HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Computer Science and Engineering
Laboratory of Computer and Information Science

Mikko Korpela

Analysis of changes in gene expression time series data

Master's Thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Technology.

Espoo, 14th February 2006

Supervisor: Professor (pro tem) Jaakko Hollmén
Instructor: Professor (pro tem) Jaakko Hollmén

HELSINKI UNIVERSITY OF TECHNOLOGY Department of Computer Science and Engineering	ABSTRACT OF MASTER'S THESIS
Author Mikko Korpela	Date 14th Feb 2006
	Pages 85
Title of thesis Analysis of changes in gene expression time series data	
Professorship Computer and information science	Professorship Code T-122
Supervisor Prof. (pro tem) Jaakko Hollmén	
Instructor Prof. (pro tem) Jaakko Hollmén	

Gene expression is the process by which genes control the biological functions of an organism through the production of proteins. DNA microarray technology enables simultaneous evaluation of the expression of tens of thousands of genes. By using multiple arrays, expression measurements can be done in different conditions and time points.

In this thesis, a gene expression data set is analysed. The data originate from experiments where the effect of asbestos on three different cell lines was studied. First, the data are subjected to various quality control methods. The work continues with descriptions of preprocessing and analysis methods.

The purpose of preprocessing is, among other things, to reduce non-biological variation in the data and to enable the comparison of measurements from different arrays. The RMA preprocessing method is used here. The method consists of the following steps: background correction, normalisation, log-transformation, and summarisation.

Preprocessed expression values are joined into time series which describe the differences between asbestos exposed and normal samples. A recent clustering method designed for short time series is employed in the analysis of the time series. The method includes the assessment of each cluster's statistical significance. The clustering scheme is laid out on an algorithmic level. An error in one part of the method is corrected, and an additional intermediate phase is also introduced.

Information available on genes is used in the analysis of clustering results. For example, it is interesting if a cluster has a significant amount of genes that have the same biological function. Also the clustering of known asbestos-related genes is studied. The implementation of the clustering algorithm is tested by repeating an experiment done on synthetic data. Finally, some results related to the asbestos data are shown. Actual conclusions are left for biologists to draw.

Keywords: bioinformatics, DNA microarray, gene expression, time series, clustering, preprocessing, asbestos

TEKNILLINEN KORKEAKOULU Tietotekniikan osasto	DIPLOMITYÖN TIIVISTELMÄ
Tekijä Mikko Korpela	Päiväys 14.2.2006
	Sivumäärä 85
Työn nimi Geeniekspressioaikasarjadatan muutosten analysointi	
Professuuri Informaatiotekniikka	Koodi T-122
Työn valvoja Ma. Prof. Jaakko Hollmén	
Työn ohjaaja Ma. Prof. Jaakko Hollmén	

Geeniekspressio tarkoittaa prosessia, jossa geenit säätelevät organismin biologisia toimintoja proteiinituotannon kautta. Geenisirutekniikka mahdollistaa kymmenien tuhansien geenien ekspression samanaikaisen arvioinnin. Useita siruja käyttämällä ekspressiomittauksia voidaan tehdä eri olosuhteissa ja aikapisteissä.

Tässä diplomityössä analysoidaan geeniekspressiodatajoukkoa. Data on peräisin kokeista, joissa tutkittiin asbestin vaikutusta kolmea erilaista solutyyppiä edustaviin näytteisiin. Ensin datan laatu tarkistetaan eri tavoin. Työ jatkuu esikäsittely- ja analyysimenetelmien kuvauksilla.

Esikäsittelyn tarkoituksena on muun muassa vähentää datassa olevaa biologisista syistä riippumatonta vaihtelua ja mahdollistaa eri siruista peräisin olevien mittausten keskinäinen vertailu. Tässä työssä käytetään RMA-esikäsittelymenetelmää, joka koostuu seuraavista vaiheista: taustakorjaus, normalisointi, logaritminen muunnos ja yhteenveto.

Esikäsitellyistä ekspressioarvoista muodostetaan aikasarjoja, jotka kuvaavat asbestikäsiteltyjen ja normaalien näytteiden välisiä muutoksia. Aikasarjojen analysointiin käytetään tuoretta klusterointimenetelmää, joka on suunniteltu lyhyille aikasarjoille ja sisältää klusterien tilastollisen merkitsevyyden arvioinnin. Menetelmä käydään läpi algoritmitasolla, ja siihen esitetään yksi korjaus sekä ylimääräinen välivaihe.

Klusteroinnin tulosten analysoinnissa käytetään hyväksi geeneistä saatavilla olevaa tietoa. Esimerkiksi on kiinnostavaa, jos jossain klusterissa on huomattavan paljon saman biologisen toiminnon toteuttavia geenejä. Myös tunnettujen asbestiin liittyvien geenien klusteroitumista tutkitaan. Klusterointialgoritmin toteutuksen toiminta testataan toistamalla synteettisellä datalla tehty koe. Lopussa esitetään joukko asbestidataan liittyviä tuloksia. Varsinaisten päätelmien tekeminen jätetään biologeille.

Avainsanat

bioinformatiikka, geenisiru, geenieksprессio, aikasarja, klusterointi, esikäsittely, asbesti

Acknowledgements

This Master's Thesis was carried out at the Laboratory of Computer and Information Science of Helsinki University of Technology (TKK).

First, I want to thank my instructor and supervisor Jaakko Hollmén for all the advice he has given me during this work and previous projects.

I wish to thank all the people from TKK, University of Helsinki, and Finnish Institution of Occupational Health that are involved in this gene expression project. My special thanks go to Leo Lahti, Pamela Lindholm, and Penny Nymark for the helpful discussions we had, both in person and through e-mail.

My gratitude also goes to the people sharing room B310 with me, and to other people in the laboratory: you are an important part of a nice and relaxed work environment.

Finally, I would like to thank my family for the invaluable support they have given me in the course of my life and studies.

Otaniemi, 14th February 2006

Mikko Korpela

Contents

Abbreviations and notations

CDF	Cumulative distribution function
cDNA	Complementary DNA
CPU	Central processing unit
DNA	Deoxyribonucleic acid
GO	Gene ontology
MIAME	Minimum information about a microarray experiment
MM	Mismatch (probe)
PDF	Probability density function
PM	Perfect match (probe)
QC	Quality control
RMA	Robust multichip average
RNA	Ribonucleic acid
SOM	Self-organising map
S, X, Y	Random variables
s, x, y	Instances of random variables or arbitrary numbers
$\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$	Vectors
x_i	i:th element of $\mathbf{x}$
$\overline{x}$	Mean of $\mathbf{x}$
$\mathbf{x} \cdot \mathbf{y}$	Dot product of vectors $\mathbf{x}$ and $\mathbf{y}$
$d(\mathbf{x}, \mathbf{y})$	Distance between vectors $\mathbf{x}$ and $\mathbf{y}$
$\mathrm{Bin}(N, p)$	Binomial distribution, #trials N, probability of success p
$\exp(\alpha)$	Exponential distribution with mean $1/\alpha$
$\mathrm{N}(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2
$\phi(x)$	PDF of $\mathrm{N}(0, 1)$ evaluated at x
$\Phi(x)$	CDF of $\mathrm{N}(0, 1)$ evaluated at x
p, p_1, p_2, p_i	Expression profiles
L, P, R, R'	Sets (of profiles)
$\lvert P \rvert$	Number of elements in a set P
$R \subset P$	R is a subset of P
$P \cup R$	Union of sets P and R
$P \cap R$	Intersection of sets P and R
$P \setminus R$	Set difference of P and R

Chapter 1

Introduction

Large amounts of data are being produced in the world today, by both individuals and organisations. Evolving technology makes the storage of ever increasing amounts of data possible, although there are some valid concerns about whether the advanced storage technologies and formats used today can stand the test of time.

The analysis of the data or *data mining* [21], however, is another story, and probably presents a bigger challenge than the storage side. As faster computers become available, the capacity available for data analysis also grows. Unfortunately, a computer doesn't function by itself, but needs precise instructions. Data analysis is based on intelligence, something that computers may have in science fiction novels but not in the real world. There is a need for people who can operate computers using familiar and new methods of data analysis. The vast amounts of data are a challenge — not only to computers, but also to data analysts.

One example of the "data explosion" can be seen in weather forecasting. The first weather forecasts relied on locally obtained data and experience gathered over time. For example, a clear sky on a winter evening is a good indicator of a cold night ahead. Now, fast communication networks and advanced observational instruments such as satellites and radars provide measurement data from all over the world. Thanks to sophisticated prediction models and increasing computer resources, the data can be used to produce more accurate and longer range forecasts than ever before.

Biological and medical research projects also produce lots of data with the help of advances in measurement technology. The analysis of these data is especially rewarding, since it can result in a better understanding of biological organisms. The discovery of the mechanism behind a disease might ultimately result in the development of a new drug or better diagnosis criteria. It is unlikely that an expert in biology or medicine would simultaneously be a skilled data analyst. It is also certain that the "average" data

analyst is not a biologist. A *collaborative mode of work* — where the strengths of different people are utilised in pursuit of a common goal — is needed. This thesis concerns itself with the analysis of *gene expression data*, one of many types of biological data studied today.

1.1 Gene expression and microarrays

The recent advancement of microarray technology is a welcome development in the study of molecular biology. With this technology, the activity of tens of thousands of genes can be measured simultaneously. It is even possible to capture information about all the genes of a given organism.

DNA microarrays [27] are used to measure RNA levels in cell samples. The measured levels can be used as an estimate of the amount of different proteins produced in cells. For a simple introduction to the relevant biological phenomena, see e.g. [23]. Simply put, DNA is copied into RNA in the nucleus of a cell. The RNA leaves the nucleus, and acts as a messenger in the production of proteins. The whole process, from DNA in a gene to RNA, and further to proteins, is called gene expression. Microarray data are often called gene expression measurements, although the results are only indirect estimates of the rate of protein production.

Proteins have several functions in living organisms. Therefore, gene expression measurements can be used to shed some light on various biological processes. The uses of microarray data reach from experiments with simple organisms like yeast to research on human cancers [22].

Gene expression measurements are typically carried out with several microarrays over a range of parameters, such as temperature, time, or exposure to some chemical agent. For more reliable results, a few replicate measurements are often made with the same parameters. All in all, with each microarray providing up to tens of thousands of data points, the amount of data collected in these experiments can best be described as huge. This calls for computer aided analysis.

1.2 Asbestos and health

Asbestos is the general name used for the following fibrous minerals: amosite, chrysotile, crocidolite, and the fibrous varieties of actinolite, anthophyllite, and tremolite. All of these except for chrysotile belong to the amphibole family of minerals. All forms of asbestos are health hazards, but amphiboles are considered more dangerous than chrysotile. [5]

The usage of asbestos dates back at least 2500 years. The earliest reports on some of the health hazards related to asbestos are about one hundred

years old [44]. Today, there is a broad consensus on the adverse health effects of asbestos. Despite that, asbestos is still used in the world. The largest consumers of asbestos in the year 2000 were Brazil, China, India, Japan, Thailand, and the former Soviet Union as a whole, especially Russia [54].

Airborne asbestos fibres enter the human body through the respiratory tract. Some inhaled fibres are later removed from the lungs, but others may move through the lungs and never leave the body [5]. When retained in the body, asbestos has the potential to cause serious diseases such as asbestosis, lung cancer, and mesothelioma [38]. Asbestosis and lung cancer are related to high cumulative exposure to asbestos, whereas mesothelioma can occur with smaller fibre burdens. Most mesothelioma cases (70–80 %) are asbestos-related. By contrast, lung cancer is not specific to asbestos and also has other major risk factors such as smoking [14].

Asbestos-related health effects can mainly be observed in people who have worked for sufficiently long periods in some particular fields like construction, shipbuilding, or asbestos manufacturing. Among these workers, exposure to asbestos is classified as "definite" or "probable" [51]. Some evidence exists that household members of asbestos workers might be at risk of developing mesothelioma. Still, cases arising from domestic exposure are rare. There is also practically no evidence of urban air pollution causing mesothelioma [33].

The usage of asbestos in Finland peaked between the 1950s and 1970s. Correspondingly, the peak in asbestos-related diseases is expected to be between the 1990s and 2020s [44]. It has been estimated that asbestos will have given rise to 2000 cancer deaths in Finland by the year 2010 [29]. About 30000 new asbestos-related cancer cases are estimated to occur annually in the population of 800 million living in Western Europe, North America, Japan, and Australia [51].

Due to the long time delay between exposure to asbestos and the development of the related diseases, there is clearly an opportunity to treat an emerging disease in its early stages. Understanding the effects of asbestos on gene expression could contribute to that goal. However, before something can be understood, it must first be studied. That should be plenty of motivation for biomedical scientists and data analysts.

1.3 Objectives and scope

The aim of the thesis is to extract information from the gene expression data that are subjected to analysis. The information should reveal significant features of the data in a way that helps collaborating biological researchers conduct further analyses and make hypotheses of the effect of asbestos on

gene expression.

Figure 1.1 is a very rough diagram about the phases of a gene expression experiment. More detailed diagrams also exist; see for example [27, p. 17]. This thesis is about computer aided data analysis, the middle phase in the

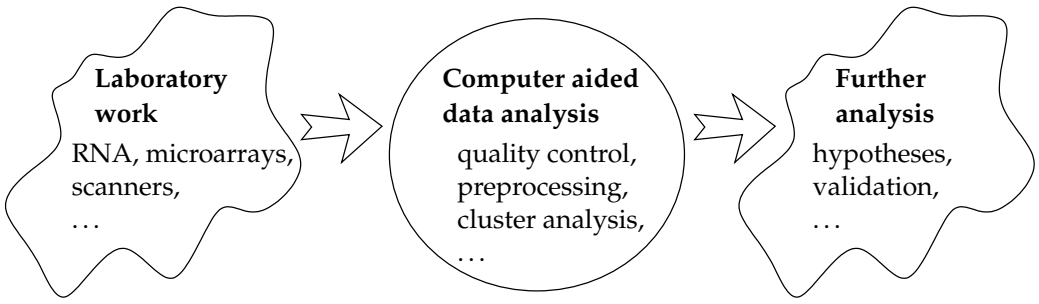


Figure 1.1: Gene expression analysis pipeline

figure. The thesis covers the basic theory of the different steps in that phase: quality control, preprocessing, and cluster analysis. The reader should be able to learn the theoretical principles of gene expression analysis to the extent that is necessary to understand what was done in the experimental part of the thesis.

The clustering algorithm discussed in Section 4.3 is covered more thoroughly than other methods applied in the course of the work. One reason for this special treatment is the importance of the algorithm for our analyses. Another reason is its novelty. There is also some value to presenting the algorithm as pseudo code in contrast to the verbal description in the article [16] that introduced the algorithm.

The first and the last phase of Figure 1.1 are performed by collaborators who are more familiar with the biological side of gene expression. The help of these experts is also valuable during the middle phase.

1.4 Structure of the thesis

After this introduction, Chapter 2 presents the data used in the thesis. Some general information about microarray data is also offered. Chapter 3 describes the first parts of the analysis process used in this thesis. The purpose of these stages is to transform the data into a form more suitable for the following phases of analysis. The comparison and co-analysis of data coming from different microarrays are enabled with the normalisation method described in this chapter. Some quality control procedures are also introduced and used to check the quality of the data.

The data analysis methods used in the thesis are described in Chapter 4. The main method is a recently developed clustering algorithm that is designed for the kind of data explored in this work. The experiments and results of the thesis are presented in Chapter 5. The chapter doesn't contain any final biological conclusions, which are better left to experts in that field. The results are more about what phenomena can be seen in the data, from a computer scientist's point of view. Further processing and validation of the results by biologists is necessary before the results are fitted into a greater biological framework.

Chapter 6 is a summary of the work done in this thesis, and offers some concluding remarks. Appendix A contains some supplementary figures and data tables produced in the course of the analyses that are better shown separately from the main text.

Chapter 2

Gene expression data

2.1 Microarray technologies

As explained in Section 1.1, microarrays are a tool for producing gene expression measurements. The following sections briefly describe two popular microarray technologies.

2.1.1 Oligonucleotide arrays

Oligonucleotide arrays are a class of microarrays mainly developed by a company called Affymetrix. The arrays are manufactured with a photolithographic masking technique, much in the same way as integrated circuits [27, pp. 77–88]. Each Affymetrix array contains up to $1.3 \cdot 10^6$ probes [1].

Here are some of the terms that are most frequently used in connection with oligonucleotide microarrays [27, pp. 77–78]:

Probe An oligonucleotide, 25 bases[1] in length, situated on a microarray. These are also called *25-mers*.

Perfect match (PM) probe A probe designed to match with a certain sequence of nucleotides in the genome

Mismatch (MM) probe A probe that is the same as its companion PM, except that one base has been switched. These are designed to measure non-specific hybridisation.

Probe pair The combination of a PM probe and the corresponding MM probe

[1]The DNA bases are adenine (A), thymine (T), guanine (G), and cytosine (C).

Probe set A set of probe pairs. In the latest Affymetrix microarrays, a probe set typically contains 11 probe pairs [1]. The PM probes of a probe set usually match to different parts of a single gene. A probe set as a whole therefore measures the expression of a gene. Since there still is some work to be done before the human genome is accurately dissected [39], not all probe sets on an array correspond to actual well-known genes.

2.1.2 cDNA arrays

Robotically spotted cDNA arrays, introduced in [46], represent the other major type of microarrays. They have both advantages and disadvantages compared to oligonucleotide arrays. Among the favourable points are customisability of the arrays and the fact that two different samples can be competitively hybridised to the same array. Disadvantages of cDNA arrays include their more variable quality and the relative difficulty of measuring absolute quantities. Instead of absolute expression values, the expression ratio of the two samples is the quantity that is typically measured with cDNA arrays. The probes on a cDNA array are about $2 \cdot 10^3$ bases long, much longer than the 25-mers in oligonucleotide arrays. [27, pp. 73–76]

2.2 Data used in the thesis

The data used in this thesis are gene expression values summarised in Table 2.1 (courtesy of Penny Nymark), acquired with Affymetrix oligonucleotide microarrays, model *GeneChip Human Genome U133 Plus 2.0*. The data are from a treatment-response experiment where the effect of asbestos on gene expression is studied. The microarrays indicate the development of the gene expression levels over time with both asbestos exposed and non-exposed samples.

In addition to the 25 microarrays listed in the table, there are also two extra arrays (A549 0 h and BEAS-2B 0 h) that are mainly used in the preprocessing stage. Thus the total number of arrays available is 27 (25 + 2). Each array has 54675 probe sets, of which 38500 correspond to "well-characterized human genes" [1]. That makes the total number of probe sets in all 27 arrays almost 1.5 million. The number of probes is roughly 20 times bigger. That is a considerable amount of data!

Table 2.2 gives the microarrays a numerical order. The same numbering scheme is used in various figures, where a simple number is more convenient than a longer description of each array. The figures are located in Chapter 5 and Appendix A. In the table, asbestos exposure is denoted with "asb."

Table 2.1: Data used in the analyses

Cell line	Description	Time points	
		asbestos exposure ($2\,\mu$g crocidolite/cm^2)	no exposure
A549	human, Caucasian, lung carcinoma	1 h 6 h 24 h 48 h (2 arrays) 7 d	1 h 6 h 24 h 48 h 7 d
BEAS-2B	primary and immortalised human bronchial epithelial cells	1 h 6 h 24 h (2 arrays) 48 h	1 h 6 h 24 h 48 h
Met5A	non-malignant transformed mesothelial cells	1 h 48 h (2 arrays)	1 h 48 h

Table 2.2: Order of arrays in some figures

1. A549 0 h	*10.* A549 asb. 48 h (2)	*19.* Beas asb. 24 h (1)
2. A549 1 h	*11.* A549 asb. 6 h	*20.* Beas asb. 24 h (2)
3. A549 24 h	*12.* A549 asb. 7 d	*21.* Beas asb. 48 h
4. A549 48 h	*13.* Beas 0 h	*22.* Beas asb. 6 h
5. A549 6 h	*14.* Beas 1 h	*23.* Met 1 h
6. A549 7 d	*15.* Beas 24 h	*24.* Met asb. 1 h
7. A549 asb. 1 h	*16.* Beas 48 h	*25.* Met 48 h
8. A549 asb. 24 h	*17.* Beas 6 h	*26.* Met asb. 48 h (1)
9. A549 asb. 48 h (1)	*18.* Beas asb. 1 h	*27.* Met asb. 48 h (2)

MIAME is a standard on the minimum information that should be published about every gene expression experiment. It aims to ease the interpretability of microarray data and the verifiability of the results derived from it [12]. Interpretability and verifiability are, without a doubt, very important principles in science. This thesis focuses on the analysis of microarray data. The details of the related microarray experiments will quite certainly be published by the experts more familiar with the whole experimental setting and the biological side of things. Hopefully the principles of MIAME will be considered then.

Chapter 3

Quality control and preprocessing

3.1 Quality control

3.1.1 Quality of data is important

Quality control (QC) is an important part of the gene expression analysis pipeline. We want our possible biological hypotheses or conclusions to be based on biologically sound data. Many spurious phenomena can emerge from noisy high-dimensional data. The analysis of data with large uncertainties requires time and money that could be better spent elsewhere. Furthermore, the possibly false results so obtained are often worse than no results at all. For these reasons, quality control is the first phase of the analysis in this thesis. Although some uncertainties will remain due to the nature of the data, any serious quality problems can hopefully be ruled out.

3.1.2 Qualitative QC

Visual inspection of intensity images of gene expression data can reveal problems resulting from faulty equipment. Microarrays themselves may have spatial artefacts, unique to each array. Additionally, microarray scanners may produce uneven results in different parts of the scanned array.

Figure 3.1 shows the pseudo image of one of the microarrays that produced the data used in the thesis. The intensity of each pixel is the logarithmically transformed expression value of a probe. The image is called a pseudo image because probe-level expression values are extracted from the actual image of the microarray. The analysis process used in the thesis bypasses this image processing stage completely, and starts with probe-level information instead.

The model of the microarray can be read from the upper left corner of the pseudo image (Figure 3.2). The image also shows an alternating dark and

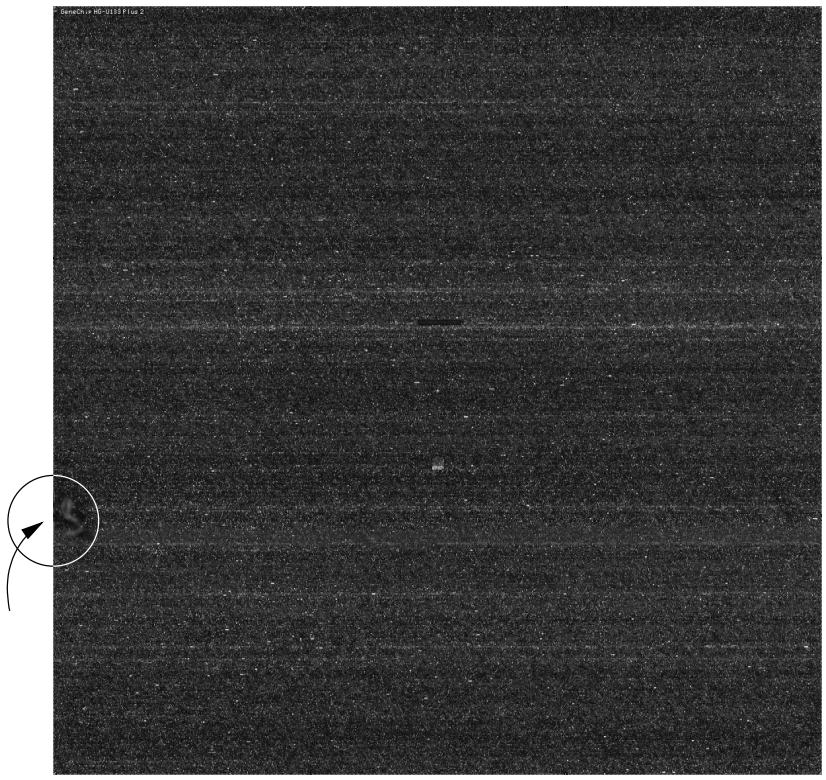


Figure 3.1: Pseudo image of a microarray (Met5A, no asbestos, 1 h)

Figure 3.2: The upper left corner of Figure 3.1 magnified

light pattern on the border. The pattern, originally in the raw image, is used in placing a virtual grid on top of the raw image [2, p. 37]. The grid divides the image area into rectangular probe cells [27, p. 80]. The expression value of each probe (visible as a pixel in the pseudo image) is computed from the intensity values of the pixels in the corresponding probe cell.

The overall quality of the image seems good, despite some horizontal striping. There is a suspicious-looking smudge in the left border region of Figure 3.1, a little below the vertical centre. Figure 3.3 is a close up view of the blurred area. The smooth shades of grey are clearly different than the noisy texture in other parts of Figure 3.1. It seems clear that the measurements from the affected part of the array do not reflect true expression values, but are somehow distorted.

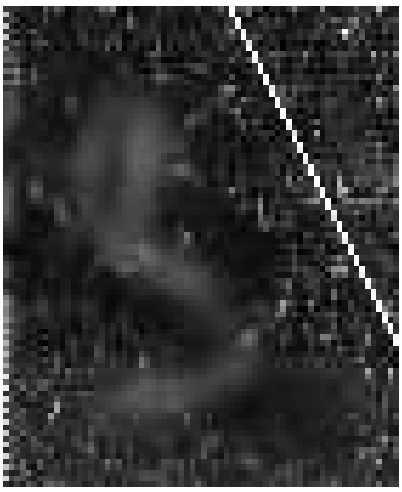


Figure 3.3: A closer look at the suspicious region of Figure 3.1. The triangle up and right from the artificial white line is considered to represent non-distorted expression values.

In order to get some kind of idea about the scale of the problem, the faulty area was roughly marked out with the white line visible in the figure. The area on the left-down side of the line was considered faulty. The PM probes[1] residing on the smeared area and the corresponding probe sets were identified. The 2231 probes were quite nicely distributed between different probe sets. Two probe sets suffered the most, but still not too much, considering that the probe sets only had three out of a total of eleven probes that were affected by the fault. Hopefully the remaining eight probes will steer the observed expression close to the true value. Most probe sets struck by the quality problem, 2141 exactly, only had one PM probe in the affected area of the microarray.

3.1.3 Quantitative QC

Although visual inspection is a great help in picking up the most glaring deficiencies in the quality of data, hard numbers are hard to beat in terms of objectivity. The following sections present a few quality control measures that are recommended by the array manufacturer and based on their MAS 5.0 preprocessing system (see Section 3.2.3).

These QC measures can easily be computed with the *simpleaffy* [58] software package, which is part of the Bioconductor [18] project. The QC results were produced with version 2.0.13 of *simpleaffy*. Bioconductor is in-

[1] MM probes are not used in our analyses

tegrated with R [42], a free software environment for statistical computing and graphics. The programming language used in R is mostly compatible with the S programming language [53].

Quantitative QC measures are objective as such, but their interpretation is quite difficult and best left to domain experts. Thus the discussion about the QC measures and their results is kept brief here.

The following QC measures are calculated on the probe or probe set level, after the image processing stage, without the actual scanned image of the microarray. Another option is quality control based on image processing. For more information on this kind of QC, see e.g. [45] and [22, pp. 24–28]. The image processing procedures tend to differ based on the type of the microarrays used. The processing of image data originating from Affymetrix arrays is typically done with proprietary software provided by the array manufacturer, as was done with the microarrays related to this thesis.

Average background

Background signal means the part of the signal measured from a microarray that is caused by auto-fluorescence and non-specific binding [2, p. 87]. Microarrays that are to be compared should have comparable background values [2, p. 38].

The average background levels measured from the data of Section 2.2 are presented in Figure A.1(a). Array 26 stands out in the crowd, but otherwise the numbers look comparable enough to the untrained eye. Considering the lack of "official guidelines regarding background" [2, p. 38], the results are not analysed any further here.

Scale factor

Scale factor or scaling factor reflects the overall expression level of an array, or more specifically the amount of scaling needed to bring the trimmed mean[2] expression level of the array to a certain level that is common to all the arrays [57, p. 4]. The scaling is applied at the probe set level [2, pp. 47–48]. It is intuitively clear that all arrays in the same experiment should ideally have similar scale factors. Comparatively large and small scale factors, indicative of low and high overall expression levels respectively, should raise some suspicions.

Affymetrix recommend that the scale factors of microarrays to be compared with each other shouldn't differ more than 3-fold [2, p. 40]. It is not

[2] A certain percentage of the smallest and largest values are excluded from the trimmed mean

entirely clear, however, what 3-fold means here. In the opinion of the author, the upper limit of range $(x, 3x)$ is 3-fold compared to the lower limit. That is, a 3-fold difference between two numbers means that the *ratio* of the larger number to the smaller one *is three*. This is very simple.

Another interpretation is that "within 3-fold" [57, p. 4] describes the range $(y, y + 3)$ in a logarithmic scale. Confusion is unavoidable, as a ratio of three suddenly becomes a ratio of eight ($2^3 = 8$) due to the use of the $\log_2$ scale in the graphical QC overview plots produced by *simpleaffy*. An arbitrarily large ratio between the largest and the smallest scale factor could be made to fit this interpretation of 3-fold by choosing a suitable logarithm base. The author thinks that this interpretation of "within 3-fold" is *wrong*. If this isn't the case, then the word "fold" must have a new implicit meaning when used in connection with scale factors.

In the author's opinion, the documentation of the *simpleaffy* package is not clear on all aspects. The use of the logarithmic scale is not motivated or even mentioned in any of the three relevant articles or manuals [34, 57, 58]. The use of the $\log_2$ scale was discovered by carefully comparing raw, non-transformed scale factors and the graphical output of the *simpleaffy* package. The use of the $\log_2$ scale is also apparent from the source code of the package. Having the source code of a program available for inspection is useful, but it shouldn't have to make up for insufficient documentation.

Figure A.2 shows the scale factors of the arrays used in this thesis, and three interpretations of a "within 3-fold" range. The scale factors are not within 3-fold in the traditional sense, as can be seen from Figure A.2(c). If arrays 23 and 25 — both measuring the Met5A cell line — are excluded (see Table 2.2 for the array numbering scheme), the scale factors of the other arrays are within 3-fold. In other words, the largest scale factor is less than three times the size of the smallest one.

Figure A.2(a) shows the scale factors in the $\log_2$ scale as in *simpleaffy*. The ± 1.5 range around the mean logarithmic scale factor is also shown. This corresponds to the interpretation of "within 3-fold" used in *simpleaffy*. The use of the mean scale factor as the centre point of the allowed range was apparent from [57]. The exact meaning of "mean" could be confirmed by reading the relevant part of the source code of *simpleaffy*. Thus, the centre of the range in Figure A.2(a) is the mean of the logarithmic scale factors, not the logarithm of the mean scale factor. As can be seen in the figure, the scale factor of array number 23 is out of the range. According to [57], *simpleaffy* should have alerted the user about this, but it didn't.

Figure A.2(b) is the same as Figure A.2(a), but the centre point of the ± 1.5 range is a little different. Now all the scale factors fit the $\log_2$ interpretation of the "within 3-fold" rule.

Number of genes called present

The MAS 5.0 preprocessing method has a "detection algorithm" where each probe set on a microarray is assigned to one of three expression classes: present, marginal, or absent [2, pp. 42–44]. The number of probe sets called present can be compared to the total number of probe sets on an array. The result is the "percent present" statistic, which should be similar across arrays in the same experiment [57, p. 4].

Figure A.1(b) shows the "percent present" statistic of each array described in Table 2.2. There is some variation in the numbers, but generally speaking they seem to be quite well in line with each other. Array number 23 has the lowest percentage of probe sets called present. As seen in Figure A.2, it also has the highest scale factor. According to [57, p. 4], these two facts together may mean that less RNA was used with array 23 than with the other arrays. Remember that there is also a clearly visible defect in the pseudo image of the array, as shown in Figure 3.1.

3′ to 5′ ratios

Some long genes are measured with several probe sets, which represent different parts of the gene's RNA transcript. The ratio of expression signals from the 3′ and 5′ probe sets[3] can be used as a QC measure [57, p. 4].

Two genes, β-actin and GAPDH, are used as quality control genes due to their ubiquitous expression in most cell types [57, p. 4]. Figure A.1(c) shows the $\log_2$ 3′ to 5′ ratios of these genes in the arrays of Table 2.2. The ratios were computed with *simpleaffy*. It is clear from the source code that the package uses $\log_2$-ratios of MAS 5.0 expression signals, but the documentation [57, pp. 4, 8–9] is again very confusing. Firstly, the use of logarithmic scale is not mentioned. Secondly, the suggested thresholds of 1.25 (GAPDH) and 3 (β-actin) would be more logical in a linear scale, since MAS 5.0 expression signal — the foundation of 3′ to 5′ ratios — follows a linear scale [2, p. 45]. Affymetrix documentation also mentions a threshold ratio of 3, but does not refer to the use of logarithms scale [2, p. 39][3, p. 5.B.19].

In addition to the 3′ to 5′ ratios, Figure A.1(c) also shows the limits suggested in [57]. Almost all the ratios lie below the limits. The use of the logarithmic scale with these ratios is quite controversial, similarly as with scale factors. Affymetrix documentation suggests that observing the variation between 3′ to 5′ ratios may be a better approach than using a strict, more or less arbitrary threshold value [3, p. 5.B.19]. In Figure A.1(c), the variation between arrays measuring the same cell line (arrays 1–12, 13–22,

[3] For an explanation on the 3′ and 5′ ends of a DNA strand, see [49, p. 9]

23–27) is quite noticeable. However, there is no single outlier array, but the variation is more of an overall phenomenon.

3.1.4 QC verdict

Qualitative QC — the microarray intensity images — didn't reveal anything too alarming, even considering the blurred region of Figure 3.1. When it comes to the results of the quantitative QC measures, the limited experience of the author doesn't allow any far reaching conclusions. The data from array 23, however, is somewhat suspicious, considering both the qualitative analysis and two quantitative QC measures: scale factor and number of genes called present (percent present). Still, the quality of the data seems to be sufficient for further analyses.

3.2 Preprocessing

3.2.1 Preprocessing — what does it mean here?

Preprocessing is not an accurately defined term. It can mean different things for different researchers. Here it stands for the computation of expression values from probe-level data. The goal is to achieve a reliable measure of the abundance of different RNA sequences. The various preprocessing methods designed for oligonucleotide arrays take probe-level data (PM and MM probes) as their input and produce a summary value for each probe set. The summary value is passed on to later stages of the analysis pipeline as the measure of expression for the gene in question.

Section 3.2.2 covers one method for preprocessing gene expression data produced with oligonucleotide arrays. The method is also used in the experimental part of the thesis. Other preprocessing methods are briefly discussed in Section 3.2.3.

3.2.2 Robust multichip average

Robust multichip average (RMA) [10, 24, 25] is a relatively simple but very powerful stochastic model based preprocessing method. It is implemented in *affy* [17], a software package also part of the Bioconductor [18] project.

The RMA expression measure can be summarised [25] in three or four steps, depending on whether log-transformation is considered a step of its own:

1. Background correction of probe values

2. Normalisation of probe values across arrays using quantile normalisation
3. Logarithmic (base 2) transformation of probe values
4. Computation of probe set level expression measures

Step 1 starts with raw probe values, and each following step uses the output of the previous stage as its input. The four steps of RMA are described here one by one.

Background correction

According to [11, pp. 16–17], the term *background correction* — when used in the context of microarrays — describes methods that should accomplish various tasks. The most obvious of the tasks is the removal of *background noise*. A background adjustment procedure should also adjust for *non-specific binding*[4] and produce expression estimates that "fall on the proper scale".

The derivation of the RMA background correction formula can be found in [11, pp. 17–20]. The basic assumptions and results are presented here. The background correction is based on a model where the observed signal S of a probe consists of two parts: $S = X + Y$. Here X is the actual signal, which is assumed to be exponentially distributed: $X \sim \exp(\alpha)$. Y is a background signal: $Y \sim \mathrm{N}(\mu, \sigma^2)$. There is an additional restriction $Y \geq 0$, which truncates the normal distribution at zero.

The background correction is given [11, p. 20] as the expectation of X conditional to the observation $S = s$:

$$\mathrm{E}(X|S = s) = a + b\frac{\phi\left(\frac{a}{b}\right) - \phi\left(\frac{s-a}{b}\right)}{\Phi\left(\frac{a}{b}\right) + \Phi\left(\frac{s-a}{b}\right) - 1} \quad . \tag{3.1}$$

Here ϕ and Φ are the probability density function (PDF) and the cumulative distribution function (CDF) of the standard normal distribution, respectively. Further abbreviations and notations in Equation (3.1) are $a = s - \mu - \sigma^2\alpha$ and $b = \sigma$. Parameters μ, σ, and α are estimated from observed probe intensities using an ad-hoc procedure described in [11, p. 21].

Since MM probes are not used in RMA, the background correction presented in Equation (3.1) is applied to *PM probes only* [11, p. 21].

According to [11, p. 21], the term $\phi\left(\frac{s-a}{b}\right)$ is close to zero and the term $\Phi\left(\frac{s-a}{b}\right)$ is close to one in most applications. This means that the numerator and the denominator in Equation (3.1) can be reduced to one term each. This is also the approach taken in *affy* (version 1.6.7), as is evident from the source code of the package.

[4] Hybridisation of a DNA sequence that is not complementary to the probe

Normalisation across arrays

Normalisation aims to remove the non-biological variation between microarrays used in the same experiment [11, p. 39]. The different scale factors of Figure A.2 — although strictly speaking an indication of variation between arrays on the probe set level, not the probe level — show that there can indeed be differences between the overall expression levels of arrays belonging to the same experiment.

RMA uses quantile normalisation [10], which is applied to background corrected PM probe values. Quantile normalisation not only unifies their means, but also makes the distribution of PM probe values identical in every array. The method has been tested against other normalisation methods and found to perform well with regard to both the quality of the results and the running time of the algorithm [10].

It is apparent that quantile normalisation involves the following assumption about the distribution of gene expression values: the distribution is roughly the same, regardless of which tissue sample is studied. Some genes are more strongly expressed in one sample than another while other genes behave in the reverse manner, and the end result is a similar distribution of expression values in all samples. The degree to which the assumption holds naturally depends on the particular samples used.

Quantile normalisation is easily done with the following four steps. The list is adapted from [10].

1. Form a matrix X with dimensions $p \times n$. Here p is the number of probes and n is the number of arrays. Each array (PM probes) is a column of the matrix.

2. Sort each column of X, which gives you X_{sort}.

3. Take the mean across rows of X_{sort}. Assign the mean to every element in the row, which gives you X'_{sort}.

4. Rearrange each column of X'_{sort} to have the same ordering as in the original X. This gives you $X_{\text{normalised}}$, which has the same distribution of numbers in each column.

Figure 3.4 illustrates the process. The numbers in each column of the example matrix are unique. Thus there are no tied ranks, which would make the sorted order of the affected column ambiguous. In a situation with tied ranks, the order of equal numbers depends on the sorting algorithm used. When there are hundreds of thousands of values in each column, as in the case of the microarray data used here, the effect of such ambiguities on any individual normalised value is probably very small.

$$\xrightarrow[\text{a matrix}]{\text{1. form}} \begin{pmatrix} 18 & 24 & 1 & 7 & 15 \\ 22 & 7 & 7 & 12 & 16 \\ 2 & 6 & 13 & 21 & 20 \\ 8 & 14 & 21 & 22 & 4 \\ 9 & 18 & 23 & 4 & 10 \end{pmatrix} \xrightarrow{\text{2. sort}} \begin{pmatrix} 2 & 6 & 1 & 4 & 4 \\ 8 & 7 & 7 & 7 & 10 \\ 9 & 14 & 13 & 12 & 15 \\ 18 & 18 & 21 & 21 & 16 \\ 22 & 24 & 23 & 22 & 20 \end{pmatrix} \xrightarrow[\text{means}]{\text{3. take}}$$

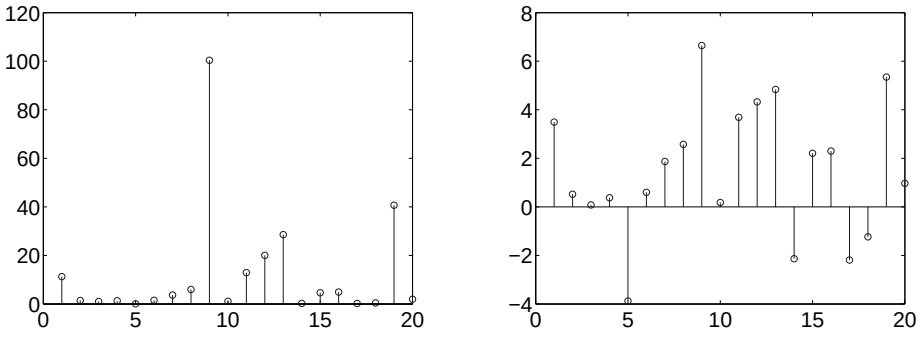
$$\ldots \begin{pmatrix} 3.4 & 3.4 & 3.4 & 3.4 & 3.4 \\ 7.8 & 7.8 & 7.8 & 7.8 & 7.8 \\ 12.6 & 12.6 & 12.6 & 12.6 & 12.6 \\ 18.8 & 18.8 & 18.8 & 18.8 & 18.8 \\ 22.2 & 22.2 & 22.2 & 22.2 & 22.2 \end{pmatrix} \xrightarrow{\text{4. unsort}} \begin{pmatrix} 18.8 & 22.2 & 3.4 & 7.8 & 12.6 \\ 22.2 & 7.8 & 7.8 & 12.6 & 18.8 \\ 3.4 & 3.4 & 12.6 & 18.8 & 22.2 \\ 7.8 & 12.6 & 18.8 & 22.2 & 3.4 \\ 12.6 & 18.8 & 22.2 & 3.4 & 7.8 \end{pmatrix}$$

Figure 3.4: Quantile normalisation (example)

Log-transformation

The third step of RMA is log-transformation, where the background corrected, quantile normalised probe values are transformed to a logarithmic scale. Log base 2 is the usual choice. [25]

Log-transformation is very important for the success of the final phase of RMA preprocessing. The most obvious effect of log-transformation is that it facilitates the comparison of data of different orders of magnitude, as seen in Figure 3.5. The differences between the small values in Figure 3.5(a) are obscured by the larger values. Figure 3.5(b) shows the same 20 numbers on a logarithmic scale. Now the comparison of the whole range of values is easier.



(a) Linear scale (b) Logarithmic scale, base 2

Figure 3.5: Effect of log-transformation

Summarisation

Summarisation is the final step of RMA preprocessing, and results in a set of probe level expression values for each array involved in the gene expression experiment. The background corrected, normalised, and log-transformed PM values $Y_{ij}^{(n)}$ are expected [25] to follow the model

$$Y_{ij}^{(n)} = \mu_i^{(n)} + \alpha_j^{(n)} + \varepsilon_{ij}^{(n)} \ , \tag{3.2}$$

where i denotes the array, j the probe, and n the probe set.

The interesting quantity in Equation (3.2) is $\mu_i^{(n)}$, the log scale expression value of probe set n in array i. Probe affinity $\alpha_j^{(n)}$ is a measure of the natural "fondness" of probe j in probe set n for the nucleotide sequence targeted by the probe set. According to [4], "all probes have different thermodynamic properties and binding efficiencies". The probe affinities are restricted by $\sum_j \alpha_j^{(n)} = 0$. The error term $\varepsilon_{ij}^{(n)}$ in Equation (3.2) has zero mean. [25]

RMA uses a robust procedure such as median polish to estimate the parameters in Equation (3.2) [25]. Robustness here means the ability to correct for outlier probes, the values of which clearly lie outside the usual range of values. Median polish is also used in the *affy* (version 1.6.7) implementation of RMA. A practical description of the algorithm is available in [11, p. 69]. The estimate of $\mu_i^{(n)}$ is the RMA expression measure [25]. The estimates of the other terms in Equation (3.2) may be dropped.

3.2.3 Other preprocessing methods

Alternatives to RMA are discussed briefly here. There are several more or less successful preprocessing methods available. Some of the better ones would well be worthy of a try with the data in this thesis. However, such hands-on experiments are left out for practical reasons.

MAS 5.0 [2, pp. 41–56] is an algorithm that is used in the current version of Affymetrix microarray analysis software (Microarray Suite = MAS). This is the obvious first choice for the preprocessing of Affymetrix microarrays. Remember that the quantitative QC measures of Section 3.1.3 are also based on MAS 5.0. Model Based Expression Index (MBEI[5]) is a preprocessing method that is based on a statistical model of PM $-$ MM probe intensities [30, 31]. Comparisons of MAS 5.0, MBEI, and RMA can be found in [24] and [25].

[5]The terms MBEI and dChip are sometimes used interchangeably. The former is a computer program that implements the MBEI expression measure.

GCRMA [60] may be considered a more advanced or more complicated version of RMA. The main difference between RMA and GCRMA is the background correction method used. The normalisation and expression value summarisation steps are the same in both methods [60, p. 14]. PerfectMatch is an implementation of another preprocessing method, which is based on a model called PDNN (position-dependent nearest neighbour) [61]. MAS 5.0, RMA, PerfectMatch, and GCRMA are compared in [59].

3.2.4 Concluding thoughts

A comparative benchmark for preprocessing methods is discussed in [13]. The accompanying web site, http://affycomp.biostat.jhsph.edu/, has over 60 benchmark results in each category (on 17th Nov 2005). There are clearly so many approaches to the preprocessing problem, that they all cannot be covered here.

None of the expression measures can decisively be declared the best. Many of the different techniques are quite well founded in theory, but the analysis of microarray data is not the easiest of endeavours. Although some comparisons have been made when "the truth" about the data is known, the relative performance of the different methods with real world data is hard to predict and varies from case to case. RMA is the method of choice in this thesis, because it is an established method with mostly good performance characteristics, and there is a free implementation available. On one hand the elegant and relatively simple theory behind RMA is quite impressive. On the other hand, some parts of RMA, at least background correction, may be too simple.

Chapter 4

Clustering gene expression time series data

4.1 Introduction

Microarray analyses differ from typical clinical studies, as shown in Figure 4.1. The number of variables (genes) in microarray data is large, while the number of cases (microarrays) is small. Using standard biostatistical techniques would require more cases than what is usually available with microarray data. Keeping that in mind, methods used in computational sciences and machine learning are more appropriate tools for analysis. [27, pp. 10–12]

4.2 Motivation for clustering and analysis of its applicability

Cluster analysis is an instrument of *descriptive modelling* in the broad field of *data mining*. A descriptive model presents the main features of the data in a convenient form. Cluster analysis aims to find "whether the data fall into distinct groups", which would mean that the data set is heterogeneous. [21, pp. 271, 293].

Clustering means the organisation of data points into groups. The clusters should be "sensible", which usually means that points in the same cluster should be somehow "similar", and points in different clusters should be "different". The discovery of differences between groups of data points and similarities within members of individual groups will hopefully lead to useful findings. Clustering belongs to the problem class of unsupervised learning or learning without a teacher. [50, p. 397]

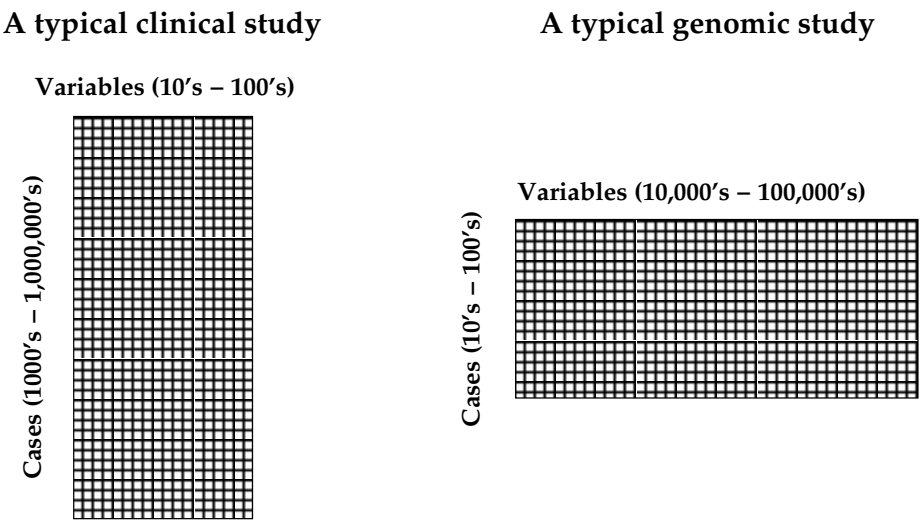


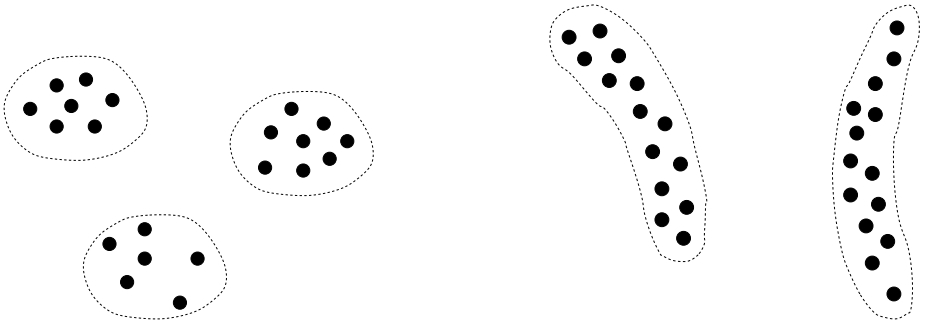
Figure 4.1: Microarray analyses deal with more variables and less cases than clinical studies [27, p. 11].

Similarity or dissimilarity of data points can be assessed in several ways. Some of the most common proximity measures between two real-valued vectors (data points) are the Euclidean distance and the inner product [50, pp. 404–409]. The "natural" type or shape of clusters varies between different data sets. The shape of clusters produced by a clustering algorithm often depends on what kind of proximity measure between a data point and a set of points is used, or on the type of representative a cluster is chosen to have [50, pp. 418–425].

Figure 4.2 shows two different shapes of clusters. When the Euclidean distance is used, the compact clusters of Figure 4.2(a) could be represented by a single point in the middle of the cluster. The elongated clusters of Figure 4.2(b) could be more accurately represented with a line. Then a data point could be assigned to the cluster with the closest point or line representative, respectively.

It seems plausible that similarly expressed genes may have a common biological function. This is the core assumption that makes clustering a widely used analysis strategy in gene expression studies [27, pp. 60–61]. Clustering can be used for hypothesis generation, among other things [50, p. 400]. The nature of the gene expression data used in this thesis is largely unknown, and clustering is used as a means to explore the data and generate hypotheses.

There are some problems or concerns related to the use of clustering in general or for the purpose of gene expression analyses. Three of them are



(a) Compact (b) Elongated

Figure 4.2: Different shapes of clusters

mentioned in [27, pp. 60–61]:

- Gene products may have many different roles under different circumstances. This is a confusing thing considering the goal of finding groups of functionally related genes.
- What does "functionally related" mean? In the end, all genes are functionally related through one broad function: keeping the organism alive.
- What is the most appropriate way to measure the similarity of expression between different genes?

A few things need to be remembered when looking at the results of clustering. Firstly, the validity of the clusters should be checked. Some clustering methods have a built-in mechanism that evaluates the statistical significance of the clusters. Secondly, the clusters only reflect the particular gene expression measurements that they are based on. All hypotheses suggested by the results of clustering must be biologically validated. [27, pp. 66–67]

Despite the concerns, clustering can be considered a good starting point for gene expression analyses. The prospect of finding functionally related groups of genes is very compelling.

4.3 Clustering short time series data

The term "time series" generally stands for a sequence of data points measured at different times. Time series can be thought in terms of changes between consecutive time points. When a set of time series is clustered, the aim is to find the typical patterns of change present in the data.

Time series data, such as the data described in Section 2.2, have an important special property. The time points have a particular order, and there could be some dependencies between them. For example, small changes in gene expression between consecutive time points might be more probable than larger variations.

Many clustering methods can be used on very different types of data. For a given purpose, a general solution may be less powerful than a more specialised one. When clustering time series data, it is beneficial to choose an algorithm that takes the special nature of the data into account. It should also be remembered that the data used in the thesis consist of only a few different time points.

4.3.1 An algorithm designed for the task

We use an algorithm designed for clustering short time series gene expression data. The data used in the thesis fit this description perfectly. The algorithm, presented in [16], doesn't seem to have a name. In this thesis, it will occasionally be called "Ernst's algorithm". There is a Java implementation of the algorithm available. It is free for non-commercial research use, and can be obtained from http://www.cs.cmu.edu/~jernst/st/ (referenced 20th Jan 2006). The following step-by-step list presents the algorithm in a nutshell:

1. Enumerate all possible model profiles. A user selectable parameter c determines the maximum amount of change units between time points. See Section 4.3.2.

2. Select a set of m distinct model profiles. See Section 4.3.3.

3. Assign each data profile to a model profile. A clustering is formed. See Section 4.3.4.

4. Identify significant model profiles (clusters) with a permutation test. See Section 4.3.4.

5. Group significant profiles. Similar model profiles are grouped together. This step is covered in Section 4.3.5, but omitted in our analyses of the gene expression data.

An interesting aspect about the algorithm is that it uses model profiles that are independent of the data. A permutation test then determines which profiles were significantly present in the data at hand. This approach appeals to the common sense: we want to select distinct model profiles that help us bring out the *different* clusters in our data.

In [16], the algorithm was found to perform favourably in comparison with another algorithm designed for time series data, CAGED [43]. According to [16], one reason for this was the tendency of CAGED to produce clusters that contain too many genes, which hinders its ability to bring out the special features of small but significant clusters.

4.3.2 Enumerating model profiles

The first step of the clustering algorithm is to enumerate all the possible expression profiles. Strictly speaking, this is hardly possible, since the number of different expression profiles is practically unlimited. The approach taken in [16] is to discretise the amount of change between consecutive time points. The first value of the model profiles is zero by design. If x is the expression value at time point t, then the value at $t + 1$ can take integer values between $x - c$ and $x + c$. Thus the maximum amount of change between consecutive time points is c units. The units are scale-invariant, which is evident from the distance measure used in the algorithm (see Section 4.3.3).

The parameter c can be called the smoothness parameter, for example. The bigger c is, the more time the clustering algorithm takes to run. The choice of c only increases the running times of the profile enumeration and profile selection phases. Other parts of the algorithm are not affected. Although it might be somewhat tempting to use a big value of c, increasing it "too much" provides diminishing returns. Tests referred to in [16] suggest that very similar results are achieved with values $c \in \{1, 2, 3\}$.

Another factor limiting the choice of c is the number of all possible model profiles:

$$|P| = (2c + 1)^{n-1} \ , \tag{4.1}$$

where n is the number of time points in a profile. With the relatively modest values of $n = 6$ and $c = 3$, $|P| = 16807$. Choosing $c = 2$ brings that number down to 3125. Having a large number of profiles is also a problem in the following phase of the algorithm, where pairwise distances between profiles are computed. The complexity of calculating all pairwise distances in a given data set is quadratic in the number of data points.

Algorithm 4.1 is a simple recursive solution for enumerating the possible model profiles. The parameters are the length of the profiles n and the smoothness parameter c. The resulting set P contains all profiles that start with a zero, have n time points, and fulfil the constraint on change per time unit set by the choice of c.

Algorithm 4.1 ENUMERATEPROFILES A recursive algorithm for enumerating all possible model profiles

```
ENUMERATEPROFILES(n, c)
 1  if n = 1
 2     then P ← {0}          ▷ a set containing one profile with the initial zero
 3     else P ← {}
 4          P_rec ← ENUMERATEPROFILES(n − 1, c)
 5          while |P_rec| > 0
 6               do let p be any profile in P_rec
 7                  P_rec ← P_rec \ {p}
 8                  let p_end be the value at the last time point of p
 9                  let S be the set of integers from p_end − c to p_end + c
10                  while |S| > 0
11                       do let s be any number in S
12                          S ← S \ {s}
13                          let q be the profile where s is appended to the end of p
14                          P ← P ∪ {q}
15  return P
```

4.3.3 Selecting distinct profiles

It is obvious that the number of model profiles needs to be reduced. With 3125 or even 16807 model profiles, the resulting clusters would be very small, and some of the clusters would be quite similar to each other. That is clearly against our goal, which is to find significant and distinct patterns of expression.

When a set R consisting of m distinct profiles needs to be selected, there are many different ways to formulate the problem. Here, true to [16], we try to maximise the minimum distance between any two profiles p_1 and p_2 in $R \subset P$, where P is the set of all possible model profiles:

$$\arg\max_{R:R\subset P,|R|=m} \min_{p_1,p_2\in R} d(p_1, p_2). \tag{4.2}$$

Here d is a distance measure. An obvious alternative would be to maximise the average of all pairwise distances in R.

The selection criterion of Equation (4.2) requires a measure for objectively judging the distance between two profiles. The distance measure used here — and throughout the different phases of this clustering algorithm — is

$$d(\mathbf{x}, \mathbf{y}) = 1 - \rho(\mathbf{x}, \mathbf{y}), \tag{4.3}$$

where $\rho(\mathbf{x}, \mathbf{y})$ is Pearson's correlation coefficient between the two profiles (vectors) $\mathbf{x}$ and $\mathbf{y}$, shown in Equation (4.4). In the equation, n is the length of the vectors.

$$\rho(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \sqrt{\sum_{i=1}^{n} b_i^2}}, \quad a_i = x_i - \overline{x},\ b_i = y_i - \overline{y} \tag{4.4}$$

Actually, it is not accurate to call Equation (4.3) a distance measure, because it does not fulfil two conditions [50, p. 404] of a *metric*: triangular inequality (see Equation (4.5)) and Equation (4.6).

$$d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}), \quad \forall \mathbf{x}, \mathbf{y}, \mathbf{z} \in X \tag{4.5}$$

$$d(\mathbf{x}, \mathbf{y}) = d_0 \quad \text{if and only if} \quad \mathbf{x} = \mathbf{y} \tag{4.6}$$

The X in Equation (4.5) is the domain of all vectors that d operates on. The d_0 in Equation (4.6) is the minimum distance between any two vectors, which in the case of Equation (4.3) is zero. Although a more accurate name for Equation (4.3) would be for example "dissimilarity measure" [50, p. 404], the name "distance measure" is often used in this thesis.

According to [16], the solution to Equation (4.2) is NP-hard. However, the solution can be approximated in polynomial time. The greedy algorithm of Algorithm 4.2 — with the distance measure of Equation (4.3) — is guaranteed to achieve a set R, where the minimum distance between profiles is at least a quarter of that in the best set R' [16].

Algorithm 4.2 SELECTVECTORSMAXMINDIST A greedy algorithm for choosing m distinct profiles (appeared in [16])

SELECTVECTORSMAXMINDIST(d, P, m)

1 let $p_1 \in P$ be the profile that always goes down one unit between time points
2 $R \leftarrow \{p_1\}$
3 $L \leftarrow P \setminus \{p_1\}$
4 **for** $i \leftarrow 2$ **to** m
5 **do** let $p \in L$ be the profile that maximises $\mathbf{min}_{p_1 \in R} d(p, p_1)$
6 $R \leftarrow R \cup \{p\}$
7 $L \leftarrow L \setminus \{p\}$
8 **return** R

There is a small problem in Algorithm 4.2 in the case of equally good profiles. More specifically, there are often several profiles, not one profile p,

that maximise the minimum distance to profiles already in R. Algorithm 4.2 doesn't take these situations into account. Instead, it assumes that there is one optimal choice for every situation, in the greedy sense.

The improved Algorithm 4.3 is a simple randomised algorithm [36]. It is a modification of Algorithm 4.2 and fixes the problem by randomly selecting one of the best candidate profiles. The user can specify the number of repeated runs with the parameter *repeats*. The results achieved with the improved algorithm are typically a little better than with the original algorithm, assuming that the ambiguity in the original algorithm is rectified in a deterministic way.

Algorithm 4.3 SELECTVECTORSMAXMINDISTRANDOM A randomised greedy algorithm for choosing m distinct profiles

SELECTVECTORSMAXMINDISTRANDOM$(d, P, m, repeats)$

```
1  dist_best ← −∞
2  for i ← 1 to repeats
3       do R_t ← SELECTHELPER(d, P, m)
4          dist_temp ← min_{(p1,p2)∈R_t×R_t} d(p1, p2)
5          if dist_temp > dist_best
6            then dist_best ← dist_temp
7                 R ← R_t
8  return R
```

SELECTHELPER(d, P, m)

```
1  let p1 ∈ P be the profile that always goes down one unit between time points
2  R ← {p1}
3  L ← P \ {p1}
4  for i ← 2 to m
5       do let p ∈ L randomly be one of the profiles that maximise min_{p1∈R} d(p, p1)
6          R ← R ∪ {p}
7          L ← L \ {p}
8  return R
```

Table 4.1 shows a comparison of the deterministic Algorithm 4.2 and the randomised Algorithm 4.3. The results of the deterministic algorithm depend on implementation details: a single profile must somehow be chosen in the case of multiple equally good options. Clearly something is done differently in the implementation used in [16], because the minimum distance

between model profiles there [16, Fig. 4a] is 0.1615 ($n = 5, c = 2, m = 50$), while our implementation of Algorithm 4.2 gives a set of profiles with minimum distance 0.1548. The results of the randomised algorithm depend on chance and the choice of *repeats*. Here, the randomised algorithm was run with 100 or 1000 *repeats*. The choice of 100 *repeats* only had to be used in the computationally demanding $n = 6, c = 3, m = 50$ case. The running time of the algorithm is obviously linear in *repeats*.

Table 4.1: Minimum distance between profiles in set R (bigger is better)

Type of algorithm	**Parameters**					
	n	5	6	5	6	3
	c	2	2	3	3	6
	m	50	50	50	50	16
Deterministic		0.1548	0.2572	0.1784	0.2843	0.0695
Randomised		0.1708	0.2929	0.1982	0.2960	0.0695

The improvement in results achieved with Algorithm 4.3 is quite small, but an improvement nonetheless. Most importantly, Algorithm 4.3 is a precisely defined solution to the profile selection problem, whereas Algorithm 4.2 is not. Using the improved algorithm is preferable to using Algorithm 4.2. After all, the minimum distance in the set of distinct profiles should be maximised, and Algorithm 4.3 gets closer to the goal than Algorithm 4.2 with a reasonable increase in running time.

Since the selection task is NP-hard, some compromises between speed and quality must be made with practical problem sizes in order to keep the computation time within practical bounds. Greedy algorithms like Algorithm 4.3 can be used to produce a satisfactory solution to a problem that is too demanding to be solved optimally.

Table 4.2 shows some measurements of the CPU time required by a MATLAB implementation of Algorithm 4.3 with a few different choices for the parameters. Remember that n and c have an effect on the running time of profile selection, because they affect the number of profiles to choose from. The times were measured on a Linux computer with Opteron 2200 MHz processors. The measurements are indicative of the scaling of the algorithm, although some program versions and other background factors might have changed between different runs, and no repeated runs were made.

Removing redundant profiles

Before the selection of distinct profiles, it might be nice to get rid of redundant profiles. Since the distance measure of Equation (4.3) is based on correlation, only the shape of profiles matters, not the scale. This applies both

Table 4.2: CPU time of Algorithm 4.3 (in seconds)

		Parameters				
Number of	n	5	6	5	6	3
repeats	c	2	2	3	3	6
	m	50	50	50	50	16
100		246	1440	1116	7806	15
1000		2486	13633	10636	–	140

to the selection of model profiles and to the clustering, which is explained in the following sections.

Algorithm 4.4 takes as input the set of all model profiles P, the smoothness parameter c, and the length of the model profiles n. The output is a set of profiles R, which doesn't contain any profiles that are a multiple of another profile. In other words, of a set of profiles that are equal with respect to the distance measure of Equation (4.3), only the flattest one is kept. The removal of redundant profiles is purely and simply a cosmetic step. As such it is an optional procedure before profile selection. The original description of the clustering algorithm doesn't have this step [16].

Algorithm 4.4 REMOVEREDUNDANT A simple algorithm for removing redundant model profiles

```
REMOVEREDUNDANT(P, c, n)
 1  R ← {}
 2  let Primes be the set of all prime numbers up to and including c
 3  while |P| > 0
 4        do let p be any profile in P
 5           P ← P \ {p}
 6           nonredundant ← TRUE
 7           let p_i, i ∈ 1, ..., n, be the values at each time point of p
 8           for each prime in Primes
 9                do if each p_i is divisible by prime
10                     then nonredundant ← FALSE
11                          break
12           if nonredundant
13             then R ← R ∪ {p}
14  return R
```

4.3.4 Clustering and finding significant profiles

The clustering of the expression profiles can be done with Algorithm 4.5. Algorithms 4.6, 4.7, and 4.8 are helper functions. The algorithms are based on their written descriptions found in [16].

Algorithm 4.5 STSCLUSTER A clustering algorithm for Short Time Series gene expression data. See algorithms 4.6, 4.7, and 4.8 for related helper functions. This is a detailed description of Ernst's algorithm [16].

STSCLUSTER$(G, P, DoGrouping, \alpha, \delta)$

1 $\triangleright$ G is an array of gene expression profiles
2 $\triangleright$ P is an array of model profiles
3 let d, $d(x, y) = 1 - \rho(x, y)$ be a distance function
4 let n be the length of each profile in G and P
5 let *perm* be the "straight" permutation $(1 \ldots n)$
6 $Idx \leftarrow$ ASSIGNTOCLUSTERS$(d, P, G, perm)$
7 let T be an array (length $|P|$) containing the column-sums of Idx
8 $[Pval, E] \leftarrow$ PERMTEST(d, T, P, G)
9 sort $Pval$ in ascending order, resulting in permutation $perm_s$
10 $R \leftarrow P$
11 apply $perm_s$ to R, E, and T
12 apply $perm_s$ to columns of Idx $\triangleright$ columns (model profiles) change places
13 $n_s \leftarrow 0$ $\triangleright$ number of significant clusters
14 **for** $i \leftarrow 1$ **to** $|R|$
15 **do if** $Pval[i] < \alpha / |R|$ $\triangleright$ Bonferroni correction
16 **then** $n_s \leftarrow n_s + 1$
17 **else break**
18 **if** *DoGrouping* $\triangleright$ Optionally group significant profiles
19 **then** $Rgrp \leftarrow$ GROUPPROFILES(R, n_s, T, d, δ)
20 **else** $Rgrp \leftarrow$ NIL
21 **return** $[R, Pval, Idx, E, n_s, Rgrp]$

The main function, presented in Algorithm 4.5, always takes as input the gene expression profiles to be clustered G, the distinct model profiles P, and a significance threshold α. Parameter *DoGrouping* determines whether similar model profiles are grouped together, and δ adjusts the sensitivity of that optional grouping. In [16], the grouping phase is considered an integral part of the algorithm. However, assigning a meaningful value to δ might be difficult. Analysis of clustering results would also be somewhat more complicated with grouped profiles. The grouping phase is discussed in more

detail in Section 4.3.5.

The clustering algorithm returns many values. R contains the same model profiles that were given in P, this time in the order of increasing P-value. *Pval* are the P-values of the model profiles, where a value small enough means that the number of data profiles belonging to the cluster represented by the model profile is significantly large. The clustering results are contained in *Idx*, a two-dimensional array, where $Idx[i, j]$ marks whether data profile i belongs to cluster j or not. It is zero if profile i doesn't belong to cluster j, and $1/n$ if the profile belongs to a total of n different clusters, one of which is cluster j. The chance of a data profile belonging to more than one cluster seems quite theoretic. In all our experiments, each data profile was assigned to a single cluster. Return value E is the expected number of genes in each cluster according to some assumptions discussed later, in Section "Permutation test". The number of significant model profiles, or significant clusters, is returned in n_s. *Rgrp* contains the groups of significant profiles, if *DoGrouping* was set to TRUE.

Algorithm 4.6 simply assigns the given data profiles G to clusters represented by the model profiles P. Each data profile is assigned to the closest model profile. The distance between profiles is measured by d, which is the distance measure of Equation (4.3) when this algorithm is called from Algorithm 4.5. As mentioned before, there is a possibility of more than one model profile having the smallest distance to a data profile, in which case the data profile is assigned equally to all of the model profiles. Input parameter *perm* determines if and how the order of time points is changed in the data profiles. Array *Cl* contains the result of the clustering.

Algorithm 4.6 ASSIGNTOCLUSTERS Helper function for Algorithm 4.5

ASSIGNTOCLUSTERS$(d, P, G, perm)$

1 let *Cl* be a $|G| \times |P|$ zero array
2 let G_s be the same as G except that each profile is permuted according to *perm*
3 let D be a $|G_s| \times |P|$ array where $D[i, j] = d(G_s[i], P[j])$
4 **for** $i \leftarrow 1$ **to** $|G_s|$
5 **do** ▷ We are looking at the i:th row of D (denoted $D[i,]$)
6 let *Idx* be the index (indices) of the minimum value(s) of $D[i,]$
7 **for** $j \leftarrow 1$ **to** $|Idx|$
8 **do** $Cl\,[i, Idx\,[j]] \leftarrow 1/\,|Idx|$ ▷ i:th row of *Cl* is altered
9 **return** *Cl*

Permutation test

Algorithm 4.7 is a permutation test [20] for determining the significance of clusters. A significant cluster is one that has many genes (data profiles) assigned to it compared to the expected number of genes according to a null hypothesis. The null hypothesis in this case is that the data have no time structure, which means that there are no dependencies between time points. [16]

Algorithm 4.7 PERMTEST Helper function for Algorithm 4.5

PERMTEST(d, T, P, G)

```
1  let n be the length of each profile in P
2  let N be the list of integers from 1 to n
3  let Perms be the set of all permutations[a] on N
4  let Cl be a |G| × |P| zero array
5  for each perm in Perms                ▷ sum up clusterings in all permutations
6       do Cl ← Cl + ASSIGNTOCLUSTERS(d, P, G, perm)
7  ▷ count total number of genes assigned to each cluster
8  let S be an array (length |P|) containing the column-sums of Cl
9  ▷ count expected number of genes according to null hypothesis
10 for i ← 1 to |P|
11      do E[i] ← S[i] / |Perms|        ▷ |Perms| = n!
12 for i ← 1 to |P|                     ▷ compute P-value for each cluster
13      do Pval[i] ← 1 − BINOCDF(T[i] − 1, |G|, E[i] / |G|)
14           ▷ BINOCDF(n, N, p) is an external function for computing the ...
15           ▷ cumulative binomial distribution function with parameters N and p at n
16 return [Pval, E]
```

[a] Using a smaller set of permutations would reduce running time and might give an acceptable approximation of the P-values.

In order to dismantle the time structure possibly present in the data, Algorithm 4.7 applies the clustering implemented in Algorithm 4.6 to all $n!$ permutations achieved by changing the order of the n time points in the expression profiles. If we let s_i^j be the number of expression profiles assigned to cluster $i = 1, \ldots, m$ in permutation $j = 1, \ldots, n!$, then $E_i = (\sum_j s_i^j)/n!$ is the expected number of profiles assigned to the cluster, when considering a random permutation. This corresponds to the idea of independent time points presented in the null hypothesis. It is worth remembering, that generally $E_i \neq |G| / m$. [16]

The parameters of Algorithm 4.7 are distance measure d, model profiles P, data profiles G, and T, which is an array containing the number of data profiles assigned to each cluster. The algorithm computes the P-value $P(X \geq T_i)$ of each cluster i having at least T_i genes. X is assumed to follow the binomial distribution $X \sim \text{Bin}(|G|, E_i/|G|)$ [16], where $E_i/|G|$ is the probability of a random profile belonging to cluster i according to the null hypothesis.

Algorithm 4.7 returns both the P-values and the expected number of genes in each cluster. These are *Pval* and E in the pseudocode, respectively.

The returned P-values are used in the main program, Algorithm 4.5. The clusters are sorted in ascending order of P-values. A Bonferroni correction ([55], [35, p. 8]) is applied to the P-values in order to compensate for the multiple statistical tests in determining the significant clusters. Thus the significance criterion for cluster i is that its P-value is smaller than the significance threshold divided by the number of clusters, i.e. $P(X \geq T_i) < \alpha/m$.

Algorithm 4.7 computes a clustering for all permutations of the data, including the "straight", original permutation. The clustering of the unaltered data is also computed in Algorithm 4.5. A small proportion of the running time of the clustering algorithm could be shaved off by eliminating the redundant computation.

Profiles starting with zero

As stated in Section 4.3.2, the first time point of each model profile used by the clustering algorithm is always zero. Behind this choice lies an assumption presented in [16]. The assumption is that a "raw" time series $\mathbf{x}$ is converted into a time series $\mathbf{y}$ containing log ratios, where the value at each time point is compared to the value at the first time point:

$$y_i = \log(x_i/x_1) = \log x_i - \log x_1. \tag{4.7}$$

The choice of the logarithm base is arbitrary and will not affect the distances between profiles. This can be seen from Equation (4.8) which gives the relationship between logarithmic values with bases a and b. Since the change of logarithm base only means a constant scaling, the distance measure of Equation (4.3) is not affected. The fact that y_1 is zero seems to go well with the idea of model profiles starting with zero. The clustering algorithm operates on the converted time series.

$$\log_b x = \frac{\log_a x}{\log_a b} \tag{4.8}$$

It can easily be seen from the definition of the correlation coefficient (see Equation (4.4)), that the distance measure of Equation (4.3) is immune to

constant biases: A profile $\mathbf{y}$ behaves exactly like an altered profile $\mathbf{y} + a$, where a is any real number. This means that the subtraction in Equation (4.7) is unnecessary. There is no real reason for model profiles to start with zero, either.

It is not clear from [16] whether the original implementation of the clustering algorithm transforms data profiles so that they start with zero. For the sake of consistency, if such a transformation is carried out, it should be done every time the data are permuted. The implementation used in this thesis, as presented in Algorithm 4.5 and the related helper algorithms, doesn't use any "start with zero" transformations.

The main point here is that profiles starting with zero are just a convention. Adding a constant value to all time points of a profile makes no difference when using the distance measure of Equation (4.3).

4.3.5 Grouping significant profiles

For the sake of completeness, grouping significant profiles is covered here. The idea of grouping similar profiles is motivated by noise. If the gene expression measurements are noisy, it might be wise to consider the possibility that a gene truly belongs to some other cluster than the one it was assigned to. Then the logical step is to split the set of significant profiles into groups of similar profiles. [16]

Algorithm 4.8 follows the description of the grouping procedure in [16]. The algorithm has five parameters. P contains the model profiles. The number of significant profiles is given in n_s. Array T contains the number of data profiles in each cluster. The distance measure used in the algorithm is d, and δ is the sensitivity parameter. The output of the algorithm is Grp, a list of profile groups starting from the group with the most member profiles and ending with the smallest group. The smallest group or groups may contain only one profile.

First, each significant profile is given its own group. Initially only profile i itself belongs to its group. Other profiles are considered as candidates for joining the group, in the order of increasing distance to profile i. Profiles are added to the group one at a time as long as the pairwise distances within the group are at most δ. This is repeated for all profiles ($i = 1, \ldots, n_s$). Basically, each profile is assigned a group of close-by profiles, which must also be close to each other. This happens on lines 6–19 of Algorithm 4.8. The groups formed here can be called temporary groups.

The procedure presented above is a classic greedy algorithm, because the order of adding profiles to a group is only locally optimal. Reaching the global optimum, which in this case could mean assigning the largest possible number of profiles into each group, is not guaranteed. Figure 4.3

Algorithm 4.8 GROUPPROFILES Helper function for Algorithm 4.5

GROUPPROFILES(P, n_s, T, d, δ)

```
1  if n_s = 0
2     then return NIL
3  let D be a n_s × n_s array where D[i, j] = d(P[i], P[j])
4  let G be a n_s × n_s zero array, except that G[i, i] = 1, i = 1 ... n_s
5  ▷ G[i, j] will indicate whether profile j belongs to the group of profile i
6  for i ← 1 to n_s                    ▷ Grow each profile's group
7       do sort D[i, ] in ascending order resulting in permutation perm
8          let I be the array 1 ... n_s permuted with perm
9          ▷ Try adding profiles to the group one candidate at a time
10         for j ← 2 to n_s
11              do let Ix be the indices of the ones in G[i, ]
12                 OK ← TRUE
13                 for k ← 1 to |Ix|
14                      do if D[I[j], Ix[k]] > δ
15                           then OK ← FALSE
16                                break
17                 if OK
18                   then G[i, I[j]] ← 1
19                   else  break
20 GroupNumber ← 1
21 while G has non-zero element(s)
22      do let S be a zero array of length n_s
23         for i ← 1 to n_s    ▷ count number of data profiles in each group
24              do S[i] ← 0
25                 let Ix be the indices of ones in G[i, ]
26                 for j ← 1 to |Ix|
27                      do S[i] ← S[i] + T[Ix[j]]
28         let ix be the index of (one of) the maximum value(s) in S
29         let Ix_grp be the indices of the ones in G[ix, ]
30         Grp[GroupNumber] ← Ix_grp     ▷ Grp must be a dynamic structure
31         G[ix, ] ← 0                   ▷ Assign zeros to a whole row
32         for i ← 1 to |Ix_grp|
33              do G[, Ix_grp[i]] ← 0    ▷ Assign zeros to a whole column
34         GroupNumber ← GroupNumber +1
35 return Grp
```

illustrates this principle with Euclidean distances on a plane. If the closest point is added to the group first, adding the other points becomes impossible. The shaded area is the intersection of the δ-radius neighbourhoods of the black dot and the white dots.

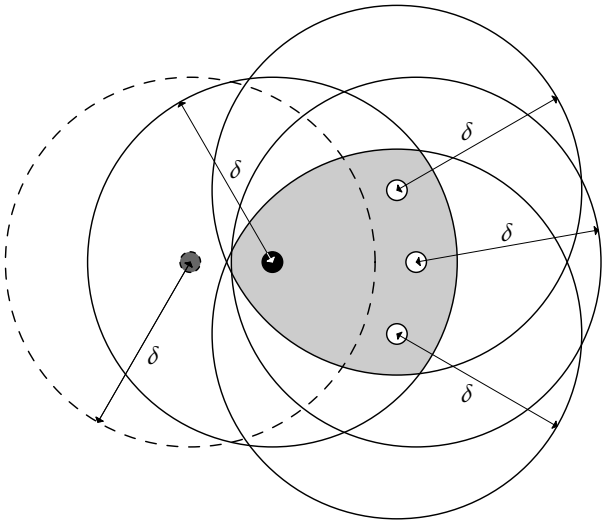


Figure 4.3: When selecting a group for the black point, a greedy algorithm will come to a dead end by choosing the closest (grey) point. δ is the distance threshold.

Figure 4.4 shows a case of five different profiles, and serves as an example of greedy profile selection when using the distance measure of Equation (4.3). The pairwise distances are shown in the matrix

$$\mathbf{D} = \begin{pmatrix} 0 & 0.3937 & 0.3937 & 0.3937 & 0.3937 \\ 0.3937 & 0 & 0.8235 & 1.2647 & 0.8235 \\ 0.3937 & 0.8235 & 0 & 0.4412 & 0.2941 \\ 0.3937 & 1.2647 & 0.4412 & 0 & 0.4412 \\ 0.3937 & 0.8235 & 0.2941 & 0.4412 & 0 \end{pmatrix},$$

where d_{ij} is the distance between profiles i and j. When choosing a group for profile 1, the greedy profile selection method starts by choosing one of the other four profiles as a candidate for membership in the group. Since the distances $d_{1j}, j = 2, \ldots, 5$, are equal, the choice will depend on implementation details. Let's assume that the distance between any pair of profiles in a group can be at most 0.5 ($\delta = 0.5$). If profile 2 is chosen, profiles 3–5 fail to meet the distance criterion, their distances to profile 2 being 0.8235, 1.2647, and 0.8235 respectively. Profiles 1, 3, 4, and 5 would make a nice group with a maximum pairwise distance of 0.4412.

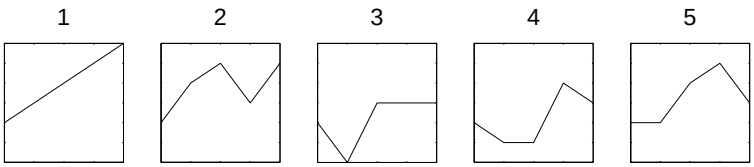


Figure 4.4: When selecting a group for profile 1, a greedy algorithm might choose profile 2 and come to a dead end

After a temporary group has been built for each profile, a final grouping is formed by selecting the largest temporary groups one by one. The largest group is the one that has the most data profiles. Since each significant profile must reside in a single final group, the profiles that belong to a selected group are removed from every other temporary group. A new group is added to the list of final groups until there are no profiles left to be grouped. This happens on lines 21–34 of Algorithm 4.8.

Identifying groups of similar model profiles seems like a good idea, yet there are some problems. As mentioned before, it is not trivial to assign a meaningful value to δ. In the biological results of [16], a value of $\delta = 0.3$ is used. The value stems from the similarity between replicate measurements on the same biological sample. If the data contain very few such replicates, as is the case with the data used in this thesis, the variability of expression profiles due to non-biological reasons is difficult to estimate.

The grouping of model profiles also complicates the analysis of the clustering results, or at least raises some questions. Should the results of the grouping be integrated into the clustering of the data profiles by combining the clusters that belong to the same group, or would some kind of separate treatment of the grouping results be more appropriate? What is the essence of Algorithm 4.8, and what kind of compromises does it make, being a greedy algorithm?

For the reasons stated above, significant model profiles are not grouped in the cluster analyses done in this thesis. Although the similarities between model profiles are not assessed in this way, the results will probably not be significantly affected.

4.4 Other approaches to clustering

Some other clustering methods applicable to gene expression analysis are discussed here in short. A nice review of various different methods is available in [37, pp. 24–36].

Self-organising map (SOM) [28] is an artificial neural network algorithm which uses a low-dimensional topologically connected lattice to represent high-dimensional data. The cluster structure of the data can be visualised with the U-matrix [52], which shows how far from each other the map units have stretched in the input data space.

Out of all clustering methods, hierarchical clustering [50, pp. 449–487] may have been [37, p. 30] the first that was applied [15] to gene expression data. There are two types of hierarchical clustering algorithms: agglomerative and divisive. Both produce a hierarchy of clusterings instead of a single clustering.

K-means is a very basic clustering method, where each data vector is assigned to the closest prototype vector. Each prototype vector is the mean of the data vectors belonging to its own cluster. Euclidean distance is often the distance measure of choice. [50, pp. 531–533]

There are also other clustering methods. One of them uses a shape-based similarity measure [8]. Another method is based on Hidden Markov Models (HMMs) [47]. Both of these are designed for gene expression time series data.

4.5 Analysis of clustering results

4.5.1 Annotation data

Annotation data (file date 21st Jun 2005) were downloaded from the Affymetrix web site, http://www.affymetrix.com/. The data file is in a comma separated values (CSV) format, where each row is a record representing one probe set. The fields of a record are enclosed in double quotes and separated by a comma. This is a very nice storage format, because it is relatively easy to extract any given field with a text oriented tool such as *awk* [6].

The annotation file contains information assembled from various public databases. The following fields were used in the experiments made for this thesis: “Gene Title”, “Chromosomal Location”, and “Gene Ontology Biological Process”. The absence of a “Gene Title” was used as a pruning criterion. This is explained in Section 5.2.2.

Chromosomal location

The “Chromosomal Location” of a probe set is the location of the corresponding nucleotide sequence or sequences in the human genome. Some possible locations are 10p12.33 — chromosome 10, band p12 in the p (petit) arm, sub-band p12.3, sub-sub-band p12.33 — and Xq26-q28 — X chromosome, bands q26 to q28 in the q (queue) arm [49, p. 49]. The amount of dif-

ferent locations was reduced by removing any sub-band information from the data in the CSV file. If the chromosomal location of a probe set could not be associated with a single band, the location was marked as missing. For example, 10p12.33 turned out as 10p12, and Xq26-q28 was marked as missing data. Also probe sets with multiple entries for chromosomal location were stripped of location information.

The aforementioned simplification procedure was applied in order to reduce the number of different locations. Without the simplification, there would be many locations with only one or a few probe sets. The reduction in the number of location categories and particularly the resulting increase in the number of probe sets in each category are desirable features when looking for enriched locations.

The simplification of chromosomal locations can also be motivated by information like 3q22.2 being too specific when looking for *areas* of the genome where something special seems to be happening. This observation stems from the fact that when looking for enriched terms, the relationship of the terms, such as proximity of locations in the genome, is not taken into account in any way. The enrichment of a term, such as a certain location, is assessed independently from other terms. Section 4.5.2 contains more information about enriched terms. When using the method described there for the analysis of enriched chromosomal locations, the scale of observation is adjusted by manipulating the detail level of the location data accordingly, as described above.

From a technical point of view, the above-mentioned simplification of chromosomal locations is a convenient way to get rid of excess details and any ambiguities in the data. However, some information obviously gets lost in the process. In the case of multiple locations (e.g. 3q12 and 5q33.1) per probe set, it seems acceptable to discard the information, as the interpretation of this kind of ambiguities is well beyond the scope of this thesis. A range like Xq26-q28, however, could be converted to the set of band-level locations in the range: Xq26, Xq27, and Xq28. Doing that would save some information from being thrown away.

Integrating the chromosomal location of genes into the analysis of clustering results can be motivated with earlier findings about the similarities in the expression of genes with nearby locations. For example, the density of genes expressed in two different types of tissues was studied in [9]. The study revealed some regions of the genome where the density in one of the tissue types was much greater than in the other. Before any major conclusions about the co-expression of close-by genes are made, it is worth remembering that local microarray artefacts such as scratches can interfere with the analysis of regularities in gene expression that are related to chromosomal location. This depends on the design of the microarrays in question [26].

It seems safe to say, that a microarray design with probes ordered according to chromosomal location is especially troublesome in this respect, and generally a bad idea.

Biological process

The "Gene Ontology Biological Process" of a probe set describes the biological process or objective the gene represented by the probe set is involved in. Examples include "inflammatory response" and "cholesterol biosynthesis". Like chromosomal location, biological process is used in the enrichment analyses done in this thesis, described in Section 4.5.2. The two other Gene Ontology (GO) categories besides "biological process" are "molecular function" and "cellular component". [7]

The biological processes were taken "as is", and not interpreted in any way. Some of the terms were quite general, while others were more detailed. As an example, consider the elaborate "branched chain family amino acid metabolism" and the plain "metabolism". Even though the former is clearly an instance of the latter, the relationship or hierarchy between different terms is not considered here.

4.5.2 Enriched terms

Clustering methods such as the one described in Section 4.3 produce a set of clusters, but how can the results be interpreted? Looking at a picture of the expression profiles in a cluster may provide some interesting information, but will leave many questions unanswered. Another way to approach the problem is to integrate some additional data with the clustering results. Annotation data such as chromosomal location and biological process, both described in Section 4.5.1, can be of great help in discovering some of the characteristic features of each cluster.

One way to describe a cluster is to find out the annotation terms that are significantly more common in the cluster than in a reference set. Let's consider a single annotation term, for example the biological process "phospholipid biosynthesis". Let n be the number of probe sets the term is associated with, and let $n + m$ be the total number of probe sets in the reference set. Now think of a randomly selected cluster containing N probe sets, the cluster being a subset of the reference set. Let x be the number of probe sets in the cluster that are associated with the annotation term. Then the probability of exactly i occurrences of the term in the cluster can be computed from the hypergeometric distribution. See Equation (4.9) for a definition of the distribution (adapted from [56]).

$$
\begin{aligned}
P(x = i) &= \frac{[\#\text{ ways for } i \text{ successes}]\,[\#\text{ ways for } N - i \text{ failures}]}{\text{total number of ways to select}} \\
&= \frac{\binom{n}{i}\binom{m}{N-i}}{\binom{m+n}{N}} \\
&= \frac{m!n!N!(m+n-N)!}{i!(n-i)!(m+i-N)!(N-i)!(m+n)!} \ .
\end{aligned}
\tag{4.9}
$$

Computing probabilities from the hypergeometric distribution, if naively implemented straight from Equation (4.9), is not feasible with the kind of numbers usually present in practical problems. The maximum exponent in the standard double precision floating point number system is 1023 (base 2) [19]. Thus the maximum number representable with the system is about $10^{1023 \cdot \log_{10} 2} \approx 10^{308}$. The factorial of 170 is the largest factorial smaller than 10^{308}. Anything bigger ($n!$, $n > 170$) goes to infinity in the floating point system.

Since the trivial implementation is practically useless, some fairly clever tricks must be employed in computing the hypergeometric distribution. There are various statistical software packages around that excel at this type of tasks, so there is little point in writing your own routine. If one decides to dig deeper in the anatomy of the hypergeometric distribution, the link between the factorial and gamma functions (see Equation (4.10)), might be a good starting point, as well as the documentation or source code (when available) of an applicable computer program.

$$\Gamma(n) = (n-1)!,\ n \in \mathbb{Z}_+ \tag{4.10}$$

Seeing that the hypergeometric distribution is discrete, the probability of seeing at least j occurrences of an annotation term in a random cluster is simply

$$P(x \geq j) = 1 - \sum_{i=0}^{j-1} P(x = i) \ , \tag{4.11}$$

where $P(x = i)$ is the probability of i occurrences as in Equation (4.9).

If the P-value $P(x \geq j)$ is small enough, then the annotation term is said to be enriched in the cluster. Usually the significance limit α of the P-value is 0.05 or 0.01. In more formal terms, let H_0 and H_1 be the null and alternative hypotheses, respectively:

H_0 : the probe sets of the cluster are randomly selected, and

H_1 : some probe sets are more likely than others.

When H_0 holds, the hypergeometric distribution is used in assessing the probability of seeing a number of terms in a cluster, as described above. Now if $P(x \geq j|H_0) < \alpha$, the null hypothesis is rejected. The annotation term that caused the rejection can be called an enriched term.

The above is fine if there is just one term to test. The probability of rejecting H_0 when it in fact is true is α. Now think of multiple terms to test for enrichment. As a direct generalisation of the above, H_0 will be rejected if the P-value of any term is smaller than α. Clearly the risk of falsely rejecting H_0 will rise. The expected number of spurious positives, i.e. terms falsely declared as enriched, will also rise as the number of comparisons grows.

One way to adjust for multiple comparisons is the Bonferroni correction [55][35, p. 8]. That means reducing the significance limit of each test to α/n, where n is the number of comparisons, for example the number of annotation terms to be tested. The Bonferroni correction is very conservative, and mostly suitable for small values of n. The probability of missing something truly significant increases as n grows. One opinion about the matter is that Bonferroni correction "creates more problems than it solves" [40].

Another option with multiple comparisons is to ignore any adjustments of α such as the Bonferroni correction. Just selecting a reasonably low α and acknowledging the possibility of spurious positives — while preserving the ability to notice truly significant features of the data — might well be an adequate solution.

Using the hypergeometric distribution as a tool for the evaluation of enriched terms is a common practice in gene expression studies. See for example [32] or [48].

4.5.3 Order of genes within a cluster

Algorithm 4.5 produces a clustering, where each data profile (gene) is assigned to a cluster, or theoretically to several clusters. When looking at a cluster, it might be useful to know how well each gene in the cluster matches with the model profile of the cluster, or which genes have a "strong" — this is defined later — expression profile that is similar to the model profile.

There is not much point in using the distance measure of Equation (4.3) at this point. Ordering the genes in a cluster in descending order of distance to the corresponding model profile would be like a kind of beauty contest. In a beauty contest final, all the contestants have already passed a series of qualifying competitions. Since all the competitors are beautiful, the selection of a single beauty queen might be considered questionable or unnecessary. In this case, all data profiles in a cluster are already quite close to the model profile with regard to Equation (4.3), closer than to any other model profile. Some members of a beauty contest jury might appreciate

tallness as a feature. In the gene expression world, there is a good reason to look for "tall" or "strong" expression profiles. A profile with large positive or negative values suggests the presence of real biological variation. Small deviations from zero can more easily be attributed to random noise.

The dot product of a data profile $\mathbf{x}$ and a model profile $\mathbf{y}$,

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} x_i y_i \tag{4.12}$$

can be used as a sorting criterion. A large value basically indicates that the data profile has strong deviations from zero to the same direction as the model profile. The genes in a cluster can be sorted in descending order of dot product. The dot products are computed between each data profile and the model profile of the cluster. This procedure leaves the most interesting, most strongly expressed genes of the cluster at the top of the list, and the genes with the relatively flat profiles near the bottom.

Figure 4.5 shows an example of model profiles ordered with the dot product method. The strongest profiles are easily distinguishable from the weakest ones. The example is from a cluster of 1108 data profiles. The strongest profile has a dot product of 11.0670 with the model profile, whereas the number is 1.8021 for the weakest profile.

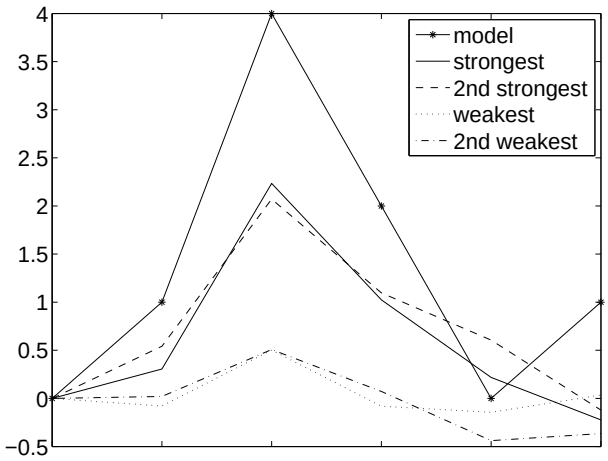


Figure 4.5: Ordering genes in a cluster according to dot product

Remember that the scale of profiles doesn't matter when using the correlation based distance measure of Equation (4.3): a profile $\mathbf{x}$ behaves in the same way as $a\mathbf{x}$, the same profile multiplied by an arbitrary positive number a. The sorting method based on dot product is a good and simple way to make use of the scale.

4.5.4 Known asbestos-related genes

A table containing information on a number of genes was received from one of our collaborators. The genes were collected from publications that studied the effect of asbestos on gene expression in connection with asbestos-induced lung cancer or mesothelioma. The corresponding probe set identifiers were also listed in the table.

The list of known asbestos-related genes can be used in the analysis of clustering results. For example, it would be interesting if such genes were mostly assigned to significant clusters, and if the genes also ranked high in the internal order of their respective clusters, as described in Section 4.5.3.

Chapter 5

Experiments and results

5.1 Synthetic data

When writing a piece of data analysis software, correctness is the most important feature requested from the program. Performance and clarity are some other desirable qualities of good code, but correctness comes first.

Some experiments presented in [16] were repeated here for the purpose of testing our implementation of Ernst's algorithm. The data used in the experiments were the same two synthetic data sets as used in [16]. Both data sets had 5000 genes with 5 time points each. In the first set, the expression value at each time point of every gene was drawn from a uniform $(10, 100)$ distribution. The other data set had 4850 genes generated in the same way. In addition to the random genes, there were 150 genes planted from the three profiles in Figure 5.1, 50 genes from each. The planted genes also had some added noise.

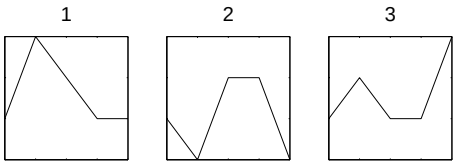
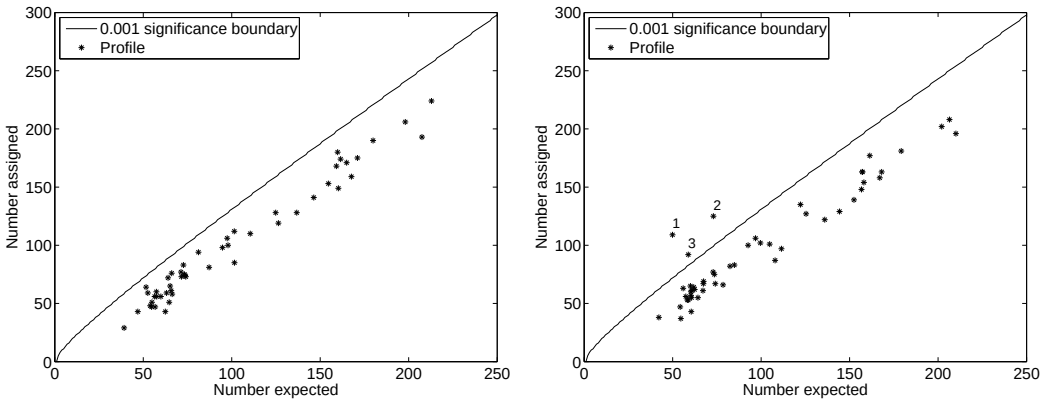


Figure 5.1: The three profiles planted in the second synthetic data set

When testing with the same model profiles as in [16, Fig. 4a], the results shown in [16, Fig. 3] were easily repeated using our implementation of the clustering algorithm. Figure 5.2(a) shows that the clustering algorithm found no significant profiles in the completely random data set. Figure 5.2(b) shows three marks above the significance line. These marks correspond with the three profiles in Figure 5.1 from which genes were planted

to the second simulated data set. In other words, the algorithm and our implementation work as expected. Figure 5.2 also shows a point emphasised in [16]: some expression profiles are more likely than others to appear by chance. Remember that the expected number of genes is computed with a permutation test (see Section 4.3.4), which has the effect of randomly ordered time points. By taking this into account, Ernst's algorithm is able to detect significant clusters, be they small or large. If the expected size of a cluster was constant, in this case $5000/50 = 100$, cluster 3 with less than 100 probe sets would certainly not be noticed as significant.



(a) Random data

(b) Random data and planted data from three profiles

Figure 5.2: Expected number of genes vs. number of genes assigned, simulated data of [16]

In [16], the clustering algorithm was also tested with some biological data. That experiment was not repeated with our implementation of the algorithm, because the details of the gene pruning criteria presented in [16] and the supplementary web site http://www.cs.cmu.edu/~jernst/st/ (referenced 2nd Dec 2005) were not quite clear enough. The computation of correlation with missing time points was not covered at all, although correlation was used as a criterion in the pruning procedure, and the data contained missing time points. The pruning procedure of [16] left 2243 out of 24192 genes for the cluster analysis. Since this result could not be duplicated here, the cluster analysis wasn't done either.

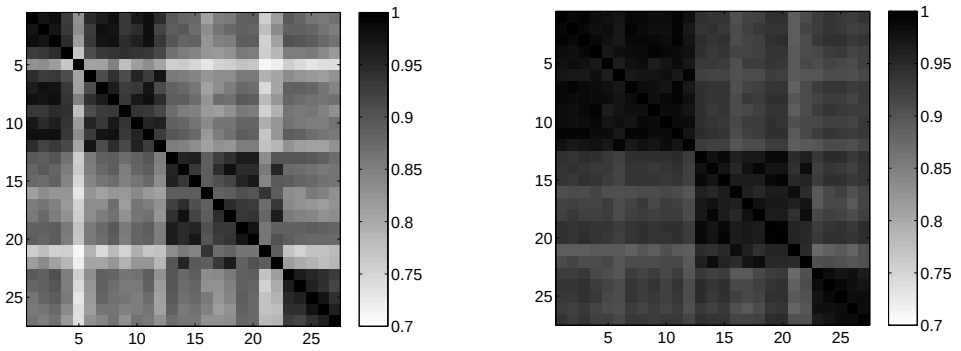
Algorithm 4.8 was not tested against reference results due to the problems described above.

5.2 Data from asbestos exposure experiment

After our implementation of Ernst's clustering method had been tested and found to be reliable, it was applied to the biological data described in Section 2.2. However, some preprocessing was needed before clustering.

5.2.1 RMA preprocessing

The raw gene expression values were RMA preprocessed (see Section 3.2.2) using version 1.6.7 of the *affy* software package. Data from all 27 microarrays was preprocessed together. The arrays measuring the Met5A cell line could have been preprocessed separately, since the Met5A expression values were not directly compared with numbers from the other cell lines (see Section 5.2.3). Figure 5.3 compares the pairwise correlations between microarrays before and after preprocessing. The numbering scheme of the arrays is shown in Table 2.2.



(a) Raw PM/MM values (probes) (b) RMA expression values (probe sets)

Figure 5.3: Effect of RMA on pairwise correlations between arrays

Figure 5.3(a) shows the correlations before RMA preprocessing. Looking at the figure, the squares formed by the three different cell lines — A549 (arrays 1–12), BEAS-2B (arrays 13–22), and Met5A (arrays 23–27) — are quite easily distinguishable. However, a distinctive feature in the figure is the exceptionally low correlation between array number 5 and the other arrays measuring the gene expression of the A549 cell line.

Figure 5.3(b) contains the pairwise correlations between the arrays, when the correlations are computed with the RMA expression values. The overall level of correlation is higher than in Figure 5.3(a), but the three squares of high correlation formed by the individual cell lines are still distinguishable.

The most notable thing is that array number 5 doesn't stand out anymore, but is highly correlated with the other A549 arrays.

5.2.2 Further preprocessing

A few additional preprocessing measures were also taken before subjecting the data to clustering. These are very simple and straightforward steps that are more naturally described here as part of the experimental setting than together with the more sophisticated procedures described in Chapter 3. Whereas RMA preprocessing deal with probe level data, the following measures operate on RMA preprocessed, probe set level data.

Averaging of replicate experiments

The data available for analyses were described in Table 2.1. For the most part there is only one microarray measurement per biological condition, but a few conditions were tested twice. It is quite clear that there isn't much to be gained from the few replicates, but it isn't wise to throw data away, either.

In those time points and biological conditions where two arrays were used, the RMA values of the arrays were simply averaged. That is, $x_i = (x_{i1} + x_{i2})/2$, where i is the index of the probe set, x_i is the new averaged value, and $\{x_{i1}, x_{i2}\}$ are the individual RMA values. Remember that RMA values are on a logarithmic scale. The following equation shows that the average of logarithmic values corresponds to the geometric mean of the values on a linear scale, or more specifically, to its logarithm:

$$\log \underbrace{\left(\prod_{i=1}^{n} a_i\right)^{1/n}}_{\text{geometric mean of } a_i} = \frac{1}{n}\log\prod_{i=1}^{n} a_i = \underbrace{\frac{1}{n}\sum_{i=1}^{n}\log a_i}_{\text{arithmetic mean of } \log a_i} . \qquad (5.1)$$

Even though the replicate arrays don't provide much protection for measurement noise, the goal of bringing out biologically relevant features of the data is still within reach. When looking at the data as a whole with the help of clustering, uncertainties regarding the exact expression levels of single genes are not likely to affect the overall results too much.

Creation of log-ratio time series

As the subject of the thesis suggests, we are interested in the changes in gene expression caused by asbestos. That being the case, it seems reasonable to integrate the two cases, "asbestos exposure" and "no asbestos exposure", so that a single number reflects the difference between them.

Consider $\mathbf{y} = \mathbf{x}_{\text{asb}} - \mathbf{x}_{\text{ctrl}}$ (see Figure 5.4), where $\mathbf{x}_{\text{asb}}$ and $\mathbf{x}_{\text{ctrl}}$ are time-series of RMA expression values of one probe set in the "asbestos" and "control" (no asbestos) cases, respectively. The time-series must naturally have the same time points. Since $\log(a/b) = \log a - \log b$, and RMA values are logarithmic, $\mathbf{y}$ can be called a log-ratio time series. Each element of $\mathbf{y}$, y_i, is the logarithmic ratio of the "asbestos" and "control" expression values in the i:th time point.

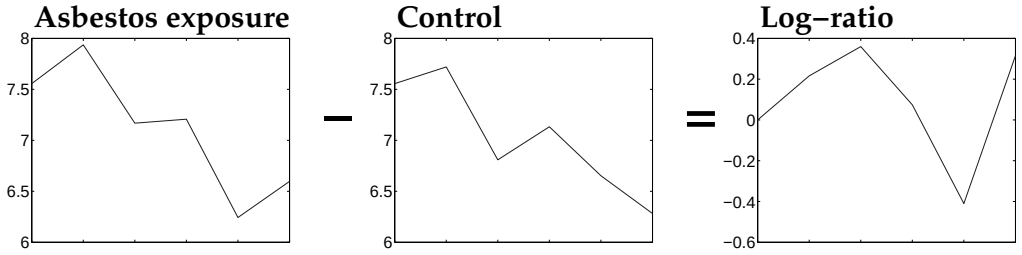


Figure 5.4: Creation of log-ratio time series

The length of the log-ratio time series varies between cell lines according to Table 2.1. There are 6, 5, and 3 time points in the expression measurements of the A549, BEAS-2B, and Met5A cell lines, respectively. These numbers include an implicit zero log-ratio added to the beginning of every time series. The zero stems from the assumption that the "asbestos" and "control" cases start out with equal expression levels at time "zero". After that, the ratios of expression in the two cases may differ.

Working with log-ratio time series is intuitive. A positive value means that a probe set is overexpressed in the "asbestos" case compared to the "control" case. A negative value means the opposite. A (theoretical) zero indicates that there is no difference between the cases. Since RMA values are on a $\log_2$ scale, the values of the log-ratio time series can easily be converted back to meaningful, linear scale ratios. For example, a log-ratio of 3 means that the "asbestos" expression is $2^3 = 8$ times as strong as the "control" expression. The absolute values of these ratios should not be taken too literally, however, due to the noise inherent in gene expression data, and the bias possibly introduced by the preprocessing method.

Pruning of probe sets

When dealing with large data sets, there is always the danger that the interesting phenomena get lost in the less interesting data or noise. In order to increase the probability of the clustering algorithm bringing out the biologically meaningful data, some probe sets were pruned out. All probe sets

fulfilling any of the following four criteria were excluded from the clustering.

Control probe set Among other data fields, the CSV file mentioned in Section 4.5.1 contained a field called "Sequence Type". Probe sets of type "Control sequence" were pruned out. These are used for quality control [2, pp. 38–39], but are not useful in assessing the expression level of genes.

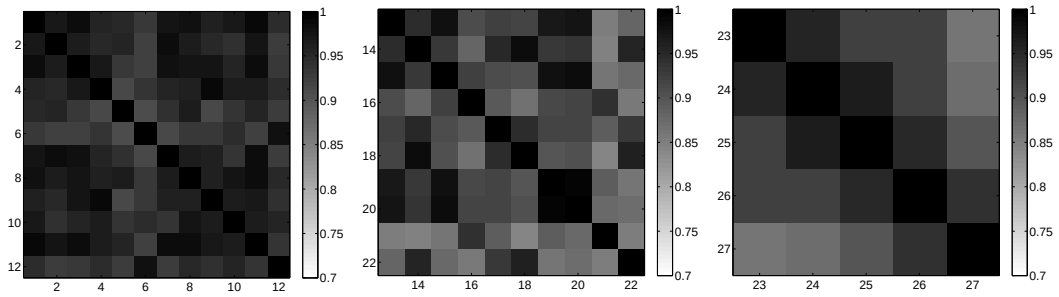
Always absent A probe set was declared "absent" in every microarray related to the clustering experiment in question. "Absent", "present", and "marginal" are the three detection calls [2, p. 44] in the MAS 5.0 preprocessing method. The calls were computed with the implementation of the MAS method provided by the *affy* [17] package in R [42].

Flat profile Absolute value of the log-ratio time series of a probe set was smaller than 0.5 in every time point. The limit corresponds to the "asbestos" linear scale expression value being $2^{0.5} = \sqrt{2} \approx 1.41$ times as large as the "control" expression, or vice versa. If the change between the two cases was always smaller, the probe set was pruned out.

Missing "Gene Title" Probe sets without a "Gene Title" in the CSV file (see Section 4.5.1) were pruned out. Generally speaking, these probably represent genes that have not been studied much. It might have been nice to include these probe sets in the clustering, but sticking with the more familiar genes still left plenty of data to be analysed.

Applying the aforementioned pruning criteria resulted in a considerable reduction in the number of probe sets. Out of the 54675 probe sets present before pruning, 12436, 16440, and 7538 probe sets survived the pruning procedure in the A549, BEAS-2B, and Met5A cell lines, respectively. When the A549 and BEAS-2B cell lines were combined by removing the last time point from the expression profiles of the A549 probe sets, the pruning procedure spared 27348 probe sets out of a total of $2 \cdot 54675 = 109350$.

Figure 5.5 shows the correlation of RMA expression values between the microarrays measuring the same cell line, for all three cell lines. The difference between these results and the corresponding squares clearly visible in Figure 5.3(b) is that these correlations have been computed after pruning. The numbering of arrays in Figure 5.5 corresponds to that in Table 2.2. The reduction in the average correlation between arrays is quite noticeable in the BEAS-2B and Met5A cell lines, where it dropped from 0.9674 to 0.9145 and from 0.9794 to 0.9209, respectively. The average correlation of the A549 arrays showed a more modest reduction, from 0.9814 to 0.9537, caused by pruning.



(a) A549 (b) BEAS-2B (c) Met5A

Figure 5.5: Correlation of RMA values between microarrays after pruning

Correlation coefficient is a simple measure of the linear dependency of two variables. Clearly, the nature of the relationship of two microarrays is rather poorly represented by a single number. Still, the reduction in correlation was expected, because the flat profiles that were pruned out contributed to a high correlation between the "asbestos" and "control" arrays of the same time point. It is quite interesting that the decrease in correlation of the A549 arrays was small compared to the other cell lines. Are the changes in gene expression between different conditions generally smaller in the A549 cell line than in the other two cell lines? Another notable thing, shown in Figure 5.5(b), is that arrays 13, 15, 19, and 20 form a group with very high correlations. Looking at Table 2.2, this doesn't come as a big surprise, since three of those four arrays represent the same time point, 24 hours.

5.2.3 Clustering

The log-ratio time series of the probe sets remaining after pruning were clustered using Ernst's algorithm, which was described in Section 4.3. A total of four different data sets were clustered:

- A549 cell line, 12436 probe sets $\times$ 6 time points
- BEAS-2B cell line, 16440 probe sets $\times$ 5 time points
- Met5A cell line, 7538 probe sets $\times$ 3 time points
- A549 and BEAS-2B combined, 27348 probe sets $\times$ 5 time points

A549 cell line

Figure A.3 shows the model profiles of the A549 clustering. The profiles are in the order[1] of decreasing significance (increasing P-value). The 12 significant profiles are shaded with grey. The number of profiles, m, was chosen to be 50, which is the same number as in the experiments in [16]. It is quite an arbitrary choice, but we think it offers a decent selection of distinct profiles. Parameter c, the maximum amount of change between the time points of a model profile, was chosen to be 3. The model profiles were acquired with the randomised version of profile selection, Algorithm 4.3.

Figure A.4 shows the data profiles associated with each model profile of the A549 clustering. The figure is quite messy and crammed with information, but it gives a general idea about the range of variation or the distribution of data profiles in each cluster.

BEAS-2B cell line

Figures A.5 and A.6 show the model profiles and the corresponding clusters of the BEAS-2B clustering. Again, the model profiles were selected with the randomised algorithm from the set of profiles bound by the rule $c = 3$. Now 16 profiles out of 50 were found to be significant.

Met5A cell line

The model profiles and clusters of the Met5A cell line are in figures A.7 and A.8. Due to the shortness of the time series, only 16 model profiles were selected. That was a rough estimate of the number of profiles needed to sufficiently represent all the different expression profiles with three time points. Three profiles were found to be significant.

Looking at Figure A.8, some of the clusters look quite similar. Figure 5.6 shows the effect of the number of selected model profiles on the minimum distance between the profiles with Algorithm 4.3, when $c = 6$ and $n = 3$. The minimum distance drops significantly when the ninth profile is selected. Comparing Figure 5.6 and Table 4.1 reveals that the $n = 3$, $m = 8$ minimum distance (0.2794) is close to the $n = 6$, $m = 50$, $c \in \{2, 3\}$ minimum distances, whereas the $n = 3$, $m = 16$ minimum distance is much worse. Picking the ninth model profile already decreases the minimum distance noticeably. Thus, $m = 8$ might be a better choice than $m = 16$, the number used in the experiment.

The model profiles of the Met5A clustering were chosen with the randomised algorithm, but parameter c of Algorithm 4.1 was increased to 6.

[1] See Section "Numerical precision" for remarks on computational limitations

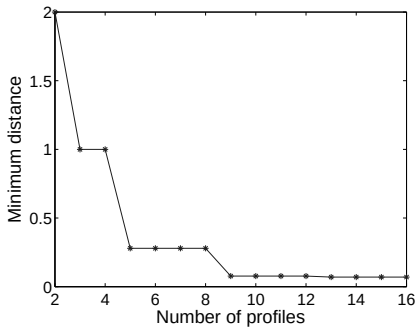


Figure 5.6: Number of profiles selected vs. minimum distance between profiles ($c = 6$, $n = 3$)

With $c = 6$ and $n = 3$, the total number of different model profiles according to Equation (4.1) is 169. Only 96 remain after the removal of redundant profiles. Compare these numbers to 16807 and 16322 profiles to choose from with $c = 3$ and $n = 6$, and you will understand why $c = 6$, $n = 3$ are easy on Algorithm 4.3, as seen in Table 4.2. Of course, the number of profiles to select, m, also affects the running time. Although the running time of Algorithm 4.3 would easily allow it, the benefits of further increasing c would probably be very small.

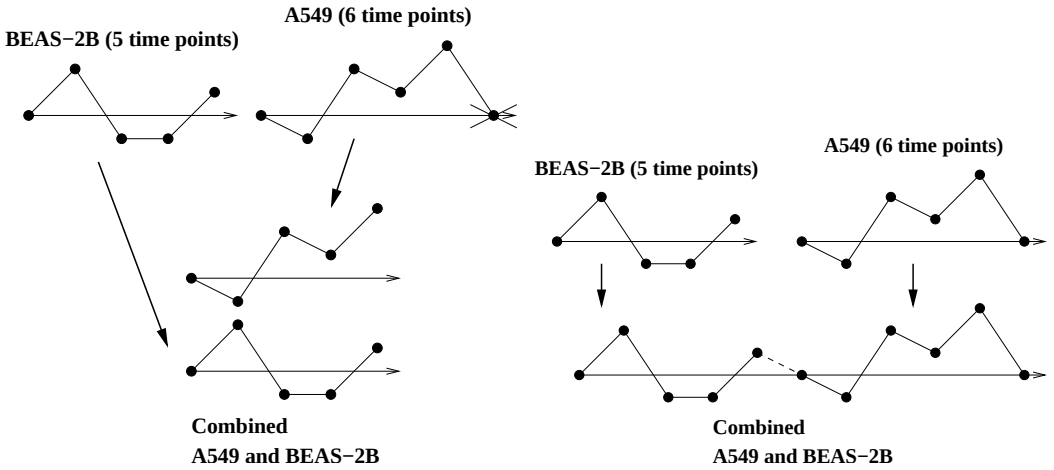
Combined A549 and BEAS-2B cell lines

Figures A.9 and A.10 show the model profiles and the corresponding clusters of the combined A549 and BEAS-2B clustering. The model profiles used here were the same 50 profiles with 5 time points each as in the BEAS-2B clustering. 13 significant profiles were found.

Originally there were two ideas for the implementation of the joint clustering. The first option, which was eventually chosen, is shown in Figure 5.7(a). The discarded option is shown in Figure 5.7(b). The options differ in the way that the data sets would be combined.

The first option included unifying the length of the time series, and keeping the time series from the two cell lines separate. This procedure roughly doubles the total number of profiles to be clustered. The A549 and BEAS-2B time series are close to each other in length: 6 and 5 time points, respectively. The time points are also the same, except for the last time point of the A549 time series (7 days), which is missing from the BEAS-2B data set. Although throwing away data is generally not recommended, dropping the last time point of the A549 data set was considered acceptable.

The second option was to join the time series together: for each probe set,



(a) Separate time series, unified length (b) Time series joined together

Figure 5.7: Two options for joint clustering of A549 and BEAS-2B

there would be one longer time series consisting of the A549 and BEAS-2B expression profiles, as shown in Figure 5.7(b). This procedure was quickly ruled out as infeasible. The clustering algorithm, or our implementation, simply cannot handle profiles as big as 11 time points. Theoretically speaking, the algorithm *could* do it, but it would take a long time on current computer hardware. See Section "Running time" for more information on the time requirements. The joint of the two original expression profiles in the middle of the longer profile would also be a potential problem. The selection of model profiles would also be painfully slow, and the model profiles would probably need to be altered to include another forced zero value.

Running time

The CPU time required by our MATLAB implementation of Algorithm 4.5 in various cases is shown in Table 5.1. The running time of the clustering algorithm depends on the particular implementation used. It also depends on the underlying hardware and software. These times were measured on a computer with 2200 MHz Opteron processors, running the Linux version of MATLAB.

The running times range from roughly one minute to over 11 hours, depending on the job. The permutation test is very computationally demanding, and quickly increases the time required by the clustering algorithm as the number of time points n is increased. When the times were measured, the clustering algorithm was set to do the grouping procedure of Algorithm 4.8 with an arbitrary value for the grouping sensitivity δ. The

Table 5.1: CPU time used in clustering (in minutes)

Data set / Cell line	Parameters			Time
	m	n	$\|G\|$	(min)
Ernst artificial data [16]	50	5	5000	46
A549	50	6	12436	689
Beas-2B	50	5	16440	155
Met5A	16	3	7538	1.3
A549 and Beas-2B	50	5	27348	252

time[2] required by the grouping algorithm was negligible, less than one second.

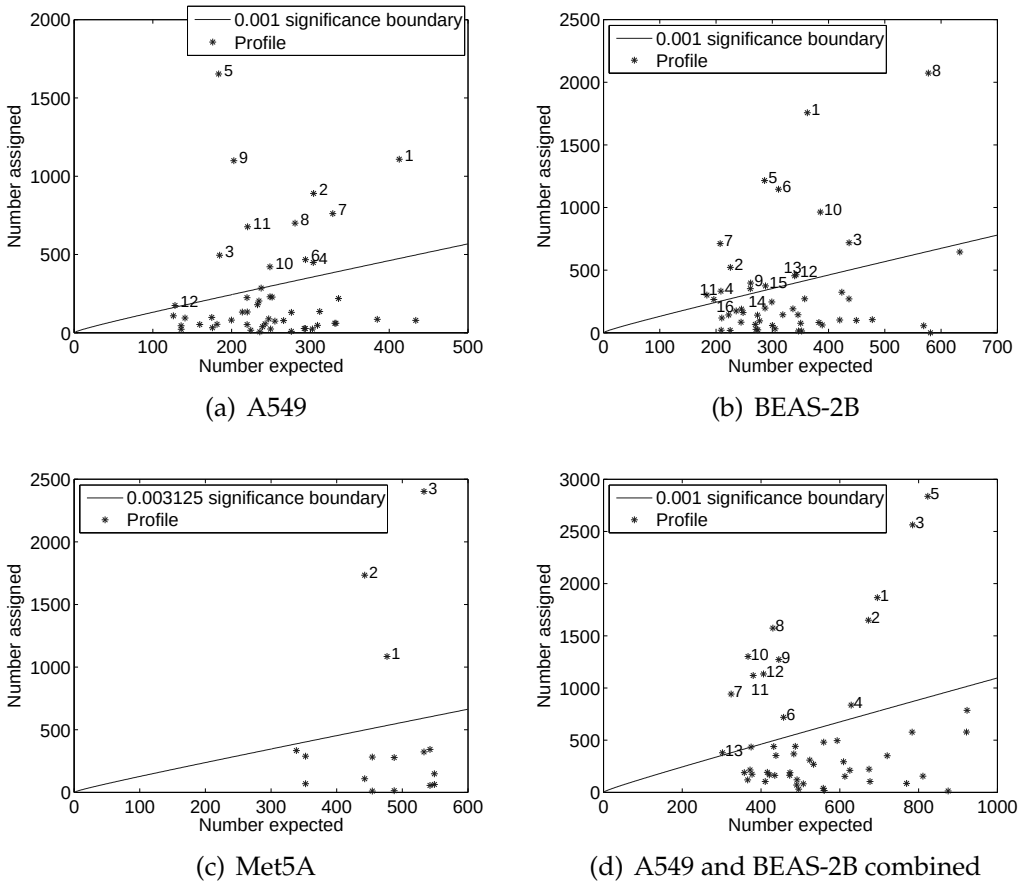
Most of the total times listed in Table 5.1 were spent by calls to Algorithm 4.6. Although the table contains only a few data points, the running time of the clustering algorithm can quite safely be declared roughly linear in both the number of genes $|G|$ and the number of calls to Algorithm 4.6 ($n! + 1$), when other parameters remain the same. Imagine that the A549 data set is expanded to include 7 time points in each expression profile instead of the present 6. By extrapolating from the numbers in Table 5.1, it can be estimated that clustering the expanded data set would take about 80 hours of CPU time. Now the part "short time series" in the title of [16] makes perfect sense.

Numerical precision

Figure 5.8 shows the expected and realised size of clusters in each of the aforementioned clustering experiments done with biological data. The Bonferroni corrected significance thresholds are also included, and the significant clusters have numerical labels. As mentioned above, the clusters are sorted in the order of decreasing significance. However, this is only theory. In practice, the precise order tends to get lost due to the limits of numerical precision.

As an example of the lost differences in significance, see profiles (clusters) 1 and 5 in Figure 5.8(a). Cluster 5 has more genes than cluster 1, even though the expected gene counts suggest otherwise. It is clear that cluster 5 is actually "more significant" than cluster 1. Obviously our implementation of Algorithm 4.7 has met some numerical limitations in the MATLAB implementation of the binomial distribution. For the record, the P-values of clusters 1–11 in the A549 clustering — with our implementation — range from $9.8948 \cdot 10^{-12}$ to $1.3200 \cdot 10^{-11}$.

[2]Measured with separate runs of the grouping algorithm



(a) A549 (b) BEAS-2B

(c) Met5A (d) A549 and BEAS-2B combined

Figure 5.8: Expected number of genes vs. number of genes assigned in each cluster, data from the asbestos experiment

In case the order of significant clusters matters, it is useful to check the "expected vs. realised" numbers like in Figure 5.8. If the significant clusters are only seen as a group with no particular order, the problems with numerical precision don't really matter. Thus there seems to be no compelling reason to further investigate the issue. One plausible explanation for the phenomenon would be the variable accumulation of error when summing up discrete probabilities represented with imprecise floating point numbers.

No grouping of profiles

The grouping of significant model profiles, described in Algorithm 4.8 was not really done with the biological data, either. The main reasons for omitting this phase were described at the end of Section 4.3.5.

Although it was never put to real use, our MATLAB implementation of

Algorithm 4.8 was tested with the biological data using various arbitrary values for parameter δ. The algorithm generally produced groups that contained "reasonably" similar profiles. The algorithm also passed a basic sanity check, namely that the output of an algorithm should change accordingly when adjusting the input. In this case, the average size of profile groups grew with increasing δ, just as was expected.

5.2.4 Analysis of clusters

The clustering results were analysed using the annotation data and the methods described in Section 4.5. The results from the various clustering experiments were also compared to each other. An outline of the results obtained during the analyses is presented in the following sections. As explained before and shown in Figure 1.1, the results are not conclusive. They will be and have been subjected to further analysis by collaborating biomedical experts. The enrichment results may be somewhat affected by the fact that some genes are measured with more than one probe set.

Biological process

The 54675 probe sets on the microarray were found to be annotated with 1921 different biological processes. The file containing the annotations was described in Section 4.5.1. For each of the four clusterings described in Section 5.2.3, the enriched biological processes were recorded. That means every biological process with a P-value smaller than 0.01 in at least one significant cluster. See Section 4.5.2 for a discussion of enriched terms.

Table 5.2 is a summary of the results. It shows the number of terms enriched in each of the four individual clusterings. The table also shows some comparative results. For example, the number of biological processes enriched in at least one cluster of the A549 clustering is 240. The number is quite big compared to the total of 1921 different terms. Fortunately, a less intimidating set of terms can be obtained by applying some set operations to the results. For example, there are 26 biological processes that are enriched in each of the four clusterings, and 28 terms enriched in each of the three clusterings corresponding to the individual cell lines. These 28 terms are listed in Table A.1.

Collecting the enriched terms from all clusters and comparing the results of the individual clusterings is only one of several different ways of trying to reduce the vast amounts of data into something that is both biologically relevant and understandable. Another option is to pick an interesting cluster and look at the terms that are enriched there. Looking at Figure 5.8(a), cluster 5 of the A549 clustering seems quite interesting due to its big ratio of

Table 5.2: Comparisons of terms enriched in significant clusters

Details of comparison	**Number of enriched terms** Biological process	Chromosomal location
Individual clustering		
A549	240	82
BEAS-2B	245	89
Met5A	90	29
A549/BEAS-2B	304	99
Union of sets		
A549 ∪ BEAS-2B ∪ Met5A ∪ A549/BEAS-2B	507	161
A549 ∪ BEAS-2B ∪ Met5A	434	145
Intersection of sets		
A549 ∩ BEAS-2B ∩ Met5A ∩ A549/BEAS-2B	26	8
A549 ∩ BEAS-2B ∩ Met5A	28	8
A549 ∩ BEAS-2B	76	34
A549 ∩ BEAS-2B ∩ A549/BEAS-2B	64	29
Difference of sets		
((Met5A \ A549) \ BEAS-2B) \ A549/BEAS-2B	21	8

assigned genes to expected genes. Table A.2 shows the 15 most significant biological processes in the cluster. Significance is measured with P-value, as usual. Cluster 12 is small but recognised as significant, thanks to the permutation test used by Ernst's clustering method. Table A.3 contains the five most significant biological processes in the cluster.

Chromosomal location

The simplification procedure used for chromosomal locations was described in Section 4.5.1. Applying the procedure resulted in 301 different categories of chromosomal locations. As with biological processes, the enriched locations were recorded. The rest of the analysis also follows that of the biological processes, as laid out in the previous section.

Table 5.2 shows the number of enriched chromosomal locations in the individual clusterings and also some comparisons. For example, there are eight terms that are enriched in each of the clusterings. Table A.4 contains these terms.

Table A.5 shows 15 most significant locations in cluster 5 of the A549 clustering. Table A.6 is similar, but shows five most significant terms in cluster 12.

Order of genes within a cluster

As explained in Section 4.5.3, the genes (probe sets) in each cluster were sorted in the order of decreasing dot product with the corresponding model profile. The results were submitted to our collaborators for further analysis. The author feels that presenting a sample of these gene lists here wouldn't be very meaningful.

Known asbestos-related genes

The table of asbestos-related genes and probe sets was mentioned in Section 4.5.4. The clustering of the probe sets was studied by recording the clusters to which each probe set in the list was assigned. Enriched clusters — that is clusters with surprisingly many asbestos-related genes compared to what can be expected under the assumption that the probe sets in the list are a random selection — were assessed similarly to enriched biological processes and chromosomal locations.

Table A.7 reflects the clustering of probe sets that are associated with genes overexpressed in conjunction with lung cancer. The table shows the number of those probe sets assigned to each significant cluster of the A549 cell line, and the probability of seeing at least as many probe sets according to the hypergeometric distribution (see equations (4.9) and (4.11)). Five of the 47 probe sets in the list are in cluster 5, and the corresponding P-value is 0.0135, which is quite low.

Table A.8 deals with probe sets (genes) overexpressed in mesothelioma, and their clustering in the Met5A cell line. Six of the 53 probe sets are in cluster 2, which gives a low P-value of 0.0065.

Chapter 6

Summary and conclusions

In this thesis, a practical data analysis task dealing with gene expression measurements was carried out. The world of microarrays and gene expression was briefly introduced. Then, a description of the data set at hand was presented. The data had been gathered using 27 microarrays with the intent of studying the effect of asbestos on three different cell lines. There were measurements from a few different time points and two conditions: asbestos exposure and no exposure.

The quality of the data was checked both qualitatively and quantitatively. Despite some quality concerns, the data were judged as eligible for further analyses. Preprocessing was covered in Section 3.2. Some of the main functions of preprocessing are adjustment for background noise and removal of non-biological variation between arrays. A well-known preprocessing method called RMA was chosen.

Chapter 4 described the analysis techniques used with the preprocessed gene expression data. Clustering as a term encompasses a host of different strategies for diving data points into distinct groups. Here, gene expression time series were the data points, and the goal was to find significant patterns of expression. A recent clustering method designed for short time series gene expression data was described, and the related algorithms were laid out in detail. One part of the clustering method, profile selection, was improved with the introduction of a simple randomised algorithm that is a modification to the original, inadequately specified algorithm. Another phase, removal of redundant profiles, was added to the clustering method. Some techniques for the analysis of clusters were also discussed.

The experiments and an overview of the results of the thesis were presented in Chapter 5. Our implementation of the clustering algorithm was tested with synthetic data and found to be reliable. After that, it was used on the asbestos exposure gene expression data set. Some measures were used to prune out some data that was not likely to contain interesting infor-

mation. The remaining gene expression time series were clustered with the previously described algorithm. The clusters were analysed by searching for overrepresented biological processes and chromosomal locations. A list of known asbestos-related genes was also used.

The practical data analysis part of this thesis started with very low-level data. Some rather advanced processing methods were needed in order to extract some information from it. The results achieved during this work do not constitute a complete understanding of the problem at hand. However, they are a good basis for further research done by people more familiar with the biological side of gene expression. This was only a link in the data analysis chain, but quite a significant one.

6.1 Future work

Any future work done by us in this gene expression project probably depends on the needs of the other scientists involved with this collaborative study. The size of our result data set is considerably smaller than that of the raw microarray measurement set we started with. Still, it can be quite difficult to find the interesting information among the results. When something attracts the domain experts' attention, they will be able to ask more specific questions about the results. Then we can do some more focused analyses on specific parts of the clustering results.

Appendix A

Figures and tables

Table A.1: Enriched bioprocesses (uncorrected P-value < 0.01), "Ernst clustering", A549, Beas-2B, and Met

Terms present in all cell lines. Only terms from significant clusters are listed.

BIOLOGICAL PROCESS
* ATP metabolism
* G2/M transition of mitotic cell cycle
* M phase
* RNA splicing
* anti-apoptosis
* apoptotic program
* cell adhesion
* cell cycle
* chromatin assembly or disassembly
* chromatin remodeling
* cytoskeletal anchoring
* fatty acid biosynthesis
* induction of apoptosis by extracellular signals
* intracellular protein transport
* mRNA processing
* nuclear mRNA splicing, via spliceosome
* nucleotide-excision repair
* positive regulation of I-kappaB kinase/NF-kappaB cascade
* protein amino acid phosphorylation
* protein folding
* protein localization
* protein transport

Table A.1: (continued)

Biological process
* protein ubiquitination
* regulation of cell cycle
* regulation of transcription from RNA polymerase II promoter
* regulation of transcription, DNA-dependent
* transcription
* ubiquitin cycle

Table A.2: 15 most important biological processes in cluster 5 of A549

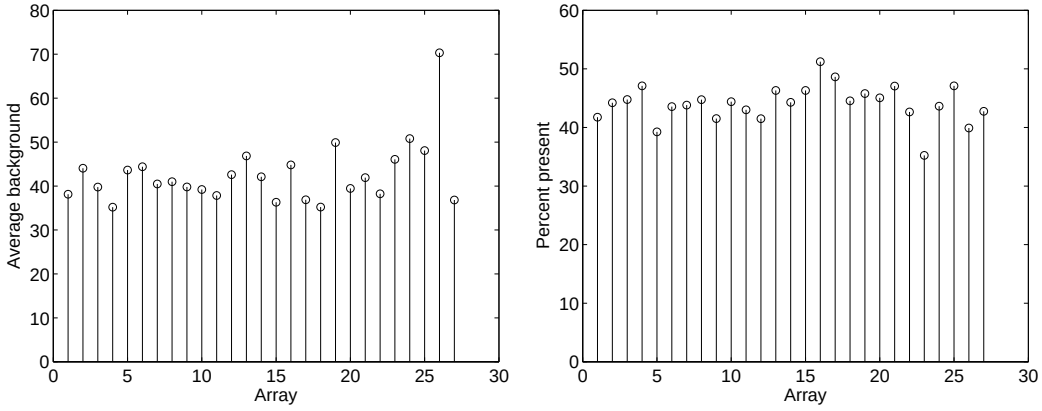
Terms are listed in the order of increasing P-value.

P-VALUE	BIOLOGICAL PROCESS
0.000000	antigen presentation, endogenous antigen
0.000000	antigen processing, endogenous antigen via MHC class I
0.000000	intracellular protein transport
0.000000	protein folding
0.000000	regulation of cell cycle
0.000002	antigen presentation
0.000002	small GTPase mediated signal transduction
0.000003	lipid metabolism
0.000008	metabolism
0.000020	rRNA processing
0.000049	protein-mitochondrial targeting
0.000099	negative regulation of cell growth
0.000151	proline biosynthesis
0.000151	ATP metabolism
0.000177	membrane fusion

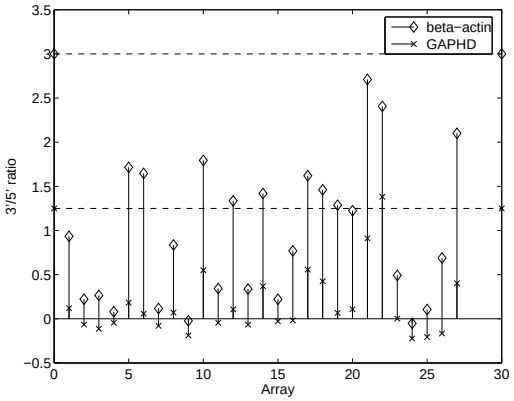
Table A.3: 5 most important biological processes in cluster 12 of A549

Terms are listed in the order of increasing P-value.

P-VALUE	BIOLOGICAL PROCESS
0.000000	glutathione biosynthesis
0.000030	receptor clustering
0.000030	synapse organization and biogenesis
0.000150	acetylcholine receptor signaling, muscarinic pathway
0.000209	clustering of voltage-gated sodium channels

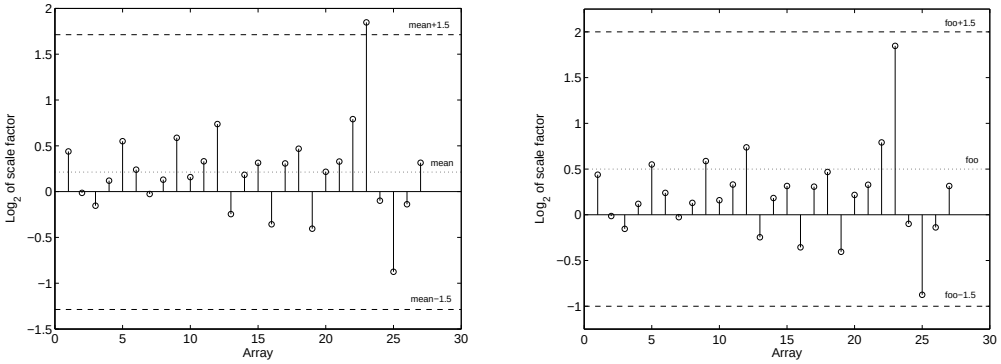


(a) Average background (b) Percent present



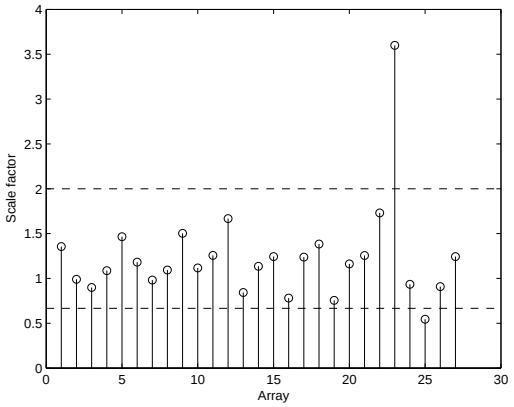
(c) 3′ to 5′ ratios ($\log_2$-ratios)

Figure A.1: Quality control statistics (see Table 2.2 for order of arrays)



(a) Log scale, range around mean logarithmic value

(b) Log scale, range around arbitrary value



(c) Linear scale, range $\left(\frac{2}{3}, 2\right)$

Figure A.2: Scale factor, different interpretations (see Table 2.2 for order of arrays)

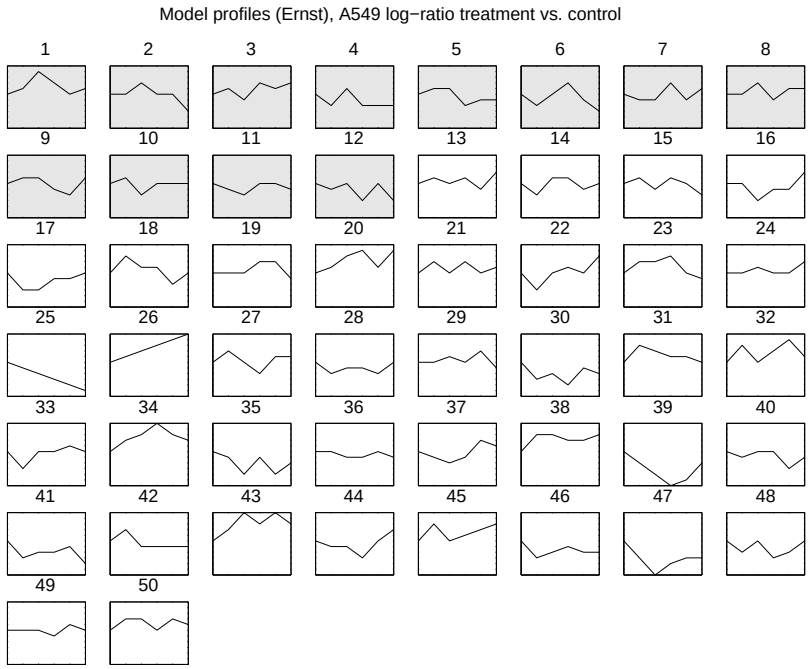


Figure A.3: Model profiles (A549), significant profiles in grey

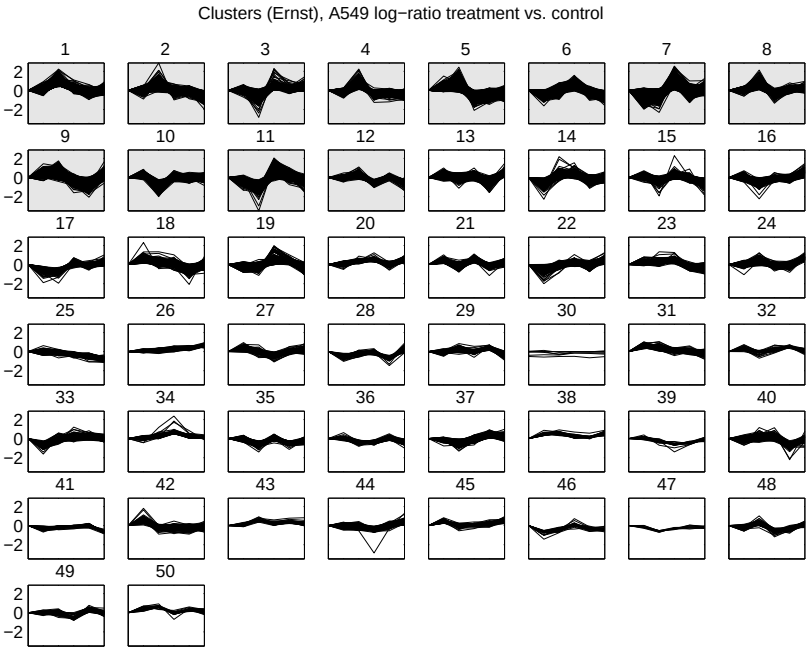


Figure A.4: Clusters (A549), significant clusters in grey

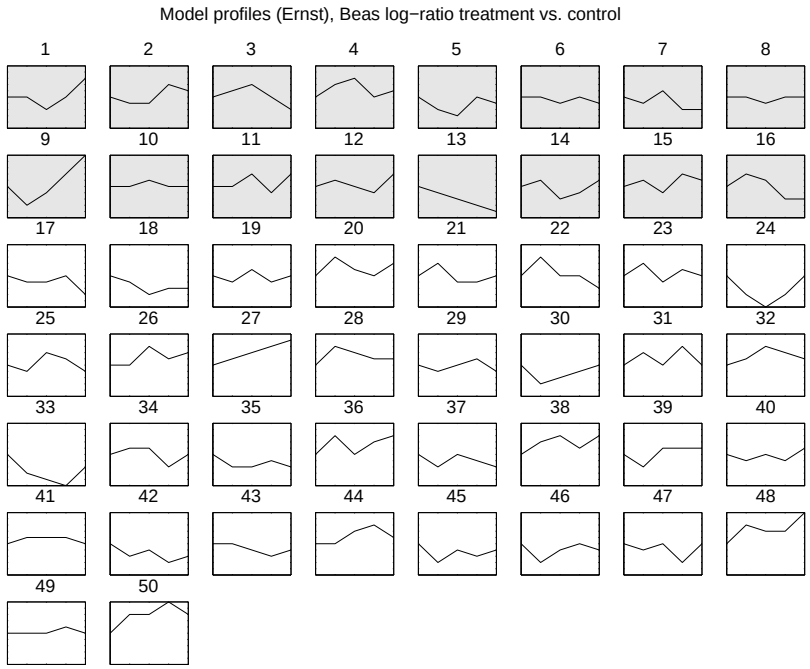


Figure A.5: Model profiles (BEAS-2B), significant profiles in grey

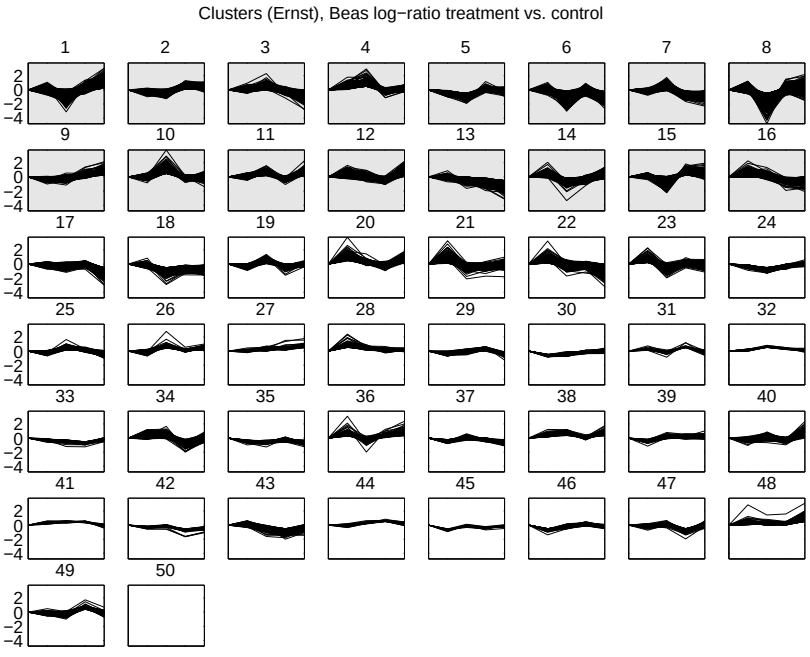


Figure A.6: Clusters (BEAS-2B), significant clusters in grey

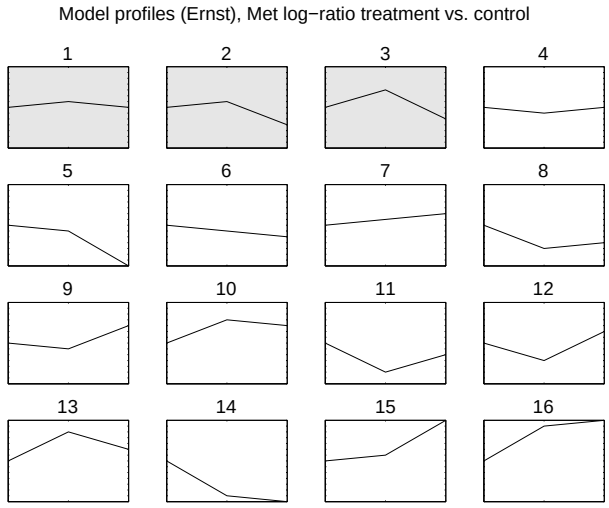


Figure A.7: Model profiles (Met5A), significant profiles in grey

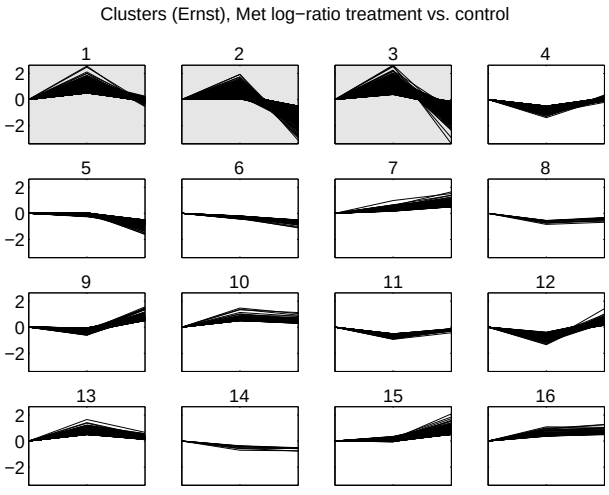


Figure A.8: Clusters (Met5A), significant clusters in grey

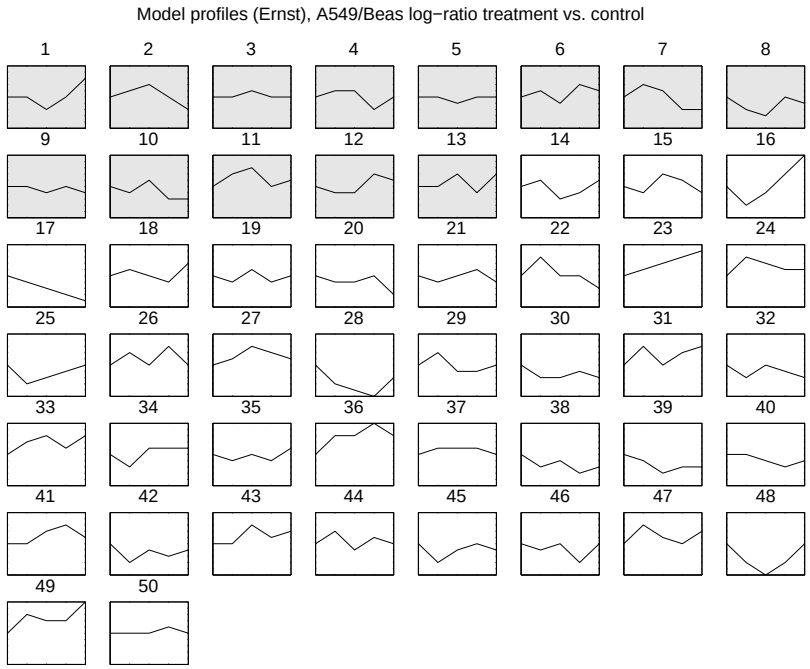


Figure A.9: Model profiles (A549 and BEAS-2B), significant profiles in grey

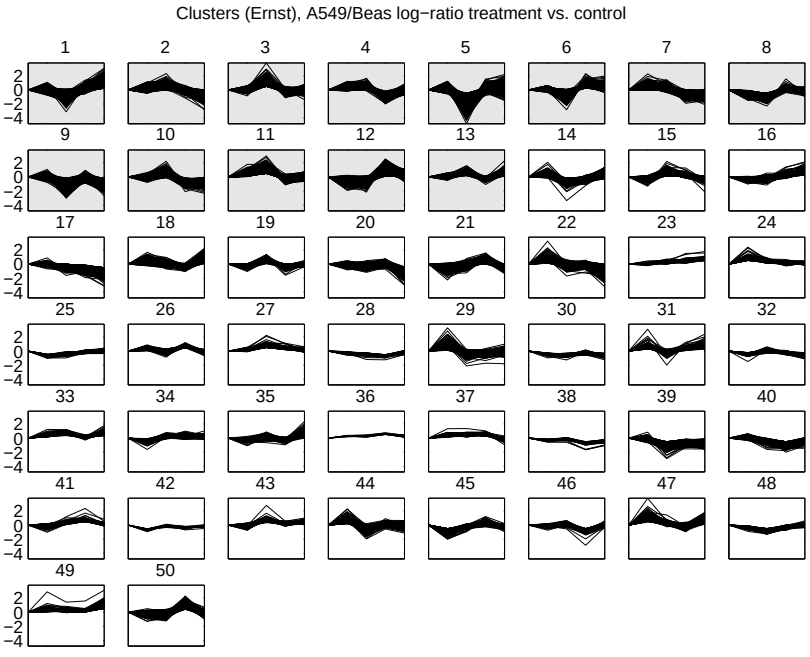


Figure A.10: Clusters (A549 and BEAS-2B), significant clusters in grey

Table A.4: Enriched chromosomal locations (uncorrected P-value < 0.01), "Ernst clustering", A549, Beas-2B, and Met

Terms present in all cell lines. Only terms from significant clusters are listed.

CHROMOSOMAL LOCATION							
4q26	5q31	8q22	11q13	16p11	17q25	19p13	19q13

Table A.5: 15 most important chromosomal locations in cluster 5 of A549

Terms are listed in the order of increasing P-value.

P-VALUE	CHROMOSOMAL LOCATION
0.000000	11q13
0.000000	19p13
0.000000	6p21
0.000000	19q13
0.000000	9q34
0.000000	20q11
0.000001	8q24
0.000001	16p13
0.000003	17q25
0.000056	7p22
0.000057	12q24
0.000104	7q11
0.000112	Xp11
0.000144	16q13
0.000292	1p36

Table A.6: 5 most important chromosomal locations in cluster 12 of A549

Terms are listed in the order of increasing P-value.

P-VALUE	CHROMOSOMAL LOCATION
0.000059	16p11
0.000156	19p13
0.000332	17q25
0.000407	19q13
0.002921	22q11

Table A.7: Clustering of asbestos-related (lung cancer, upregulated) probe sets (47 in total), A549

P-values are from the hypergeometric distribution, $n + m = 54675$

CLUSTER #	CLUSTER SIZE	PROBE SETS ASSIGNED	P-VALUE
1	1108	1	0.6181
2	890	2	0.1780
3	495	2	0.0677
4	449	2	0.0571
5	1653	5	0.0135
6	467	0	1
7	761	0	1
8	700	0	1
9	1099	2	0.2437
10	421	0	1
11	677	2	0.1151
12	174	0	1

Table A.8: Clustering of asbestos-related (mesothelioma, upregulated) probe sets (53 in total), Met5A

P-values are from the hypergeometric distribution, $n + m = 54675$

CLUSTER #	CLUSTER SIZE	PROBE SETS ASSIGNED	P-VALUE
1	1085	1	0.6545
2	1734	6	0.0065
3	2403	5	0.0826

Bibliography

[1] Affymetrix, Inc., Santa Clara, CA, USA. *Data Sheet, GeneChip Human Genome Arrays*, 2004.

[2] Affymetrix, Inc., Santa Clara, CA, USA. *GeneChip Expression Analysis: Data Analysis Fundamentals*, 2004.

[3] Affymetrix, Inc., Santa Clara, CA, USA. *GeneChip Expression Analysis Technical Manual*, 2004.

[4] Affymetrix, Inc., Santa Clara, CA, USA. *GeneChip Expression Platform: Comparison, Evolution, and Performance*, 2004.

[5] Agency for Toxic Substances and Disease Registry (ATSDR). Toxicological profile for asbestos. Technical report, U.S. Department of Health and Human Services, Public Health Service, Atlanta, GA, USA, 2001.

[6] A. V. Aho, B. W. Kernighan, and P. J. Weinberger. *The AWK Programming Language*. Addison-Wesley, 1988.

[7] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, May 2000.

[8] R. Balasubramaniyan, E. Hullermeier, N. Weskamp, and J. Kamper. Clustering of gene expression data using a local shape-based similarity measure. *Bioinformatics*, 21(7):1069–1077, 2005.

[9] A. Bisognin, S. Bortoluzzi, and G. Danieli. Detection of chromosomal regions showing differential gene expression in human skeletal muscle and in alveolar rhabdomyosarcoma. *BMC Bioinformatics*, 5(1):68, 2004.

[10] B. Bolstad, R. Irizarry, M. Astrand, and T. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

[11] B. M. Bolstad. *Low Level Analysis of High-density Oligonucleotide Array Data: Background, Normalization and Summarization*. PhD thesis, University of California, Berkeley, 2004.

[12] A. Brazma, P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, C. Stoeckert, J. Aach, W. Ansorge, C. A. Ball, H. C. Causton, T. Gaasterland, P. Glenisson, F. C. P. Holstege, I. F. Kim, V. Markowitz, J. C. Matese, H. Parkinson, A. Robinson, U. Sarkans, S. Schulze-Kremer, J. Stewart, R. Taylor, J. Vilo, and M. Vingron. Minimum information about a microarray experiment (MIAME)—toward standards for microarray data. *Nature Genetics*, 29:365–371, Dec. 2001.

[13] L. M. Cope, R. A. Irizarry, H. A. Jaffee, Z. Wu, and T. P. Speed. A benchmark for Affymetrix GeneChip expression measures. *Bioinformatics*, 20(3):323–331, 2004.

[14] P. De Vuyst. Guidelines for attribution of lung cancer to asbestos. In Proceedings of an International Expert Meeting on Asbestos, Asbestosis and Cancer [41], pages 92–96.

[15] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.

[16] J. Ernst, G. J. Nau, and Z. Bar-Joseph. Clustering short time series gene expression data. *Bioinformatics*, 21(Suppl. 1):i159–168, 2005.

[17] L. Gautier, L. Cope, B. M. Bolstad, and R. A. Irizarry. affy–analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics*, 20(3):307–315, 2004.

[18] R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. H. Yang, and J. Zhang. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80, 2004.

[19] D. Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, 23(1):5–48, 1991.

[20] P. Good. *Permutation Tests — A Practical Guide to Resampling Methods for Testing Hypotheses*. Springer Series in Statistics. Springer-Verlag, second edition, 2000.

[21] D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. The MIT Press, Cambridge, MA, USA, 2001.

[22] S. Hautaniemi. *Studies of Microarray Data Analysis with Applications for Human Cancers*. PhD thesis, Tampere University of Technology, 2003.

[23] A. Hughes. The central dogma and basic transcription. http://cnx.rice.edu/content/m11415/1.5/, July 2003. Web-document (Connexions Web site). Referenced 27th Jan 2006.

[24] R. A. Irizarry, B. M. Bolstad, F. Collin, L. M. Cope, B. Hobbs, and T. P. Speed. Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Research*, 31(4):e15—, 2003.

[25] R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003.

[26] Y. Kluger, H. Yu, J. Qian, and M. Gerstein. Relationship between gene co-expression and probe localization on microarray slides. *BMC Genomics*, 4(1):49, 2003.

[27] I. S. Kohane, A. T. Kho, and A. J. Butte. *Microarrays for an Integrative Genomics*. The MIT Press, Cambridge, MA, USA, 2003.

[28] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, 3rd edition, 2001.

[29] S. Langård and T. Haug. Screening and early diagnosis of asbestosis and asbestos-related cancers. In Proceedings of an International Expert Meeting on Asbestos, Asbestosis and Cancer [41], pages 85–91.

[30] C. Li and W. H. Wong. Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proceedings of the National Academy of Sciences*, 98(1):31–36, 2001.

[31] C. Li and W. H. Wong. Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application. *Genome Biology*, 2(8):research0032.1–0032.11, 2001.

[32] A. E. Loraine, M. L. Salmi, S. C. Stout, and S. J. Roux. Gene-ontology-based analysis of gene expression changes in early development of ceratopteris spores. In *CSB Workshops*, pages 95–96. IEEE Computer Society, 2005.

[33] J. C. McDonald and A. D. McDonald. The epidemiology of mesothelioma in historical context. *European Respiratory Journal*, 9(9):1932–1942, 1996.

[34] C. J. Miller. *Description of simpleaffy: easy analysis routines for Affymetrix data*, May 2005. Electronic documentation from *simpleaffy* v. 2.0.13.

[35] R. G. Miller. *Simultaneous statistical inference*. Springer-Verlag, New York, NY, USA, second edition, 1981.

[36] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, Cambridge, UK, 1995.

[37] J. Nikkilä. *Exploratory cluster analysis of genomic high-throughput data sets and their dependencies*. PhD thesis, Helsinki University of Technology, 2005.

[38] J. E. Parker. Radiological criteria: The use of chest imaging techniques in asbestos-related diseases. In Proceedings of an International Expert Meeting on Asbestos, Asbestosis and Cancer [41], pages 28–40.

[39] E. Pennisi. Gene counters struggle to get the right answer. *Science*, 301(5636):1040–1041, 2003.

[40] T. V. Perneger. What's wrong with Bonferroni adjustments. *British Medical Journal*, 316(7139):1236–1238, 1998.

[41] *Proceedings of an International Expert Meeting on Asbestos, Asbestosis and Cancer*, number 14 in People and work research reports, Helsinki, Finland, 1997. Finnish Institute of Occupational Health.

[42] R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005. ISBN 3-900051-07-0.

[43] M. F. Ramoni, P. Sebastiani, and I. S. Kohane. Cluster analysis of gene expression dynamics. *Proceedings of the National Academy of Sciences*, 99(14):9121–9126, 2002.

[44] J. Rantanen. Global asbestos epidemic — is it over? In Proceedings of an International Expert Meeting on Asbestos, Asbestosis and Cancer [41], pages 1–4.

[45] S. Ruosaari and J. Hollmén. Image analysis for detecting faulty spots from microarray images. In S. Lange, K. Satoh, and C. H. Smith, editors, *Proceedings of the 5th International Conference on Discovery Science*

(DS 2002), volume 2534 of *Lecture Notes in Computer Science*, pages 259–266. Springer-Verlag, 2002.

[46] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235):467–470, Oct. 1995.

[47] A. Schliep, A. Schonhuth, and C. Steinhoff. Using hidden Markov models to analyze gene expression time course data. *Bioinformatics*, 19(Suppl. 1):255i–263, 2003.

[48] P. Spellman and G. Rubin. Evidence for large domains of similarly expressed genes in the drosophila genome. *Journal of Biology*, 1(1):5, 2002.

[49] T. Strachan and A. P. Read. *Human molecular genetics*. Garland Science, New York, NY, USA, 3rd edition, 2004.

[50] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, San Diego, CA, USA, second edition, 2003.

[51] A. Tossavainen. Asbestos, asbestosis and cancer. Exposure criteria for clinical diagnosis. In Proceedings of an International Expert Meeting on Asbestos, Asbestosis and Cancer [41], pages 8–27.

[52] A. Ultsch and H. Siemon. Kohonen's self organizing feature maps for exploratory data analysis. In *Proc. INNC'90, Int. Neural Network Conf.*, pages 305–308, Dordrecht, Netherlands, 1990. Kluwer.

[53] W. N. Venables and B. D. Ripley. *S Programming*. Springer-Verlag, New York, NY, USA, 2000.

[54] R. L. Virta. Worldwide asbestos supply and consumption trends from 1900 to 2000. Technical Report 2003-83, U.S. Department of the Interior, U.S. Geological Survey, 2003.

[55] E. W. Weisstein. Bonferroni correction. http://mathworld.wolfram.com/BonferroniCorrection.html. From *MathWorld*—A Wolfram Web Resource. Web-document. Referenced 15th Dec 2005.

[56] E. W. Weisstein. Hypergeometric distribution. http://mathworld.wolfram.com/HypergeometricDistribution.html. From *MathWorld*—A Wolfram Web Resource. Web-document. Referenced 11th Nov 2005.

[57] C. Wilson, S. D. Pepper, and C. J. Miller. *QC and Affymetrix data*. Paterson Institute for Cancer Research. Electronic documentation from *simpleaffy* v. 2.0.13.

[58] C. L. Wilson and C. J. Miller. Simpleaffy: a BioConductor package for Affymetrix quality control and data analysis. *Bioinformatics*, 21(18):3683–3685, 2005.

[59] Z. Wu and R. A. Irizarry. Preprocessing of oligonucleotide array data. *Nature Biotechnology*, 22:656–658, June 2004.

[60] Z. Wu, R. A. Irizarry, R. Gentleman, F. M. Murillo, and F. Spencer. A model based background adjustment for oligonucleotide expression arrays. Number 1 in Johns Hopkins University, Dept. of Biostatistics Working Papers. May 2004.

[61] L. Zhang, M. F. Miles, and K. D. Aldape. A model of molecular interactions on short oligonucleotide microarrays. *Nature Biotechnology*, 21:818–821, July 2003.