mapped a selection of probe sets from the alternative CDFs to the original Affymetrix CDFs to check whether there is a link between multiple components in a probe set. In addition, the probe sequences were mapped to the Mouse genome v.10 by a fast sequence alignment tool `bowtie` (Langmead *et al.*, 2009) to check the exon locations of the probes and number of exons corresponding to the probe sets in



(a)

(b)

Figure 6.4: (a) ICC for the probe sets with 3 components ($n = 660$): blue dots correspond to the ICC from basic model, the green, black and red dots represent the ICC for each of the component; (b) ICC for the probe sets with 4 components ($n = 52$): blue dots correspond to the ICC from basic model, the dots of four other colors represent the ICC for the four components.

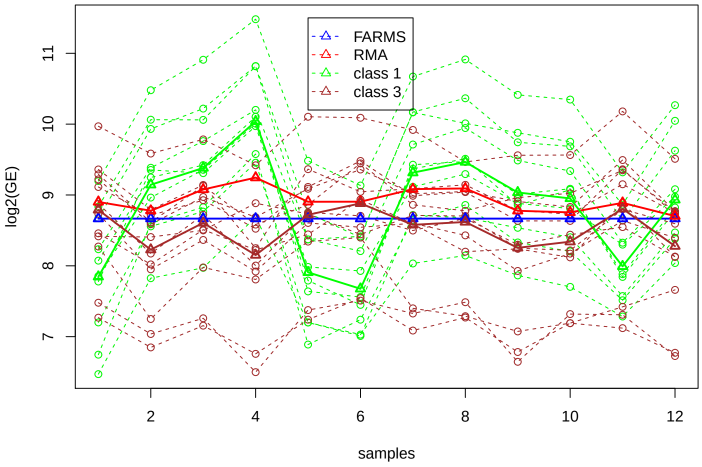
alternative CDFs. The results are shown in Table 6.1.

Table 6.1 presents only a selection of probe sets which are non-informative by the basic model, but become informative if the component-specific effects are taken into account. The probe set 20971_at has two informative components and is shown together with summarized profiles in Figure 6.5(a). The probe set 77945_at has one informative and one non-informative component shown in Figure 6.5(b).

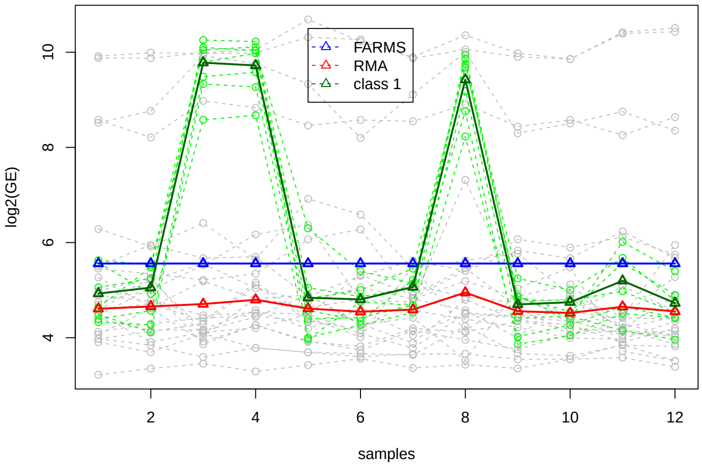
It should be noted, that the number of components in the mixture model is different from the number of corresponding Affymetrix probe sets and in addition different from the number of exons corresponding to the probe sets. It implies that the discovered grouping of probes in a probe sets depends on the experiment. In some cases it can be caused by alternative splicing, then component of a probe set can be interpreted as a probe set for a specific exon or a group of exons. Yet, in many cases it will not be related to the alternative splicing, rather to other reasons, e.g., to the failed binding of transcripts.

Table 6.1: The Sialin data. Mapping of probe sets with multiple components to the Affymetrix probe set annotation. The basic ICC is given in the column ρ_0, the vector of ICCs from the components is given in the column $\boldsymbol{\rho}$.

Probe set name aCDFs	size in aCDFs	$\sum$ probe sets in Affymetrix	$\sum$ mapped exons	ρ_0	$\boldsymbol{\rho}$
		2 components			
77945_at	30	3	-	0.23	$(0.98;\ 1.4 \cdot 10^{-9})$
20971_at	22	2	1	0.17	(0.95; 0.66)
103784_at	29	3	1	$2 \cdot 10^{-9}$	(0.87;0.07)
100503572_at	31	3	1+intron	$8 \cdot 10^{-10}$	(0.82;0.26)
104681_at	11	1	1	$2 \cdot 10^{-9}$	(0.33;0.79)
105504_at	11	1	1	$9 \cdot 10^{-9}$	(0.23;0.83)
74549_at	40	4	1	0.44	(0.05;0.71)
29819_at	21	2	2	0.6	$(0.83;8 \cdot 10^{-9})$
		3 components			
13003_at	41	4	5	0.04	(0.83;0.05;0.76)
16565_at	28	2	3	0.1	(0.73;0.55;0.05)
26943_at	72	5	-	0.24	(0.69;0.05;0.49)
18991_at	22	2	1	0.1	(0.61;0.48;0.64)
13193_at	43	4	4	0.04	(0.61;0.11;0.58)
103711_at	11	1	1	0.001	(0.64;0.38;0.32)
14950_at	64	5	3	0.07	(0.77;0.02;0.61)
		4 components			
100037258_at	53	5	5	0.005	(0.67;0.61;0.08;0.64)
13386_at	22	2	1	0.04	(0.66;0.06;0.62;0.49)
14264_at	56	6	1	0.06	(0.6;0.011;0.52;0.58)
14348_at	33	3	1	0.06	(0.75;0.46;0.09;0.62)
15378_at	55	4	2	0.11	(0.82;0.85;0.31;0.82)
16180_at	44	4	4	0.1	(0.89;0.87;0.45;0.43)
17988_at	55	6	1	0.17	(0.8;0.63;0.3;0.89)
17909_at	42	4	7	0.17	(0.7;0.57;0.32;0.73)
16001_at	54	4	1	0.12	(0.53;0.67;0.22;0.52)



(a) 20971_at

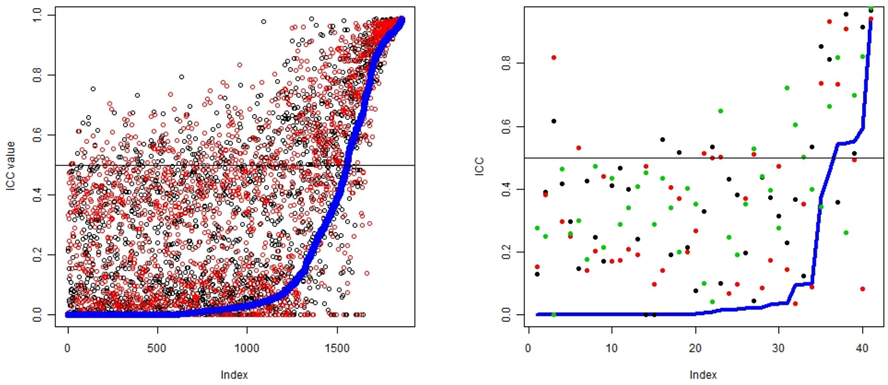


(b) 77945_at

Figure 6.5: Sialin data. Probes level intensities of the probe sets 20971_at and 77945_at, and the summarized values by FARMA and RMA. The probes from the informative components are summarized separately by median polish.

6.3.2 The Platinum Spike dataset

After applying the same procedure for the Platinum Spike dataset and removing components with less than three probes, $17,051$ out of $18,952$ probe sets had one component, $1,860$ probe sets had two components and 41 probe sets had three components. In comparison to the probe sets in the Sialin data, annotated by the alternative CDFs, the number of components in Platinum Spike data is substantially lower. The low number of components can be explained by the probe set size of twenty-two probes per probe set. In addition, in an experiment such as the Platinum Spike study, we would expect to see mostly single-component probe sets. The detection of probe sets with two or three components due to expression values can be explained by technical artifacts, like unspecific binding or incorrectly printed probe sequence for some of the probes. Figure 6.6 shows the estimated ICCs for the probe sets with two and three components.



(a)

(b)

Figure 6.6: (a) ICC for the probe sets with 2 components ($n = 1,860$): blue dots correspond to the ICC from basic model, the black and red dots represent the ICC for each of the component; (b) ICC for the probe sets with 3 components ($n = 41$): blue dots correspond to the ICC from basic model, the dots of the three other colors represent the ICC for the three components.

If we compare a number of informative probe sets in the basic model to the total number of informative probe sets taking into account informative components, it is considerably lower ($2,461$ vs. $3,640$, respectively). It illustrates that the violation of one of the basic filtering assumptions leads to the more stringent filtering criteria of

Table 6.2: The Platinum Spike data. A selection of non-informative probe sets with informative components. The basic ICC is given in the column ρ_0, the vector of ICCs from the components is given in the column $\boldsymbol{\rho}$.

Probe set name	ρ_0	$\boldsymbol{\rho}$
2 components		
1623800_at	0.03	$(0.94;9 \cdot 10^{-8})$
1626188_at	0.008	$(0.91;7.8 \cdot 10^{-8})$
1626290_at	0.27	(0.94;0.25)
1627248_at	0.29	(0.93;0.72)
1627971_s_at	0.21	(0.99; 0.71)
1630784_a_at	0.19	(0.91; 0.59)
1633474_x_at	0.15	$(0.95; 7.6 \cdot 10^{-8})$
1635500_a_at	0.09	(0.9; 0.86)
1635886_s_at	0.14	(0.91; 0.62)
1636544_at	0.24	(0.97; 0.57)
1637133_x_at	0.12	(0.93; 0.22)
1637325_at	0.002	$(0.97; 1.9 \cdot 10^{-8})$
1638495_at	0.04	$(0.99;9.1 \cdot 10^{-8})$
1639452_at	$1.3 \cdot 10^{-8}$	$(0.97; 3 \cdot 10^{-7})$
3 components		
1625943_at	$6.8 \cdot 10^{-10}$	$(0.62; 0.8; 6.7 \cdot 10^{-9})$
1628404_at	$8.7 \cdot 10^{-10}$	(0.86; 0.76; 0.38)
1629240_at	$1.1 \cdot 10^{-9}$	(0.1; 0.61; 0.68)
1636783_at	0.002	(0.62; 0.15; 0.51)

the probe sets and potentially influences the downstream analysis. We have selected several probe sets which are non-informative in the basic model, but have at least one informative component, in Table 6.2.

The probe sets 1623800_at, 1626188_at, 1626290_at, 1633474_x_at, 1637133_x_at, 1637325_at, 1638495_at, and 1639452_at have one informative and one non-informative components. When the values of all probes in the probe set 1638495_at are taken for summarization, the probe set does not capture the difference between two spiked in groups, however, if summarization is performed on the informative component then there is a clear signal in the informative part of a probe set as illustrated in Figure 6.7.

The probe sets 1625943_at, 1628404_at, 1629240_at, 1636783_at have three com-

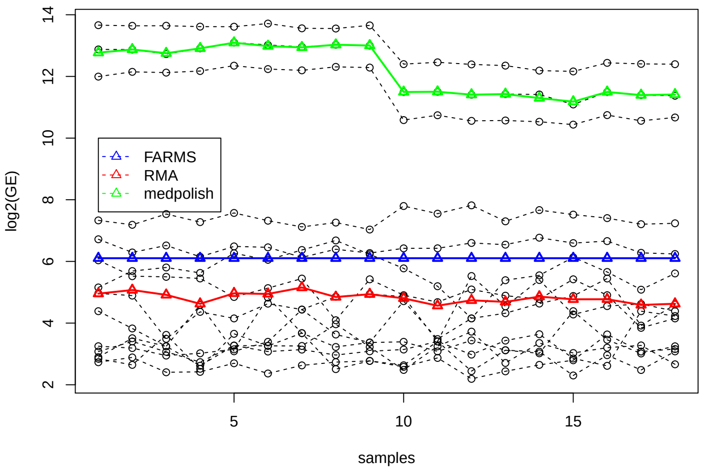


Figure 6.7: Platinum Spike data. A plot of probes intensities in 1638495_at probe set and the summarized values by FARMA and RMA. Both probe sets are declared non informative by the basic model. The three probes from the informative component are summarized by median polish.

ponents, out of which at least one is informative. The scatterplot of the probe set 1625943_at is shown in Figure 6.8. It is remarkable that in this particular probe set the class 1 and class 2 probe sets show anti-correlated patterns, which implies that the class 1 and the class 2 probes target transcripts from different biological mixtures.

Such patterns cannot be distinguished by the robust summarization methods such as FARMS or RMA, as shown in Figure 6.9, where the three probes from the class 1 have up-regulation and the three probes from the class 2 have down-regulation in the first nine samples, respectively.

6.4 Discussion

In this chapter we have presented the results of an ongoing research on the data-driven probe set definition based on the Affymetrix GeneChip data. The results presented in this chapter and the patterns detected in the data have not yet been presented in the literature.

In order to tackle a problem of discordant values of probes in a probe set due to

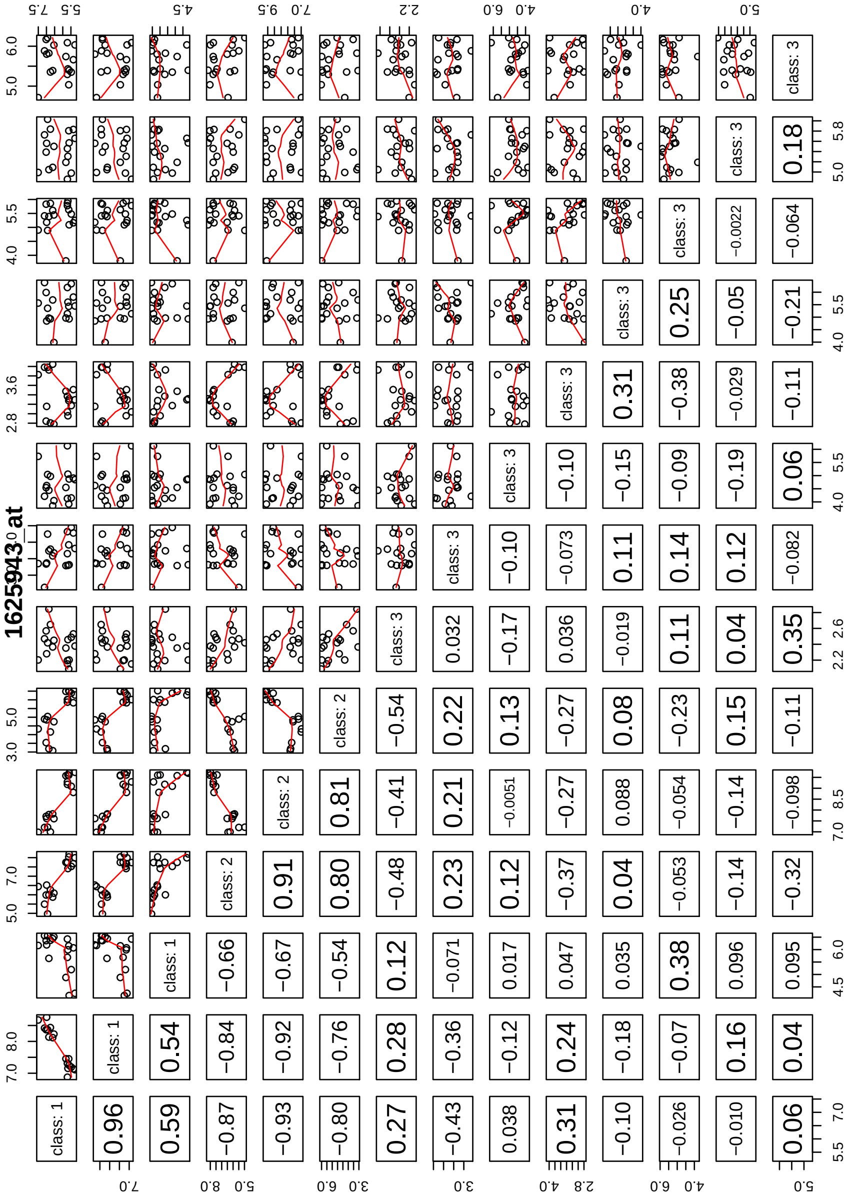


Figure 6.8: Platinum Spike data. A scatterplot of probes intensities in 1625943_at probe set with two informative components (class 1 and class 2) out of the three components identified by the mixture model.

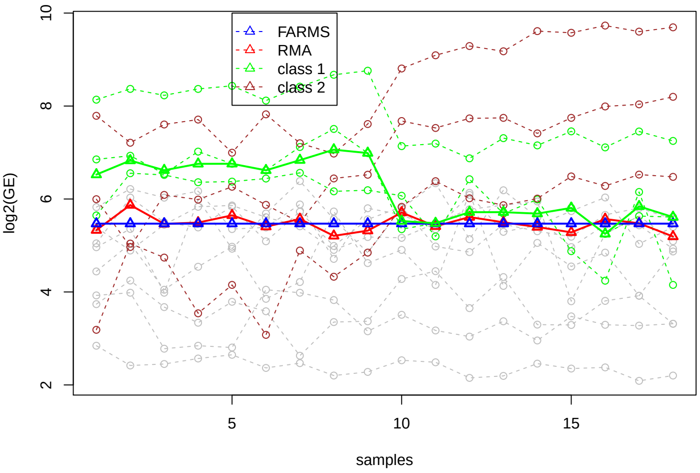


Figure 6.9: Platinum Spike data. A plot of probes intensities in the probe set 1625943_at with the overlayed summarized values by FARMA and RMA. The probes from the classes 1 and 2 are summarized separately by median polish.

either a biological process such as alternative splicing or technical artifacts such as manufacturing errors of probe sequences, we have used a mixture model to identify the number of components per probe set. We have shown that the re-definition of the grouping within a probe set allows the probe set previously called uninformative to be considered for the downstream analysis. It is important to mention that the definitions of probe sets provided by Affymetrix, which is based on earlier versions of genome builds, and by alternative CDFs do not solve the problem of all probes in a probe set targeting the same transcript.

The linear mixed-effects model can incorporate exon or Affymetrix probe set information as a fixed covariate and estimate variance components specific for each exon or each corresponding Affymetrix probe set. The formulation of the linear mixed-effects model does not change, it is the definition of a group s, which changes. Instead of a mixture-component, the group s can be an exon or an Affymetrix probe set indicator. The informative exons can be selected for further analysis of the differential expression without any other additional steps.

The flexibility of the linear mixed-effects model framework allows us to identify and incorporate various components of a probe set as a covariate for estimation of random effects in each component. Moreover, if the detection of alternatively spliced transcripts is not the goal of the analysis and the focus is on detection of differentially expressed genes, then the group difference can be modeled as a covariate in the model.

Part II

Biclustering analysis of gene expression experiments

Chapter 7

Biclustering methods: an introduction

In the last decade an explosion in data collection has been observed. The collections of biological data sets containing genetic data are still expanding. Other areas such as marketing data are collected in large amounts as well. Therefore, the development of computer-intensive techniques in statistics and data mining is required in order to extract meaningful information from all the data being collected. In this part of the dissertation, we continue to focus on gene expression experiments and, in particular, on the identification of *local expression patterns* in the expression matrix.

Early work in exploration and data mining of gene expression data was mostly done by unidimensional techniques, such as one-way clustering of genes or samples. Nowadays, biologists are also interested in local patterns of gene expression, e.g., which groups of genes are exclusively co-expressed in a group of samples. For exploration of local clusters of both genes and samples, two-dimensional techniques are used. Among the most popular solutions of this problem are biclustering algorithms (Madeira and Oliveira, 2004; Prelic *et al.*, 2006).

In Section 7.1, we formulate the biclustering problem and discuss the general approaches for biclusters identification. Section 7.2 describes biclustering methods used for testing, evaluation and data analysis. In Section 7.3, we provide an overview of statistical aspects in biclustering and the research questions addressed. In Section 7.4, we present the case studies that will be analyzed in this part of the dissertation.

7.1 Biclustering problem formulation

Madeira and Oliveira (2004) defined a bicluster and the biclustering problem in the following way. Let $\boldsymbol{A}$ be an $m \times n$ expression matrix, given by

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ . & \dots & . \\ . & \dots & . \\ . & \dots & . \\ a_{m1} & \dots & a_{mn} \end{bmatrix}, \tag{7.1}$$

where row i represents the expression levels for gene i on the microarray, column j represents array j of a sample under a certain condition. Our aim is to find a submatrix $\boldsymbol{D}$ or a set of p submatrices $\boldsymbol{D}_k, k = 1, \dots, p$, for which genes are co-regulated and co-expressed under a subset of conditions. A perfect biclustering algorithm, thus, would search for all possible submatrices in the data matrix and select the most important ones according to a certain criterion on homogeneity. The computation of all possible submatrices of data matrix is computationally intractable and, therefore, heuristic algorithms should be applied to optimize the runtime and provide an approximation to the ideal solution.

A more formal definition of a bicluster is given below (Madeira and Oliveira, 2004).

Definition 1

Let $\boldsymbol{A}_{m \times n}$ be a data matrix, X is a set of rows, Y is a set of columns. Let $I \subseteq X$, $J \subseteq Y$. Then a $k \times s$ submatrix $\boldsymbol{D}_{IJ} \subseteq A$ is a *bicluster*, where $I = \{i_1, \dots i_k, \}$ $(k \leq m)$ and $J = \{j_1, \dots, j_s\}$ $(s \leq n)$, and some specific characteristics of homogeneity of all elements in a bicluster is satisfied.

The precise definition of the co-expression is not available, but if genes are expected to be co-expressed, it must be reflected in their intensity levels. In general, a bicluster would be a submatrix of intensity values that are correlated or homogeneous in any other way. Madeira and Oliveira (2004) classifies biclusters in five major types described below. Figure 7.1 shows parallel coordinates plots for the different types of biclusters. The bicluster type is *constant values*, if all elements of bicluster are equal (Figure 7.1a); *constant values on rows or columns*, if all elements of every row/column are equal (Figures 7.1b and c); *coherent values*, if each row and column can be obtained by adding a constant or by multiplying by a constant value (Figures 7.1d, e and f); *coherent evolutions*, if the elements of the matrix are treated as symbolic values, ranks or discretized values. Coherent evolutions biclusters represent the most general class and cover constant and coherent biclusters.

If the data matrix is assumed to have a set of submatrices, then different types of the underlying structure of the bicluster set can be considered. We will focus on several of them: *exclusive row and column* biclusters, *non-overlapping* biclusters and *arbitrarily positioned overlapping* biclusters (Madeira and Oliveira, 2004). Schematically, the structures of bicluster sets are shown in Figure 7.2.

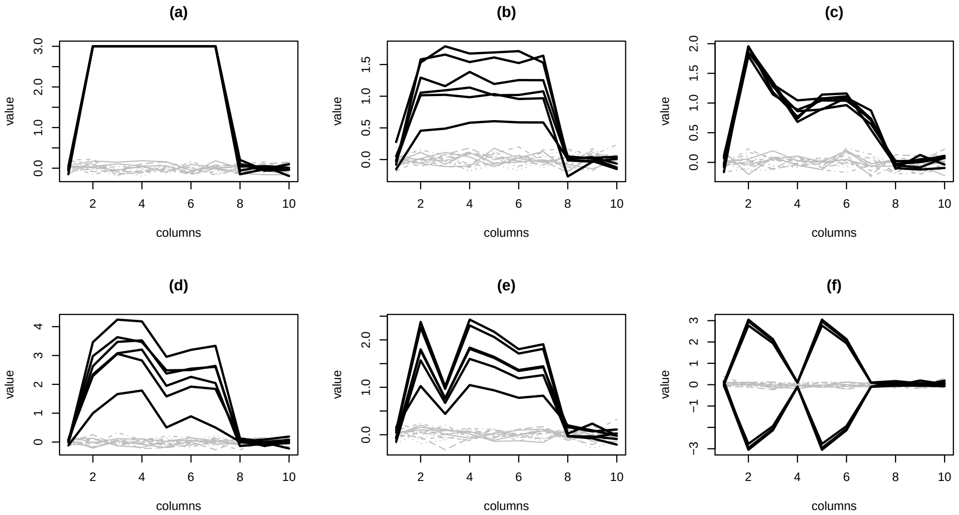


Figure 7.1: Parallel coordinates plots for a 6 × 6 bicluster in a 20 × 10 data matrix: (a) constant values, (b) constant rows, (c) constant columns, (d) coherent values, (e) coherent values under multiplicative model with correlated row profiles, (f) coherent values under multiplicative model with correlated and anticorrelated profiles. The plots show row profiles, each line plots rows values with background rows in gray, and bicluster rows in black.

Finding exclusive biclusters can be solved by clustering approaches, since the re-ordering or rows or columns will lead to blocks of rows or columns. For the overlapping biclusters, re-ordering is not trivial and usually for these data structures biclustering is considered to be more beneficial. The assumptions of a certain bicluster structure and the way biclusters are distributed in the data are determined by the method. Since many various assumptions on homogeneity of biclusters can be made, the number of algorithms is increasing. In the next section we will describe a selection of biclustering methods.

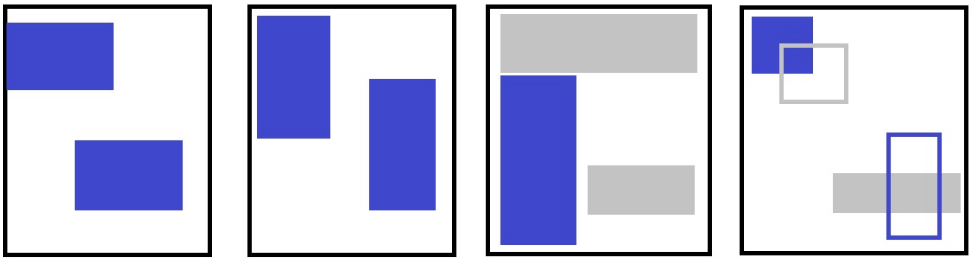


Figure 7.2: Types of structures for the sets of biclusters (from left to right): exclusive rows biclusters; exclusive columns biclusters; non-overlapping non-exclusive biclusters; arbitrary positioned overlapping biclusters.

7.2 Biclustering algorithms

The number of methods for finding biclusters is large and has been growing since the first biclustering method was published in 2000. Until now, the number of publications related to biclustering has been increasing exponentially, as demonstrated in Figure 7.3.

It is impossible to give an exhaustive list of all methods and to apply all of them to

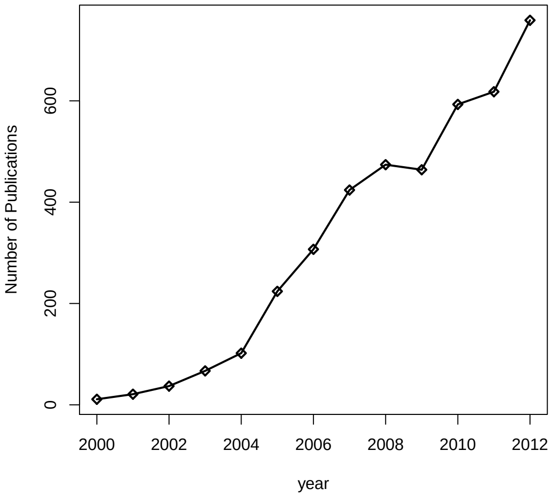


Figure 7.3: Number of publications on biclustering in period of 2000-2012. Data are taken from Google Scholar, search term ”biclustering” (May, 2013).

the data. Table 7.1 shows the methods, which are known to have a software implementation. We have focused our research on the following algorithms: δ biclustering, flexible overlapping biclustering, Plaid, Xmotifs, Bimax, spectral biclustering, Factor analysis for bicustering acquisition (FABIA), qualitative biclustering (QUBIC), iBBiG, and iterative search algorithm (ISA). These methods are well-described in data mining and statistics literature and are implemented in R. In Section 7.2.1 we discuss biclustering methods for discrete data and in Section 7.2.2 we cover biclustering methods for continuous data.

Table 7.1: List of biclustering methods with available software implementation.

Name	Methods
Implemented in R	
`biclust`	Plaid, BiMAX, δ, xMOTIFs, xQuest, Spectral (Kaiser *et al.*, 2011)
`Fabia`	FABIA, FABIAs, FABIAp, matrix factorization (Hochreiter *et al.*, 2010)
`NMF`	non-smooth non-negative matrix factorization (Pascual-Montano *et al.*, 2006a)
`s4vd`	s4vd (Sill *et al.*, 2011)
`qubic`	QUBIC (Li *et al.*, 2009)
`iBBigs`	iBBiG (Gusenleitner *et al.*, 2012)
`isa2`	ISA (Ihmels *et al.*, 2004)
`BicARE`	FLOC (Yang *et al.*, 2005)
`biclustGUI`	Plaid, BiMAX, δ, xMOTIFs, xQuest, Spectral, FABIA
`eisa`	ISA(Ihmels *et al.*, 2004)
`FacPad`	factor analysis for pathways (Ma and Zhao, 2012a)
`ThreeWayPlaid`	Plaid for three-dimensional data (Mankad and Michailidis, 2013)
`fastICA`	fast independent component analysis (Marchini *et al.*, 2012)
`cMonkey`	cMonkey (Reiss *et al.*, 2013)
Implemented in C, C++, C-sharp, Visual C++	
`NIM`	noisy itemset mining (Mouhoubi *et al.*, 2011)
`SS-CoSBI`	semi-supervised coherent and shifted bicluster identification (Teng and Tan, 2012)

Table 7.1 – continued from previous page

Name	Methods
`barycenter`	barycenter biclustering (Nisar *et al.*, 2009)
`DeBi`	differentially expressed biclustering (Serin and Vingron, 2011)
`Plaid`	Plaid (Lazzeroni and Owen, 2002)
`BBC`	Bayesian Plaid (Gu and Liu, 2008)
`SOM`	self-organizing maps (Marcos A.S. da Silva AND, 2013)
`Rncluster`	noise-robust biclustering (Ahn *et al.*, 2008)
`treebic`	hierarchical generative biclustering (Caldas and Kaski, 2011)
`TriMAX`	triadic concept analysis (Kaytoue *et al.*, 2011)
`QBBC`	query-based biclustering algorithm (Alqadah *et al.*, 2012)
`DRCluster`	differentially co-expressed constant rows cluster (Wang *et al.*, 2013)
	Implemented in Java
`BiCAT`	BiMAX, δ, OPSM, xMOTIF, ISA (Barkow *et al.*, 2006)
`BiGGesTS`	biclustering of GE time series data (Gonçalves *et al.*, 2009)
`SAMURAI`	SAMURAI (Okada *et al.*, 2007; Okada and Fujibuchi, 2007)
`bioNMF`	non-negative matrix factorization (Pascual-Montano *et al.*, 2006b)
`Expander`	SAMBA (Ulitsky *et al.*, 2010)
`cHawk`	cHawk (Ahmad, 2007)
`BiMine`	BiMine (Ayadi *et al.*, 2009)
`BiBit`	BiBit (Rodriguez-Baena *et al.*, 2011)
`BicFinder`	BicFinder (Ayadi *et al.*, 2012)
`CoBi`	CoBi (Roy *et al.*, 2013)
`BiCAT-Plus`	BiMAX, δ, OPSM, xMOTIF, ISA, MSBE (Al-Akwaa *et al.*, 2009)
`Anaconda`	ISA with modifications (Moura *et al.*, 2007)
`cMMR`	cMonkey (Kacmarczyk *et al.*, 2011)
`bicluster editing`	Bicluster editing (Sun P, 2012)
	Implemented in MatLab
`Spectral co-clustering`	Spectral co-clustering (Dhillon, 2001)

Table 7.1 – continued from previous page

Name	Methods
`BinaryBiclustering`	LAS for binary data (Lock, 2011)
`Ensemble biclustering`	ensemble biclustering (De Smet and Marchal, 2011)
`BiVisu`	split-merge algorithm, MSR, ACV minimization (Cheng *et al.*, 2007)
`RoBA`	robust biclustering (Tchagang and Tewfik, 2005)
`SIMBIC+`	SIMBIC+ (Bagyamani *et al.*, 2013)
`PIGA`	parallel immune genetic algorithm (Guifen *et al.*, 2007)
`LAS`	large average submatrices (Shabalin *et al.*, 2009)
	Implemented in Python
`BiBench`	δ, Plaid, OPSM, ISA, Spectral, xMOTIFs, BiMax, BCC, COALESCE, CPB, QUBIC and FABIA (Eren *et al.*, 2012)
`GEMbiclustering`	fast biclustering and parallel computing (Imhoff, 2006)
`RobinViz`	co-expression biclustering (Aladag *et al.*, 2011)

7.2.1 Biclustering of discrete data

Biclustering methods for discrete data such as iBBiG (Gusenleitner *et al.*, 2012) and BiMax (Prelic *et al.*, 2006) assume that the data have been binarized before the analysis, whereas Xmotifs (Murali and Kasif, 2003) and QUBIC (Li *et al.*, 2009) allow for a set of discrete values. The expected structure of the discovered biclusters is the most general one, coherent evolutions. However, simpler structures, such as constant values, constant rows or columns, or coherent values can be generated by these methods. In addition, all of these methods are designed to discover overlapping biclusters. We describe biclustering methods for discrete data and the main parameter settings for each method in the next section.

7.2.1.1 BiMAX

Prelic *et al.* (2006) developed an algorithm that works on the binarized version of the data matrix. An expression value is set to one if there is a change with respect to the control setting and to zero otherwise. If the control setting is unavailable, the threshold is applied based on the distribution of data values. A BiMAX bicluster spans a submatrix of 1's which cannot be a part of a larger submatrix of 1's, i.e., maximal inclusion bicluster. The BiMAX algorithm's goal is to find such maximal inclusion biclusters in a binary matrix by applying divide-and-conquer strategy. The algorithm is implemented in R and is available as the method `BCBimax()` in `biclust` package.

Parameters

For achieving the optimal time performance of the BiMAX algorithm, the size of the biclusters generated during the search process is constrained. In practice, before running the algorithm, the minimal number of columns and rows to be included into a bicluster must be specified. The number of biclusters should be passed as a parameter. At each iteration, the algorithm compares the result to the maximal inclusion bicluster found in the previous step. The algorithm terminates when either the boundary level of the number of biclusters has been achieved or when no maximal inclusion biclusters can be found.

7.2.1.2 iBBiG

Iterative binary bi-clustering of gene sets (iBBiG) was recently proposed by Gusenleitner *et al.* (2012). Similar to BiMAX, it looks for submatrices of 1's. However, it works under assumption of noisy data, which means that binarization is not perfect, hence, a number of 0's are tolerated within a bicluster. The method uses an iterative genetic algorithm to obtain biclusters of maximal size and entropy. The algorithm is implemented in R and is available as the `iBBiGs` package on Bioconductor.

Parameters

The most important parameter for iBBiG is a module fitness score, α, which is designed to control bicluster homogeneity and size. The input parameters for the genetic algorithm, such as a population size, a mutation rate, a success ratio, a selection pressure, and a stop criterion are reported to have little effect on the results. The number of biclusters has to be specified as well. It is recommended to use the upper boundary, so that algorithm can stop before the maximum number of biclusters has been reached.

7.2.1.3 xMotifs

xMotifs looks for conserved gene expression motifs in a discretized version of the gene expression matrix (Murali and Kasif, 2003). The underlying model assumes that gene can be expressed in a finite number of states, for example, up- and down-regulated. States of the gene expression could also be defined by a fold change, hence, quantile discretization of the original data matrix with log-transformed values would represent various levels of fold changes. The conserved gene expression motif is a submatrix of a maximum size, for which the values within each row are equal to the same level. In order to cover all the columns in the data completely using xMOTIFs, the following iterative procedure is used: find the largest xMOTIF in the data, remove the samples that satisfy this motif from the data, find the largest motif in the remaining data, and continue until all samples satisfy some motif. Thus, xMOTIFs looks for exclusive columns biclusters. The algorithm is implemented in R and is available as the method `BCXmotif()` in the `biclust` package.

Parameters

The algorithm's input does not require the number of biclusters, but a number of samples n_s to be randomly selected as a seed, a number of sets of size s_d of samples n_d to be randomly selected from the samples that are not in n_s. The minimum fraction of samples α in xMotif and a number of clusters to be returned have to be given as an input.

7.2.1.4 QUBIC

Qualitative biclustering (Li *et al.*, 2009) operates on discrete data with more than two levels and searches for submatrices of correlated values. Two rows are correlated if the values within any row are the same, i.e., 1's in one row and 2's in another. Thus, BiMAX and iBBiG biclusters are special cases of QUBIC biclusters, where all rows are expected to have 1's. QUBIC allows for anti-correlated patterns within a row, i.e., rows with -1 , -2 etc. xMOTIFs are special cases of QUBIC biclusters, where only correlated patterns are considered.

The search for correlated groups of rows under a subset of columns is performed using a graph-theory based algorithm, looking for the heaviest subgraphs. The weights for graphs are given by correlation between row values across columns. Essentially, the method uses seeds, which create an initial bicluster. The initial bicluster is expanded in both row and column direction, so that pairwise correlations between rows are maintained under the maximum consistency. The process is iterative and goes over all seeds specified as an input parameter. The algorithm was implemented in R and is available as the package `rqubic` on CRAN.

Parameters

If seeds for initalization of biclusters are not user-specified, they are constructed automatically. Maximum consistency is a control parameter for a proportion of tolerated 0 values within a bicluster. The range of possible ranks controls the number of levels which should enter the bicluster. The percentage of the regulating conditions of each gene controls the size of a bicluster. The maximum number of biclusters should be specified as the upper boundary for a number of biclusters.

7.2.2 Biclustering methods for continuous data

The methods for continuous data are additive methods such as δ-biclustering (Cheng and Church, 2000), FLOC (Yang *et al.*, 2005), Plaid (Lazzeroni and Owen, 2002) or multiplicative methods such as spectral biclustering (Kluger *et al.*, 2003), FABIA (Hochreiter *et al.*, 2010) and ISA (Ihmels *et al.*, 2004). δ-biclustering can find non-overlapping biclusters, and FLOC is an extension of δ-biclustering which allows for overlap in both row and column directions. Plaid, FABIA, and ISA can discover overlapping biclusters.

7.2.2.1 The δ biclustering (CC) algorithm

Cheng and Church (2000) proposed the first biclustering method for gene expression data analysis based on an ANOVA-like model together with a node deletion algorithm for finding δ-biclusters. Let $\boldsymbol{A}_{IJ}$ be a submatrix in data matrix $\boldsymbol{A}$, i.e., a δ-bicluster $(I = (i_1 \dots i_k);\ J = (j_1 \dots j_s))$, for which the mean squared residual score (MSR) H_{IJ} is given by

$$H_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} r_{ij}^2,$$

where $r_{ij} = a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}, i \in I, j \in J$.

In order to find δ-biclusters in the data matrix $\boldsymbol{A}$, the algorithm starts with the full matrix and calculates MSR. The goal is to minimize the MSR by deleting/including genes or samples in the matrix. The first step is to compute the MSR for each action and determine which of the addition/deletion maximally reduces the score. The computational approach of choosing all possible deletion and additions of row and column, known as a *brute-force approach* for finding the best action, would be computationally intractable. Therefore, Cheng and Church (2000) developed two algorithms, namely single-node deletion and multiple-node deletion, which are the *greedy search* algorithms, i.e., suboptimal ways of computing the best action that do not guarantee the discovery of an optimal solution. These two algorithms choose a row and a column with the largest MSR and remove them from the data matrix. This process is run iteratively until the submatrix with the MSR less than δ is found. After node deletion, the resulting δ-bicluster may not be maximal, in the sense that some rows and columns may be added without increasing the score. Therefore, node addition is performed by adding genes and samples one by one and computing the MSR. If the MSR at a current step does not exceed the MSR at a previous step, the column/row is added to the bicluster.

The algorithm is implemented in R and is available as the method `BCCC()` in the `biclust` package.

Parameters

The most important parameter for this method is the threshold δ. It is dependent on the total variability of the data, taking into account the assumed variability of the noise and the variability of bicluster values.

7.2.2.2 Flexible overlapped biclustering (FLOC)

The FLOC algorithm is a generalization of the δ-biclustering method that can detect k overlapping biclusters simultaneously.

The method starts with k initial biclusters, for which rows and columns are randomly selected with a probablity ρ. At the second stage, the algorithm updates the initial biclusters in order to improve the MSR of a bicluster. Iteratively, each row and column are examined to determine the best action (removal or addition) towards reducing the average MSR. Since there are k biclusters, there are k potential actions for each row and column. Similar to the original δ-biclustering, the action is chosen so that the gain function $Gain(\boldsymbol{A}_{IJ}, \boldsymbol{A}_{I'J'})$ is maximized. The gain function is determined by

$$Gain(\boldsymbol{A}_{IJ}, \boldsymbol{A}_{I'J'}) = \frac{H_{IJ} - H_{I'J'}}{\delta / H_{IJ}} + \frac{|I||J| - |I'||J'|}{|I||J|}.$$

Here, $H_{I'J'}$ is the MSR after an action has taken place and δ is a threshold parameter for residue. The gain function is, thus, a function of the MSR reduction and bicluster volume increment.

After the best action is determined, a bicluster with the best action is stored and sequentially, all the remaining biclusters are updated. If there is no improvement in the current iteration, the process terminates; otherwise, the iterative update continues.

The algorithm is implemented in R and is available as the `BicARE` package.

Parameters

The FLOC algorithm depends on the input parameters, such as r - residue threshold (equivalent of δ), ρ - the inclusion probability controlling the size of initial biclusters, $M(N)$ - the minimal number of rows (columns) to be included into bicluster, k - the number of biclusters and the maximumn number of iterations t.

7.2.2.3 Plaid

The Plaid model (Turner *et al.*, 2005) presents the data structure as a sum of layers. The model includes a background layer to capture the global effects. Afterwards, the method constructs a series of layers that represent biclusters. At each level, a newly discovered layer represents an additional effect that has not been captured by the general model.

In the Plaid model, the element $a_{ij}, i = 1, \dots m; j = 1 \dots n$ of the data matrix is modeled by

$$a_{ij} = \Theta_{ij0} + \sum_{k=1}^{p} \Theta_{ijk}\rho_{ik}\kappa_{jk} + \epsilon_{ij},$$

where k is the layer index, p is the number of biclusters, Θ_{ijk} is a sum of overall mean, row, and column effects in layer k, ρ_{ik} and κ_{jk} are indicator variables for a bicluster membership, i.e., $\rho_{ik} = 1, i \in k$ and $\kappa_{jk} = 1, j \in k$ and zero otherwise. For a given bicluster, the mean structure Θ_{ijk} can take four possible forms:

$$\Theta_{ijk} = \begin{cases} \mu_k, \\ \mu_k + \alpha_{ik}, \\ \mu_k + \beta_{jk}, \\ \mu_k + \alpha_{ik} + \beta_{jk}. \end{cases} \qquad (7.2)$$

The four parametrizations of the bicluster structure in (7.2) express biclusters with constant values, constant rows, constant columns, and coherent values. The unknown parameters to be estimated, i.e., μ, α, and β correspond to the overall effect, the row and the columns effect of the bicluster, respectively.

The estimation method for the Plaid model proposed by Lazzeroni and Owen (2002) and improved by Turner *et al.* (2005) is an iterative algorithm. The background layer is fitted first, then bicluster-specific layers are added one at a time. Within a layer, parameters are updated iteratively. The number of iterations should be specified as an input parameter. After the model parameters are updated, the layer sum of squares is calculated and the permutation test is performed.

The permutation layer is a built-in protection of Plaid method from discovery of random biclusters. At each step of permutation, the sum of layers for the same block is calculated. If the layer sum of squares is significantly larger than sums of square of layers in shuffled data, it is added into the model and the algorithms proceeds. The algorithm works until a specified number p if reached or no more significant layers can be found.

The algorithm is implemented in R and is available as the method `BCPlaid()` in the `biclust` package.

Parameters

Before running Plaid, one of the models from (7.2) should be specified. The other internal algorithm parameters are R^2-like thresholds for each layer, the number of permutations to perform for each layer, the number of iterations and back-fit iterations.

7.2.2.4 Spectral biclustering

Spectral biclustering (Kluger *et al.*, 2003) is a method of searching for multiplicative biclusters of coherent values, i.e., when each element of a bicluster can be defined as a product of the three terms: over-all mean (μ), row-specific (α_i) and column-specific (β_j) means:

$$a_{ij} = \mu \times \alpha_i \times \beta_j.$$

The spectral biclustering is based on a singular value decomposition (SVD) of the normalized data matrix. If the constants in an eigenvector can be sorted to produce a step-like structure, the column clusters can be identified accordingly. The left and right eigenvectors corresponding to the largest eigenvalue are expected to provide the optimal clustering of rows and columns. In the next step, the data is projected on the best two or three eigenvectors and k-means clustering (Hartigan and Wong, 1979) is run to get the grouping. The algorithm is implemented in R and is available as the method `BCpectral()` in the `biclust` package.

Parameters

Before running Spectral biclustering, the normalization procedure, the number of eigenvectors to use for the projection, and the number of clusters should be specified. The number of eigenvectors depend on the way data have been normalized.

7.2.2.5 Factor Analysis for Bicluster Acquisition (FABIA)

Hochreiter *et al.* (2010) proposed a sparse factor analysis method, FABIA, for the discovery of multiplicative biclusters. The assumption of sparseness comes from the gene expression data, where normally only a small fraction of genes is active under a small subset of conditions.

A factor model for the expression matrix $\boldsymbol{A}$ with p factors can be described as in (7.3):

$$\boldsymbol{A} = \sum_{k=1}^{p} \boldsymbol{\lambda}_k \boldsymbol{Z}_k + \boldsymbol{\varepsilon}, \tag{7.3}$$

where additive random noise is assumed to be normally distributed, $\boldsymbol{\varepsilon} \sim N(0, \boldsymbol{\Psi})$, $\boldsymbol{Z}_k$ represents the kth factor and $\boldsymbol{\lambda}_k$ is the vector of the factor loadings for $\boldsymbol{Z}_k$. This model assumes that $\boldsymbol{\Psi}$ is a diagonal matrix, i.e., the error terms $\boldsymbol{\varepsilon}$ are independently and normally distributed given the p factors in the model. Furthermore, $\boldsymbol{Z}$ and $\boldsymbol{\Psi}$ are assumed to be independent which implies that the noise is independent of the signal strength.

Using the factor model, biclusters can be obtained in the following manner. If $\boldsymbol{\lambda}_k$ is *sparse*, i.e., has many zero components, and $\boldsymbol{Z}_k$ is sparse, then the component $\boldsymbol{\lambda}_k \boldsymbol{Z}_k$ can be viewed as a bicluster. Observations that are non-zero and samples that are non-zero are members of the jth bicluster. Hence, specifying the number of factors is equivalent to specifying the number of biclusters.

FABIA is implemented in R and is available as the `fabia` package on Bioconductor.

Parameters

The sparseness should be specified, which depends on the noise level in the data, the factor loadings, and the size of the dataset. The number of factors or biclusters is an important parameter, which can be set to the upper boundary. Further, the normalization parameters should be specified and if there are restrictions on a number of biclusters where a gene can belong to.

7.2.2.6 Iterative signature algorithm (ISA)

Iterative signature algorithm or ISA looks for the multiplicative biclusters or transcriptional modules in the data (Ihmels *et al.*, 2004). Similar to FABIA, it assumes that the data can be decomposed in a set of modules, where each row or column has its score, similar to a factor loading and a factor score. However, in contrast to FABIA, where all factors are updated simultaneously, it looks for one bicluster at a time and repeats its search iteratively.

The starting point of ISA is initialization of the gene score vector with some values, e.g., based on the available pathway or a gene set of interest. Then, the scores for rows and columns are updated until the values within a given bicluster are coherent. In the last step, a threshold is applied on the scores vector to obtain row and column membership in biclusters.

ISA is implemented in the R package `isa2` and is available on CRAN.

Parameters

To run ISA, a matrix of seeds should be given, together with the threshold vectors for rows and column. The number of columns in the matrix of seeds is equal to the number of biclusters. The method of normalization of the data can be specified as well.

7.3 Statistical aspects of biclustering analysis

The biclustering methods discussed in Section 7.2 constitute a very small fraction of the methods proposed in the literature. In this dissertation we focus on a systematic comparison between the performance of several biclustering methods and propose new diagnostics tools for biclusterig solutions. In what follows we give more details about the research questions addressed in this part of the dissertation.

7.3.1 Robust biclustering solution

An important property of a biclustering method is its robustness to the random initialization. As described earlier, many methods, such as Plaid, ISA, Spectral biclustering, QUBIC, xMOTIFS and others use random seeds to start the bicluster search. Since we can view any biclustering solution as an approximation of a true bicluster set in the data, different initializations often lead to different results.

To make a biclustering procedure robust, ensemble methods, which have been applied in classification area, can be involved. The general idea is that a number of parameters or initialization seeds are chosen and a given biclustering method is run on data multiple times. Hanczar and Nadif (2012) have developed an ensemble biclustering method which in essence clusters the binary output of multiple biclustering runs.

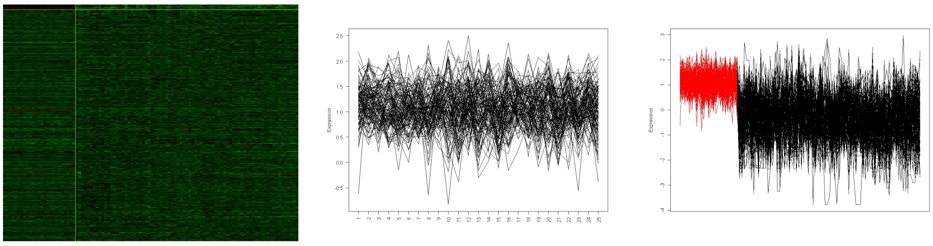
An alternative way is to asses the similarity of all biclusters in the output and to perform the clustering of biclusters based on the similarity measure of choice.

Robustness of the biclustering solution is discussed in Chapters 8 and 9 in the dissertation. In Chapter 8 we present the results of a simulation study conducted to investigate the robustness property of a biclustering solution. In Chapter 9 we describe in detail how the similarity of biclusters can be computed and how the clustering is performed on multiple runs of biclustering together with the description of a software package to perform the analysis in R.

7.3.2 Diagnostics of a biclustering solution

Despite of a large number of biclustering methods, the diagnostics of a biclustering solution has not received a lot of attention in data mining and statistical literature.

There were several attempts to use internal indices to evaluate the quality of a bicluster, such as MSR and average residue (Lee *et al.*, 2011). This type of measures is well-suited for additive biclustering methods and are less specific for other types of biclusters, including coherent evolutions and multiplicative biclusters. Chia and Karuturi (2010) used linear models to develop scoring methods independent of the



(a) Heatmap of the dataset with identified bicluster (b) Gene profiles within bicluster samples (c) Bicluster gene profiles across all samples

Figure 7.4: Illustration of the differential co-expression framework.

biclustering algorithm and data structure. Their approach is based on the differential co-expression framework of Kostka and Spang (2004). In practice, analysis of differential expression would look for genes under- or over-expressed with respect to the conditions outside the biclusters (difference in the first moments between two groups of conditions). The differential co-expression, however, looks at the difference in the second moments, i.e., variances of gene expression values within and outside bicluster. It is related to the other type of internal indices mentioned in Lee *et al.* (2011), where similarity of values within bicluster is compared to similarity of values outside a bicluster.

Figure 7.4 illustrates the differential co-expression framework: the heatmap of a synthetic dataset with a bicluster is shown in Figure 7.4(a), gene profiles for the arrays within bicluster are plotted in Figure 7.4(b), the expression levels inside the bicluster (red lines) are compared with the expression levels of the same genes outside the bicluster (black lines) in Figure 7.4(c). In this example, there is also a difference in the first moments, i.e., means. However, due to the larger variability outside the bicluster, these genes would not be considered differentially expressed. The other important aspect of biclustering is the type of bicluster. For the additive as well as the multiplicative models different types of biclusters include constant values, constant rows or columns, coherent values, and coherent evolutions. Hence, there is a clear need for a tool that can infer about bicluster structure based on the expression values.

In Chapter 10 we present the concept of differential co-expression and the derived ranking and stratification scores. Afterwards, we propose a number of scores based on a two-way ANOVA model and bootstrapping procedure for ranking and stratification of biclusters based on their co-expression patterns.

7.4 Case studies for biclustering

The illustration of the methodology discussed in this part of the dissertation uses the Affymetrix microarray data. We consider three gene expression data sets, which have been provided by the Broad Institute and were previously analyzed in Hochreiter *et al.* (2010) and Hoshida *et al.* (2007).

7.4.1 Dutch breast cancer dataset

The breast cancer data set (van 't Veer *et al.*, 2002) aimed at discovering a predictive gene signature for the outcome of a breast cancer therapy. The four distinct groups of patients were included in the study: 34 patients who developed distant metastases within 5 years, 44 patients who were disease-free after the period of 5 years, 18 patients with BRCA1 germline mutations and 2 from BRCA2 carriers. In van 't Veer *et al.* (2002) several gene signatures were found for patients from different groups, which can be viewed as biclusters. On the other hand, Hoshida *et al.* (2007) found three candidate subclasses related to BRCA1 mutation, lymphocytic infiltration and early occurrence of distant metastasis. For the biclustering analysis the dataset was preprocessed according to the guidelines in Hochreiter *et al.* (2010) and resulted in the expression matrix with 98 samples and $1,213$ genes.

7.4.2 Diffuse large-B-cell lymphoma (DLBCL)

DLBCL data set (Rosenwald *et al.*, 2002) aimed at predicting the survival after chemotherapy. The authors identified three cancer subgroups based on gene-expression: germinal-center B-cell-like, activated B-cell-like, and type 3 diffuse large-B-cell lymphoma. These subgroups were related to the survival rate of the patients. Hoshida *et al.* (2007) discovered three subtypes according to relevant molecular mechanisms. The aim of biclustering methods is to discover these subgroups with corresponding gene signatures. The expression matrix after preprocessing consists of 180 samples and 661 genes (Hochreiter *et al.*, 2010).

7.4.3 Pilot colon cancer data set

The goal of the pilot project in colon cancer cell line was aimed to investigate the compound induced gene expression. The compounds in the study were selected based on the phenotypic screen and known to be active in colon cancer treatment. These compounds had chemical structure similar to the kinase inhibitors, however coming from a number of different classes determined by the chemists. From the experiment data, the fold changes are obtained with respect to the DMSO (control treatment, when only vehicle without a compound is applied to the cell culture). The data contains 241 compounds and $2,289$ genes after I/NI and fold-change based filtering.

Chapter 8

Comparison between biclustering methods: a simulation study

In order to evaluate and to compare the performance of the biclustering algorithms discussed in the previous chapter in terms of robustness to noise, initialization parameters and the type of signal, a simulation study was carried out.
This chapter starts with motivating examples in Section 8.1. In Section 8.2 the simulation setting will be introduced. The quality evaluation of each biclustering solution as used in this study is discussed in 8.3. Section 8.4 presents results of the simulation followed by Section 8.5 discussing parameter setting of the biclustering algorithms. This chapter is concluded by a general discussion in Section 8.6.

8.1 Motivating examples

We use the Breast Cancer dataset for illustration how different biclustering methods obtain various biclusters in the data. The summary of discovered biclusters by Plaid, Spectral biclustering, xMOTIF, FLOC, δ biclustering, BiMAX and FABIA is given in Table 8.1. Spectral biclustering has discovered three biclusters of different size compared to the three biclusters of Plaid. xMOTIF did not discover any biclusters in the Breast cancer data, whereas δ biclustering discovered five biclusters. FLOC, an extension of δ biclustering, has discovered five biclusters. BiMax has acquired five biclusters on a binary version of the Breast cancer data (the threshold for absolute values was set to 0.75), which had considerably smaller size compared to the biclusters of Plaid, Spectral biclustering, or δ biclustering. FABIA has estimated five factors in the data and after applying the threshold of 1.5 to factor scores, five non-empty biclusters were obtained.

Table 8.1: The biclusters discovered by seven different methods in the Dutch Breast Cancer data, k is the number of biclusters discovered by a method.

	Number of rows	Number of columns
Plaid, `model=m+a+b`, $k = 3$		
1	60	28
2	38	21
3	6	18
Spectral, $k = 3$		
1	19	26
2	19	22
3	19	22
δ biclustering, $\delta = 0.1, k = 5$		
1	137	44
2	67	34
3	49	27
4	27	27
5	27	20
FLOC, $k = 5$		
1	192	26
2	175	30
3	208	29
4	192	29
5	191	37
BiMAX, $k = 5$		
1	6	5
2	6	5
3	6	5
4	6	5
5	6	5
FABIA, $k = 5$		

Continued on next page

Table 8.1 – continued from previous page

	Number of rows	Number of columns
1	226	7
2	352	5
3	379	18
4	220	11
5	176	7

Several conclusions can be drawn from these results. First, biclusters from different methods vary in size. Second, while some biclustering methods like Plaid, BiMAX, xMOTIF, or Spectral biclustering have a stopping criterion and report fewer biclusters than a maximum number of biclusters specified, FABIA and δ biclustering report as many biclusters as they have been requested to extract. Third, some biclustering methods give highly overlapping biclusters (FLOC, BiMAX, Spectral biclustering), which should be refined before the biological interpretation. In the end, there are some samples and genes, which appear in different biclusters across all methods for the given dataset, however, there are more of those samples and genes, which are exclusively found to belong to a bicluster discovered by one or another method.

Since the underlying biclustering structure is usually not known in the novel datasets, we need to evaluate biclustering methods by means of some benchmark data. A simulation study, which has been carried out within our research project, is presented below.

8.2 Simulation setting

The aim of the simulation study in this chapter is to investigate:

- sensitivity of bicluster solutions to the signal/noise levels;
- ability of biclustering methods to discover different types of biclusters;
- robustness of biclustering methods to the random noise and initialization seeds.

We have chosen seven methods for inclusion in the simulation study: Plaid, BiMAX, xMOTIF, δ-biclustering, Spectral biclustering, FLOC, and FABIA.

8.2.1 Dimensionality of simulated data

In every simulation setting, the dimensionality was set to 500 rows and 50 columns, which is substantially lower than for the datasets generated by recent microarary experiments. However, it is an important step before considering higher dimensions, as methods failing in lower dimensions are highly unlikely to perform well when the number of rows and columns are scaled to thousands. Moreover, some of the methods like δ-biclustering have been developed in the very beginning of gene expression data and are not expected to scale well to matrices with thousands of rows and columns.

8.2.2 Simulated biclusters

We have chosen six arbitrarily positioned overlapping biclusters of different sizes. Three types of biclusters were constructed: constant values, constant columns, and coherent values. The generation model for each bicluster is listed in Table 8.2.

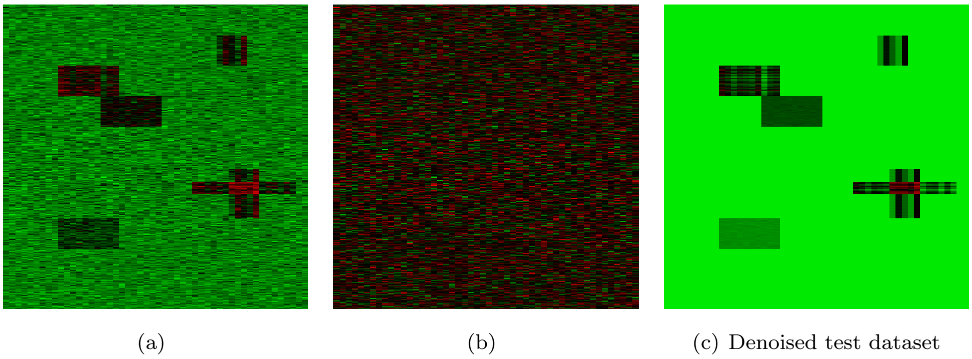
The values within biclusters were set to be both up- and down-regulated with respect to the background. The variance of 0.1 within a bicluster reflects variability of gene expression values for the bicluster samples.

Table 8.2: Models for simulating signal in biclusters.

bicluster type	size	model
constant	50×10	$a_{ij} \sim N(2, 0.1)$
constant	50×10	$a_{ij} \sim N(6.5, 0.1)$
constant columns	70×5	$\beta_j \sim N(1, 1)$
		$a_{ij} \sim N(-3 - \beta_j, 0.1)$
constant columns	50×5	$\beta_j \sim N(1, 1)$
		$a_{ij} \sim N(5 + \beta_j, 0.1)$
coherent	20×17	$\alpha_i \sim N(1, 1); \beta_j \sim N(1, 1)$
		$a_{ij} \sim N(7 + \alpha_i - \beta_j, 0.1)$
coherent	50×10	$\alpha_i \sim N(1, 1); \beta_j \sim N(1, 1)$
		$a_{ij} \sim N(2 + \beta_j, 0.1)$

8.2.3 The background noise

The additive noise was modeled through normally distributed values with mean 0 and variance 0.5. An example of a dataset with implanted biclusters over the Gaussian noise is given in Figure 8.1.


(a)

(b)

(c) Denoised test dataset

Figure 8.1: A heatmap of one of the simulated datasets: (a) test dataset with noise, (b) additive noise samples from normal distribution, (c) simulated signal.

8.2.4 Number of iterations

In order to evaluate the overall performance of the methods and their robustness to different noise values, 100 data sets have been simulated. For the investigation of the initial values effect, a simulation study consisting of 100 datasets was conducted, where for each dataset, the algorithms were run 100 times.

8.3 Quantifying performance of biclustering methods

8.3.1 Similarity indices

To check how close a discovered bicluster is to the original one, we need to compute a similarity in terms of their elements, i.e., by checking how many of the originally implanted genes and conditions have been discovered. There are several ways to quantify the goodness of overlap: (1) sensitivity - a number of genes and conditions discovered from the original bicluster with respect to the number of genes and conditions

in the original bicluster, (2) specificity - a number of genes and conditions discovered from the original bicluster with respect to the number of genes and conditions in the discovered bicluster, and (3) F_1 measure or the Jaccard coefficient, which is a harmonic mean of sensitivity and specificity (a number of genes and conditions shared by discovered and simulated bicluster with respect to the total number of genes and conditions in both biclusters) (Turner *et al.*, 2005; Prelic *et al.*, 2006).
Let g_X be the number of genes in a bicluster X, s_X be the number of samples in the bicluster, and $n_X = g_X \cdot s_X$, then

$$sensitivity = \frac{g_{A \cap B}}{g_A} \times \frac{s_{A \cap B}}{s_A},$$

$$specificity = \frac{g_{A \cap B}}{g_B} \times \frac{s_{A \cap B}}{s_B},$$

$$F_1 = \frac{g_{A \cap B} s_{A \cap B}}{\sqrt{g_A s_A g_B s_B}}.$$

Sensitivity measures the proportion of data values in B that are also in the target bicluster A. The specificity is the proportion of A that has been retrieved in B.

8.3.2 Comparison of multi-sets

While conducting the pairwise comparison of biclusters is a straightforward task, so that comparison of biclustering algorithms discovering one bicluster in the data with a single simulated bicluster can be done just by calculation of one index per pair, the comparison of multiple biclusters in the output with a single one becomes a more complex task. There is an extra complexity when more than one bicluster is created and more than one bicluster is discovered. Several strategies can be used to evaluate a biclustering solution as a whole.
Following Turner *et al.* (2005), Prelic *et al.* (2006) proposed to take the average of a maximum of quality measures. Let K be the set of p target biclusters and R be the set of retrieved biclusters by an algorithm. Then, for a given quality measure f,

$$f(R) = \frac{1}{p} \sum_{K} \max_{b \in R} f(b).$$

If the simulated biclusters have been fully identified by a method, then $f(R) = 1$. In general, $f(R) \in [0, 1]$. In the calculation of $f(R)$, no one-to-one assignment of the biclusters from the discovered set to the biclusters of the simulated set is performed. That is, if a method discovers one large bicluster containing two simulated ones, it will contribute to the sum of the maxima twice.

8.4 Simulation results

8.4.1 Robustness to the random noise

After 100 iterations with a fixed random seed for the starting values of the algorithms, the quality measures were obtained and plotted for each type of the bicluster. In this section, we present the results for coherent biclusters (Figure 8.2). The results for constant and constant columns biclusters are similar to the results of coherent bilcusters and, therefore, are not presented here.

On average, specificity of Plaid and δ biclusters was higher than the sensitivity, which implies that the discovered biclusters were smaller in terms of the number of rows and columns than the true ones. In contrast, BiMAX and FABIA had higher sensitivity implying that the discovered biclusters were larger than the true ones. xMOTIF was not able to discover biclusters and FLOC had a very low sensitivity and specificity. Plaid had the best average performance for coherent biclusters.

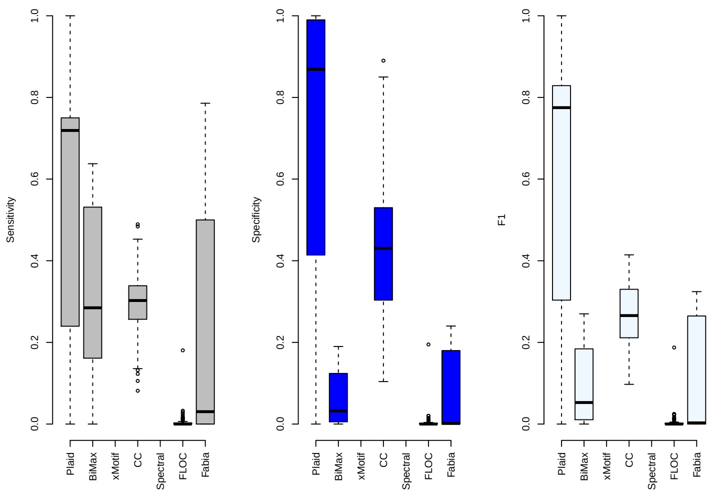


Figure 8.2: Quality measures (seed fixed, data different) for the seven methods on coherent biclusters.

In general, most of the methods give variable results depending on the noise. The most variable methods were FABIA and Plaid. Therefore, in a real data setting, denoising is a very important step to start from.

8.4.2 Robustness to the initial seed

Noise coming from random data is unavoidable in real life. It is not surprising to observe variability in the detection of target biclusters when the data are different. However, an additional problem of the majority of heuristic approaches is that they use a random seed for initialization. As a result, a recovery of true biclusters can vary with the random seed. Figure 8.3 displays the average F_1 measure together with its standard error based on results from 100 datasets. The complete overview of simulation results is presented in Table 8.3.

Table 8.3 summarizes the results of the simulation study per type of the simulated biclusters. As can be observed from Figure 8.3 and Table 8.3, the average similarity indices of Plaid and FABIA are the largest. FLOC, δ-biclustering, and xMOTIFs have small values for all bicluster types while Spectral biclustering did not find any biclusters in the simulated datasets (therefore, no results are available for the Spectral biclustering).

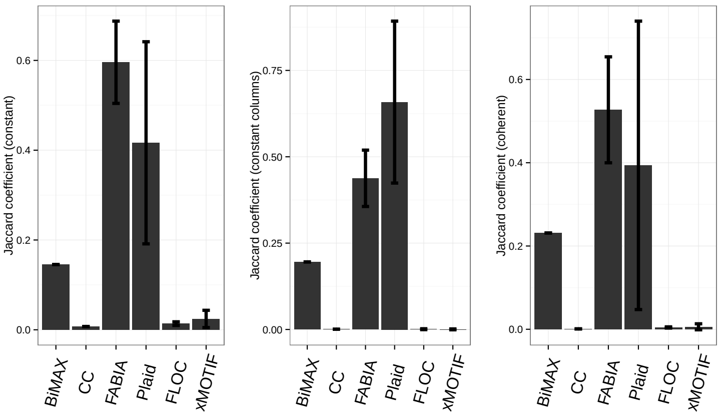


Figure 8.3: Average trend and standard error of F_1 (or Jaccard coefficient) for the biclusters discovered in each simulated dataset according to the type of simulated biclusters: (a) constant values, (b) constant columns and (c) coherent values. The results are based on 100 simulated data sets.

Apart from the average trends, variability should be taken into account. BiMAX and δ biclustering showed the same results regardless of the random seed initialization. The BiMAX implementation uses an exact algorithm, which is more dependent on how binarization occurred. δ-biclustering is a deterministic algorithm, which depends more on the value of δ, rather than the starting values. FLOC and xMOTIFs had

Table 8.3: Results of the simulation study, based on 100 data sets, 100 iterations each (mean - is the average performance for all generated datasets, together with the standard error; SD is the average standard deviation for 100 data sets based on 100 initial values, with the standard error).

	Constant			Constant columns			Coherent		
	spe	sen	F_1	spe	sen	F_1	spe	sen	F_1
BiMAX									
mean	0.21	0.13	0.15	0.27	0.19	0.196	0.42	0.17	0.23
(s.e.)	(0.21)	(0.18)	(0.18)	(0.29)	(0.28)	(0.25)	(0.21)	(0.09)	(0.13)
SD	0	0	0	0	0	0	0	0	0
(s.e.)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
FABIA									
mean	0.66	0.58	0.59	0.53	0.38	0.44	0.64	0.49	0.53
(s.e.)	(0.02)	(0.04)	(0.03)	(0.02)	(0.018)	(0.018)	(0.09)	(0.04)	(0.06)
SD	0.06	0.1	0.09	0.096	0.083	0.082	0.21	0.1	0.13
(s.e.)	(0.021)	(0.029)	(0.028)	(0.043)	(0.033)	(0.034)	(0.04)	(0.02)	(0.03)
Plaid									
mean	0.48	0.42	0.416	0.72	0.64	0.66	0.45	0.36	0.39
(s.e.)	(0.03)	(0.03)	(0.03)	(0.033)	(0.027)	(0.029)	(0.04)	(0.035)	(0.038)
SD	0.31	0.22	0.23	0.28	0.22	0.23	0.4	0.32	0.35
(s.e.)	(0.02)	(0.02)	(0.019)	(0.018)	(0.022)	(0.02)	(0.014)	(0.015)	(0.014)
δ biclustering									
mean	0.05	0.012	0.007	0.005	0.001	0.0007	0.0007	0.001	0.0009
(s.e.)	(0.005)	(0.009)	(0.006)	(0.002)	(0.006)	(0.003)	(0.001)	(0.003)	(0.002)
SD	0	0	0	0	0	0	0	0	0
(s.e.)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)
FLOC									
mean	0.0007	0.011	0.014	0.0004	0.008	0.0008	0.0002	0.032	0.004
(s.e.)	(0.0046)	(0.0091)	(0.006)	(0.0049)	(0.0055)	(0.0028)	(0.0014)	(0.003)	(0.0019)
SD	0.002	0.04	0.0041	0.007	0.013	0.013	0.0008	0.01	0.001
(s.e.)	(0.00026)	(0.005)	(0.0004)	(0.0001)	(0.0023)	(0.0002)	(0.0001)	(0.002)	(0.0002)
xMOTIFs									
mean	0.04	0.026	0.024	0.001	0.0002	0.0003	0.011	0.006	0.006
(s.e.)	(0.02)	(0.03)	(0.019)	(0.034)	(0.026)	(0.029)	(0.009)	(0.007)	(0.007)
SD	0.027	0.023	0.019	0.004	0.0008	0.001	0.014	0.008	0.007
(s.e.)	(0.0096)	(0.015)	(0.009)	(0.005)	(0.001)	(0.002)	(0.007)	(0.006)	(0.004)

low variability with the poor performance in the current simulation setting. FABIA had low variability coupled with high average performance. Plaid had the largest variability of the results. The standard deviation for the hundred data sets was on average around 0.3. It implies that this method heavily depends on the starting values. Turner *et al.* (2005) mentioned that start-up values affected the results, however, since they did not run the algorithm with various random seeds, no conclusion about how variable the results were, was made.

8.4.3 Recovery of various types of biclusters

Table 8.3 reveals several important patterns:

- xMOTIF has hardly detected the true biclusters, with relatively larger values of similarity indices for constant biclusters.
- Plaid showed on average higher performance on biclusters with constant columns. The coherent and constant values biclusters were detected with large sensitivity and specificity.
- FABIA outperformed Plaid in terms of sensitivity and specificity on the coherent and constant values biclusters. Therefore, despite the fact that FABIA is designed for multiplicative coherent structures, it can determine all three simulated bicluster types.
- FLOC had comparably low performance as the δ-biclustering algorithm.
- BiMAX specificity was the highest for coherent values biclusters, couples with a moderate sensitivity. It suggests, that BiMAX outputs smaller biclusters with the elements form the true biclusters. The small size of BiMAX biclusters was observed for the real-life data sets as well.

8.5 Sensitivity of the bicluster solution to the choice of the parameter settings

The choice of parameters is an important step in order to gain specificity and sensitivity of a biclustering method. In most situations, parameters are data-dependent, thus, extra tuning of each method is necessary to make these methods comparable. It could be due to the wrong settings that methods like xMOTIF, FLOC or Spectral could not discover underlying patterns in the simulated data. In this section we discuss the possible impact of parameter settings on the performance of the different biclustering methods discussed in the chapter.

BiMAX

BiMAX algorithm works with the binarized version of the data. The results of the method would be in general affected by various thresholds. Normally, if the threshold is not specified, the median of the absolute values is used as a cut-off point. In the simulation setting of the current study, the threshold was set to 2, enabling the binarization procedure to distinguish between the noise and the data points from the biclusters. The other set of parameters, specifying the bicluster structure in the data, can also affect the result. The minimal number of columns and rows would affect the results by discovering more biclusters with fewer rows and columns. Furthermore, while the number of biclusters would add extra biclusters, it would not influence the structure of the bicluster sets contained in a larger set. In other words, an increase in number of biclusters to search would create a superset containing a set of lower order.

xMOTIF

Similarly to the BiMAX algorithm, xMOTIF does not work with the original version of the data, but with a discretized one. The special preprocessing step, specifying a number of discretization levels influence the results dramatically. In the current study, the quantile discretization parameter was set to 7. It could be the case that, in order to improve the performance of the method, more levels for discretization would be necessary. However, it is not the only parameter that influences the outcome. The number of randomly selected samples and the number of samples to join the xMOTIF could be not sufficient for eliminating dependency on the starting values. In any case, the failure of finding biclusters in the data by xMOTIF is subject to further investigation.

The δ biclustering

The quality of biclusters detected by δ-biclustering can vary with the parameter δ, which was set to 0.75 in the current study. This choice was driven by the major guidelines given in the original paper of Cheng and Church (2000). The authors used datasets with discovered clustering patterns in the previous research to determine the expected score. Additional investigation on the δ parameter was performed via permutations of the data matrix (Cheng and Church, 2000). In the current study, the value of δ was set to 0.75 since the score of implanted biclusters varied around this value depending on the realization of random sampling. For the overlapping biclusters it could be the case that the score was larger than δ so the biclusters could not be identified. It should be mentioned that in this complex situation the choice of parameters is a difficult task. Therefore, simplification of the original setting to dataset with one bicluster would be desirable in order to evaluate what is the range of δ-values that can be suitable for finding biclusters in the data.

FLOC

FLOC has comparable low performance as the initial δ-biclustering most probably due to non-optimal default settings of parameters. The minimal number of rows/columns to be included into bicluster was probably too high, as well as the number of biclusters, which was set to 20.

Plaid

As was already mentioned before, the instability of the Plaid algorithm could be caused by several parameters, such as the number of iterations for the start-up values, the number of back-fitting iterations, and the number of iterations within each layer. However, Turner *et al.* (2005) did not use high values for iteration settings and still drew major conclusions about the performance of their method. Even though the method is suitable for complex structures and would reflect biological processes in its models, the variability caused by the starting values can deter user from making any definitive conclusions. An additional investigation was conducted by increasing the number of iterations (the results are shown in Figure 8.4). Even an increase in number of iterations for starting up layers, within a layer, and back fitting still did not help to decrease the variability within the same dataset.

Spectral biclustering

Spectral biclustering settings depend on the normalization method (we used the recommended log-normalization) and on the number of eigenvectors (set to 3) used for the projection of the data and clustering. In our simulation study, the Spectral biclustering algorithm did not detect any bicluster. Since Spectral biclustering is based on the singular vector decomposition of the data matrix, random noise could influence the decomposition to a large extent and, thus, cause the failure of the method to detect biclusters in the data. However, the major problem could be related to the k

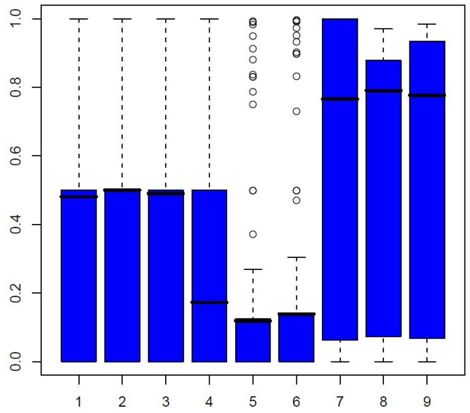


Figure 8.4: The performance of Plaid on one dataset after 100 runs when the number of iterations was increased (1-3: Sensitivity, Specificity, F_1 measure for constant values; 4-6: Sensitivity, Specificity, F_1 measure for constant columns; 7-9: Sensitivity, Specificity, F_1 measure for coherent values). The increase in numbers did not reduce the variability of the method.

means clustering used in the post-projection step. This clustering method is known to be unstable in a presence of noise. Moreover, the Spectral biclustering assumes that the data can be partitioned in the checkerboard-like structure, which is not the case in this simulation setting.

FABIA

FABIA is a promising method in the field of bicluster analysis since it can discover a wide range of bicluster types. The major parameter of interest is the number of factors (biclusters). However, the bicluster size is not restricted and depends on a threshold. Varying the threshold t_Z for factors around the default value of 0.5 ($t_Z \in [1; 2]$) did not affect the performance at all, whereas a threshold for loadings perturbed the results to a minor extent. From the prior investigation, it has been observed that the thresholds do not play the most important role in discovery of true biclusters since they are used as an ad-hoc procedure.

8.6 Discussion

We have demonstrated the differences between seven biclustering methods by applying them to the Dutch breast cancer dataset. We have observed that the discovered biclusters vary in size both in terms of the number of genes and the number of samples. Furthermore, the number of resulting biclusters varied per method. It should be expected, since the biclustering methods in this study are determined to find different structures. For example, the BiMAX works on the binary version of the data and would discover only biclusters of genes which are passing the threshold. The Dutch breast cancer dataset is sparse, containing low number of expression values which are above 1.5, thus the resulting biclusters are in general smaller. On the other hand, methods, which allow coherent values, discover larger structures.

The variability in biclustering solutions have motivated us to carry out a simulation study to evaluate how biclustering methods perform when a given bicluster is constant, coherent, or has constant values structure. We have observed that the coherent biclusters are better recognized by Plaid and FABIA.

The other important evaluation setting was related to the evaluation of the biclustering algorithms with respect to the different initialization seeds. We have seen that, in addition to the variability coming from the random noise, extra variability was induced from the initialization parameters. In case of Plaid, the variability was the highest. It implies that for the highly variable methods in terms of initialization, ensemble techniques can provide robust results. The ensemble method for discovery of robust biclusters is discussed in Chapter 9.

The parameter setting for each simulation task is a challenge. The optimization of the parameters for biclustering algorithms for the simulated data sets is a topic of future research.

Chapter 9

Software development for obtaining robust biclustering results

Stability of a blustering solution is a central issue which highly influences the ability to interpret the results of a biclustering analysis. Filippone *et al.* (2009) argued that the stability of biclusteting algorithms can be affected by initialization, parameter settings, and perturbations such as different realization of random noise in the data set. The initialization affects mostly the algorithms relying on local search procedures. In such cases, the outcome of a biclustering procedure is highly dependent on the choice of initial values. Most of the algorithms try to overcome this drawback by using a large number of initial values (seeds) and then finding the best biclusters in the final output set (Murali and Kasif, 2003; Bergmann *et al.*, 2003; Shi *et al.*, 2010). This procedure does not guarantee an optimal solution of biclustering problem, though. In practice, running a chosen biclustering algorithms several times on the same dataset will give a different output. It remains still up to the analyst to chose the most reliable biclusters or the most robust to the specification of initial values, as well as the parameters.

Shi *et al.* (2010) introduced a novel procedure for obtaining robust biclusters from a set of initial values. The idea of authors is appealing and very intuitive and can be applied to any biclustering method, which provides 'hard' biclusters, i.e., biclusters where membership is a binary indicator. Shi *et al.* (2010) introduced the 'super'-biclusters concept, which can be obtained from the hierarchy of the biclusters. To construct the hierarchy of biclusters, Shi *et al.* (2010) proposed to use the Jaccard

index.
In this chapter we introduce the R package `superbiclust` which has been developed in order to perform a robust bicluster analysis. The `superbiclust` can perform a robust analysis of R objects produced by the biclustering R packages `biclust`, `fabia` and `isa2`, as well as any other biclustering output that gives a number of biclusters and row/column membership for a given bicluster.
We start this chapter with a motivating example in Section 9.1. The algorithm used for the robust analysis and the architecture of the package are described in Section 9.2. The practical implementation in R is illustrated in Section 9.3. The results are discussed in Section 9.4.

9.1 Motivating example

As mentioned above, stability of a biclustering solution is one of the two major issues related to the interpretation of the results. For illustration, let us consider the Plaid model proposed by Turner *et al.* (2005). The Plaid model requires initialization of binary bicluster memberships. Even though several attempts were performed in order to stabilize the initialization, this implementation turned out to give unstable results when multiple runs were considered.
We run the Plaid model on the colon cancer dataset. The model was run 50 times with the same parameter setting. Figure 9.1 shows that the number of biclusters per run discovered by the Plaid model varied from 3 to 23, with the most runs resulting in about 10 biclusters.
Nevertheless, Turner *et al.* (2005) argued that Plaid was able to pick up important structures. This statement was supported by the simulation study discussed in the previous chapter.

9.2 Algorithm and implementation

The following algorithm is implemented in the R package `superbiclust` for a robust analysis:

1. Run the biclustering algorithm with N different starting values. Output: the list of obtained binary memberships for rows and columns.

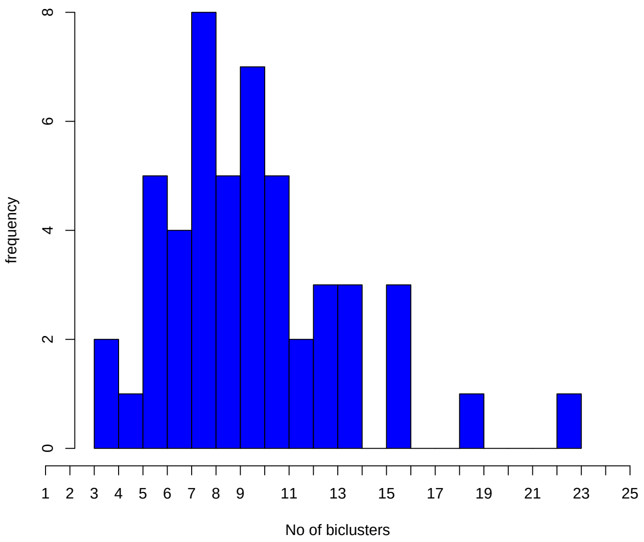


Figure 9.1: Distribution of number of biclusters per run discovered in 50 runs of the Plaid model on the colon cancer data. Parameter setting is fit.model = y   m + a + b, row.release = 0.75, col.release = 0.7, shuffle = 50, back.fit = 5, max.layers = 50, iter.startup = 100, iter.layer = 100.

2. Construct a similarity matrix, using all pairwise comparisons between discovered biclusters.
3. Calculate the hierarchy of the biclusters based on the similarity matrix.
4. By selecting a cut-off height, cut the dendrogram and obtain classes.
5. For each class, make a prototype bicluster, taking either union of all elements or an intersection.

The package contains methods for obtaining distance function based on the similarity indices, constructing the hierarchical clustering of biclusters and obtaining robust biclusters. The architecture of the R package `superbiclust` is shown in Figure 9.2. The details of the implementation are given below.

9.2.1 Distance function

In theory, any of the similarity indices can be used to construct the distance measure. Hallez *et al.* (2009) mentioned three types of comparison indices for sets as well as multisets. Let A and B be two sets of objects (two biclusters, two sets of rows and column, sets of biclusters etc.). The notation that is used in this section corresponds to the notation in discrete mathematics, where $\bar{A}$ is a complement to the set A, $A\bar{\Delta}B$ is a complement to the symmetrical set difference and $|A|$ is the number of elements in set A. We use a function $g(\cdot)$, which is a non-decreasing function that maps the set of objects to the interval $[0;1]$. In addition we denote by $f(\cdot)$ a commutative mapping to $[0;1]$, satisfying $f(A,B) \geq g(A \cap B)$. For more details about the properties of the mapping $g(\cdot)$ and $f(\cdot)$ in a general setting we refer to Hallez *et al.* (2009).

1. Inclusion indices. The inclusion indices $I(A,B)$ indicate to which extent one set contains another. The most general form is given by

$$I(A,B) = \frac{g(\bar{A} \cup B) - g(\bar{A})}{1 - g(\bar{A})}. \tag{9.1}$$

   Examples of inclusion indices are sensitivity or specificity. As Hallez *et al.* (2009) mentioned, the inclusion indices can be used whenever it is sufficient that the set belonging to one object is a part of the set belonging to another object. For example, if we are interested in the extent to which the original bicluster is contained in a discovered bicluster or vice versa.

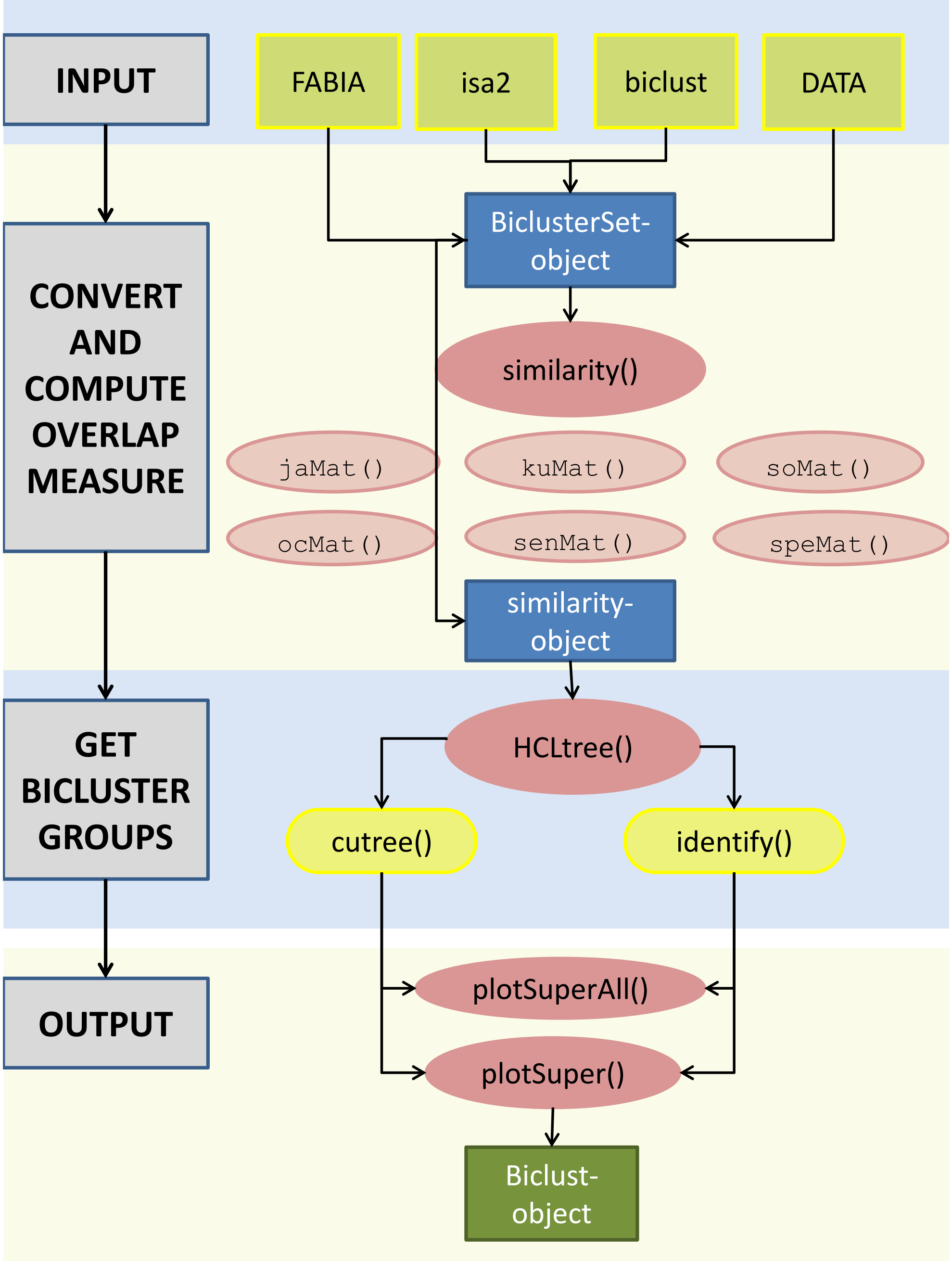


Figure 9.2: Architecture of `superbiclust` R package and the analysis workflow for obtaining robust biclusters or studying structure of biclustering output. External input and output objects are given in green blocks. Internal objects (S4 and S3) are drawn in blue blocks. External functions are given in yellow rounded blocks. Internal main functions are in red ovals and the help functions are in light shaded red ovals.

2. Partial matching indices. The partial matching indices are symmetrical indices that evaluate the intersection of two sets:
$$PM(A,B) = \frac{g(A \cup B)}{f(A,B)}, \tag{9.2}$$
where $f(\cdot)$ should satisfy certain axiomatic condition. As an example of such a function, $f(A,B) = min(|A|,|B|)$ can be used. Partial matching indices are useful in object matching if a minimal overlap between many valued attributes is required. For biclustering output it does not have a straightforward application, because we are interested in maximal overlap.

3. Similarity indices. The similarity indices evaluate the symmetrical difference between two sets. There are three types of these indices. The first type of the indices is based on $g(A\bar{\Delta}B)$ and is given by
$$S(A,B) = \frac{g(A\bar{\Delta}B) - g(A\bar{\cup}B)}{1 - g(A\bar{\cup}B)}. \tag{9.3}$$
The Jaccard index $Ja = |A \cap B|/|A \cup B|$ is an example of a similarity index of the first type, if $g(\cdot) = |\cdot|$ is a uniform probability measure (set cardinality).
The second type of similarity indices is based on a symmetrical function f of $g(\bar{A} \cap B)$ and $g(A) \cap \bar{B}$:
$$S(A,B) = \frac{f(g(A \cup \bar{B}), g(B \cup \bar{A})) - f(g(\bar{A}), g(\bar{B}))}{1 - f(g(\bar{A}), g(\bar{B}))}. \tag{9.4}$$
The Sorensen index given by
$$So = \frac{2|A \cap B|}{|A| + |B|}$$
and is an example of the second type similarity index.
The third type of similarity indices is based on a symmetrical combination of an index given by
$$S(A,B) = h(I(A,B), I(B,A)), \tag{9.5}$$
where $h(\cdot,\cdot)$ is a commutative function satisfying $h(0,0) = 0$ and $h(x,y) = 1 \iff x = y = 1$. The Kulczynski index
$$Ku = \frac{1}{2}\left(\frac{|A \cap B|}{|A|} + \frac{|B \cap A|}{|B|}\right)$$
and the Ochiai index
$$Oc = \frac{|A \cap B|}{\sqrt{|A||B|}}$$
are similarity indices of the third type.

The similarity indices are equivalent to each other. However, some of them can attain maximum faster than the others, depending on the type and symmetric function that is used. Simulation was performed to illustrate the equivalence of the indices.

In all simulated situations, the similarity indices were monotonically increasing with the proportion of overlap between the two sets. The Jaccard index expressed the slowest growth with respect to the overlap, giving always the lower boundary of the similarity indices. Sensitivity and specificity could remain high or reach maximum when one of the sets was completely included in the other. The differences between similarity indices were observed when one of the set had much lower cardinality compared to the other one (Figures 9.3(a)-9.3(c)). This observation suggests that when the discovered bicluster is much smaller than the original bicluster, high differences between the values of similarity indexes are observed.

This in turn proves the concept of inclusion indexes. In the evaluation of biclustering methods, the inclusion indices can give a better idea to which extent the discovered bicluster is contained in the original one, or what is the proportion of the original bicluster discovered by the method. The similarity indices, on the other hand, provide additional information about the ratio of overlap with respect to the total number of elements contained in both biclusters. In any case, the higher the value of a similarity index, the more elements of the original bicluster were discovered by a method.

Since all similarity indices are equivalent, we can focus on the most widely used one, which is the Jaccard index. We define the distance measure between two biclusters A and B as

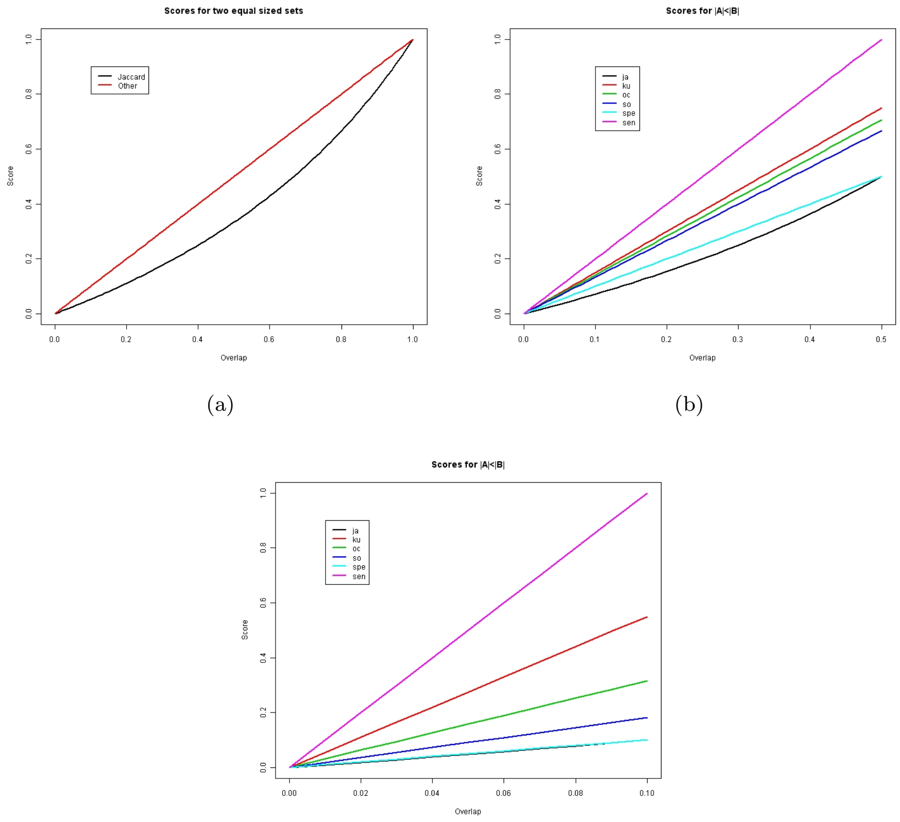
$$dist(A;B) = 1 - Jaccard(A;B). \tag{9.6}$$

It is evident that in case of a total overlap of all elements in two biclusters, the distance between them is 0. When no overlap is present, the distance is equal to 1.

9.2.2 Choice of the link function

There are several choices to make while constructing a hierarchical tree. When two biclusters are merged, the distance between them is used to calculate the height of the tree. When two sets of biclusters need to be merged, a link function, such as average link, single link, or complete link can be used. We choose complete link for merging. The complete-link distance between two clusters of biclusters C_i and C_j in a tree is the maximum distance between any object in C_i and any object in C_j. This distance is defined by the two most dissimilar objects,

$$dist(C_i;C_j) = \max_{A,B}\{dist(A,B)|A \in C_i, B \in C_j\}. \tag{9.7}$$



(a)

(b)

(c)

Figure 9.3: Various comparison scores in three different situations: (a) biclusters A and B have same number of elements, (b) the cardinality of a bicluster A is 10% compared to the cardinality of a bicluster B ($|A| = 0.1|B|$), (c) the cardinality of a bicluster A is 50% compared to the cardinality of a bicluster B ($|A| = 0.5|B|$).

9.2.3 Construction of a hierarchical tree and obtaining bicluster prototypes

The construction of a tree is performed by the R package `hclust`. The visualization function is also available within a forementioned package. After the tree is constructed, the user needs to select the cutting threshold for obtaining the bicluster prototypes. Default setting is 0.5, i.e., when biclusters share at least 50% of their elements. When the tree is cut, clusters are obtained and user gets a histogram that displays the distribution of cluster sizes. There is more interest in the classes with the highest number of biclusters. The reason for selecting clusters with the largest number of members is that regardless of the initialization, the algorithm is able to discover them in the dataset. These are so-called robust biclusters. Biclusters found with a low frequency are spurious and initialization-dependent. The classes with large number of biclusters are used to construct prototypes of the 'super'-biclusters. The prototype can be either the intersection of all elements, belonging to all biclusters within a class, or the union of them. Intersection of elements will result in a tight, core bicluster, since the elements are common for all biclusters in the class. On the other hand, union of the elements will result in larger 'super'-biclusters, which contain all elements (all genes and samples) that belong at least to one of them.

The same hierarchical tree can be used to obtain unique biclusters from any biclustering method. If the user is interested in obtaining only unique biclusters from the output of a given biclustering algorithm the tree should be cut at the first non-zero height of the tree. The prototype for a class of any given bicluster is any class representative, since the tree has been cut just above zero.

9.3 The R package `superbiclust`

9.3.1 Robust analysis for the Plaid model

The full Plaid model $\Theta_{ijk} = \mu_k + \alpha_{ik} + \beta_{jk}$ was applied to the Breast cancer data. The algorithm was run with 100 different seeds for initialization of starting values. Depending on the initialization, one to three biclusters where discovered in each run of Plaid. The results from the list of `Biclust` objects are combined by using the function `combine`.

```
>PlaidSetResult <-combine(PLAID[[1]], PLAID[[2]])
>for(i in 3:(length(PLAID)-1)){
>        tmp <- combine(PLAID[[i]], PLAID[[i+1]])
>        PlaidSetResult <- combine(PlaidSetResult,tmp)
}
>PlaidSetResult<- BiclustSet(PlaidSetResult)
>PlaidSetResult
An object of class BiclustSet
Number of Clusters found:  288
First  5  Cluster sizes:
                   BC 1 BC 2 BC 3 BC 4 BC 5
Number of Rows:      60   40    6   60   40
Number of Columns:   28   21   18   28   21
```

In total, 288 biclusters were found and stored in `PlaidSetResult`. At the next step we compute the Jaccard similarity matrix for the Plaid results in two dimensions by setting `type` parameter to `"both"`.

```
>JaMatr<- similarity(res,index="jaccard", type="both")
```

Afterwards, the hierarchy of biclusters is obtained by the function `HCLtree()` and plotted with `plot()`.

```
>PlaidBiclustTree <- HCLtree(JaMatr)
>plot(PlaidBiclustTree)
```

The dendrogram in Figure 9.4 shows that there are only three groups of biclusters, each of them are joined at height 0. Thus, the Jaccard index for all biclusters within group equals 1. It is also remarkable that the three groups are non-overlapping - they are all joined at the height of 1, with no common elements between groups. Based on this analysis it can be concluded that according to the Plaid model, there are three distinct biclusters in the data. To obtain the groups and the super biclusters we can either use the `cutree` function or the `identify` function.

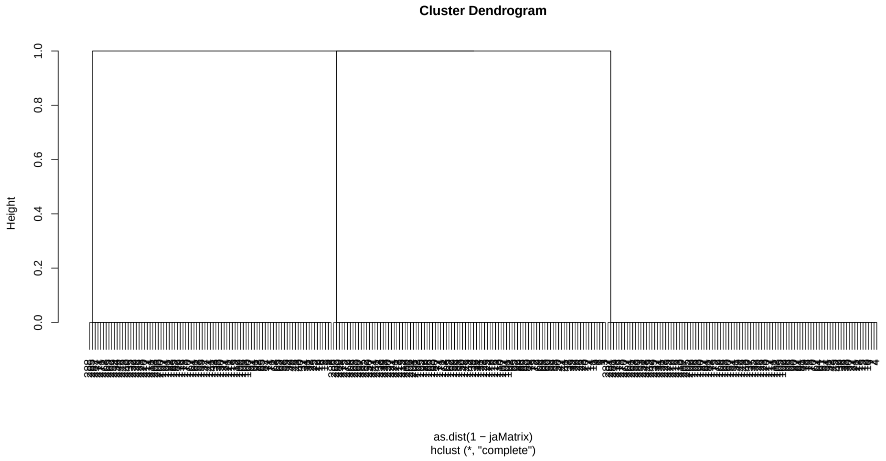


Figure 9.4: The hierarchy of biclusters generated with 100 different seeds by full Plaid model: the tree indicates three distinct non-overlapping groups of biclusters of Plaid.

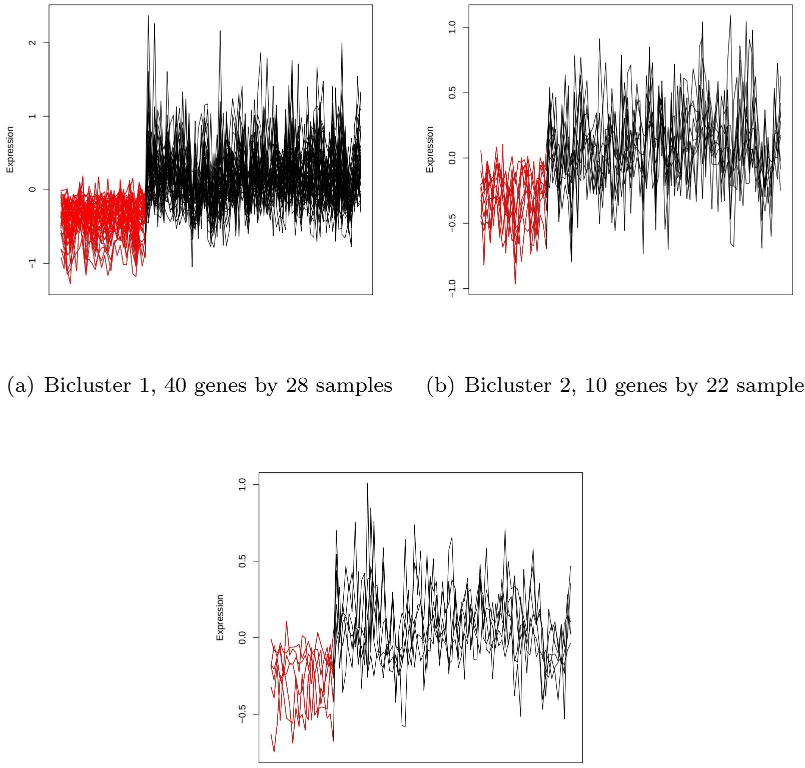
```
>Idx <- identify(PlaidBiclustTree)
>length(Idx[[1]])
100
>length(Idx[[2]])
100
>length(Idx[[3]])
100
```

In 100 runs, each of the biclusters was discovered. For the Breast cancer dataset, the Plaid solution was stable in all runs. To obtain the robust biclusters and plot their profiles, the functions `plotSuper` and `plotsuperAll` are applied to the tree of biclusters.

```
>plotSuperAll(which(clusters==Idx[[1]]), as.matrix(XBreast),
 BiclustSet=PlaidSetResult)
> superbiclustersPlaid
Number of Clusters found:  3
First  3  Cluster sizes:
```

```
                      BC 1 BC 2 BC 3
Number of Rows:         40   60    6
Number of Columns:      21   28   18
```

These biclusters are shown in Figure 9.5. In Section 9.3.4 we will illustrate how we can combine robust biclusters of Plaid with the biclusters discovered by ISA and FABIA using the `superbiclust` functionality.



(a) Bicluster 1, 40 genes by 28 samples (b) Bicluster 2, 10 genes by 22 samples

(c) Bicluster 3, 6 genes by 21 samples

Figure 9.5: Gene expression profiles of discovered biclusters, selected bicluster samples are in red.

9.3.2 Robust analysis of ISA

ISA was applied to the Breast cancer data set with 500 seeds and five different thresholds for rows and columns. The routine from the package isa2 was used to select the best biclusters. The resulting set contained 97 biclusters. It could be still possible that there are some highly overlapping and even identical biclusters in this output.

```
>ISAResult <- isa.biclust(ISA)
>resISA <- BiclustSet(ISAResult)
>resISA

An object of class BiclustSet
Number of Clusters found:  97
First  5  Cluster sizes:
                   BC 1 BC 2 BC 3 BC 4 BC 5
Number of Rows:     355  400  133  449  375
Number of Columns:    5    4    3    2    2

>JaMatrISA <- similarity(resISA,index="jaccard", type="both")
>ISABiclustTree <- HCLtree(JaMatrISA)
>plot(ISABiclustTree)
```

The hierarchy of biclusters in Figure 9.6 shows that, indeed, there are 16 pairs of exactly the same biclusters - they are joined at the height of 0 in the tree. The rest of the biclusters are joined at heights higher than 0.5, which means that there is a large overlap, but they must be different biclusters. There are also 31 non-overlapping groups of biclusters, joined at the height of 1. By cutting the tree at any non-zero length, we can obtain non-redundant biclusters. In Section 9.3.4 we will check how these biclusters are related to the Plaid and FABIA biclusters.

9.3.3 Fabia: overlap between biclusters

The FABIA method was applied to the breast cancer data using a maximum number of the biclusters equal to 25. The resulting number of non-empty biclusters was 22.

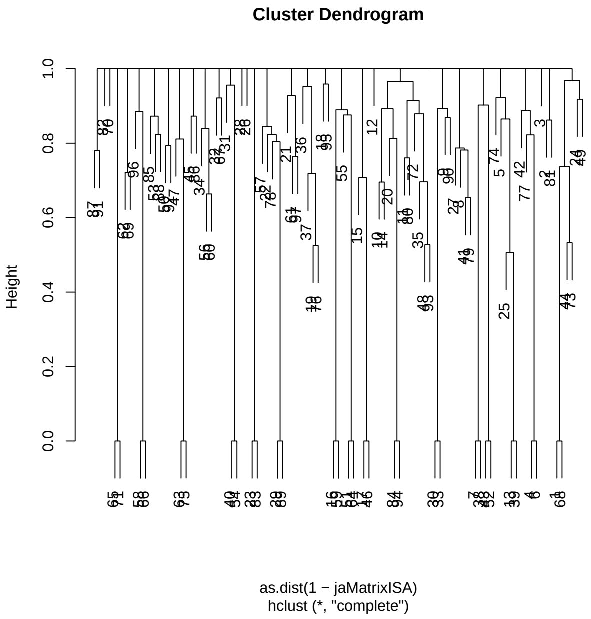


Figure 9.6: The hierarchy of 97 best biclusters generated by ISA using different thresholds and seeds.

```
>resFABIA <- BiclustSet(FABIA)
>resFABIA
An object of class BiclustSet
Number of Clusters found:  22
First  5  Cluster sizes:
                   BC 1 BC 2 BC 3 BC 4 BC 5
Number of Rows:     186   65  161  149  150
Number of Columns:   38   36   32   36   30
>JaMatrFABIA<- similarity(resFABIA,index="jaccard", type="both")
>FABIABiclustTree <- HCLtree(JaMatrFABIA)
>plot(FABIABiclustTree)
```

To check for the overlapping results, we plotted the dendrogram in Figure 9.7. Some overlap between all biclusters is observed, however, it is not large. The minimum height of the tree is around 0.9, which means that, for the most overlapping biclusters, about 10% of the elements are shared. All FABIA biclusters, thus, can be used as prototypes, due to the low overlap for the Breast cancer dataset. However for other datasets, applying `superbiclust` procedure to FABIA output may prove useful. In the next chapter we combine all results together to see how the discovered biclusters are related to each other.

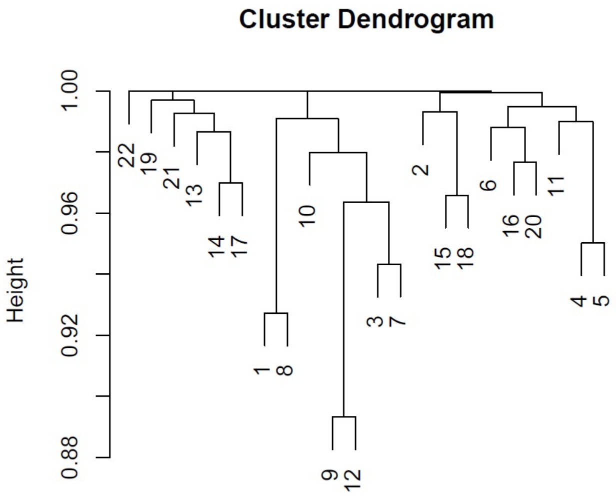


Figure 9.7: The hierarchy of 22 biclusters generated by FABIA.

9.3.4 Biclustering analysis combining several methods

Three robust biclusters of Plaid, 22 biclusters of FABIA, and 97 biclusters of ISA were combined to construct the dendrogram to check if there is overlap between the methods. The constructed tree is shown in Figure 9.8. The clusters in the tree are mostly formed by the biclusters generated from the same method. There is almost no overlap between the Plaid and ISA or FABIA biclusters. A slightly higher overlap is observed for the ISA and FABIA biclusters, even though the highest overlap is 20% of bicluster elements. We have observed that the Plaid biclusters have substantially lower number of rows. To check whether there is similarity of biclusters in terms of rows or columns, we can construct similarity trees of biclusters based on rows and columns separately. Similar to the overall Jaccard index, the row Jaccard index shows that there is a very small overlap between methods in terms of discovered rows. However, in terms of columns, there has been considerably larger overlap between

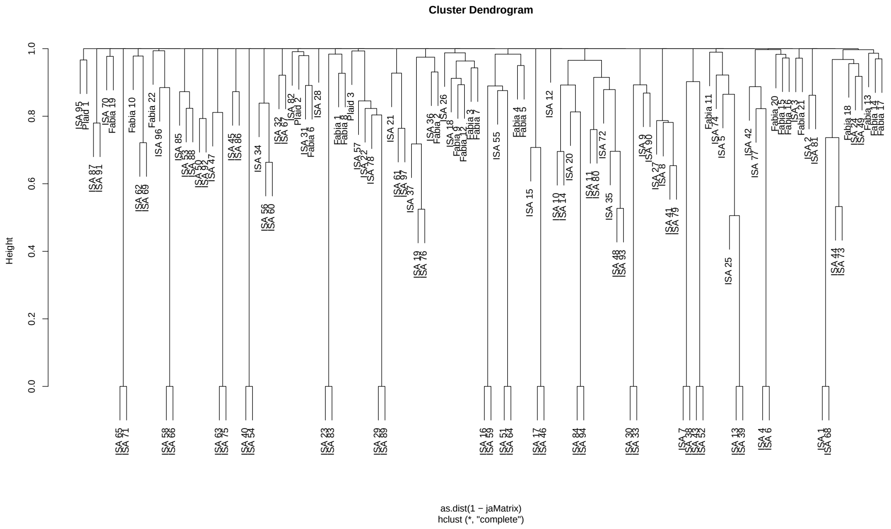


Figure 9.8: The hierarchy of biclusters generated by FABIA, ISA and Plaid.

columns of Plaid and FABIA biclusters (Figure 9.9). Robust biclusters 1 and 2 of Plaid had more than 50% overlap in terms of columns with FABIA biclusters 7 and 12, whereas robust bicluster 3 of Plaid had almost 50% overlap with bicluster 18 of ISA and more than 40% overlap with FABIA bicluster 19.

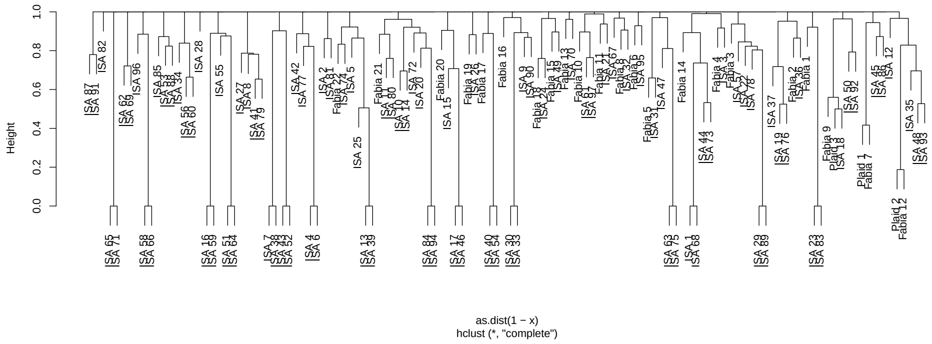


Figure 9.9: The hierarchy of biclusters columns generated by FABIA, ISA and Plaid.

9.4 Discussion

We have presented the implementation of a robust biclustering procedure in the R package `superbiclust`. A series of similarity indices were described and compared. The choice of the link function and the way robust biclusters can be obtained were illustrated on three different biclustering methods.

The `superbiclust` package contains functions to convert the output of different algorithms, which are usually stored in various formats. It allows the user to define the cut-off value by using a specific height, which can be chosen after the structure of biclusters has been plotted, or by using `identify()` on the plotted tree, which automatically creates groups of biclusters.

This tool is useful for obtaining robust results in biclustering methods unstable to some initialization parameters and in addition to investigate the degree of overlap of biclusters in any biclustering method. In addition, it can be applied to check if there are similar biclusters discovered by different methods.

The similarity measure which has been computed for the examples in this chapter used overlap in two dimensions, rows and columns. However, if a particular focus of analysis are subsets of rows or columns, the similarity can be computed in one dimension, either rows or columns by specifying `type` parameter in the `similarity function`. This takes in general less time compared to the two-dimensional overlap.

Chapter 10

Diagnostic tools for biclusters

This chapter is dedicated to the development of diagnostic tools for biclustering solutions. The diagnostics methods for biclustering output are scarce and method-specific. The diagnostic tools presented in this chapter can be used for the whole class of additive biclustering methods discussed in Chapter 7. We present diagnostic tools based on a two-way ANOVA model and extend the scoring approach proposed by Chia and Karuturi (2010). The framework is general and uses concepts of differential co-expression, rather than focusing on a particular biclustering method.

The chapter is organized as follows. In Section 10.1 we discuss the differential co-expression framework of Kostka and Spang (2004) and the stratification procedure of Chia and Karuturi (2010). Subsequently, new scoring and diagnostic tools based on a two-way ANOVA model are introduced in Section 10.2. The proposed methods are illustrated on biclustering results of the Breast cancer and the DLBCL datasets in Section 10.3. The results are discussed in Section 10.4.

10.1 Scores for stratification of biclusters based on differential co-expression

10.1.1 Additive biclustering and differential co-expression

The differential co-expression concepts were introduced in Section 7.3. In this section, we explicitly formulate the model for the expression levels within a submatrix, which was considered by Kostka and Spang (2004).

Without any loss of generality, let $\boldsymbol{A}_{m\times n}$ be the expression matrix and let the *ij*-

th entry, a_{ij}, be the expression measurement of gene i on array j, $i = 1, \dots, m$, $j = 1, \dots, n$. The expression matrix can also be seen as the real-value matrix of n variables and m conditions. Let a subgroup of rows (genes) be denoted by I and a subgroup of columns (arrays, also referred to as samples or conditions) be denoted by J and the submatrix formed by the subgroup of I rows and J columns be denoted by $\boldsymbol{A}_{IJ}$. As mentioned in Chapter 7, expression values in a submatrix (bicluster) are modelled by the following linear model:

$$a_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij}, \;\; i \in I, \; j \in J. \tag{10.1}$$

Here, μ is an overall (constant) effect, α_i is a gene effect, β_j is an array effect, and ε_{ij} are independent identically distributed normal errors, $\varepsilon_{ij} \sim \mathrm{N}(0, \sigma^2)$. For a known submatrix, the maximum likelihood estimators for the unknown parameters of this model are $\hat{\mu} = a_{..}$, $\hat{\alpha}_i = a_{i.} - a_{..}$, $\hat{\beta}_j = a_{.j} - a_{..}$, where $a_{..} = 1/(n_I \cdot n_J) \sum_{i \in I, j \in J} a_{ij}$ is the overall mean, $a_{i.} = 1/(n_J) \sum_{j \in J} a_{ij}$ is the average of gene i and $a_{.j} = 1/(n_I) \sum_{i \in I} a_{ij}$ is the average of array j.
Kostka and Spang (2004) developed a differential co-expression algorithm to discover submatrices (i.e., biclusters) in the data matrix with the elements that follow a common linear model (10.1). This type of bicluster would have a specific structure, which is different from the structure outside a bicluster. The same idea holds for the other additive biclustering methods such as Plaid, δ-biclustering, and FLOC.

10.1.2 Stratification of biclusters (Chia and Karuturi, 2010)

The additive model (10.1) can be also used for the biclustering diagnostics as demonstrated by Chia and Karuturi (2010). Within the differential co-expression framework, a bicluster k can be viewed as a submatrix with n_I genes and n_{J1} arrays, other arrays, which are not a part of the bicluster, are denoted by J_2, and their number is $n_{J2} = n - n_{J1}$. To evaluate how well the selected genes form a bicluster on J_1 compared to the J_2 conditions, Chia and Karuturi (2010) proposed to measure the co-expression of genes within the bicluster and differential co-expression of genes between the arrays within and outside the bicluster. To estimate the strength of co-expression within the bicluster, the linear model (10.1) is fitted to both groups of arrays (inside and outside the bicluster) and the parameter estimates of genes and array effects are used as quantitative measure for the co-expression strength. Note that Chia and Karuturi (2010) define the genes and arrays effects for a bicluster k as $a_{i.}$ and $a_{.j}$, respectively. Based on these estimates, Chia and Karuturi (2010) proposed two scores to measure the co-expression strength of a bicluster:

$$\begin{aligned} T_l(k) &= \tfrac{1}{n_I} \sum_{i \in I} (a_{i.l})^2 - \tfrac{E_l(k)}{n_{Jl}}, \\ B_l(k) &= \tfrac{1}{n_{Jl}} \sum_{j \in J_l} (a_{.jl})^2 - \tfrac{E_l(k)}{n_I}, \end{aligned} \tag{10.2}$$

where the residual effect $E_l(k)$ is given by

$$E_l(k) = \frac{1}{(n_I - 1)(n_{Jl} - 1)} \sum_{i \in I, j \in Jl} (a_{ijl} - a_{i.} - a_{.j} + a_{..})^2, \tag{10.3}$$

and $l = 1, 2$ (with 1 denoting within-bicluster arrays and 2 denoting outside-bicluster arrays). Chia and Karuturi (2010) defined three major types of biclusters: (1) T type, which has strong gene effect (constant rows), (2) B type, which has strong array effect (constant columns), and (3) μ type (constant values), which has both strong array and gene effect. Thus, for a given bicluster k, the stratification score is given by

$$TS(k) = log\left(\frac{T_l(k) + \theta}{B_l(k) + \theta}\right), \tag{10.4}$$

where $l = 1$ if $SB > 0$ and $l = 2$, if $SB < 0$, where SB is a differential co-expression score given by

$$SB(k) = log\left(\frac{\max(T_1(k) + \theta, B_1(k) + \theta)}{\max(T_2(k) + \theta, B_2(k) + \theta)}\right), \tag{10.5}$$

where θ is a small offset constant to avoid extreme values and negative arguments for the logarithm.

Chia and Karuturi (2010) proposed to use a threshold τ for stratifying the biclusters as (1) T-type (i.e., having dominant gene effect) for $TS > \tau$, (2) B-type (i.e., having dominant array effect) for $-TS > \tau$ or (3) μ type (i.e., both gene and array effects) for $|TS| < \tau$. The value τ is chosen arbitrarily.

10.2 Stratification of biclusters based on a two-way ANOVA model with one observation per cell

Within a given bicluster, the linear model defined in (10.1) is a two-way ANOVA model with one replicate per cell. Hence, it can be used to decompose the variability within a specific bicluster k into three sources: (1) the variability between the genes, (2) the variability between the arrays, and (3) the residual variability. These sources of variability can be estimated using the mean square errors, denoted as $MSG(k)$,

$MSA(k)$, and $MSE(k)$ for mean sum of squares of genes, the mean sum of squares of arrays, and the mean square error, respectively, given by

$$MSG(k) = \frac{n_{J1}}{n_I - 1} \sum_{i \in I} (a_{i.} - a_{..})^2, \tag{10.6}$$

$$MSA(k) = \frac{n_I}{n_{J1} - 1} \sum_{j \in J1} (a_{.j} - a_{..})^2, \tag{10.7}$$

$$MSE(k) = \frac{1}{(n_I - 1)(n_{J1} - 1)} \sum_{i,j} (a_{ij} - a_{i.} - a_{.j} + a_{..})^2. \tag{10.8}$$

In ANOVA terms, $E_1(k)$ defined in (10.3) is the mean residual sum of squares ($MSE(k)$) calculated within the bicluster. Further, the scores of rows and columns proposed by Chia and Karuturi (2010) are closely related to the mean squares errors in the two-way ANOVA model. In particular, the following relationship holds:

$$T_l(k) = \tfrac{n_I - 1}{n_I} MSG(k) + (a_{..l})^2 - \tfrac{MSE(k)}{n_{Jl}}.$$

Similarly,

$$B_l(k) = \tfrac{n_{Jl} - 1}{n_{Jl}} MSA(k) + (a_{..l})^2 - \tfrac{MSE(k)}{n_I}.$$

The details of the relationship of the T_l and the (MSG, MSE) score are provided below:

$$\begin{aligned}
T_l(k) &= \tfrac{1}{n_I} \textstyle\sum\limits_{i \in I} (a_{i.l})^2 - \tfrac{E_l(k)}{n_{Jl)}} = \tfrac{1}{n_I} \textstyle\sum\limits_{i \in I} ((a_{i.l} - a_{..l}) + a_{..l})^2 - \tfrac{MSE(k)}{n_{Jl}} = \\
&= \tfrac{1}{n_I} \textstyle\sum\limits_{i \in I} (a_{i.l} - a_{..l})^2 + \tfrac{2}{n_I} \textstyle\sum\limits_{i \in I} ((a_{i.l} - a_{..l}) \cdot a_{..l}) + \tfrac{1}{n_I} \textstyle\sum\limits_{i \in I} (a_{..l})^2 - \tfrac{MSE(k)}{n_{Jl}} = \\
&= \tfrac{n_I - 1}{n_I} MSG(k) + \tfrac{2a_{..l}}{n_I} \textstyle\sum\limits_{i \in I} (a_{i.l}) - \tfrac{2}{n_I} \textstyle\sum\limits_{i \in I} (a_{..l})^2 + (a_{..l})^2 - \tfrac{MSE(k)}{n_{Jl}} = \\
&= \tfrac{n_I - 1}{n_I} MSG(k) + (a_{..l})^2 - \tfrac{MSE(k)}{n_{Jl}}.
\end{aligned}$$

Based on the results presented above, in the next section we describe a new approach for stratification of biclustering methods.

10.2.1 Stratification scores

10.2.1.1 Rows and columns effects

The mean sums of squares (10.6) - (10.8) are used to compute F statistics for each effect (genes, arrays) included in the model. Let

$$F(G) = MSG(k)/MSE(k)$$

and

$$F(A) = MSA(k)/MSE(k)$$

be the F statistics to test for the effect size of gene and array within a bicluster k, respectively. The $F(G)$ and $F(A)$ statistics are related to the types of biclusters described in Madeira and Oliveira (2004). In case of constant values bicluster, both $F(G)$ and $F(A)$ are expected to be low, for constant rows (columns) biclusters $F(A)$ ($F(G)$) is low, in case of coherent values both $F(G)$ and $F(A)$ are expected to be high. Table 10.1 gives a summary of different bicluster models and the corresponding F statistics, which should be significant. The evaluation of significance of the F statistics is presented in Section 10.2.3.

Table 10.1: Stratification of types of biclusters according to the significance of F statistics, $F(G \times A)$ is discussed in Section 10.2.1.2.

Bicluster Type	Model	Score
constant values	$a_{ij} = \mu + \varepsilon_{ij}$	None
constant rows	$a_{ij} = \mu + \alpha_i + \varepsilon_{ij}$	$F(G)$
constant columns	$a_{ij} = \mu + \beta_j + \varepsilon_{ij}$	$F(A)$
coherent values	$a_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij}$	$F(G)$ and $F(A)$
multiplicative	$a_{ij} = \mu + \alpha_i + \beta_j + \delta \times (\alpha_i\beta_j) + \varepsilon_{ij}$	$F(G \times A)$

10.2.1.2 Multiplicative effect

The three bicluster models introduced in Section 10.2.1.1 describe strictly additive biclusters. In contrast, multiplicative structure implies interaction between genes and arrays. Hence, the two-way ANOVA model (10.1) should be extended to

$$a_{ij} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \varepsilon_{ij}, \;\; i \in I, \; j \in J, \tag{10.9}$$

where γ_{ij} is the gene×array interaction effect. The model specified in (10.9) is not identifiable since there is only one observation per cell. Tukey (1949) proposed a test statistic for departures from the main effect model by assuming multiplicative effect for the interaction. Hence, we re-write the interaction model (10.9) as

$$a_{ij} = \mu + \alpha_i + \beta_j + \delta \times (\alpha_i\beta_j) + \varepsilon_{ij}. \tag{10.10}$$

In case of departures from additivity (a model, which is valid for coherent values biclusters), δ is expected to be high. Tukey's test statistic, $F(G \times A) = (A^2)/(B \times C)$, can be used for inference on the non-additivity effect, where

$$A = \sum_{ij} a_{ij} \times (a_{i.} - a_{..}) \times (a_{.j} - a_{..}), B = \sum_{i} (a_{i.} - a_{..})^2, \text{ and } C = \sum_{j} (a_{.j} - a_{..})^2.$$

If the $F(G \times A)$ statistic is large it indicates the presence of a multiplicative structure. The semi-parametric bootstrap procedure for evaluating significance of Tukey's statistic is described in Section 10.2.3.

10.2.2 Illustration of two scoring systems

To highlight the similarities and the differences between the two stratification methods, several biclusters with various structure have been simulated: constant values, constant rows and columns, coherent values, and multiplicative. The bicluster structures are visualized in Figure 10.1.

The TS score used for stratification according to Chia and Karuturi (2010) is plotted against the F statistics for rows and columns and the Tukey test statistic in Figure 10.2.

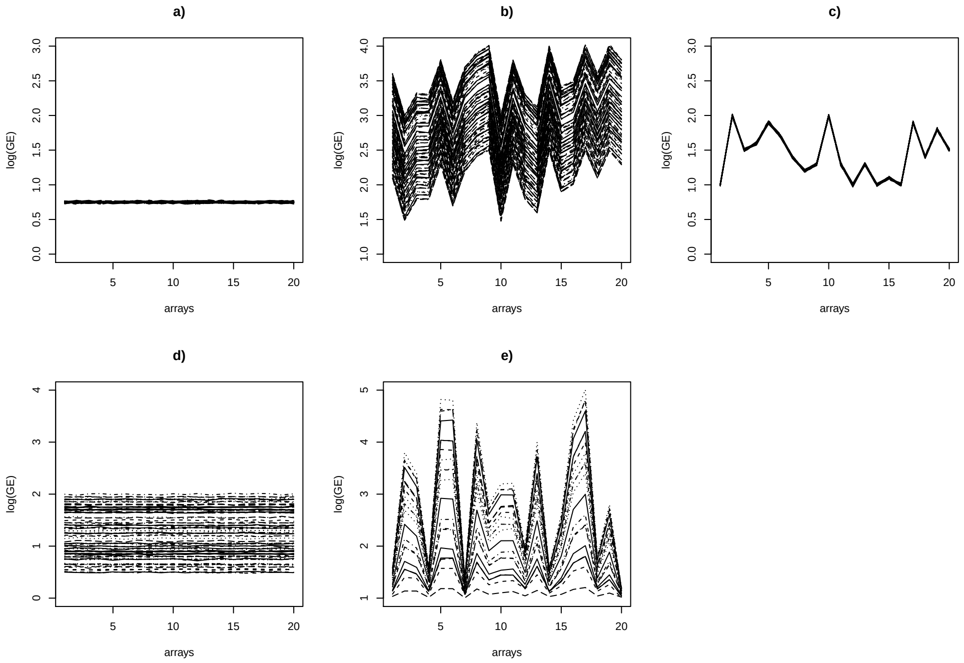


Figure 10.1: Expression profiles of genes in biclusters of 5 structure types: (a) constant values, (b) constant rows, (c) constant columns,(d) coherent values, (e) multiplicative.

As can be observed from Figure 10.2, TS has a very low score for all biclusters and does not exceed 0.15 in the absolute value. Even if the threshold proposed by Chia and Karuturi (2010) is low, the TS stratification score would not be able to distin-

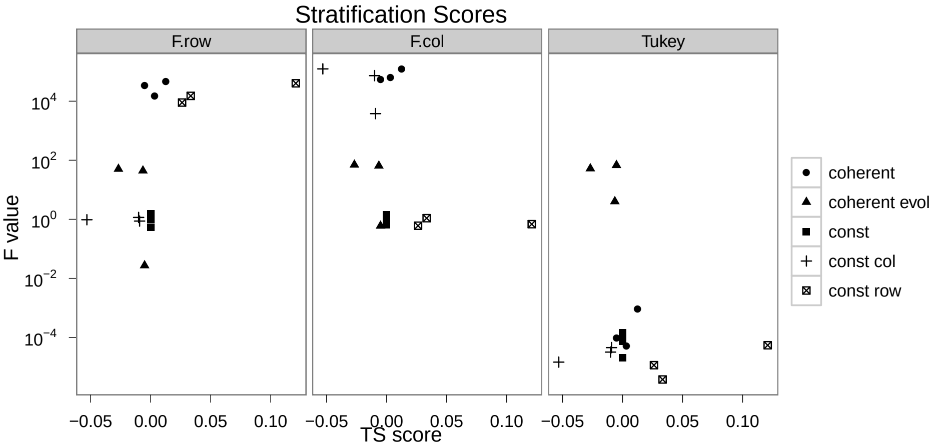


Figure 10.2: Comparison of two scoring systems: F statistic for row effect vs. TS score.

guish various effects present within the bicluster. Hence, TS will most likely put all biclusters in one group. However, all simulated biclusters have a structure different from random noise, with different effects present. In contrast to the F values differentiation, TS score puts all these various structures in one group, namely, μ type biclusters.

10.2.3 Significance of stratification scores

The asymptotic distribution of the F test statistics cannot be used in order to calculate the p values for the observed test statistics. The reason is that the dimension of the biclusters discovered by a bicluster algorithm is a random variable. We have shown in a simulation study presented in Chapter 8 that not all I rows and J columns in a true bicluster are discovered by a particular biclustering algorithm. In what follows we propose a non-parametric test for the genes and arrays effects and a semi-parametric method for the multiplicative effect.

Permutation test for gene and array effects

To investigate whether the observed F statistics within biclusters are large enough to discriminate between different structures, we propose the permutation procedure for estimation of significance of the row and column effects within a bicluster.

Since a bicluster in the data is a result of optimization problem, i.e., minimization of the MSE for δ biclustering or maximization of a linear additive model in Plaid biclustering, the observed $F(G)$ and $F(A)$ are extremes for the given number of columns and rows in the bicluster. To evaluate how extreme are the observed F statistics, we construct the reference distribution based on the maxima of column and row effects for the rows and columns within a bicluster.

Let us consider the permutation test for the array effect, $F(A)$. Under the assumption of no array effect for the bicluster genes, we permute arrays for bicluster genes. Then, B submatrices of the same size as the original bicluster are sampled from the permuted data and the maximum of the array-effect F statistic is calculated. The B maxima of $F(A)$ statistics are taken for the reference distribution.

Algorithm for calculating p-values for column effects.

```
Select the $n_I \times n$ submatrix of the data matrix.
For $1 : B$
permute each row and create a 'null' submatrix $n_I \times n$.
    For $1 : P$
randomly select $n_J$ columns from a 'null' submatrix and calculate $F^*(A)$
    Calculate $F^*_{max}(A)_b = max\{F^*(A)_1, \dots, F^*(A)_p\}$.
    Compute $p$-value

        $$p(A) = \frac{\sum (F^*_{max}(A)_b > F(A))}{B}$$

.
```

The significance of the row statistic $F(G)$ is obtained in a similar way.

Semi-parametric bootstrap for the row and column effects

To test the significance of the non-additive effect we assume $\delta = 0$ in model (10.10), i.e., there is no extra structure in the residuals of the additive model. Under the null hypothesis we resample the residuals of two-way ANOVA and construct a bootstrap bicluster with additive effects from the original bicluster. In each iteration, $F(G \times A)$ is calculated and the bootstrap p-value for Tukey's test statistic is obtained based on B bootstrap re-samplings.

Semi-parametric bootstrap algorithm for calculating p-values for non-additive effect.

Within a given $n_I \times n_J$ bicluster fit the model $a_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij}$ and get the residuals

$$r_{ij} = a_{ij} - (\mu + \alpha_i + \beta_j).$$

```
for 1 : $B$
1. Sample $n_I \times n_J$ residuals $r^*$ and calculate bootstrap bicluster elements by $a^*_{ij} = \mu + \alpha_i + \beta_j + r^*_{ij}$;
2. Calculate $F(G \times A)_b$ for the bootstrap bicluster.
Compute $p$-value
```

$$p(G \times A) = \frac{\sum (F(G \times A)_b > F(G \times A)_{obs})}{B}.$$

10.3 Results

In practice, the gene expression data have high level of noise and detecting structure within a given bicluster is a non-trivial task. In this section we apply stratification methods to gain insight into the structure of discovered biclusters in real life data. The following biclustering methods were used for the analysis: Plaid with four different types of layers ((1) `m` - basic, (2) `m+a` - column effect, (3) `m+b` - row effect, and (4) `m+a+b` - full model), δ-biclustering with $\delta = 0.1$, FLOC with $\delta = 0.05$, and FABIA with five factors.

10.3.1 Dutch breast cancer dataset

In this section we use the Breast cancer dataset to compare the two methods for stratification of biclusters. The diagnostics scores and the results of tests of the significance of structure within biclusters are summarized in Table 10.2.

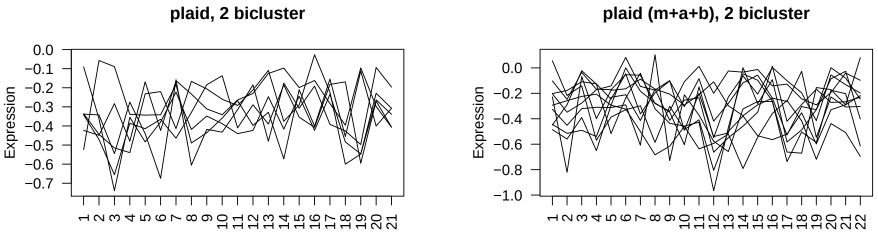
For illustration, we consider the output of two Plaid models: a model for discovery of constant biclusters and a full model for discovery of coherent-values biclusters. Both models resulted in three biclusters; let us focus on a specific bicluster of the constant bicluster model denoted as $Plaid_2$ and a bicluster of the full model denoted as $Plaidmab_2$. The biclusters are displayed in Figure 10.3. Both biclusters have a low TS score (-0.001 and 0.037, respectively), due to the similar magnitude of T

Table 10.2: Diagnostics for biclusters obtained from the Breast cancer dataset (p^* are based on the asymptotic F distribution, p are the resampling based p values).

bicluster	$F(G)$	p^*	p	$F(A)$	p^*	p	$F(G \times A)$	p^*	p	T_1	B_1	TS
$Plaid_1$	4.9	0.0004	0.01	1.11	0.35	0.99	0.002	0.97	0.72	0.104	0.102	0.02
$Plaid_2$	1.34	0.25	1	1.14	0.32	1	0.027	0.87	0.19	0.11	0.11	−0.001
$Plaid_3$	3.09	0.06	1	0.79	0.71	1	0	0.99	0.98	0.06	0.057	0.02
$Plaidma_1$	36.25	0	0	5.58	0	0	0.34	0.56	0.0001	0.22	0.19	0.12
$Plaidmb_1$	8.36	0	0	4.02	0	0	0.09	0.76	0.004	0.09	0.09	−0.003
$Plaidmb_2$	0.32	0.81	1	2.09	0.01	0.35	0.04	0.85	0.03	0.04	0.04	−0.06
$Plaidmab_1$	45.27	0	0	11.08	0	0	0.35	0.55	0	0.18	0.15	0.17
$Plaidmab_2$	10.38	0	0	3.31	0	0	0.002	0.96	0.75	0.1	0.1	0.037
$Plaidmab_3$	32.92	0	0	0.55	0.79	1	0.07	0.79	0.02	0.08	0.06	0.26
$FLOC_1$	0.67	0.99	1	3.24	0.0002	0.03	1.31	0.25	0.0001	−0.002	0.001	−0.19
$FLOC_2$	0.8	0.99	1	10.95	0	0	0.63	0.43	0.1	−0.001	0.003	−0.39
$FLOC_3$	1.76	0	0	4.79	0	0	50.63	0	0	0.002	0.001	−0.23
$FLOC_4$	0.82	0.99	1	4.2	0	0	0.01	0.93	0.78	−0.001	0.001	−0.21
$FLOC_5$	1.05	0.29	1	2.04	0.09	1	0.04	0.85	0.52	0.001	0.001	−0.26
δ_1	4.3	0	0	13.2	0	0	0.31	0.58	0	0.005	0.005	−0.45
δ_2	6.6	0	0	41.95	0	0	0.3	0.58	0	0.01	0.015	-0.94
δ_3	6.4	0	0	4.5	0	0	0.05	0.82	0.025	0.005	0.004	-0.43
δ_4	8.8	0	0	6.9	0	0	0.05	0.82	0.021	0.003	0.003	0.56
δ_5	4.13	0	0	13.5	0	0	0.004	0.95	0.54	0.002	0.005	-0.23
$FABIA_1$	1.05	0.37	0.99	362.8	0	0	0.52	0.47	0.004	0.007	0.22	−2.6
$FABIA_2$	0.89	0.53	0.99	6.63	0	0	3.32	0.07	0.001	0.08	0.16	−0.65
$FABIA_3$	2.34	0.07	0.99	1.7	0.17	0.99	1.12	0.29	0.01	0.12	0.01	0.14
$FABIA_4$	17.7	0	0	2.77	0.003	0.23	0.07	0.79	0.36	0.28	0.13	0.72
$FABIA_5$	1.03	0.42	0.99	3.05	0.006	0.42	3.14	0.08	0.0004	0.003	0.01	−0.44

and B scores: $T(Plaid_2) = 0.111$, $B(Plaid_2) = 0.111$, and $T(Plaidmab_2) = 0.103$, $B(Plaidmab_2) = 0.099$. The two-way ANOVA based scores reveal more details about the structure within a bicluster. For the bicluster $Plaid_2$, $F(A)$ and $F(G)$ are not significant ($p(A) = 1, p(G) = 1$). Hence, this bicluster is classified as a constant-values bicluster. However, the bicluster $Plaidmab_2$ has low p values both row and column effect: $F(G) = 10.38, p = 0$ and $F(A) = 3.30, p = 0$, suggesting the coherent-values structure within the bicluster (see also Figure 10.3(b)). Thus, it is not evident that biclusters with the same magnitude of the TS, B, and T scores have the same structure. The p values are more refined tools for identifying the bicluster type compared to the TS scores.

Next, we check if these biclusters have a multiplicative effect using the Tukey's statistic. None of the two biclusters have a large interaction effect: $F(G \times A) = 0.027, p = 0.869$ for $Plaid_2$, $F(G \times A) = 0.002, p = 0.961$ for $Plaidmab_2$. Therefore we can conclude that the bicluster from the basic model has the constant-values structure and the bicluster from the full model has the coherent-values structure.



(a) (b)

Figure 10.3: Gene profiles for the discovered biclusters in the Dutch breast cancer data: (a) the second bicluster of Plaid `model=m`; (b) the second bicluster of Plaid `model = m+a+b`.

Profile plots of biclusters are shown in Figure 10.4.

More interestingly, a non-additive biclustering method FABIA identifies a bicluster with a significant interaction effect ($F(G \times A) = 3.14, p = 0.0004$), whereas depending on the value of the threshold chosen for TS score it is classified as either a B type or a μ type. The bicluster is shown in Figure 10.5(a). In addition, the third bicluster found by FABIA ($FABIA_3$) is classified as a constant-row bicluster (with $p(A) = 0.23$ and $p(G \times A) = 0.36$). Another example of a bicluster with a significant interaction is the FLOC bicluster 3, which is detected as a μ type or could be T type, if the low cut-off value τ is chosen for the threshold. This bicluster is plotted in Figure 10.5(b). The profile plot shows that identification of bicluster structure is non-trivial.

10.3.2 The DLBCL dataset

The Plaid algorithm discovered two biclusters with the models `m` and `m+a` and one bicluster with the models `m+b` and `m+a+b`. Both the δ-biclustering with $\delta = 0.1$ and FLOC with $\delta = 0.05$ discovered five biclusters. In general, the FLOC output contained more genes in a bicluster compared to the δ-biclusters, but the number of columns was lower. It can be explained by the difference in the residual threshold value, which

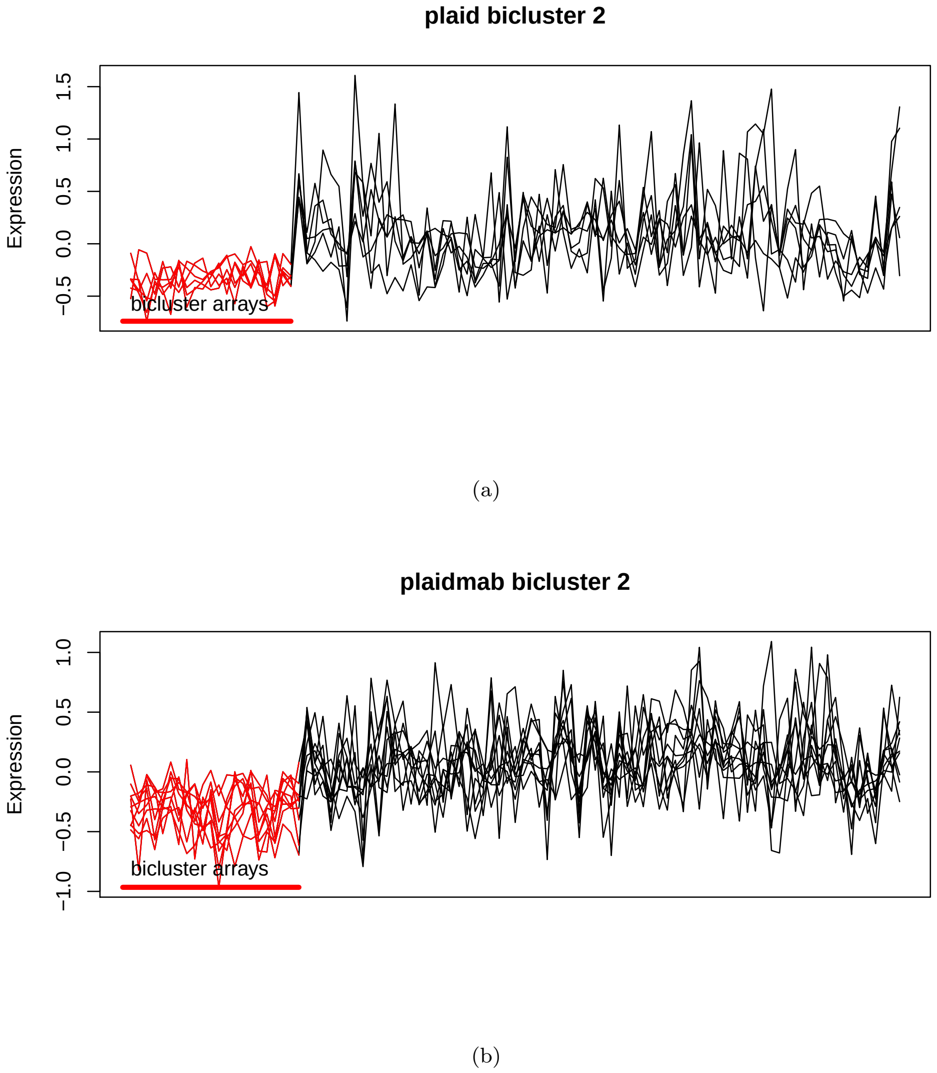
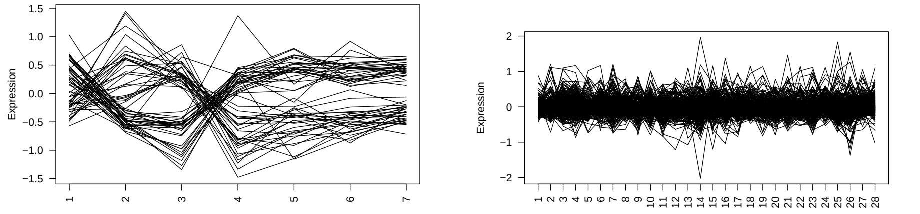


Figure 10.4: Gene profiles for the discovered biclusters in the Dutch breast cancer data: (a) the second bicluster of Plaid `model=m`; (b) the second bicluster of Plaid `model = m+a+b`.

was higher than in the case of δ-biclustering.

The stratification scores are summarized in Table 10.3. There were only two biclusters of constant columns discovered, namely the FLOC biclusters 2 and 3. The low p values for the gene and array effects suggest the coherent-values structure for all biclusters of the Plaid and δ-biclustering, as well as the biclusters 1, 4 and 5 of FLOC. Low p



(a) (b)

Figure 10.5: Gene profiles for the discovered biclusters in Breast Cancer data: (a) the fifth bicluster of FABIA; (b) the third bicluster of FLOC.

values for Tukey statistic indicate multiplicative structure of FABIA biclusters, which is expected since FABIA searches for multiplicative biclusters. In contrast, the second and the fifth biclusters of FLOC, the Plaid bicluster `m+b` and the δ-biclusters $1 - 4$ have a significant non-additive term, which indicate that not all structure in the data was captured by the additive model. However, the presence of noise in the data can lead to the discovery of non-additive patterns by additive methods.

TS scores detect all δ-biclusters and FLOC-biclusters as T-type and biclusters of Plaid as μ-type. The FABIA biclusters 1, 2, and 5 are T-type and biclusters 3 and 4 are of B type.

10.4 Discussion

In this chapter we have presented a bicluster stratification framework based on the differential co-expression and the bicluster classification of Madeira and Oliveira (2004). The F statistics from a two-way ANOVA model with Tukey's statistic of non-additivity estimate additive and non-additive effects of genes and arrays within a given additive bicluster. To estimate the significance of effects within a bicluster we have used resampling-based p-values. Based on the significance of the F statistics, a bicluster is defined to have a gene or array effect if the gene or array p-values are smaller than 0.05. The calculation of p-values takes into account the size of a bicluster and depends on the number of genes and arrays within a bicluster.

Based on the outlined methodology we could see that the additive biclustering methods have indeed discovered the biclusters with additive structure in both data sets. However, the co-expression structure was not always coinciding with the expected

Table 10.3: Diagnostic scores for the biclusters of the DLCBL data. (p are resampling based p values, p^* are based on the asymptotic F distribution)

Bicluster	F(G)	p^*	p	F(A)	p^*	p	$F(G \times A)$	p^*	p	T_1	B_1	TS
$plaid_1$	5.44	0	0	3.95	0	0	0.41	0.05	0.05	1.03	1.02	−0.01
$plaid_2$	4.56	0	0	3.11	0	0	0.07	0.77	0.32	0.55	0.55	0.01
$plaidma_1$	21.25	0	0	17.38	0	0	0.39	0.53	0.19	1.37	1.5	0.095
$plaidma_2$	7.52	0	0	2.4	0	0.02	0.08	0.78	0.57	1.1	1.09	−0.01
$plaidmb_1$	7.85	0	0	11.25	0	0	4.25	0.04	0	1.27	1.17	−0.08
$plaidmab_1$	21.51	0	0	29.81	0	0	0.52	0.47	0.12	0.99	0.99	−0.001
$FLOC_1$	1.63	0	0.01	5.52	0	0	2.61	0.11	0.06	0.11	0.025	−0.85
$FLOC_2$	1.19	0.1	0.96	10.42	0	0	5.34	0.02	0.007	0.06	0.007	−1.46
$FLOC_3$	0.87	0.81	0.99	7.56	0	0	0.26	0.61	0.54	0.95	−0.004	−2.23
$FLOC_4$	1.43	0	0.04	16.34	0	0	0.003	0.95	0.95	0.06	0.012	−1.22
$FLOC_5$	1.51	0.001	0.03	8.54	0	0	2.18	0.14	0.03	0.04	0.007	−1.31
δ_1	4.2	0	0	95.04	0	0	0.43	0.5	0.03	0.07	0.004	−1.88
δ_2	10.21	0	0	23.29	0	0	1.31	0.25	0.003	0.04	0.03	−1.99
δ_3	2.07	0	0.03	42.77	0	0	0.73	0.39	0.007	0.12	0.012	−2.45
δ_4	7.68	0	0	37.91	0	0	3.86	0.049	0	0.168	0.09	−2.59
δ_5	8.89	0	0	13.14	0	0	0.0004	0.98	0.95	0.95	0.04	−1.64
$FABIA_1$	3.11	0	0	326.63	0	0	50.66	0	0	0.889	0.25	−1.23
$FABIA_2$	2.03	0	0	376.31	0	0	3.04	0.08	0.006	1.525	0.044	−3.36
$FABIA_3$	10.12	0	0	11.78	0	0	61.3	0	0	0.322	0.67	0.71
$FABIA_4$	3.55	0	0	64.21	0	0	16.5	0	0	0.810	0.29	−1.002
$FABIA_5$	24.52	0	0	1.45	0.16	0.99	3.85	0.05	0.006	0.26	1.44	1.68

output. This could be explained mostly by the global data structure that might not have strong local gene and array component. In such a case an analyst would like to consider some biclustering methods that require less parameters and lead to better results. For example, Plaid biclustering optimized for discovering coherent values obtained a bicluster with negligible gene and array effects in other words, a constant-values bicluster. It implies that the biclustering methods targeting discovery of constant value biclusters are more appropriate. Unexpectedly, there was an additive biclustering method that discovered biclusters with a strong non-additive effect of genes and arrays. On the other hand, a multiplicative method found some biclusters with a weak non-additive component.

It must be mentioned, that the p-values calculation was carried out conditionally on the bicluster. These p-values do not allow direct ranking of the biclusters or biclustering methods. In addition, if a biclustering method discovers a bicluster in the data without actual biclusters as a result of optimization problem, the p-values can still be significant. Furthermore, due to the fact that we take out the overall mean within a bicluster, p-values cannot distinguish potentially interesting constant biclusters from the submatrices of noise in randomly generated data. Therefore, we cannot say whether the discovered bicluster is a true bicluster or a submatrix of random values.

Our work is closely related to the stratification framework of Chia and Karuturi (2010). To our knowledge, this is the only related work in the area of bicluster structure diagnostics. In the current chapter, we have shown how the preceding stratification scores relate to the F scores in case of additive bicluster structure. The scores of Chia and Karuturi (2010) look for the dominant effect in the bicluster, whereas the pair $(F(A), F(G))$ focuses on effects size. We have observed that summarizing the gene and array effects by a log-ratio leads to the loss of information about bicluster structure and may create ambiguous classification. We have shown that the TS stratification score cannot distinguish a constant structure from a coherent one. In addition, the previous method is not well-suited for a multiplicative or a non-additive structure while Tukey's test can detect non-additive effects. It should be noted that Tukey's test serves as an indication of the non-additive effect and further investigation of the multiplicative structure within a particular bicluster is needed.

There is still an agreement between our method and the one of Chia and Karuturi (2010). In the trivial case, when the co-expression structure within a bicluster has one dominant effect (array or gene), the log-ratio statistic would be pointing to the same direction as the p-value. A similar agreement is observed when the coherent-values bicluster has a difference in mean expression levels for the columns inside and outside of a bicluster. Applying both procedures allows analyst to choose the most interesting biclusters in terms of differential co-expression.

In conclusion, the proposed stratification method is the diagnostic tool for a biclustering output when little prior knowledge is available about the biological processes in the exploratory gene expression datasets and an analyst wishes to check what kind of structure has been discovered after applying biclustering algorithm to the data. Investigation of bicluster structures may direct the choice of an analyst to a more appropriate method or parameter setting, i.e., specifying the correct model for the Plaid algorithm.

Part III

Joint unsupervised analysis of multiple high throughput data

Chapter 11

Exploratory analysis of integrated data sources using biclustering methods: introduction

In order to investigate a molecular profile of a disease or to discover new therapeutic targets, large amount of data is collected. To fully understand the underlying molecular mechanism of biological processes, the data are collected at different levels: DNA, RNA, transcription factors, proteins and metabolites (Cao *et al.*, 2008). The number of collections in the high throughput data is also increasing, for example, Cancer Genome Atlas, Leukemia Genome Atlas, and others. Using evidence from different data sources, the power of analysis can be increased and more insight into the biology can be gained.

On the other hand, to discover potentially effective drugs to treat complex diseases, information about drug activity can be integrated with the associated genomic activity in a cell. In addition, it is important to discover if a particular molecular structure of a drug can cause a specific desirable on non-desirable effect in a cell. This type of scientific question lead to creation of the high-dimensional databases such as the ChEMBL (Gaulton *et al.*, 2012) and CMAP (Lamb, 2006, 2007).

The research questions in both characteristics of a disease and discovery and design of new effective drugs can be different. Yet, they share one common aspect: the integration of a number of high dimensional datasets is indispensable in order to gain deeper understanding of the cellular processes. The bottleneck in the analysis of the datasets is the availability of appropriate tools. In this part of the dissertation we illustrate how biclustering methods can be used in order to reveal underlying structure in the integrated high-dimensional datasets.

A number of review papers have been written to describe and compare the methods for genomic data integration. For example, Lahti *et al.* (2013) and Huang *et al.* (2012) provided reviews on copy number and gene expression integrative methods and mentioned that there is still a lack of methods for the analysis of integrated data sources. The integration of the chemical and biological data is still an area of the ongoing research. However, some methods developed for the integrative analysis of genomic data can be translated to the setting of chemo- and bioinformatics data.

This chapter will introduce the concepts and studies for the integrative data analysis. We focus on the exploratory analysis, which is the first step in the data analysis and is unsupervised. First, in Section 11.1 we introduce the concepts of gene expression data integration across various compound screening projects. Section 11.2 is focused on the discovery of new therapeutic targets. Section 11.3 provides more details on the concepts of analysis of chemical structure and gene expression data. Finally, to introduce the analysis tools for exploration of the joint datasets, Section 11.4 presents the concepts of joint biclustering.

11.1 Integrated database for drug design

Gene expression compound profiling becomes a routine in the early drug discovery experiments. However, at the moment, the gene expression data is not generalizable due to the selection of compounds for a given experimental setting. For instance, a narrow class of compounds (e.g., kinase inhibitors) is considered for cancer cell lines, which mostly shows expected elevation in gene expression. Nevertheless, provided the increasing amount of compounds with gene expression profiles, exploring data across a variety of projects can pinpoint potential toxicity or discover unspecific compound effects (Chengalvala *et al.*, 2007).

Currently, there are attempts to create public and commercial databases containing gene expression profiles for a wide range of compounds used to treat various diseases (Lamb, 2006; Davis *et al.*, 2011). Given availability of such big and heterogeneous datasets in terms of underlying chemical structure and biological processes, it is important to be able to integrate smaller scale gene expression experiments with large databases. There are a number of benefits resulting from such an integration: new and less well-studied compounds in terms of their molecular profile have potential connection to the well-characterized and studied compounds in other databases. It can highlight side-effects, efficacy, and provide more information about the mechanism of action of a new compound or a set of compounds. As a result it can save time and reduce costs of the drug development process.

This topic will be discussed in Chapter 12 where we illustrate how information contained in a publicly available data set can be combined with *in-house* gene expression experiments.

11.2 Integration of miRNA and mRNA datasets

Various sources of biological information are often combined in the studies to gain insight in the gene regulatory pathways and deeper understanding of the transcription processes on cellular level (Ma and Zhao, 2012b). In transcriptomics research, miRNA and mRNA are profiled on the same tissue or condition of interest. In addition, copy number alteration, SNP assays, and methylation data can be added in the study. Concerning miRNA-mRNA relationship, the goal of such studies is two-fold: 1) to discover how miRNAs regulate mRNAs under a condition of interest; 2) to select joint miRNA-mRNA biomarkers that are responsible for a phenotypic subtype.

The analysis of various data sources can be performed in two ways: handling all datasets separately or jointly. There are advantages and disadvantages to both approaches. Separate analyses result in an output that is not straightforward to integrate. In case of differential expression, it could be possible to select the differential miRNAs first, select differentially expressed mRNAs, and then correlate their profiles. In this setting a grouping variable (disease-control, survival rate, etc.) should be available, which is not always the case. For instance, when a novel cancer subtype needs to be discovered, the grouping variable is not clearly defined and this approach cannot be used. A possible solution is to cluster samples based on their miRNA profiles and mRNA profiles separately and then look for the clusters that would have similar samples in both datasets. In this case, the clustering may not be perfect and relationship between the clusters will not be evident. Moreover, the local correlation patterns may not be taken into account. Therefore, biclustering could be a better approach for selecting the subset (subtype) of samples active on the subset (a signature) of mRNA or miRNA. The biclustering can be performed on both mRNA and miRNA jointly, or on two datasets separately. The joint biclustering analysis received little attention in the integrative literature, mostly due to the comparability issues related to the miRNA and mRNA datasets, since mRNA data has a higher variability

in expression values than miRNA data.
An integrated analysis of miRNA and mRNA data sets is presented in Chapter 13.

11.3 Integration of chemical structure and gene expression datasets

Changes in the molecular structure of a compound can result in drastic changes in the activity of the compound. Therefore, it is of ultimate interest to know which type of changes in some parts of molecule can lead to a certain effect at the genetic or phenotypic level. The major challenge in obtaining data for the chemical structure of the compounds is to decipher a three-dimensional molecule into a numeric vector. Many attempts have been undertaken in the field of chemoinformatics and combinatorial chemistry to find methods for converting a 3D structure into a series of numbers, which can be further used for the analyses. One of the ways to represent a molecule is to compute its fingerprints in terms of substructures of various size (Rogers and Hahn, 2010). Fingerprint is a binary representation of a molecular structure where 1 is the code for present substructure, which is referred to as a fingerprint feature (can be an atom, a or an atom with several bonds), and 0 codes for an absent fingerprint feature. The larger the chemical space from which compounds are taken, the longer are fingerprints of molecules and the dimensionality can reach up to a hundred of thousands. For compounds from the same chemical class, where only a number of changes is made to the structure, the fingerprints are shorter and the length of a vector with non-redundant features is a number of hundreds.
The challenge in the exploration of high-dimensional fingerprint data and high-dimensional gene expression data is to discover, if any, relationship between certain fingerprint features and a change in gene expression. The feature-by-feature and gene-by-gene approaches provide a good indication but are in general time consuming and do not take into account the correlation structure between genes.
In Chapter 14 we present a joint biclustering analysis of fingerprints on gene expression data, which aims at discovering subsets of fingerprint features and genes with the same profile across subsets of compounds.

11.4 Joint biclustering

The main focus of a biclustering method is to discover a subgroup of genes (or other biological measurements) co-expressed or co-regulated under a subgroup of conditions (drugs or disease status). In Part II of the dissertation we have discussed applications of biclustering to one high dimensional dataset at a time. In this section, we briefly introduce the setting of multiple datasets and two important aspects of integrated analysis, namely, data merging and preprocessing.

Data merging

Since the input for any biclustering algorithm is one data matrix, the multiple datasets need to be merged into one data matrix for further exploration. Let us consider two data matrices, $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. The data merging is performed by the dimension they have in common, e.g., genes, samples, compounds.

In Chapter 12 we consider integrated analysis of gene expression data sets, where $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are gene expression matrices, which have various compound profiles and share the dimension of genes. Thus, the merging is performed by the gene dimension. It should be noted that the merged dataset contains only the genes which are measured in both data sets and the missing values for the genes are not allowed.

In Chapter 13, the integrated analysis of paired miRNA-mRNA data is considered. In this setting, two datasets share the dimension of samples. Hence, the integrated data set is obtained by merging $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ by the sample dimension. Similar to the case of gene expression data, the samples in the integrated data set should have profiles in both $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$.

In Chapter 14, the exploration of compounds structure and associated changes in gene expression are considered in a joint way. The fingerprint matrix $\boldsymbol{V}$ and the discretized gene expression matrix $\boldsymbol{X}$ share the compounds dimension. Therefore, the merging occurs by the compound dimension. All compounds in the merged data set should have both a fingerprint vector and the gene expression profile.

Data preprocessing

Whether the expression levels of genes need to be normalized or not prior to biclustering largely depends on the method, that is applied. FABIA is a multiplicative model which assumes standardized values, whereas Plaid is an additive biclustering method which does not require standardized values. Normalization methods, specific to the integrated analysis of gene expression data sets and miRNA-mRNA analysis, are discussed in Chapters 12 and 13.

For the analysis of joint fingerprint feature-gene expression data, the discretization of the gene expression data is crucial for obtaining meaningful biclusters. In Chapter 14, the discretization is discussed so that the joint data can be further used as input for discrete biclustering methods.

Chapter 12

Biclustering methods to assist drug design processes

When gene expression profiles are obtained for a novel set of compounds, it is of high priority to check if the acquired profiles are similar to the well-studied small molecule therapeutics. Rather than querying novel compounds one by one, we propose to use joint biclustering as an exploratory step of the new compound set and available drug profiles. In this chapter we focus on a procedure to obtain joint biclusters between *in-house*, i.e., a set of compounds developed by a company, and the CMAP data. In Section 12.1, we discuss the concept of CMAP project. Section 12.3 aims at providing the overview of necessary preprocessing steps. Section 12.4 describes the biclustering procedure for the integrated data. Finally, the proposed method is presented in Section 12.6.

12.1 Connectivity map

The connectivity map (CMAP) project is a collection of gene expression profiles of more than a thousand of small molecule therapeutics (Lamb, 2006, 2007). The goal of the project is to establish relations among diseases, physiological processes, and drug activity driven by gene signatures. The data collection has been published as a large public database and it has a built-in tool for matching a gene expression pattern of a drug in development with gene signatures of known drugs.

More than a thousand of drugs with various molecular targets, clinical indication, proximal and distal to gene expression are profiled in CMAP. The drugs included are the FDA approved drugs as well as known bioactive compounds. The concentration and time treatment for a set of compounds were kept fixed. Another set of drugs was profiled using varying time and concentration in order to choose the best sensitivity of a cell line to a drug.

Four cell lines were chosen for the profiling, namely, the breast cancer epithelial cell line MCF7, which is used throughout the world as a reference cell line, the prostate cancer epithelial cell line PC3, non-epithelial leukemia cell line HL60, and non-epithelial melanoma cell line SKEML5.

Gene expression has been profiled by means of two microarray types: Human Genome HGU133a and high throughput HTHGU133a arrays.

12.2 Joint biclustering of gene expression profiles across projects

In this section we introduce the integrated dataset that will be analyzed in the remainder of the chapter. Let $\boldsymbol{X}_1$ be a $m \times n_1$ expression matrix obtained from the CMAP data. Note that this could be the complete CMAP expression matrix or a submatrix. Let $\boldsymbol{X}_2$ be a $m \times n_2$ *in house* expression matrix. Note that $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ share the gene dimension. The integrated expression matrix $\boldsymbol{Y}$ is a $m \times (n_1 + n_2)$ matrix given by

$$\boldsymbol{Y} = [\boldsymbol{X}_1 \boldsymbol{X}_2] \,.$$

The aim of the analysis in this chapter is to find submatrices, in which gene expression is similar for a subgroup of compounds from both projects (e.g., company-specific and CMAP). A schematic view of the joint analysis in this case is shown in Figure 12.1.

The aim of the analysis presented below is to find biclusters which contain both *in-house* and CMAP compounds.

12.3 Data preprocessing

The major challenge in combining two different projects and getting sensible joint biclusters is the data merging. Two projects may investigate activity of compounds on a different cell line. Therefore, fold changes with respect to the controls should be considered for the analysis. Nevertheless, the variability of gene expression may be different due to the concentrations of the compound applied to the cell lines and the

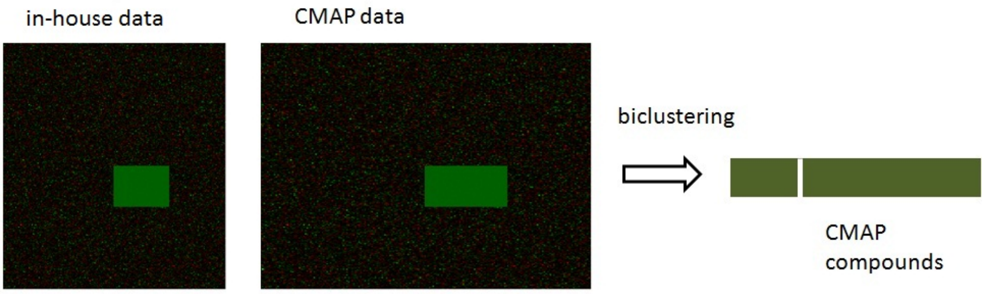


Figure 12.1: Joint biclustering on two datasets: *in-house* compound set and CMAP data are joined by genes and analyzed by biclustering. The output is a bicluster, containing compounds from the new set and CMAP

distribution of expression values remains different. It has to be mentioned that the CMAP data is sparse, i.e., there are many low expression values, which may not be the case in the *in-house* data.

The batch-effect correction should also be considered for the datasets like CMAP in order to avoid obtaining biclusters caused by batch effects. Filtering is another essential step, since the data have to be standardized before the analysis by biclustering methods like FABIA.

In what follows we describe several preprocessing steps.

12.3.1 Gene filtering

The first gene filtering method which is applied is the I/NI call, described in a detail in Chapter 3. Only the probe sets, which pass the filtering criterion are kept within each project. In addition, fold change or variance-based filtering can be applied for further refinement of the gene set.

Fold-change filter is based on the expert knowledge and requires a fixed threshold value. Fold changes are obtained by taking a ratio of gene expression for a given compound versus gene expression of the control sample. If multiple control samples are available in the project, a median or spatial median is taken as a baseline. A typical threshold value is chosen to be equal to 2, corresponding to the two-fold change from the control. Since there are no replicates in a compound screening projects, it is required that a gene should have a fold change higher than the chosen threshold in at least one sample.

Variance-based filter requires multiple control samples. The variance of control samples is compared to the variance of the samples treated with compounds. If the variance of gene expression values is higher in the control group, the gene is excluded from further analysis. In this case, special care should be taken to remove outliers from the control samples, which can substantially affect the filtering procedure.

12.3.2 Normalization

The decision whether to perform additional normalization of the fold changes or not largely depends on the biclustering method which should be applied and its ability to discover certain data structures. FABIA requires data to be centered and normalized. This type of data standardization should be performed prior to the merging step, as otherwise original values would be disturbed. To illustrate it, let us consider a gene G which can be of potential interest within an *in-house* project. The number of samples in the *in-house* project is 36, and the number of samples in the CMAP data is more than $5,000$. The densities of G within the *in-house* project and the CMAP data are shown in Figure 12.2 (left panel). After normalization, both distributions should be comparable. However, it certainly disturbs the original structure. Thus, if the main interest is in discovering a bicluster with a gene G (and a number of genes which have similar expression pattern), which has either up-regulated or down-regulated values in both projects, normalization will prevent biclustering algorithm from discovering such a bicluster. By putting most of the expression values within the *in-house* project close to zero, the *in-house* compounds will not contribute much to the bicluster.

12.4 Biclustering data analysis

We apply three different biclustering methods, which are able to discover a range of structures: unsupervised additive biclustering Plaid, unsupervised multiplicative biclustering by FABIA, and semi-supervised biclustering ISA. The choice of parameters for each method is explained below.

12.4.1 Exploring additive biclusters by Plaid

Discovering biclusters over columns of two different projects by additive biclustering methods has certain advantages in interpretation. Constant joint biclusters have similar fold changes for genes in both projects. The data centering is not directly required, if the bicluster has the constant-rows or coherent-values structure. Coherent or constant-column joint biclusters can address differences in fold changes between

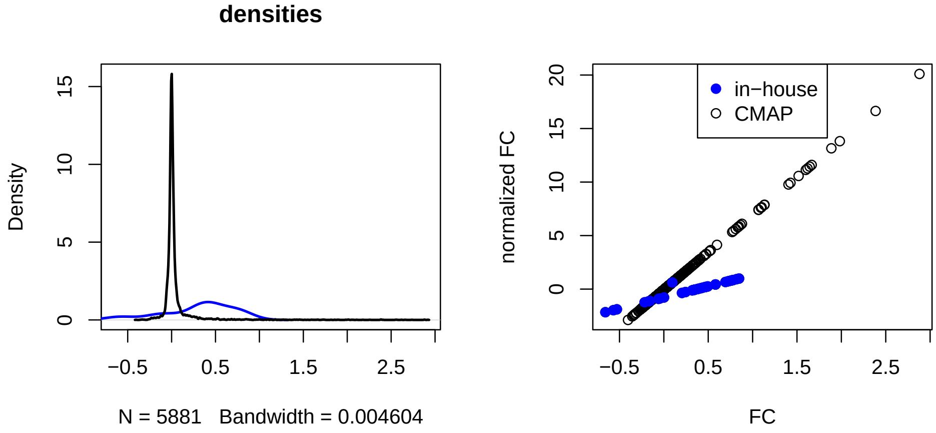


Figure 12.2: Effect of preprocessing on gene G. Left panel: the densities of gene expression within in-house experiment and CMAP data. Right panel: effect of normalization on the original fold changes within in-house and CMAP data sets.

projects as long as the fold changes co-vary in a similar way. Therefore all four type of layers were used to obtain Plaid biclusters.

Plaid is set to discover biclusters with a high tolerance of noise, so that the filtering R^2 criterion described in Chapter 7 is set to 0.5. The biclusters were obtained by the robust biclustering explained in Part II of the dissertation.

12.4.2 FABIA biclustering

For the FABIA method, the thresholds for obtaining hard biclusters are selected based on the evaluation of distribution of scores and loadings for a particular bicluster. To extract the biclusters, the most outlying samples and genes in terms of scores and loadings are selected. The *ad-hoc* bicluster extraction procedure can be fine-tuned to focus on biclusters with both *in-house* and CMAP compounds.

1. For each bicluster k, extract a scores vector $\boldsymbol{Z}_k$.

2. Split $\boldsymbol{Z}_k$ into two vectors $\boldsymbol{Z}_{in-house}$ and $\boldsymbol{Z}_{CMAP}$ and obtain $q_{.25}$ and $q_{.75}$, the 25th and 75th quantiles.

3. Select the threshold τ_Z for the factor scores. A typical value for τ_Z is $max(|q_{.25}|, |q_{.75}|)$.

If the scores of both *in-house* compounds and the CMAP compounds pass the threshold, the bicluster is considered to be a joint *in-house*-CMAP bicluster. It should be noted that if $\boldsymbol{Z}_{in-house}$ and $\boldsymbol{Z}_{CMAP}$ are both positive or negative, the corresponding compounds have the similar effect on the genes in a bicluster, i.e., up- or down-regulation. If $\boldsymbol{Z}_{in-house}$ and $\boldsymbol{Z}_{CMAP}$ values are of opposite sign, the corresponding compounds have the opposite effect on the gene expression.

12.4.3 ISA

The ISA biclustering is a semi-supervised method. In order to start an ISA biclustering, initialization with a random seed is needed. If a specific gene set is given, or a research question related to certain pathways is investigated, then ISA genes loadings s_i can be initialized by a binary vector with 1 for genes belonging to a gene set/pathway of interest or 0 otherwise. The resulting bicluster is considered to be pathway/gene set driven and can be interpreted as a compound-induced pathway expression.

The ISA obtains bicluster membership by applying a set of thresholds to the gene loadings and samples scores. The bicluster for each threshold value is obtained and the most similar biclusters are joined. If two modules for the different thresholds are similar, then the larger one, the one with the smaller threshold is kept.

12.5 Biological interpretation of biclusters

For the genes identified in biclusters, the gene set enrichment analysis can be performed by using the Gene Ontology (GO) database (Ashburner *et al.*, 2000). There are two major options for performing gene set enrichment analysis (GSEA): either on gene set itself or on the associated p-values. In Section 12.5.1, we describe the hypergeometric test for hard biclusters, like in Plaid. In Section 12.5.2, we describe the MLP analysis of the soft biclusters discovered by ISA.

12.5.1 GSEA for hard biclusters

For the hard biclusters in the result set of the Plaid procedure we apply the hypergeometric test (Falcon and Gentleman, 2008) for enrichment of GO terms. For each gene set we construct Table 12.1.

The p-value for the bicluster genes are obtained from the Fisher exact test and the top most significant pathways are extracted for possible interpretation of the compounds in terms of their gene expression.

Table 12.1: The two-way table for testing overrepresentation of a GO category among bicluster genes.

	present in GO category	absent in GO category
bicluster genes	n_{11}	n_{10}
non-bicluster genes	n_{01}	n_{00}

12.5.2 GSEA for soft biclusters

In case of a soft bicluster membership of genes, a probability can be put on each gene belonging to a bicluster. The probability vector $(p_1, \ldots, p_m)$ for genes in ISA bicluster are converted genes loadings, $p_i = 1 - |s_i|$.

The vector of probabilities for each ISA bicluster can be further used for the MLP analysis (Raghavan *et al.*, 2010). As described in detail in Raghavan *et al.* (2006, 2012), for each GO category, the the Mean-Log-P (MLP) statistic is calculated as

$$MLP = \sum_{g \in GOcat} \frac{-log(p)}{n_{g \in GOcat}}.$$

To determine the significance of the MLP statistic, a permutation test is performed by resampling original p-values and repeating the calculation of the MLP statistic for each GO category. Afterwards, a critical value is chosen to determine whether the GO category is considered to be significant. The choice of a critical value is GO category size dependent, therefore an additional empirical procedure based on permutation and smoothing is run to obtain critical values.

The significant GO terms can further facilitate the interpretation of biclusters and characterization of the compounds inducing changes in gene expression.

12.5.3 Compound set analysis

Comparative genomic database contains information on various drugs and their gene expression activity (Davis *et al.*, 2011). For a subset of compounds, selected to belong to a bicluster, it can be checked whether a certain pathway is activated by the compounds in the bicluster. Providing this information to the analyst can facilitate interpretation of biclusters in terms of mechanism of action.

12.6 Results

12.6.1 Preprocessing of the data

Data selection and gene filtering

In general, including multiple concentrations, treatment durations, and cell lines is not a problem for a biclustering method except for the computation time. However, for the illustration of the analysis concepts and prospective application of joint biclustering, only one concentration and treatment duration per compound is chosen. The sample with the largest variance in gene expression is included in the final set. Nevertheless, for general exploration purposes all concentrations and treatment durations should be used since various genes can respond to a certain amount of compound at different times. The platform which is used for this comparison is HT HGU133a and the chosen cell line is the prostate cancer PC3 cell line.
The resulting number of samples in CMAP data used for the current analysis is $1,161$. The number of *in-house* samples is 94. Concerning gene filtering, joining the I/NI calls for the CMAP and *in-house* data resulted in 442 genes.

Data normalization

Normalization is carried out gene by gene for all compounds from the CMAP and *in-house* data. The normalization of each vector of expression values $\boldsymbol{Y}_i$ for a gene i is performed by

$$\boldsymbol{Y}'_i = \frac{\boldsymbol{Y}_i - \bar{\boldsymbol{Y}_i}}{\sigma(\boldsymbol{Y}_i)},$$

where $\sigma(\boldsymbol{Y}_i)$ is the standard deviation of the vector of expression values, $\boldsymbol{Y}_i$.

12.6.2 Robust Plaid results

For obtaining robust results for each parameter setting, 100 runs were performed. Table 12.2 gives summary of the robust biclusters of Plaid.

The bicluster of coherent values is shown in Figure 12.3. The three genes of the bicluster are up-regulated in both groups.

12.6.3 FABIA results

We have chosen the exclusive-row option of FABIA algorithm, which restricts a given gene to be included in one bicluster. The number of cycles for FABIA was set to 5000

Table 12.2: Joint biclusters of Plaid

Bicluster	No. of in-house compounds	No. of CMAP compounds	No. of genes	Top active pathway
m	7	6	167	regulation of gene expression
$m + a$ 1	1	47	5	branched-chain aliphatic amino acid transport
$m + a$ 2	1	47	218	regulation of gene expression
$m + a$ 3	40	3	17	-
$m + b$ 1	5	53	2	-
$m + b$ 2	5	53	11	-
$m + a + b$	4	45	3	virus-infected cell apoptosis

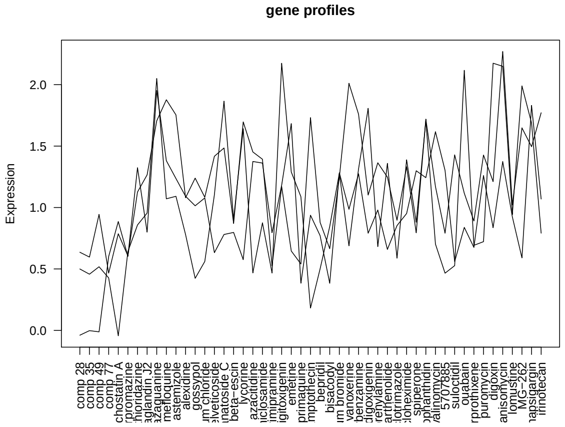


Figure 12.3: Gene profiles for a joint bicluster of coherent values discovered by Plaid.

and the number of biclusters was 25.

FABIA discovered 12 joint biclusters in the data. The biplots for the joint biclusters 3 and 4 (coded as 6 and 7 in the original FABIA output) are shown in Figure 12.4. On the plot a clear separation of a group of *in-house* and CMAP samples is observed and after applying threshold to factor scores the joint bicluster is confirmed.

For bicluster scores, the threshold of 2 has been chosen for both in-house and CMAP data samples and the threshold of 0.25 for factor loadings. The summary statistics and the top activated gene pathways from the CTG database are shown in Table 12.3. It must be mentioned that CTG does not contain pathway information for all compounds in CMAP, therefore the selected pathway are provided merely as indication. However, integration of various databases into the interpretation step gives some idea about the activity of compounds in a bicluster.

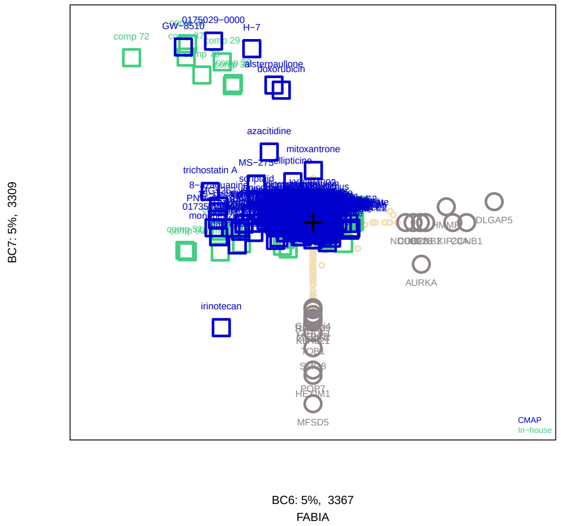


Figure 12.4: Biplot of the joint biclusters 3 and 4.

12.6.4 ISA results

The ISA bicluster was initialized with the four genes of scientific interest, however in the resulting bicluster only one of them was present. The compound scores of the *in-house* project for the given ISA bicluster were considerably lower than for the CMAP, which means that the expression values were higher within CMAP than within the *in-house* data set. Initialization with other random seeds did not result in joint biclusters either.

12.7 Discussion

In this chapter we presented a joint analysis of gene expression data obtained for an *in-house* compound set and the CMAP data. The aim of the analysis if to link between *in-house* compounds and CMAP compounds based on the similarity of the gene expression profiles. We have shown that unsupervised biclustering methods allow us to identify a subset of *in-house* and CMAP compounds for which subsets of genes are co-expressed.

The advantage of the biclustering is that it does not require prior information about pathways or groups of compounds in order to extract useful and biologically relevant

Table 12.3: Joint Biclusters of FABIA.

Bicluster	No. of in-house compounds	No. of CMAP compounds	No. of genes	Top CTG pathway(s)
1	10	23	49	apoptosis
2	8	43	23	signal transduction
3	17	36	19	apoptosis, immune system
4	7	10	28	apoptosis
5	11	44	11	apoptosis, immune system
6	54	29	6	apoptosis, pathways in cancer
7	7	45	9	immune system
8	1	62	7	immune system, signal transduction
9	8	31	8	apoptosis
10	8	51	6	apoptosis
11	11	70	5	signal transduction, immune system, apoptosis
12	5	28	1	apoptosis

information about underlying cellular processes, observed after compound treatment. However, in case the pathway or a group of genes is known, a semi-supervised approach can be used, such as ISA, FacPad, or FABIA with specific factor scores initialization. Biclustering in this case can be considered as the first exploratory tool to re-define the pathways or gene modules for further analysis with the standard methods implemented for CMAP data.

The power of biclustering analysis enables discovery of relevant pathways and gives an automated procedure for finding connection with CMAP gene expression profiles.

Chapter 13

Integrated analysis of miRNA and mRNA data

In case of the miRNA and mRNA data analysis we assume that it is possible to normalize expression profiles to carry out the simultaneous biclustering on the joint dataset and discover corresponding miRNAs and mRNAs for subsets of samples. We propose to use the FABIA biclustering, which takes into account local correlation patterns and selects subgroups of samples related to the miRNAs and mRNA co-regulation.

The NCI-60 dataset used for the illustration of the joint miRNA-mRNA analysis is presented in Section 13.1. In Section 13.2 we discuss the concept of the joint biclustering in the context of the joint miRNA-mRNA data. In Section 13.3, the normalization of the data and the analysis outline are discussed. The results are presented and discussed in Sections 13.4 and 13.5.

13.1 The NCI60 panel

The National Cancer Institute panel (NCI-60) consists of 60 human cancer tumor cell lines from 9 tissues of origin, including melanoma (ME), leukemia (LE), and cancers of breast (BR), kidney (RE), ovary (OV), prostate (PR), lung (LC), central nervous systems (CNS), and colon (CO). The gene expression profiling was performed on several platforms (Gmeiner *et al.*, 2010). However, for the current analysis we use Agilent data according to Liu *et al.* (2010). Specifically, the expression levels of $\sim$

$21,000$ genes and 723 human miRNAs were measured by 41,000-probe Agilent Whole Human Genome Oligo Microarray and the 15,000-feature Agilent Human microRNA Microarray V2.

13.2 Joint biclustering of miRNA and mRNA data

Consider n samples, from which both miRNAs and mRNAs have been extracted and profiled. The two datasets are stored as two matrices: $\boldsymbol{X}_1$ of size $m_1 \times n$ and $\boldsymbol{X}_2$ of size $m_2 \times n$, respectively. The datasets share the sample dimension (e.g. samples come from the same biological source), and we are interested in discovering a link between miRNA and mRNA expression. Thus, the joint miRNA-mRNA data matrix $\boldsymbol{Y}$ is a $(m_1 + m_2) \times n$ matrix:

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{bmatrix}.$$

We assume that there is a subgroup of samples with similar molecular profiles in both datasets. The biclustering methods are run on a joint data matrix to discover this subgroup together with a subgroup of corresponding miRNAs and mRNAs. The example is schematically illustrated in Figure 13.1.

The choice of FABIA as a biclustering algorithm for the current analysis is based on its ability to discover both positively and negatively correlated expression profiles within one bicluster. The factor loadings of FABIA can indicate whether the values within a bicluster are correlated or anti-correlated. Figure 13.2 shows loadings for FABIA biclusters run on a hypothetical dataset of gene expression and miRNA expression separately and jointly. The joint bicluster has negative loadings for genes and positive loadings for miRNAs, which reflects their anti-correlation.

13.3 Methods

13.3.1 Data preprocessing

Filtering of miRNAs and genes

Similar to the procedure of Liu *et al.* (2010), the quantile filtering is used. The idea is to select highly expressed genes with high variability, which are potentially interesting for the biclustering procedure (Bourgon *et al.*, 2010).

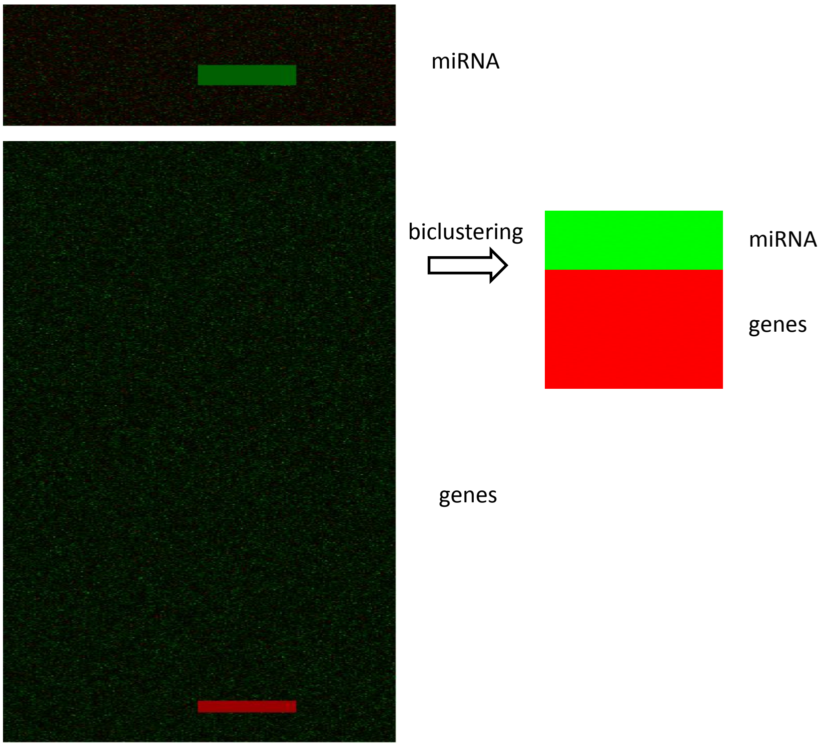


Figure 13.1: An illustrative example of joint biclustering on two datasets: miRNAs and mRNAs are joined by subjects and analyzed by biclustering. The output is a bicluster, containing both miRNAs and mRNAs expression.

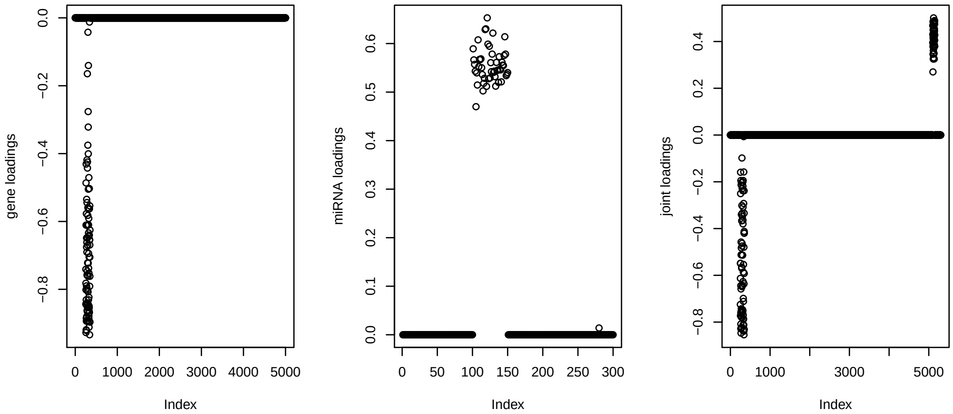


Figure 13.2: A hypothetical example of bicluster loadings based on FABIA output. Left panel: gene loadings; central panel: miRNA loadings; right panel: joint loadings.

Let m be the total number of genes and $\boldsymbol{Y}_i, i = 1 \ldots m$ be the vector of expression values for the ith gene. Compute maximum $max(\boldsymbol{Y}_i)$ and inter-quartile range $IQR(\boldsymbol{Y}_i)$ for each gene and $q_{max} = q_{0.75}(max(\boldsymbol{Y}_i))$ and $q_{IQR} = q_{0.75}(IQR(\boldsymbol{Y}_i))$. If $max(\boldsymbol{Y}_i) > q_{max}$ and $IQR(\boldsymbol{Y}_i) > q_{IQR}$ then the gene is selected, otherwise it is excluded from further analysis. For the NCI-60 data, this filtering step resulted in $2,455$ genes. The 422 probes of miRNAs summarized by the median to the unique 306 miRNA IDs were used for the analysis.

Normalization of datasets

The miRNAs are known to be in lower abundance in a cell compared to the mRNA, which is reflected in the intensity values. Therefore, all rows in the joint data are normalized to have mean zero and variance equal to one.

13.3.2 Joint biclustering

FABIA for joint miRNA-mRNA data

As mentioned in Section 13.2, miRNA and mRNA expression values are supposed to be anticorrelated and $\boldsymbol{\lambda}_k$ in model (7.3) can be either positive or negative, therefore, FABIA should be highly suitable for the discovery of joint miRNA-mRNA modules. After the FABIA solution is obtained, the extraction of biclusters is performed. The contribution of miRNAs and genes to each FABIA bicluster is evaluated based on the values of factor loadings. The higher a loading value for a given miRNA/mRNA, the more relevant this miRNA/mRNA is for a given module. In general, FABIA biclusters are so-called soft biclusters. Therefore, a threshold is needed to obtain a joint miRNAs-mRNAs module.
To select the threshold and get joint miRNA-mRNA modules from the joint data we use the following procedure.

1. For each bicluster k, get miRNA and mRNA loadings $\boldsymbol{\lambda}_{miRNAs}$ and $\boldsymbol{\lambda}_{genes}$; and obtain 100 quantiles of both vectors. Let q^{-}_{max} be the maximum non-zero quantile negative loadings values and q^{+}_{min} be the minimum non-zero quantile for the positive loadings values.

2. Plot the loadings and select the most extreme miRNAs loadings in absolute values. In real life data, most of the $\boldsymbol{\lambda}_{miRNAs}$ will be zero or close to zero. Hence, we define a threshold $\tau = max(|q^{-}_{max}|; q^{+}_{min})$.

3. Similar to previous step, select the most outlying positive and negative mRNA loadings.

The extraction of cell lines belonging to a bicluster is based on the most outlying factor scores, both positive and negative, for a given bicluster. Thus, the joint miRNA-mRNA module is determined by miRNAs and mRNAs, which have opposite (negatively correlated) expression patterns on a subset of samples.
It should be mentioned, that if all loadings are close to zero, i.e., when τ is close to zero neither miRNA nor miRNAs should be extracted, the same holds for the factor scores.

13.4 Results

13.4.0.1 Obtaining hard biclusters from the FABIA solution

In total, 25 biclusters have been discovered in the NCI-60 panel data. In this section we illustrate how the joint miRNA-mRNA modules have been extracted based on the loadings and scores from the solution of FABIA. The boxplots of miRNA and mRNA loadings as well as the cell lines' scores are shown in Figure 13.3. Let us illustrate the first two factors from the FABIA solution, i.e., the first two columns of the loadings matrix $\boldsymbol{\lambda}$ and the first two rows of the scores matrix $\boldsymbol{Z}$. Figure 13.4 shows sorted factor loadings and scores for the two biclusters. The first bicluster is dominated by mRNAs, while miRNAs have considerably lower values for the loadings. On the other hand, the second bicluster has comparable loadings for both miRNAs and mRNAs, which contribute to the joint bicluster. To select the miRNAs and mRNAs for the joint miRNA-mRNA module, we set the threshold for miRNA and mRNA loadings to 0.25. The factor scores of the first bicluster have high positive and negative values, whereas in the second bicluster only negative scores have extreme values. As it can be observed from the factor scores of the second bicluster, seven cell lines pass the threshold of -1.

13.4.0.2 Joint miRNA-mRNA modules discovered in the NCI-60 data

After the joint miRNA-mRNA modules extraction procedure, 10 joint modules have been discovered. The joint biclusters are presented in Table 13.1. As we can see from Table 13.1, the joint miRNA-mRNA modules 3, 4, 9 and 10 have overlap in columns dimensions. In addition, the miRNA let-7a is present in joint miRNA-mRNA modules 3, 7, 8 and 10. Several studies confirm activity of let-7a miRNAs in cancer (Kim

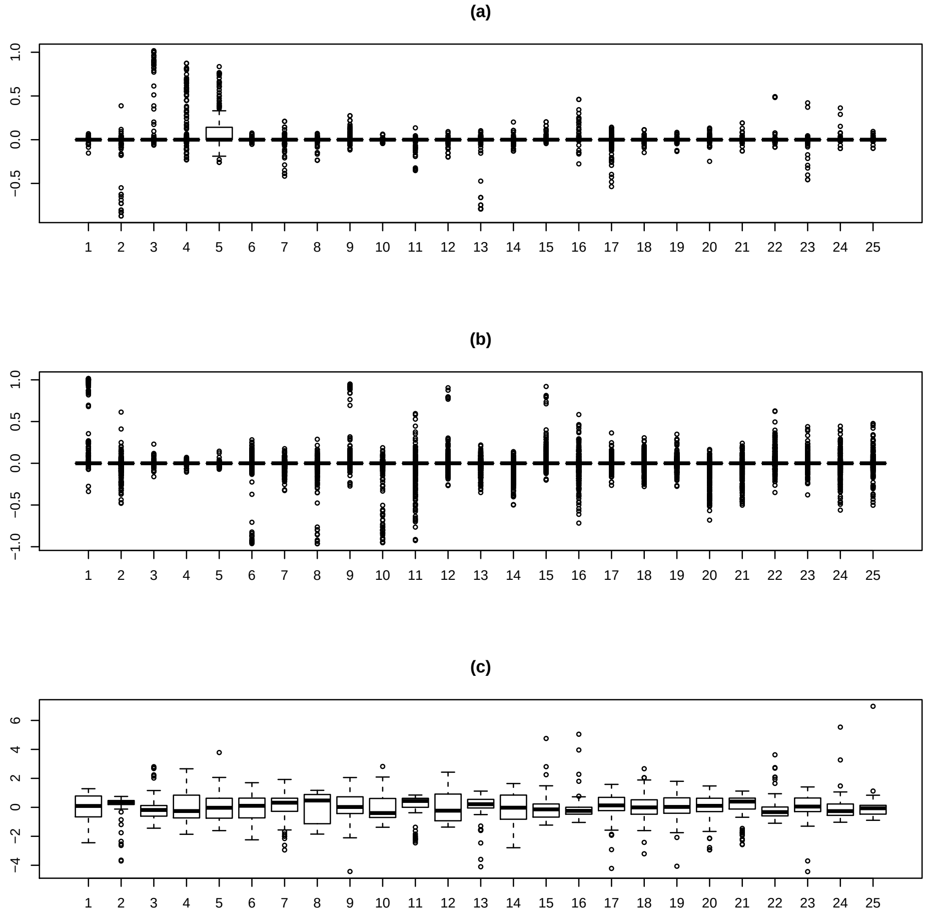


Figure 13.3: The boxplots of loadings for miRNA and mRNA, and scores for cell lines for 25 factors of FABIA: (a) miRNA loadings; (b) mRNA loadings; (c) scores.

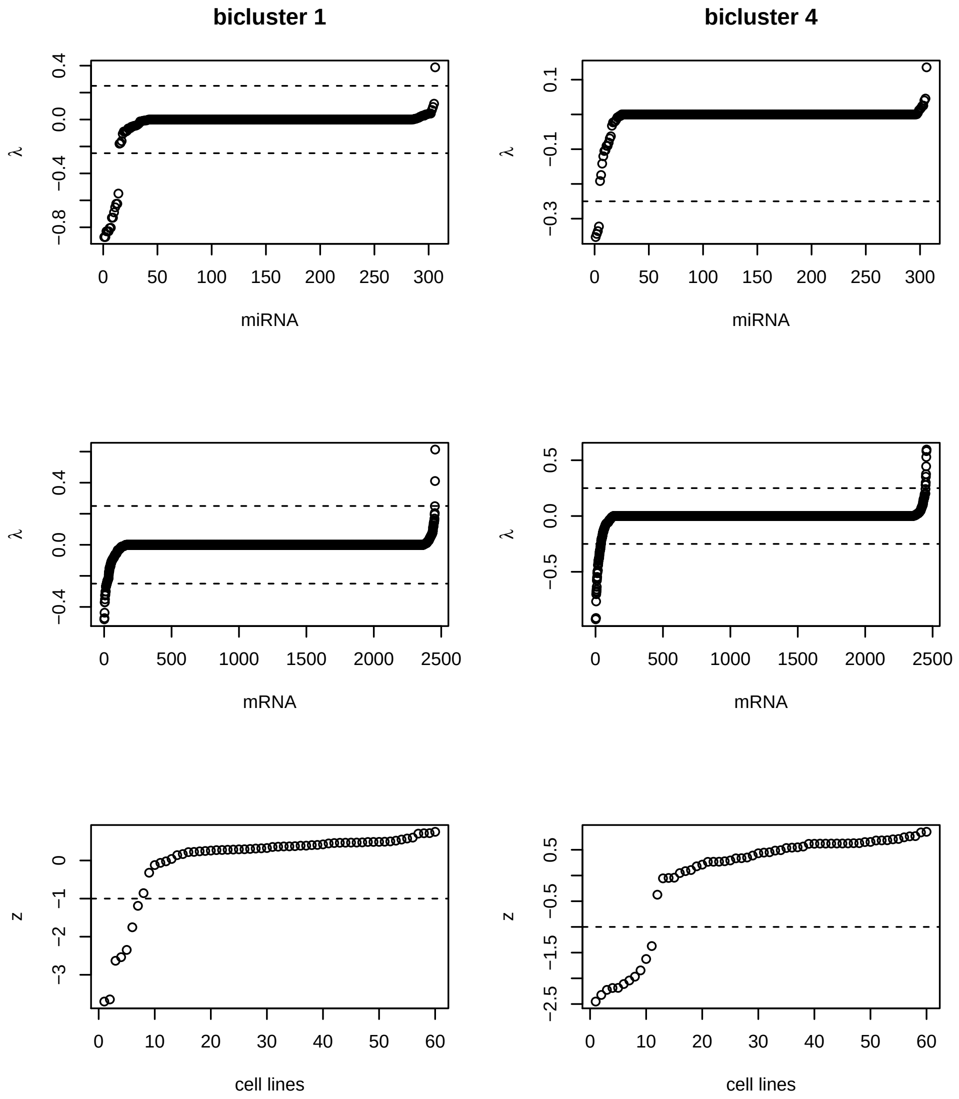


Figure 13.4: Sorted factor loadings for miRNAs and mRNAs, sorted factor scores for cell lines. Left panel: joint bicluster 1, right panel: joint bicluster 4. The dashed lines represent the cut-off values: ±0.25 for factor loadings and ±1 for factor scores

et al., 2012). In what follows we discuss two joint modules in more details.

Table 13.1: NCI-60 dataset. Joint miRNA-mRNA modules.

Bicluster	No. of miRNA	No. of genes	Cell lines
1	15	21	Skin cancer: MALME_3M, _MEL_2, SK_MEL_28, SK_MEL_5, UACC_257, UACC_62, MDA_MB_435, MDA_N
2	7	5	Breast cancer: HS578T, BT_549, CNS-glioma: SF_295, SF_539, SNB_75, U251, lung cancer: A549, HOP_62, and Renal cancer: ACHN
3	2	22	Breast cancer: MCF7, colon cancer: KM12, leukemia: K_562, lung cancer: A549, NCI_H522, ovary cancer: IGROV1, OVCAR_3
4	4	48	Breast cancer: MCF7, T47D, colon cancer: COLO205, HCC_2998, HCT_116, HCT_15, HT29, KM12, lung cancer: NCI_H322M, ovary cancer: OVCAR_3, OVCAR_4
5	9	7	CNS-glioma: SF_268, colon cancer: HCT_15, leukemia: K_562, ovary cancer: OVCAR_3, OVCAR_4, OVCAR_8, NCI_ADR_RES, renal cancer: SN12C
6	5	42	Leukemia: CCRF_CEM, HL_60, MOLT_4, RPMI_8226
7	6	2	Colon cancer: HCC_2998, melanoma: LOXIMVI, SK_MEL_5, leukemia: CCRF_CEM, K_562, RPMI_8226, SR
8	4	25	Colon cancer: COLO205, HCC_2998, CO.HT29, CO.KM12, SW_620, lung cancer: A549, ovary cancer: OVCAR_5
9	7	6	Breast cancer: MCF7, T47D, leukemia: CCRF_CEM, K_562, melanoma: MDA_N
10	2	26	Breast cancer: MCF7, colon cancer: HT29, leukemia: HL_60, K_562, RPMI_8226

Module 1

Module 1 has a clear separation of eight melanoma cell lines from the rest (Figure 13.5). The cell line LOXIMVI does not share similar molecular profile on selected mRNAs and miRNAS. The bicluster genes NODAL, GHRH, CARD11 belong to the cell adhesion and cytokine production pathways. In addition, the gene GLO1 (A_32_P53822) is related to malignant melanoma, however, this gene was downregulated in the eight cell lines determining this miRNA-mRNA module. Furthermore, the upregulated genes in the bicluster CHCHD1 (A_32_P51082), FN1 (A_32_P201723), C14orf10 (A_32_P18357), C1orf80 (A_32_P182662) are known to be related to the skin cancer (Uhlen *et al.*, 2010). The miR-96 downregulated for the eight melanoma samples is known to inhibit melanoma cell activity (Poell *et al.*, 2012). The miR-506-514 cluster, which has overexpression, is known to be active in melanoma samples as well (Streicher *et al.*, 2012).

Module 4

Module 4 is represented by a subgroup of epithelial cell lines: ovarian (OVCAR_3, OVCAR_4), colon (COLO205, HCC_2998, HCT_116, HCT_15, HT29, and KM12), lung (NCI_H322M) and breast (MCF7, T47D). It contains genes and miRNAs known to be involved in the epithelial-mesenchymal transition, e.g., the miRNAs hsa-mir-141, hsa-mir-200c, hsa-mir-205 suppress via ZEB the expression of the gene VIM, which is an effect of miRNA regulation (Miska, 2008). The data have shown that there is a high anti-correlation between the gene VIM and miRNAs (Figure 13.6).

In addition to the observed differences in the expression values of the miRNAs and mRNAs within and outside the cell lines included in the joint modules 1 and 4, we compare a module-specific local correlation with a global correlation structure. Figure 13.7 shows comparison of global correlations (the values are obtained from all samples) and local correlations (values are obtained within bicluster samples only). For module 4 there is difference in the miRNA and mRNA expression values (as shown in Figure 13.6), but there was no module-specific local correlation. Furthermore, in module 1 we observed higher anti-correlation values within the bicluster cell lines in addition to the difference in expression levels (see Figure 13.5). Thus, next to the changes in expression levels, FABIA can determine the joint miRNA-mRNA modules with the differences in correlation structure.

13.5 Discussion

We have presented a procedure for the discovery of joint miRNA-mRNA modules by a biclustering technique, FABIA. While FABIA has been successfully applied to and was shown to discover relevant modules in gene expression data, the discovery of joint miRNA-mRNA modules is carried out for the first time. Two important aspects of the joint FABIA biclustering are addressed in this work: the normalization of the miRNA and mRNA expression data and post-processing of the FABIA loadings to extract anti-correlated miRNA-mRNA modules. The normalization step disturbs the original expression values and the so called *flat* expression profiles have a chance to enter the bicluster. For this reason, we have filtered the small variance genes and miRNAs prior to the normalization.

The advantages of applying FABIA on joint miRNA and mRNA expression data are: (1) it is fully unsupervised method, which does not depend on some external grouping of cell lines (or any other conditions), genes and miRNAs; (2) it uses correlation structure of the data and extracts correlated as well as anti-correlated expression patterns; (3) compared to the global correlation analysis, FABIA can detect local correlation patterns and in some cases can be computationally more attractive than global correlation analysis on all samples in the data; and (4) FABIA allows for a two-dimensional overlap of modules, i.e., allowing miRNAs and genes to be a part of multiple modules as well as various conditions affecting multiple regulatory pathways.

While the proposed analysis is a very promising exploratory tool for paired miRNA-mRNA data, it has certain limitations. At the moment we cannot fully automate the process of extraction of miRNA-mRNA modules. Therefore, as a solution we use a semi-automated way for the extraction of samples, genes and miRNAs belonging to the joint miRNA-mRNA modules based on the plots and quantiles of loadings and scores from the FABIA output. Furthermore, it should be pointed out that FABIA assumes the same correlation structure within miRNA and mRNA data, which might not necessarily be the case in the data. To further extend the method for joint miRNA-mRNA modules discovery, we can use the loadings of miRNA and mRNA from FABIA as the input parameter for the integrative factor analysis (iFAD), which uses two separate correlation matrices for the estimation of loadings and scores (Ma and Zhao, 2012b).

In conclusion, we believe that joint biclustering is a technique which can be used for the exploratory analysis of multiple omics datasets, especially in the setting when supervised analysis cannot be carried out due to heterogeneous conditions or incomplete annotation of grouping variables, such as cancer subtype, treatment with compounds of various nature, and others.

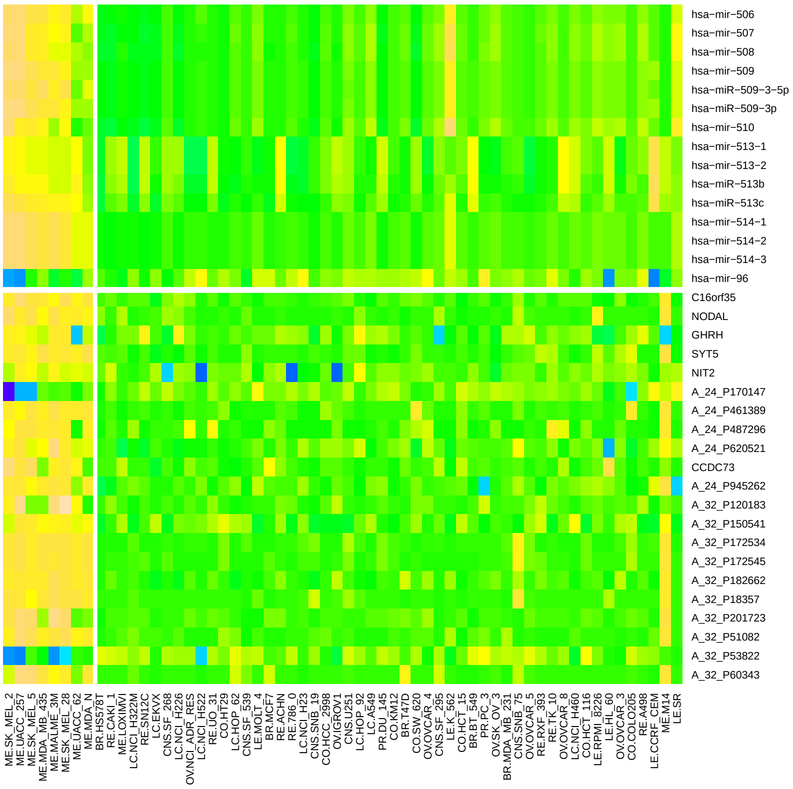


Figure 13.5: Heatmap of normalized expression values for features in the joint miRNA-mRNA module 1. The eight melanoma cell lines defined by the module show anti-correlation pattern for miR-96 and A_32_p51082 (CHCHD1), A_32_p201723 (FN1), A_32_P18357 (C14orf10), A_32_P182662(C1orf80); the anti-correlation is detected for the miRNA-506-514 cluster with A_24_P170147 and A_32_P53822.

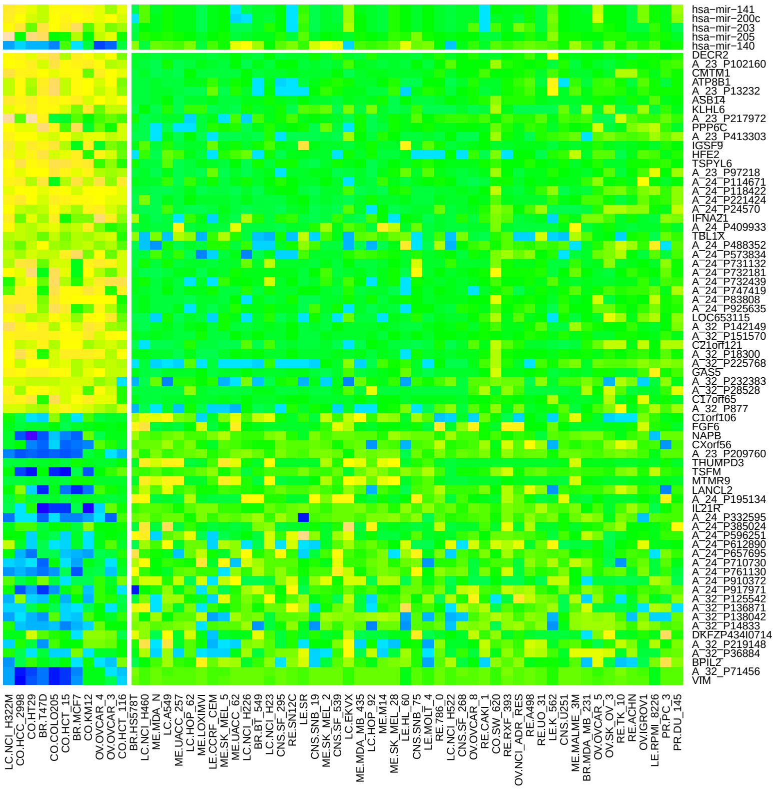


Figure 13.6: Heatmap of normalized expression values for features in the joint miRNA-mRNA module 4. The miRNAs 141, 200c, 205 are known to be involved in the epithelial-mesenchymal transition and regulate the expression of VIM gene via ZEB. Note that ZEB gene expression was not available in the Agilent gene expression dataset.

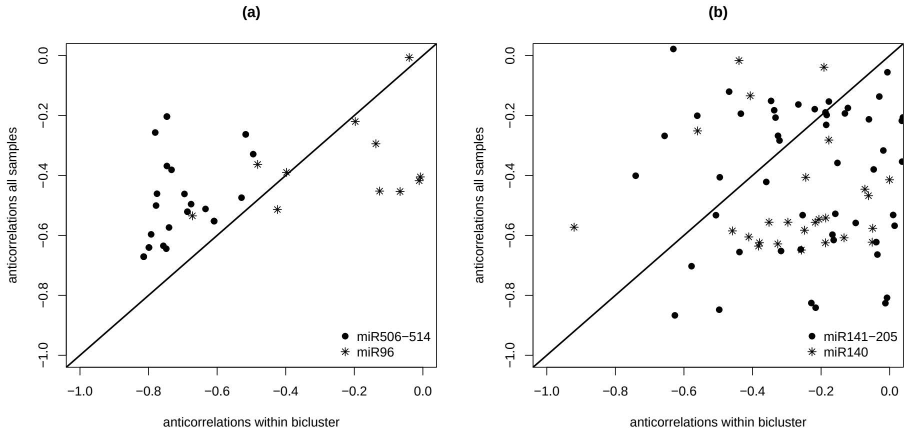


Figure 13.7: Comparison of global and local miRNA-mRNA anti-correlations based on joint miRNA-mRNA modules obtained by FABIA: (a) joint miRNA-mRNA module 1; (b) joint miRNA-mRNA module 4. The joint miRNA-mRNA module 1 has higher anti-correlation within bicluster compared to the global anti-correlation; the joint miRNA-mRNA module 4 has comparable anti-correlation within bicluster samples and global anti-correlation.

Chapter 14

Joint biclustering analysis of discrete data

In the previous chapters we have focused on the analytic tools for integration of expression measurements on continuous scale and the interpretation of resulting biclusters. In this chapter we present a case study for the joint analysis of discrete and continuous data. Discrete data, which we analyze in this chapter, are the matrix of fingerprints introduced in Chapter 11. The underlying assumption is that there is a relationship between compound substructure and its transcriptional and phenotypic activity. In order to discover this relationship, the data integration is essential. Similar to the situations before, joint biclustering is an analysis method that can be used to explore relations between different datasets.

The analysis presented in this chapter is based on a joint gene expression-fingerprint dataset obtained from a specific drug development project. The fingerprints of compounds are introduced in Section 14.1. In Section 14.2 we describe joint biclustering setting for the case study. For the integration of gene expression and fingerprint data, both datasets should be standardized. Section 14.3 addresses strategies for gene expression data discretization and binarization. Section 14.4 describes algorithms used for the analysis and their parameter settings. The results are presented in Section 14.5 followed by a discussion in Section 14.6.

14.1 Fingerprints of a compound set

The analysis presented in this chapter is a part of a large-scale discovery project (QSTAR), which aims at developing novel methodologies for the integration of three high-dimensional data types: gene-expression, chemical-structure and bioassay data. Fingerprint features are one form of the local compound encoding, where the whole molecule is described as a list of local substructure patterns (fingerprint features or FPFs). Within QSTAR, the Extended-Connectivity Algorithm (ECFP) for calculating FPFs for compounds was used, since this method performs well for most chemical questions in encoding the chemical concepts for further data mining.

The ECFP fingerprint algorithm starts from a single atom and expands in circles (topological in the 2D graph) to the next level of connected atoms. It captures the bond orders of the bonds between atoms and ensures substructures are normalized, which assigns each substructure a unique number. The expansion of the spheres stops by default after eight expansion steps, i.e., ECFP8 FPFs are calculated. This process is then started from every single atom in a compound. A detailed description of the algorithm can be found in Rogers and Hahn (2010) and the schematic view is presented in Figure 14.1.

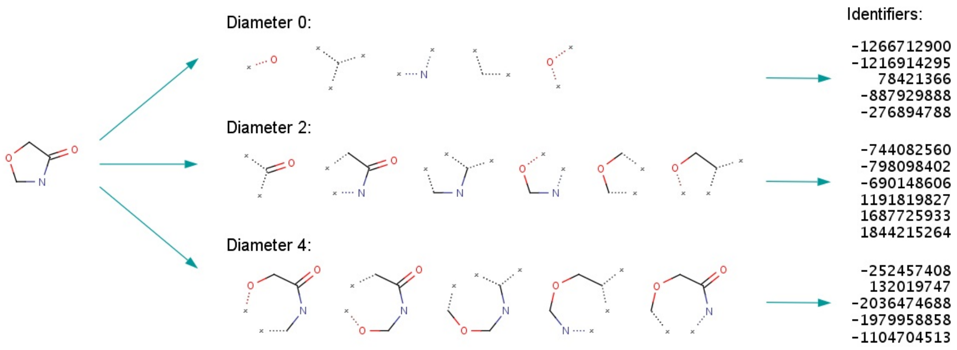


Figure 14.1: The ECFPx fingerprint calculation method starts at a single atom and expands iteratively to the next topological level of connected atoms till a maximum threshold x is reached (the default setting for x is 8). The fingerprint of a whole molecule is a combination of all FPFs executing this method from every single atom. Source: ChemAxon (2013).

From the description of the algorithm follows that, if a larger fingerprint feature is present, then all substructures of it are also present. When fingerprint features are calculated for a set of compounds, a binary matrix $\boldsymbol{V}$ is obtained, where each column represents a compound and each row is a value of fingerprint feature. If the differences between compound structures are not large, then many rows will have the same entries. These entries can be removed from the data matrix for the analysis since they do not carry any information with respect to the differentiation of a structure. The resulting number of rows in the matrix $\boldsymbol{V}$ varies and is compound-set dependent.

14.2 Joint biclustering for chemical structure and gene expression data integration

Since fingerprint features are used to describe compound structures, the main question that arises is, if we can find a subset of fingerprint features, which is responsible for a certain change in gene expression across a subset of compounds. The subgroup of compounds and genes is not known in advance and biclustering is a natural choice for the exploration in this setting. Going one step further, discrete biclustering can jointly explore fingerprint-features and gene-expression data, discovering biclusters of fingerprint features related to the gene expression. Integrative analysis of binary fingerprint features and gene expression is numerically similar to the integration of SNP and expression data and the methods developed and well-investigated in the former setting can be translated to the latter one.

Let $\boldsymbol{V}$ be a $m_f \times n$ matrix of fingerprint features and $\boldsymbol{X}$ be a $m_g \times n$ discrete matrix of gene expression for n compounds in a study. The joint $(m_f + m_g) \times n$ data matrix $\boldsymbol{U}$ is used to find the submatrices of binary values containing both genes and fingerprint features.

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{X} \end{bmatrix}$$

An alternative way to integrate fingerprint features with gene expression data is to binarize gene expression according to the presence or absence of a particular fingerprint. The general idea is to construct a gene expression signature of each fingerprint feature. The genes in a signature are coded as 1 and not in a signature are coded as 0. Afterwards a discrete biclustering method, such as iBBigs or QUBIC can be applied. A bicluster in this setting is a subset of fingerprint features (a certain substructure), for which a subset of genes are differentially expressed given the set of compounds in the experiment.

14.3 Preprocessing of gene expression data

A number of strategies can be used for obtaining discrete or binary values for the original gene expression data. In the first step, the fold changes are obtained for the gene expression with respect to the control samples. Afterwards, one of the possibilities, such as quantile-based discretization, up- and down-regulation discretization, or binarization can be chosen depending on the biclustering method of choice.

14.3.1 Binarization

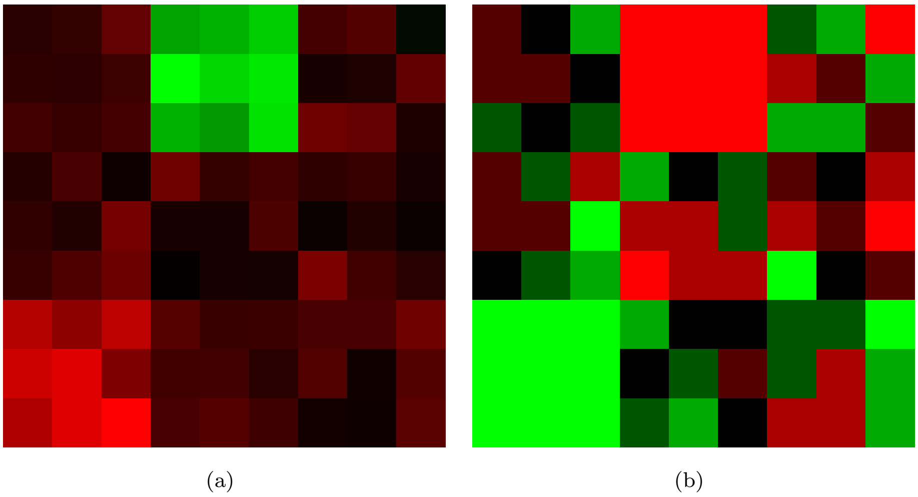
The binarization of the data was originally related to the hypothesis test of differential expression. However, in a typical compound profiling experiment replicates are not available and no testing is possible. Therefore, a gene is called expressed and coded as 1 if the absolute value of its fold change passes a given threshold τ, otherwise it is called unexpressed and coded as 0. The threshold τ is data-dependent and should be chosen after studying the data distribution, the overall distribution of expression values in a dataset, as well as the gene expression variability per compound.

14.3.2 Up- and down-regulation

This type of discretization requires a threshold for calling a value up- or down-regulated. The threshold is data-dependent and should be chosen with respect to the underlying data distribution. If a given value of fold change is larger than $\tau_1 > 0$, then we call it up-regulated and code as 1, if it is smaller than $\tau_2 < 0$, we call it down-regulated and code as -1, otherwise a value is called unexpressed and coded as 0.

14.3.3 Quantile-based discretization

For biclustering methods, looking for specific patterns or motifs in the data, such as xMOTIF, gene expression data should be discretized in a number of levels Q, fixed upfront. The quantile discretization computes Q quantiles in the data and replaces the original gene expression values with the quantile numbers. To illustrate how quantile discretization is performed, we simulate a 9×9 dataset of fold changes, where three genes are upregulated in three samples and three genes are downregulated in three samples. The heatmap of the data is shown in Figure 14.2(a). The quantiles of the data matrix suggest that a high number of levels for discretization is needed, for example, $Q = 7$ proves to be suitable for further analysis (Figure 14.2(b)), so that a pattern-matching algorithm can discover groups of up-regulated and down-regulated genes.



(a) (b)

Figure 14.2: Hypothetical gene expression data. Fold changes (a) before and (b) after quantile discretization.

14.3.4 Data merging

After the gene expression data have been converted to binary or discrete values, the datasets are ready to join. The common dimension in both datasets is compounds dimension, thus, merging occurs by columns. Each column in a merged dataset contains m_g gene expression values and m_f fingerprint features values, which results in a $(m_f + m_g) \times n$ data matrix. It should be noted, that the matrix of fingerprint features may contain duplicates due to similarity of compound structures under investigation or computation of fingerprint features. Duplicates of the FPFs should be removed in order to refine search of the biclusters in the vectors of unique fingerprint features.

14.4 Biclustering methods

For the joint analysis of gene expression data and fingerprints, we have selected two methods: QUBIC (Li *et al.*, 2009) and iBBiG (Gusenleitner *et al.*, 2012). The BiMAX (Prelic *et al.*, 2006) and xMOTIFs (Murali and Kasif, 2003) methods were less performing in a number of studies and even though originally these methods were applied to the data, we do not present the results here.

The joint biclustering analysis is performed under an assumption that there is a group of compounds with a subset of common features, which induce similar gene expression activity.

14.4.1 QUBIC

QUBIC (Li *et al.*, 2009) is a biclustering method working on discretized dataset. We have chosen three levels of discretization based on data quantiles. The method looks for biclusters where each pair of rows has either the same discrete or the values of the opposite sign (thus, including down- and up-regulated expression values) with a possibility of some error. It is a desirable feature of the algorithm, taking into account that fingerprint features may not always be present for all compounds in a subgroup, or the discretization process may result in errors.

The maximum number of biclusters has been set to 200, the tolerance parameter to 0.9, and the biclusters with more than 0.75 overlap have been filtered out.

14.4.2 iBBiG

iBBiG is a biclustering method for gene signatures for certain conditions (Gusenleitner *et al.*, 2012). We apply binarization procedure with a threshold of $\tau = 0.75$ to run the algorithm.

Most of the parameter settings were default, since they do not affect the performance of the algorithm according to Gusenleitner *et al.* (2012). We have set the number of biclusters to 100, however, the algorithm has its own criterion for evaluation of the actual number of biclusters, which can be lower.

For the final ranking of the iBBiG biclusters, we use the Fisher's exact test for the enrichment of 1's within a bicluster compared to the columns outside a bicluster given bicluster rows. Let a bicluster contain m_I rows and n_J columns. Then, for each bicluster we construct a contingency table (Table 14.1), where $n_{0w} + n_{0out} + n_{1w} + n_{1out} = m_I \times n_J$. Afterwards, the Fisher exact test is performed for each constructed contingency table. The ranking is carried out according to the lowest p-value of a bicluster.

Table 14.1: A contingency table used to obtain the ranking score of an iBBiG bicluster.

	0's	1's
within bicluster	n_{0w}	n_{1w}
outside bicluster	n_{0out}	n_{1out}

14.5 Results

The resulting dataset contained 35 samples, 137 unique fingerprint features and 595 genes, which passed the fold change filtering criterion of $FC > 0.5$ in at least one out of 35 samples.

14.5.1 QUBIC results

The QUBIC discovered 56 biclusters, out of which 27 were identified as joint biclusters. From the resulting joint biclusters we selected the biclusters with more than two compounds. The summary of the biclustering results are presented in Table 14.2.

Table 14.2: The joint biclusters of QUBIC with more than two compounds.

Bicluster	No. of FPFs	No. of genes	No. of compounds
1	7	390	7
2	28	353	6
3	21	225	9
4	5	202	10
5	5	182	11
6	18	293	5
7	7	203	4
8	27	26	13
9	7	80	7
10	45	22	9
11	5	47	10
12	29	53	4
13	9	1	26
14	14	33	5
15	2	21	9
16	45	4	4
17	36	1	4
18	17	2	6
19	20	16	3
20	12	15	3

The joint bicluster 15 is plotted in Figure 14.3. Since a number of 0's is tolerated, the last compound has two fingerprint features missing. Both correlated and anti-correlated genes are included in the bicluster.

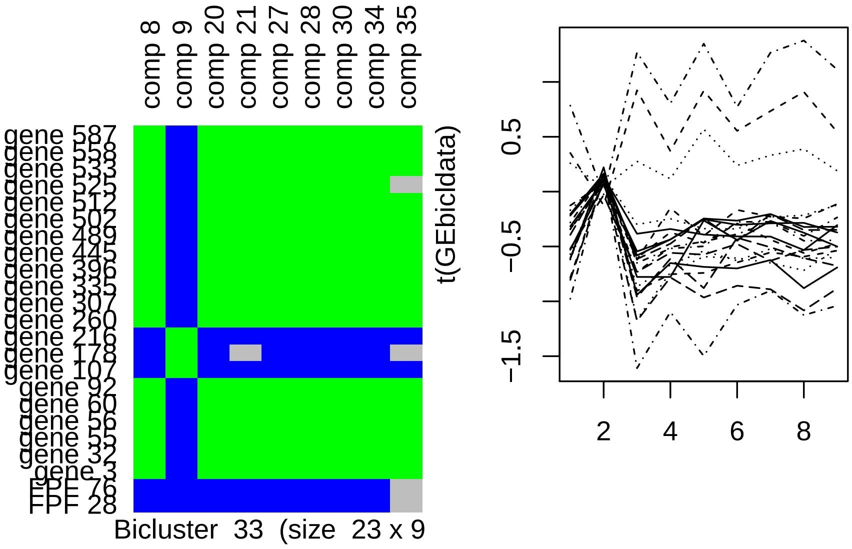


Figure 14.3: The joint bicluster 15 of FPF and gene expression. For FPFs, gray represents 0, blue represents 1; for gene expression gray is for non-expressed genes, blue is for upregulated genes, green is for down-regulated genes.

14.5.2 iBBiG results

Only one joint bicluster with more than two compounds was discovered by iBBiG when the threshold was set to 0.75. It contained four FPFs and 101 genes. The remaining joint biclusters had only two compounds and are not listed here.
Relaxing binarization threshold to 0.5 resulted in larger biclusters. The summary is given in Table 14.3, where we output only larger biclusters, in terms of columns. The rank given is related to the overall ranking of 41 joint biclusters.

Table 14.3: The joint biclusters of iBBiG.

Bicluster	No. of FPFs	No. of genes	No. of compounds	Fisher test	Rank
1	22	202	8	0	1
2	24	1	5	1.3×10^{-10}	8
3	20	47	3	3.8×10^{-93}	26
4	10	2	6	10^{-24}	3
5	18	1	6	7×10^{-40}	11
6	11	1	3	6×10^{-14}	22
7	4	1	3	1.5×10^{-4}	13

The top ranked bicluster is shown in Figure 14.4. The compounds of this bicluster are a subset of the reported QUBIC bicluster compounds. For the compounds 34 and 35 most of the FPFs are absent, while being present in the others.

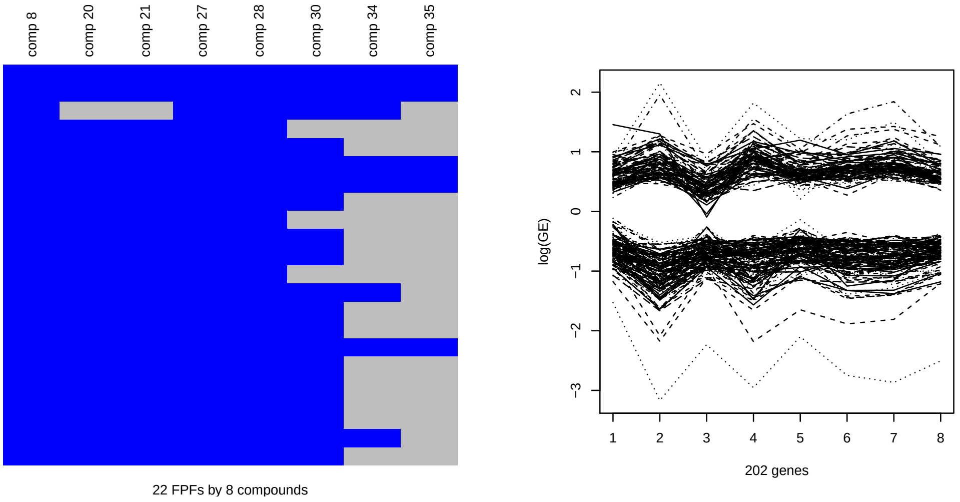


Figure 14.4: Joint bicluster 1 of FPF and gene expression. Grey represents 0 (absent FPF), blue - 1 (present FPF).

14.6 Discussion

One of the most important question in the early drug development projects is if the chemical structure of the compounds set under investigation influences gene expression. An identification of genes whose expression levels are influenced by certain chemical structures (or substructures) can lead to better understanding of desirable or undesirable properties of compounds under investigation. In this chapter we have presented the approach of joint exploration of chemical structure and gene expression profiles of therapeutic compounds. We have discussed the approaches to discretize gene expression data to make it comparable to the FPF data and to perform the joint biclustering. We have applied recent biclustering methods designed for the discrete data. The QUBIC biclustering deserves a special attention in the exploration of this type of datasets: it can discover both up- and down-regulated genes together with present or absent FPFs for a given subset of compounds. The iBBiG biclustering looks for biclusters of expressed genes and the present fingerprint features. Both biclustering algorithms allow 0 entries within a bicluster, which is a desirable property, that some other methods, such as BiMAX and xMOTIF, lack.

The QUBIC biclusters are, in general, larger in the number of compounds present, and the iBBiG bicluster set contained many biclusters with only two compounds. The iBBiG method was designed to look for the subtle signal potentially masked by some stronger signal in the sparse binary data. Therefore it is expected that more biclusters of smaller size can be identified in the joint FPF-gene expression dataset.

Chapter 15

Conclusions and future research

15.1 Part I: Probe-level analysis of Affymetrix microarray data

In Part I we have presented the modeling framework for the probe level analysis of Affymetrix GeneChip data. The modeling framework is based on the linear mixed models with experiment-specific and random effects. We have shown how the LMM framework can be applied to the different experimental settings.

The most straightforward application of the probe-level LMM is the differential gene expression analysis. The probe-level LMM accounts for the correlation of probes intensities and provides estimates for the fixed effects of interest. Moreover, estimates of the between-array variance can be used to infer about unaccounted factors. This type of information is lost after the summarization step. Hence, the probe-level LMM can be used to perform a quality control of the data and indicate the presence of technical artifacts, such as batch effects, or clusters of samples according to the similar biological origin, such as age or gender.

In a simulation study we have shown that if the probe level data satisfies marginal normality, then probe-level LMM had the same power and FDR as the differential expression tests on summarized data, with an exception of the setting with small sample size (12) and small probe set size (11), where the probe LMM had slightly higher power.

Using the information obtained from the probe-level LMM, the experiment-specific probe sets can be selected, i.e., the probe sets which are after the adjustment for the experimental factors have low between-array variation. In general, four types of probe-set patterns have been observed in the real-life data sets: (1) non-informative probe sets, which have low agreement in probe intensities, reflected in low adjusted and unadjusted ICCs; (2) experiment-specific probe sets, which have correlated probe intensities caused by an experimental factor, reflected in high unadjusted ICCs and low adjusted ICCs; (3) probe sets, sensitive to technical artifacts, which have correlated probe intensities caused by some technical artifact, reflected in high adjusted and high unadjusted ICCs; (4) hybrid probe sets, which have correlated probe intensities due to both experimental factors and technical artifacts.

The LMM framework can be further extended to model the probe sets, where the assumption of all probes capturing the same target transcript is violated. A mixture model was used in order to estimate the number of components in a probe set, and the posterior probabilities were used to obtain the component membership (class label) for each probe in the probe set. The class labels were further used for the definition of the variance-covariance structure in the probe-level LMM. The estimates of the within-class ICCs were used for the updated definition of probe sets. We have illustrated how experiment-specific probe set definition can be performed based on the output of the probe-level LMM. For example, when a probe set is composed of multiple components, it can be divided into several different probe sets. Afterwards, these probe sets can be used for the update of the chip definition files (CDF). Further, the standard downstream analysis of the gene expression data (summarization, test for differential expression etc.) can be carried out based on the updated CDFs.

Microarrays have been applied in the gene expression experiments for almost two decades, and most of the research questions related to the analysis of microarray data have been solved. Nevertheless, there are several remaining research questions to consider. For example, to conduct a permutation-based method for the probe-level analysis, the runtime of the probe-level SAM should be optimized. It can result in a higher application rate of the probe-level analysis or included as a standard microarray analysis tool. Furthermore, a larger simulation study should be performed to check the performance of the probe-level LMM in the setting of non-normally distributed probe intensities. In addition, it is advisable to develop a software package, containing a wrapper of the probe-level LMM for the automated analysis in more general settings, other than considered in this dissertation. In the analysis presented in the first part of the thesis, the RMA and FARMS methods were used for summarization while the LMM was used for the probe-level analysis. A future research line could be the development of the LMM as a summarization model as well. This will allow us, similar to FARMS, to use the same model for filtering (using probe-level data) and summarization.

15.2 Part II: Biclustering analysis of gene expression experiments

In Part II we have introduced concepts of biclustering analysis and presented the overview of biclustering methods. Biclustering is a relatively new data analysis tool and it considered to be a data mining tool. However, classical statistical procedures, used for the reduction of dimensionality (factor analysis, principal component analysis, canonical correlation analysis and their derivatives or modifications), have been shown to be highly appropriate in the biclustering setting as well.

A simulation study was conducted in order to compare the performance of several biclustering methods in terms of stability and robustness for the initial values of the algorithms. It has been observed that some biclustering methods are highly unstable with respect to the initialization parameters. For example, Plaid had high performance on average, however, the variability of its specificity and sensitivity was the highest. In order to make the Plaid biclustering robust to the random initialization, we have used an ensemble algorithm. The ensemble procedure has been coded in `R` and is available as the `superbiclust` package on the `CRAN`. The ensemble of biclustering results can be used for obtaining robust biclusters with respect to the other algorithm-specific parameters (such as the number of biclusters, sparseness, filtering criteria, model fit criteria, etc.) and even for comparison of the bicluster sets obtained from different biclustering methods.

The central work of Part II is the diagnostics tools for biclusters. In the current biclustering literature, the evaluation of the biclusters quality is performed by their biological relevance. However, there is a need of statistical tools, which can evaluate the significance of the discovered patterns. We have described one of the current approaches, the strength of the differential gene co-expression, and extended the available method of Chia and Karuturi (2010) to the identification of significant effects in a bicluster. The diagnostic method discussed in Part II is a part of an ongoing research and a simulation study should be conducted in order to investigate the distribution of the p values obtained from the two-way ANOVA model. In addition, the influence of the size of the expression matrix on the bicluster discovered by a given method should be investigated as well.

Since the diagnostics of biclusters is a recent research area, many questions remain open. It would be of further interest to use the results of the structure diagnostics and extend it to the multivariate tests of differential-co-expression, such as the Hotelling test or F-test. This might be an alternative for the currently available ranking procedure of Chia and Karuturi (2010). In addition, the application of linear mixed models can be considered for both structure and differential co-expression diagnostics. Another open question is the method-independent evaluation of biclustering methods in a simulation study, where several parameters of the data generation will be varied. The starting point is the setting described in Eren *et al.* (2012), where one bicluster at a time is considered with varied structure and signal-to-noise ratio. The study can be further extended to the setting with two biclusters and variable the overlap. Eventually, the newly developed methods with a software implementation should be added to the simulation study.

15.3 Part III: Joint unsupervised analysis of high throughput data

In Part III we have extended the single-dataset biclustering to joint biclustering by combining data sets, which have one common dimension (features or samples). We have shown how biclustering can be applied to the gene expression data, collected from different projects or measured by different microarray types. Joint biclustering was applied to the setting, where both miRNA and mRNA were profiled for the same set of biological samples. Eventually, we presented a novel application of joint biclustering, i.e., the integrative analysis of gene expression and chemical structure data.

The central problem in joint biclustering is the normalization of the expression data for different projects. Joint biclusters cannot be discovered without appropriate preprocessing and availability of control samples for each batch, cell line, or array type. The separate normalization of miRNAs and mRNAs expression values is important, since miRNAs is found in lower abundance in the cell compared to the mRNA.

The discretization of the gene expression data is essential for joint biclustering of the FPF-gene expression data. The parameters for discretization should be chosen according to the requirements and properties of a biclustering algorithm. It is important that the numerical definition of the up- and down-regulated genes reflects the biological signal in the data.

The other aspect, considered in Part III, is obtaining joint biclusters by the biclustering methods, which output continuous bicluster membership. The *ad-hoc* procedure for selecting the threshold values for miRNA-mRNA joint biclusters and joint biclusters for *in-house* and the CMAP data has been discussed. Nevertheless, the extraction and interpretation of biclusters from the FABIA output remains an open research question and should be further investigated.

Joint biclustering has not yet been applied to the data with partially matching dimensions, e.g., bioassay and gene expression data, bioassay and chemical structure data, or a joint ”bioassay - gene expression - fingerprint feature” data. It is mainly due to the incomplete bioassay data. Further investigation of the performance of biclustering algorithms in presence of missing data is needed. To our knowledge, no biclustering methods dealing with missing data have been reported so far.

Another important aspect of integrating data across projects is the increasing dimensionality in terms of samples. Most biclustering methods have been developed for data sets of a moderate sample size. Hence, the performance and runtime efficiency should be evaluated in the high-dimensional setting. This could be carried out by means of a simulation study with a large number of columns and rows and a sparse signal.

Bibliography

Ahmad, W. (2007) chawk: An efficient biclustering algorithm based on bipartite graph crossing minimization.

Ahn, J., Yoon, Y. and Park, S. (2008) Rn-cluster: Discovering coherent biclusters which is robust to noise. In: *Proceedings of the 2008 International Conference on Biocomputation, Bioinformatics, and Biomedical Technologies*, BIOTECHNO '08, 131–136. Washington, DC, USA: IEEE Computer Society.

Al-Akwaa, F., Ali, M. and Kadah, Y. (2009) Bicat-plus: An automatic comparative tool for biclustering of gene expression data obtained using microarrays. In: *Radio Science Conference, 2009. NRSC 2009. National*, 1–8.

Aladag, A. E., Erten, C. and Sozdinler, M. (2011) Reliability-oriented bioinformatic networks visualization. *Bioinformatics*, **27**, 1583–1584.

Alqadah, F., Bader, J. S., Anand, R. and Reddy, C. K. (2012) Query-based biclustering using formal concept analysis. In: *SDM*, 648–659. SIAM / Omnipress.

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

Ayadi, W., Elloumi, M. and Hao, J.-K. (2009) A biclustering algorithm based on a bicluster enumeration tree: application to dna microarray data. *BioData Mining*, **2**, 9.

Ayadi, W., Elloumi, M. and Hao, J.-K. (2012) Bicfinder: a biclustering algorithm for microarray data analysis. *Knowledge and Information Systems*, **30**, 341–358.

Bagyamani, J., Thangavel, K. and Rathipriya, R. (2013) Comparison of biological significance of biclusters of simbic and simbic+ biclustering models. *ACEEE Int. J. on Information Technology,*, **3**.

Barkow, S., Bleuler, S., Prelic, A., Zimmermann, P. and Zitzler, E. (2006) Bicat: a biclustering analysis toolbox. *Bioinformatics*, **22**, 1282–1283.

Benjamini, Y. and Hochberg, Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, **57**, 289–300.

Bergmann, S., Ihmels, J. and Barkai, N. (2003) Iterative signature algorithm for the analysis of large-scale gene expression data. *Phys. Rev. E*, **67**, 031902.

Bourgon, R., Gentleman, R. and Huber, W. (2010) Independent filtering increases detection power for high-throughput experiments. *Proceedings of the National Academy of Sciences*, **107**, 9546–9551.

Caldas, J. and Kaski, S. (2011) Hierarchical generative biclustering for microrna expression analysis. *Journal of Computational Biology*, **18**, 251–261.

Calza, S., Raffelsberger, W., Ploner, A., Sahel, J., Leveillard, T. and Pawitan, Y. (2007) Filtering genes to improve sensitivity in oligonucleotide microarray data analysis. *Nucleic Acids Research*, 1–10.

Cambon, A. C., Khalyfa, A., Cooper, N. G. and Thompson, C. M. (2007) Analysis of probe level patterns in affymetrix microarray data. *BMC Bioinformatics*, **8**.

Cao, K.-A. L., Rossouw, D., Robert-Granié, C. and Besse, P. (2008) A sparse pls for variable selection when integrating omics data. *Statistical Applications in Genetics and Molecular Biology*, **7**, 1544–6115.

ChemAxon, L. (2013) Ecfp - extended-connectivity fingerprints. URL `http://www.chemaxon.com/jchem/doc/user/ECFP.html`. Version 6.0.0.

Cheng, K., Law, N., Siu, W. and Lau, T. (2007) Bivisu: software tool for bicluster detection and visualization. *Bioinformatics*, **23**, 2342–2344.

Cheng, Y. and Church, G. M. (2000) Biclustering of expression data. In: *ISMB-00 Proceedings*, 93–103.

Chengalvala, M. V., Chennathukuzhi, V. M., Johnston, D. S., Stevis, P. E. and Kopf, G. S. (2007) Gene expression profiling and its practice in drug development. *Current Genomics*, **8**, 262–270.

Chia, B. K. H. and Karuturi, R. K. M. (2010) Differential co-expression framework to quantify goodness of biclusters and compare biclustering algorithms. *Algorithms for Molecular Biology*, **5**, 23.

Chu, T.-M., Weir, B. and Wolfinger, R. (2002) A systematic statistical linear modeling approach to oligonucleotide array experiments. *Mathematical Biosciences*, **176**, 35–51.

Clarke, P. A., te Poele, R. and Workman, P. (2004) Gene expression microarray technologies in the development of new therapeutic agents. *European journal of cancer*, **40**, 2560–2591.

Clement, L., De Beuf, K., Thas, O., Vuylsteke, M., Irizarry, R. A. and Crainiceanu, C. M. (2012) Fast wavelet based functional models for transcriptome analysis with tiling arrays. *Statistical Applications in Genetics and Molecular Biology*, **11**, 1–36.

Cope, L. M., Irizarry, R. A., Jaffee, H. A., Wu, Z. and Speed, T. P. (2004) A benchmark for affymetrix genechip expression measures. *Bioinformatics*, **20**, 323–331.

Dai, M., Wang, P., Boyd, A. D., Kostov, G., Athey, B., Jones, E. G., Bunney, W. E., Myers, R. M., Speed, T. P., Akil, H., Watson, S. J. and Meng, F. (2005) Evolving gene/transcript definitions significantly alter the interpretation of genechip data. *Nucleic Acids Research*, **33**, e175.

Davis, A. P., King, B. L., Mockus, S., Murphy, C. G., Saraceni-Richards, C., Rosenstein, M., Wiegers, T. and Mattingly, C. J. (2011) The comparative toxicogenomics database: update 2011. *Nucleic Acids Research*, **39**, D1067–D1072.

De Beuf, K., Pipelers, P., Andriankaja, M., Thas, O., Inze, D., Crainiceanu, C. and Clement, L. (2012) Analysis of tiling array expression studies with flexible designs in bioconductor (wavetiling). *BMC Bioinformatics*, **13**, 234.

De Neve, J., Thas, O., Clement, L. and Ottoy, J.-P. (2009) A semiparametric unified approach for the detection of differential gene expression in microarrays. In: *Joint Statistical Meetings, Abstracts.*

De Neve, J., Thas, O. and Ottoy, J.-P. (2013) Goodness-of-fit methods for probabilistic index models. *Communications in Statistics - Theory and Methods*, **42**, 1193–1207.

De Smet, R. and Marchal, K. (2011) An ensemble biclustering approach for querying gene expression compendia with experimental lists. *Bioinformatics*, **27**, 1948–1956.

Dhillon, I. S. (2001) Co-clustering documents and words using bipartite spectral graph partitioning. In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, 269–274. New York, NY, USA: ACM.

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

Eren, K., Deveci, M., Küçüktunç, O. and Çatalyürek, Ü. V. (2012) A comparative analysis of biclustering algorithms for gene expression data. *Briefings in Bioinformatics*.

Falcon, S. and Gentleman, R. (2008) *Bioconductor Case Studies*, chap. Hypergeometric Testing Used for Gene Set Enrichment Analysis, 207–220. Springer New York.

Filippone, M., Masulli, F. and Rovetta, S. (2009) Stability and performances in biclustering algorithms. In: *Computational Intelligence Methods for Bioinformatics and Biostatistics* (Eds. F. Masulli, R. Tagliaferri and G. M. Verkhivker), chap. Stability and Performances in Biclustering Algorithms, 91–101. Springer-Verlag.

Forcheh, A. C., Verbeke, G., Kasim, A., Lin, D., Shkedy, Z., Talloen, W., Göhlmann, H. W. and Clement, L. (2012) Gene filtering in the analysis of illumina microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, **11**, 1544–6115.

Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B. and Overington, J. P. (2012) Chembl: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, **40**, D1100–D1107.

Gmeiner, W. H., Reinhold, W. C. and Pommier, Y. (2010) Genome-wide mrna and microrna profiling of the nci 60 cell-line screen and comparison of fdump[10] with fluorouracil, floxuridine, and topoisomerase 1 poisons. *Molecular Cancer Therapeutics*, **9**, 3105–3114.

Gonçalves, J. P., Madeira, S. C. and Oliveira, A. L. (2009) Biggests: integrated environment for biclustering analysis of time series gene expression data. *BMC Research Notes*, **2**.

Gu, J. and Liu, J. (2008) Bayesian biclustering of gene expression data. *BMC Genomics*, **9**, S4.

Guifen, C., Baocheng, W. and Helong, Y. (2007) The implementation of parallel genetic algorithm based on matlab. In: *Advanced Parallel Processing Technologies*, 676–683.

Gupta, R., Greco, D., Auvinen, P. and Arjas, E. (2010) Bayesian integrated modeling of expression data: a case study on rhog. *BMC Bioinformatics*, **11**, 295.

Gusenleitner, D., Howe, E. A., Bentink, S., Quackenbush, J. and Culhane, A. C. (2012) ibbig: iterative binary bi-clustering of gene sets. *Bioinformatics*, **28**, 2484–2492.

Hallez, A., Bronselaer, A. and De Trez, G. (2009) Comparison of sets and multisets. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **17, suppl. 1**, 153–172.

Hanczar, B. and Nadif, M. (2012) Ensemble methods for biclustering tasks. *Pattern Recognition*, **45**, 3938–3949.

Hartigan, J. A. and Wong, M. A. (1979) Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **28**, 100–108.

Hochreiter, S., Bodenhofer, U., Heusel, M., Mayr, A., Mitterecker, A., Kasim, A., Khamiakova, T., Van Sanden, S., Lin, D., Talloen, W., Bijnens, L., Göhlmann, H. W. H., Shkedy, Z. and Clevert, D.-A. (2010) Fabia: factor analysis for bicluster acquisition. *Bioinformatics*, **26**, 1520–1527.

Hochreiter, S., Clevert, D.-A. and Obermayer, K. (2006) A new summarization method for affymetrix probe level data. *Bioinformatics*, **22**, 943–949.

Hoshida, Y., Brunet, J.-P., Tamayo, P., Golub, T. R. and Mesirov, J. P. (2007) Subclass mapping: Identifying common subtypes in independent disease data sets. *PLoS ONE*, **2**, e1195+.

Huang, N., Shah, P. K. and Li, C. (2012) Lessons from a decade of integrating cancer copy number alterations with gene expression profiles. *Briefings in Bioinformatics*, **13**, 305–316.

Hughes, J., Rees, S., Kalindjian, S. and Philpott, K. (2011) Principles of early drug discovery. *British Journal of Pharmacology*, **162**, 1239 –1249.

Ihmels, J., Bergmann, S. and Barkai, N. (2004) Defining transcription modules using large-scale gene expression data. *Bioinformatics*, **20**, 1993–2003.

Imhoff, L. (2006) Parallel biclustering algorithm - fast algorithm for finding all biclusters in a gem. URL `https://github.com/KronicDeth/biclustering/blob/master/INSTALLATION`.

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U. and Speed, T. P. (2003) Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostat*, **4**, 249–264.

Irizarry, R. A., Wu, Z. and Cawley, S. (2004) *affycomp: Graphics Toolbox for Assessment of Affymetrix Expression Measures*. R package version 1.26.0.

Kacmarczyk, T., Waltman, P., Bate, A., Eichenberger, P. and Bonneau, R. (2011) Comparative microbial modules resource: Generation and visualization of multi-species biclusters. *PLoS Comput Biol*, **7**, e1002228.

Kaiser, S., Santamaria, R., Khamiakova, T., Sill, M., Theron, R., Quintales, L. and Leisch, F. (2011) *biclust: BiCluster Algorithms*. URL `http://CRAN.R-project.org/package=biclust`. R package version 1.0.1.

Kasim, A., Shkedy, Z. and Khamiakova, T. (2012) Semi-finite components mixture model for probes on affymetrix microarrays. *Tech. rep.*, Durham University, Hasselt University.

Kasim, A., Shkedy, Z., Lin, D., Sanden, S. V., Talloen, W., Göhlmann, H. W., Bijnens, L., Clevert, D.-A., Hochreiter, S., Amaratunga, D. and Heydebreck, A. V. (2010) Informative or noninformative calls for gene expression: A latent variable approach. *Statistical Applications in Genetics and Molecular Biology*, **9(1)**.

Kaytoue, M., Kuznetsov, S. O., Macko, J., Jr., W. M., and Napoli, A. (2011) Mining biclusters of similar values with triadic concept analysis. In: *The Eighth International Conference on Concept Lattices and Their Applications* (Eds. N. A. and V. V.), 175–190. INRIA Nancy Grand Est and LORIA Nancy, France.

Kim, S.-J., Shin, J.-Y., Lee, K.-D., Bae, Y.-K., Sung, K., Nam, S. and Chun, K.-H. (2012) Microrna let-7a suppresses breast cancer cell migration and invasion through downregulation of c-c chemokine receptor type 7. *Breast Cancer Research*, **14**, R14.

Kluger, Y., Basri, R., Chang, J. T. and Gerstein, M. (2003) Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Research*, **13**, 703–716.

Kostka, D. and Spang, R. (2004) Finding disease specific alterations in the co-expression of genes. *Bioinformatics*, **20 Suppl. 1**, i194–i199.

Lahti, L., Schäfer, M., Klein, H.-U., Bicciato, S. and Dugas, M. (2013) Cancer gene prioritization by integrative analysis of mrna expression and dna copy number data: a comparative review. *Briefings in Bioinformatics*, **14**, 27–35.

Lamb, J. (2006) The connectivity map: Using gene-expression signatures to connect small molecules, genes, and disease. *Science*, **313**, 1929–1935.

Lamb, J. (2007) The connectivity map: a new tool for biomedical research. *Nat Rev Cancer*, **7**, 54–60.

Langmead, B., Trapnell, C., Pop, M. and Salzberg, S. (2009) Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biology*, **10**, R25.

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

Lee, Y., Lee, J. and Jun, C.-H. (2011) Stability-based validation of bicluster solutions. *Pattern Recognition*, **44**, 252 – 264.

Leek, J. T., Scharpf, R. B., Corrada Bravo, H., Simcha, D., Langmead, B., Johnson, W. E., Geman, D., Baggerly, K. and Irizarry, R. A. (2010) Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews*, **11**, 733–739.

Li, G., Ma, Q., Tang, H., Paterson, A. H. and Xu, Y. (2009) Qubic: a qualitative biclustering algorithm for analyses of gene expression data. *Nucleic Acids Research*, **37**, e101.

Lin, D., Shkedy, Z. and Burzykowski, T. (2008) *Classification, dose-response modelling, and the evaluation of biomarker in a microarray setting.* Ph.D. thesis, Hasselt University.

Liu, H., D'Andrade, P., Fulmer-Smentek, S., Lorenzi, P., Kohn, K. W., Weinstein, J. N., Pommier, Y. and Reinhold, W. C. (2010) mrna and microrna expression profiles of the nci-60 integrated with drug activities. *Molecular Cancer Therapeutics*, **9**, 1080–1091.

Lock, E. F. (2011) Biclustering extensions. *Tech. rep.*, University of North Carolina. URL `http://www.unc.edu/~lock/software/Biclustering_Extensions.pdf`.

Lockhart, D. J., Dong, H., Byrne, M. C., Follettie, M. T., Gallo, M. V., Chee, M. S., Mittmann, M., Wang, C., Kobayashi, M., Norton, H. and Brown, E. L. (1996) Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, **14**, 1675–1680.

Lu, J., Kerns, R. T., Peddada, S. D. and Bushel, P. R. (2011) Principal component analysis-based filtering improves detection for affymetrix gene expression arrays. *Nucleic Acids Research*, **39**, e86.

Ma, H. and Zhao, H. (2012a) Facpad: Bayesian sparse factor modeling for the inference of pathways responsive to drug treatment. *Bioinformatics*, **28**, 2662–2670.

Ma, H. and Zhao, H. (2012b) ifad: an integrative factor analysis model for drug-pathway association inference. *Bioinformatics*, **28**, 1911–1918.

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

Mankad, S. and Michailidis, G. (2013) Biclustering three-dimensional data arrays with plaid models. URL `http://dept.stat.lsa.umich.edu/~smankad/`. Http://dept.stat.lsa.umich.edu/ smankad/.

Marchini, J. L., Heaton, C. and Ripley, B. D. (2012) *fastICA: FastICA Algorithms to perform ICA and Projection Pursuit.* URL `http://cran.r-project.org/web/packages/fastICA/`. R-package v. 1.1-16.

McCall, M. N., Bolstad, B. M. and Irizarry, R. A. (2009) Frozen robust multi-array analysis (frma). *Johns Hopkins University, Dept. of Biostatistics Working Papers*, **Working Paper 189.**

McCall, M. N., Bolstad, B. M. and Irizarry, R. A. (2010) Frozen robust multiarray analysis (frma). *Biostatistics*, **11**, 242–253.

McGee, M. and Chen, Z. (2006) New spiked-in probe sets for the affymetrix hgu-133a latin square experiment. *COBRA Preprint Series*, **5**.

Miska, E. A. (2008) Micrornas - keeping cells in formation. *Nat Cell Biol*, **10**, 501–502.

Mouhoubi, K., Letocart, L. and Rouveirol, C. (2011) Itemset mining in noisy contexts: A hybrid approach. In: *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*, 33–40.

Moura, G., Pinheiro, M., Arrais, J., Gomes, A. C., Carreto, L., Freitas, A., Oliveira, J. L. and Santos, M. A. (2007) Large scale comparative codon-pair context analysis unveils general rules that fine-tune evolution of mrna primary structure. *PLoS ONE*.

Murali, T. and Kasif, S. (2003) Extracting conserved gene expression motifs from gene expression data. In: *Pacific Symposium on Biocomputing*, vol. 8, 77–88.

Nisar, A., Ahmad, W., keng Liao, W. and Choudhary, A. (2009) High performance parallel/distributed biclustering using barycenter heuristic. In: *Proceedings of the Ninth SIAM International Conference on Data Mining*, 1050–1062.

Okada, Y. and Fujibuchi, W. (2007) Mining a large-scale microarray database for similar gene expression modules to find distant relationships between down syndrome and huntington's disease. In: *CAMDA 2007*.

Okada, Y., Fujibuchi, W. and Horton, P. (2007) A biclustering method for gene expression module discovery using closed itemset enumeration algorithm. *IPSJ Transactions on Bioinformatics*, **48**, 39–48.

Pascual-Montano, A., Carazo, J., Kochi, K., Lehmann, D. and Pascual-Marqui, R. D. (2006a) Nonsmooth nonnegative matrix factorization (nsnmf). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**, 403–415.

Pascual-Montano, A., Carmona-Saez, P., Chagoyen, M., Tirado, F., Carazo, J. and Pascual-Marqui, R. (2006b) bionmf: a versatile tool for non-negative matrix factorization in biology. *BMC Bioinformatics*, **7**, 366.

Pawitan, Y., Bjöhle, J., Amler, L., Borg, A.-L. and et al. (2005) Gene expression profiling spares early breast cancer patients from adjuvant therapy: derived and validated in two population-based cohorts. *Breast Cancer Res*, **7**, R953–64.

Pinheiro, J. C. and Bates, D. M. (2000) *Mixed Effects Models in S and S-Plus*. Springer.

Poell, J. B., van Haastert, R. J., de Gunst, T., Schultz, I. J., Gommans, W. M., Verheul, M., Cerisoli, F., van Noort, P. I., Prevost, G. P., Schaapveld, R. Q. J. and Cuppen, E. (2012) A functional screen identifies specific micrornas capable of inhibitinghuman melanoma cell viability. *PLoS ONE*, **7**, e43569.

Prelic, A., Bleuler, S., Zimmermann, P., Wille, A., Buhlmann, P., Gruissem, W., Hennig, L., Thiele, L. and Zitzler, E. (2006) A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, **22**, 1122–1129.

Raghavan, N., Amaratunga, D., Cabrera, J., Nie, A., Qin, J. and McMillian, M. (2006) On methods for gene function scoring as a means of facilitating the interpretation of microarray results. *Journal of Computational Biology*, **13**, 798–809.

Raghavan, N., Bondt, A., Verbeke, T. and Amaratunga, D. (2012) Gene set analysis as a means of facilitating the interpretation of microarray results. In: *Modeling Dose-Response Microarray Data in Early Drug Development Experiments Using R* (Eds. D. Lin, Z. Shkedy, D. Yekutieli, D. Amaratunga and L. Bijnens), Use R!, 181–191. Springer Berlin Heidelberg.

Raghavan, N., De Bondt, A. M., Talloen, W., Moechars, D., Göhlmann, H. W. H. and Amaratunga, D. (2007) The high-level similarity of some disparate gene expression measures. *Bioinformatics*, **23**, 3032–3038.

Raghavan, N., Verbeke, T., with contributions by Javier Cabrera, A. D. B., Amaratunga, D. and Casneuf, T. (2010) *MLP*. R package version 0.99.5.

Reiss, D. J., Baliga, N. S. and Bonneau, R. (2013) *cMonkey integrated biclustering algorithm*. URL `http://err.bio.nyu.edu/cmonkey/`. R package.

Rodriguez-Baena, D. S., Perez-Pulido, A. J. and Aguilara-Ruiz, J. S. (2011) A biclustering algorithm for extracting bit-patterns from binary datasets. *Bioinformatics*, **27**, 2738–2745.

Rogers, D. and Hahn, M. (2010) Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, **50**, 742–754. PMID: 20426451.

Rosenwald, A., Wright, G., Chan, W. C., Connors, J. M., Campo, E., Fisher, R. I., Gascoyne, R. D., Muller-Hermelink, H. K., Smeland, E. B., Giltnane, J. M., Hurt, E. M., Zhao, H., Averett, L., Yang, L., Wilson, W. H., Jaffe, E. S., Simon, R., Klausner, R. D., Powell, J., Duffey, P. L., Longo, D. L., Greiner, T. C., Weisenburger, D. D., Sanger, W. G., Dave, B. J., Lynch, J. C., Vose, J., Armitage, J. O., Montserrat, E., Lopez-Guillermo, A., Grogan, T. M., Miller, T. P., LeBlanc, M., Ott, G., Kvaloy, S., Delabie, J., Holte, H., Krajci, P., Stokke, T. and Staudt, L. M. (2002) The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. *N Engl J Med*, **346**, 1937–1947.

Roy, S., Bhattacharyya, D. K. and Kalita, J. K. (2013) Cobi: Pattern based co-regulated biclustering of gene expression data. *Pattern Recognition Letters*, –.

Sasidharan Nair, P. and Vihinen, M. (2012) Varibench: A benchmark database for variations. *Human Mutation*.

Serin, A. and Vingron, M. (2011) Debi: Discovering differentially expressed biclusters using a frequent itemset approach. *Algorithms for Molecular Biology*, **6**, 18.

Shabalin, A. A., Weigman, V. J., Perou, C. M. and Nobel, A. B. (2009) Finding large average submatrices in high dimensional data. *The Annals of Applied Statistics*, **3**, 985–1012.

Shi, F., Leckie, C., MacIntyre, G., Haviv, I., Boussioutas, A. and Kowalczyk, A. (2010) A bi-ordering approach to linking gene expression with clinical annotations in gastric cancer. *BMC Bioinformatics*, **11**, 477.

Sill, M., Kaiser, S., Benner, A. and Kopp-Schneider, A. (2011) Robust biclustering by sparse singular value decomposition incorporating stability selection. *Bioinformatics*.

Marcos A.S. da Silva AND, Antonio M.V. Monteiro AND, G. C. (2013) *SOM-Code : Design Patterns and Generic Programming in the Implementation of Self-Organizing Maps*. URL `http://somcode.cpatc.embrapa.br/doc/index.html`.

Smyth, G. K. (2004) Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, **3**.

Stevens, J. R., Bell, J. L., Aston, K. I. and White, K. L. (2010) A comparison of probe-level and probeset models for small-sample gene expression data. *BMC Bioinformatics*, **11**.

Stevens, J. R. and Nicholas, G. (2012) Assessing numerical dependence in gene expression summaries with the jackknife expression difference. *PLoS ONE*, **7**, e39570.

Streicher, K. L., Zhu, W., Lehmann, K. P., Georgantas, R. W., Morehouse, C. A., Brohawn, P., Carrasco, R. A., Xiao, Z., Tice, D. A., Higgs, B. W., Richman, L., Jallal, B., Ranade, K. and Yao, Y. (2012) A novel oncogenic role for the mirna-506-514 cluster in initiating melanocyte transformation and promoting melanoma growth. *Oncogene*, **31**, 1558–1570.

Su, A. I., Cooke, M. P., Ching, K. A., Hakak, Y., Walker, J. R., Wiltshire, T., Orth, A. P., Vega, R. G., Sapinoso, L. M., Moqrich, A., Patapoutian, A., Hampton, G. M., Schultz, P. G. and Hogenesch, J. B. (2002) Large-scale analysis of the human and mouse transcriptomes. *Proceedings of the National Academy of Sciences of the United States of America*, **99**, 4465–70.

Subramaniam, S. and Hsiao, G. (2012) Gene-expression measurement: variance-modeling considerations for robust data analysis. *Nature Immunology*, **13**, 199–203.

Sun P, Guo J, B. J. (2012) Integrated simultaneous analysis of different biomedical data types with exact weighted bi-cluster editing. *Journal of Integrative Bioinformatics*, **17**, 197.

Talloen, W., Clevert, D.-A., Hochreiter, S., Amaratunga, D., Bijnens, L., Kass, S. and Göhlmann, H. W. (2007) I/ni-calls for the exclusion of non-informative genes: a highly effective filtering tool for microarray data. *Bioinformatics*, btm478.

Tchagang, A. and Tewfik, A. (2005) Robust biclustering algorithm (roba) for dna microarray data analysis. In: *Statistical Signal Processing, 2005 IEEE/SP 13th Workshop on*, 984–989.

Teng, L. and Tan, K. (2012) Finding combinatorial histone code by semi-supervised biclustering. *BMC Genomics*, **13**, 301.

Tukey, J. W. (1949) One degree of freedom for non-additivity. *Biometrics*, **5 (3)**, 232–242.

Turner, H., Bailey, T. and Krzanowski, W. (2005) Improved biclustering of microarray data demonstrated through systematic performance tests. *Computational Statistics and Data Analysis*, **48**, 235–254.

Tusher, V. G., Tibshirani, R. and Chu, G. (2001) Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences*, **98**, 5116–5121.

Uhlen, M., Oksvold, P., Fagerberg, L., Lundberg, E., Jonasson, K., Forsberg, M., Zwahlen, M., Kampf, C., Wester, K., Hober, S., Wernerus, H., Bjorling, L. and Ponten, F. (2010) Towards a knowledge-based human protein atlas. *Nat Biotech*, **28**, 1248 – 1250.

Ulitsky, I., Maron-Katz, A., Shavit, S., Sagir, D., Linhart, C., Elkon, R., Tanay, A., Sharan, R., Shiloh, Y. and Shamir, R. (2010) Expander: from expression microarrays to networks and functions. *Nat. Protocols*, **5**, 303–322.

van 't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R. and Friend, S. H. (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.

Verbeke, G. and Molenberghs, G. (2002) *Linear Mixed Models for Longitudinal Data.* New-York: Springer.

Wang, M., Shang, X., Li, X., Li, Z. and Liu, W. (2013) Efficient mining differential co-expression constant row bicluster in real-valued gene expression datasets. *Applied Mathematics & Information Sciences*, **7**, 587–598.

Xie, L., Xie, L., Kinnings, S. L. and Bourne, P. E. (2012) Novel computational approaches to polypharmacology as a means to define responses to individual drugs. *Annual Review of Pharmacology and Toxicology*, **52**, 361–379.

Yang, J., Wang, H., Wang, W. and Yu, P. (2005) An improved biclustering method for analyzing gene expression. *International Journal on Artificial Intelligence Tools*, **14(5)**, 771–789.

Zhu, Q., Miecznikowski, J. C. and Halfon, M. S. (2010) Preferred analysis methods for affymetrix genechips. ii. an expanded, balanced, wholly-defined spike-in dataset. *BMC Bioinformatics*, **11**, 285.

Samenvatting

Al meer dan tien jaar verandert biotechnologische vooruitgang de manier waarop biomedisch en pharmaceutisch onderzoek worden uitgevoerd. Gelet op de toenemende ontwikkelingskosten voor nieuwe geneesmiddelen en de desondanks grote uitval van kandidaatgeneesmiddelen in de klinische fase vanwege gebrek aan efficaciteit of aanwezigheid van bijwerkingen, kunnen nieuwe technologieën helpen bij de selectie van de meest veelbelovende verbindingen voor volgende stadia van de geneesmiddelenontwikkeling. Met name genexpressie experimenten staan bekend omwille van hun bruikbaarheid bij identificeren en begrijpen van de werking en de bijwerkingen van de verbindingen. Bij kanker, neurologische aandoeningen en andere complexe ziektes zijn vaak veel genen en biologische pathways betrokken. Om kennis over geactiveerde pathways te verkrijgen, hetzij gewenste werking, hetzij ongewenste bijwerking, worden grote hoeveelheden genexpressie-experimenten uitgevoerd. Aanvullend op *in-house* gegevens, d.w.z. gegevens binnen een bepaald bedrijf verzameld, kan een farmaceutisch bedrijf profiteren van publieke databanken, zoals de Connectivity Map en de Comparative Toxicogenomics Database. Het efficiënte gebruik van databanken, zij het publieke of commerciële, vereist goede instrumenten voor de data voor te bereiden en te analyseren. Dit proefschrift is gewijd aan de statistische aspecten van zulke instrumenten.

Het proefschrift bestaat uit drie delen. Deel I focust op de analyse op probe-niveau van Affymetrix microarray data. De Affymetrix microarrays kwantificeren de expressie van een gen via meerdere oligonucleotides (*probes*) van 25 basisparen lang die samen een *probe set* vormen. Deel II behandelt de biclusteringanalyse van genexpressiedata. Deel III introduceert joint biclusteringsmethoden voor data-integratie en exploratie.

In Deel I presenteren we het modelleerkader voor de probe-niveau analyse van Affymetrix GeneChip data. Het modelleerkader is gebaseerd op lineaire mixed modellen (LMM) met experiment-specifieke en random effecten. We laten zien hoe het LMM-kader in verschillende experimentele situaties kan worden toegepast. Het probe-niveau LMM houdt rekening met de correlaties tussen probe-intensiteiten en geeft schattingen voor de vaste effecten die van belang zijn. Bovendien kunnen schattingen van de variantie tussen reeksen worden gebruikt om conclusies te trekken over factoren die het model niet heeft meegenomen. Dit soort informatie gaat verloren wanneer alle probe data samengevat worden in één getal. Daarom kan het probe-niveau LMM gebruikt worden om een kwaliteitscontrole van de gegevens uit te voeren en de aanwezigheid van technische artefacten, zoals batch-effecten, aangeven. Door middel van simulatiestudies tonen we aan dat als de probe-niveau data voldoen aan marginale normaliteit, het probe-niveau LMM even goed presteert met een even goede FDR als de differentiële expressie testen op samengevatte gegevens.

Het LMM-kader wordt verder uitgebreid om probe-sets te modelleren, waarbij de aanname dat alle probes hetzelfde doel-transcript meten wordt geschonden. Een mixture model wordt gebruikt om het aantal componenten in een probe set te schatten, en de posterior waarschijnlijkheden werden gebruikt om de componentsamenstelling (klasse-label) voor elke probe in de probe set. De schattingen van de intra-class correlaties (ICC's) worden gebruikt voor de geactualiseerde definitie van de probe sets. We illustreren hoe een experiment-specifieke probe-set definitie kan worden uitgevoerd, gebaseerd op de uitkomst van het probe-niveau LMM. Als bijvoorbeeld een probe set uit meerdere componenten bestaat, kan hij worden onderverdeeld in een aantal verschillende probe sets. Daarna kunnen deze probe sets worden gebruikt voor de actualisering van de bestanden die per chip type probes annoteren (CDFs). Zo kan de standaard downstream analyse van de genexpressiedata (samenvatting, test voor differentiële expressie, etc.) uitgevoerd worden op basis van de geactualiseerde CDFs.

In Deel II introduceren we concepten van biclusteringanalyse en presenteren een overzicht van bestaande biclusteringsmethoden. Biclustering is een relatief nieuwe data-analysetechniek die wordt beschouwd als een data mining methode. Anderzijds zijn klassieke statistische procedures die gebruikt worden om de dimensionaliteit te reduceren (factor analyse, principale componentenanalyse, canonische correlatieanalyse en daarvan afgeleide analyses) ook zeer geschikt. Een simulatiestudie werd uitgevoerd om de prestatie van verschillende biclusteringsmethoden te vergelijken wat betreft de stabiliteit en robuustheid voor verschillen in de initialisatiewaarden. Men heeft gevonden dat sommige biclusteringsmethoden zeer instabiel zijn met betrekking tot deze initialisatieparameters. Om een biclusteringsmethode robuust te maken voor random initialisatiewaarden, hebben we gebruik gemaakt van een ensemble-algoritme. De ensembleprocedure is geprogrammeerd in R en beschikbaar als het `superbiclust`-pakket op de CRAN.

Het centrale onderwerp van Deel II betreft diagnostische methoden voor biclusters. In de huidige biclustering-literatuur wordt de biclusterkwaliteit bepaald door biologische relevantie. Er is echter een behoefte aan statistische methoden, die de significantie van de gedetecteerde patronen kunnen evalueren. Wij beschrijven één van de huidige aanpakken, de differentiële gen co-expressie, en breiden de beschikbare methode van Chia and Karuturi (2010) uit tot de identificatie van significante effecten in een bicluster.

In Deel III breiden we biclustering van afzonderlijke datasets uit tot joint biclustering door het combineren van datasets, die een gemeenschappelijke dimensie (kenmerken of monsters) hebben. We laten zien hoe biclustering kan worden toegepast op genexpressie-data uit verschillende projecten of gemeten met verschillende microarray types. Joint biclustering werd toegepast in een studie waarbij zowel miRNA en mRNA geprofileerd waren voor dezelfde set van biologische monsters. Tot slot presenteren we een nieuwe toepassing van joint biclustering, namelijk de geïntegreerde analyse van gen-expressie en chemische-structuur data.