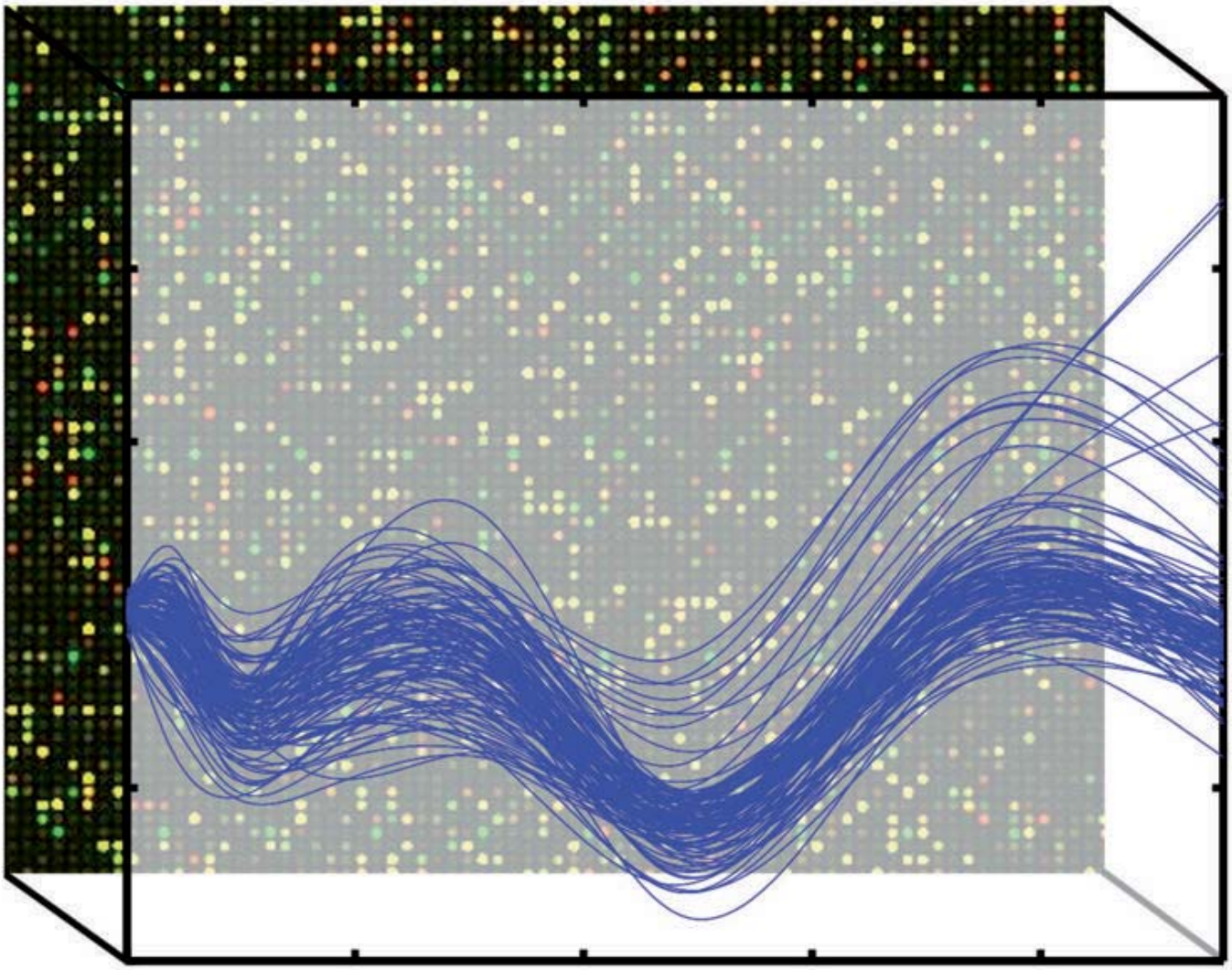
Christian Salzig

Modeling of Gene Expression Time Courses and Identification of Gene Interaction Networks



Cuvillier Verlag Göttingen
Internationaler wissenschaftlicher Fachverlag

Dissertation im Fachbereich Mathematik der Technischen Universität Kaiserslautern

Modeling of Gene Expression Time Courses and Identification of Gene Interaction Networks

Christian Salzig

1. Gutachter: Prof. Dr. Dieter Prätzel-Wolters
2. Gutachter: Prof. Dr. Jürgen Franke

Datum der Disputation: 06.01.2011

Vom
Fachbereich Mathematik der Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften
(Doctor rerum naturalium, Dr. rer. nat.)
genehmigte Dissertation

D 386

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.
1. Aufl. - Göttingen : Cuvillier, 2011
Zugl.: (TU) Kaiserslautern, Univ., Diss., 2011

978-3-86955-629-1

Nonnenstieg 8, 37075 Göttingen
Telefon: 0551-54724-0
Telefax: 0551-54724-21
www.cuvillier.de

1. Auflage, 2011
Gedruckt auf säurefreiem Papier

978-3-86955-629-1

To my parents

Abstract

This thesis was written within the Rhineland-Palatinate Research Program for Universities *Wissen schafft Zukunft*. Two-channel microarray experiments for different organisms and under different treatments were run in cooperation with the departments Biology and Chemistry of the University of Technology in Kaiserslautern, the *Center for Nanostructure Technology and Biomolecular Technology* Kaiserslautern, the *Institute of Biotechnology and Drug Research Kaiserslautern* (IBWF) and the *Fraunhofer Institute for Industrial Mathematics Kaiserslautern* (ITWM). The task of the ITWM and thus the task of the PhD work was the design and the analysis of the experiments as well as the mathematical modeling of the resulting data.

The thesis is based on the experiments of the IBWF concerning the rice blast disease fungus Magnaporthe grisea, whose gene expression patterns during host infection and the first 24 hours of growth shall be ascertained.

After presenting the basics of genetics and microarray technology as well as a short overview of the considered organism, sources of variance during the data generation are described. Since there was nearly no data for Magnaporthe available in literature at the beginning of the project, a two-step-concept for design of experiments is presented.

Therefore, one run of microarray experiments was made as a first screening. Thereafter additional experiments are made in mature time periods in which a large number of changes in gene expressions occurred. Normalization techniques for reducing technical variances are described and applied to the data. The resulting, adjusted data is analyzed using the non-parametric Fisher-Pitman-Test, since a normal distribution of the data was not assumed. Furthermore, the minimal sample size for detecting significantly differentially expressed genes at all is calculated.

As a next step the interpolation of the microarray time course data using cubic smoothing splines is proposed. Again, as for the statistical analysis, the medians of the single measurements are used as an estimator for the interpolation, since no normal distributed data was assumed. The estimator accuracy, needed for spline smoothing, is calculated using an exact version of the bootstrapping method.

Thereafter several clustering methods as well as distance measures suitable for gene expression data are presented and applied to the Magnaporthe data. The results are evaluated using internal validity indices. For later needs in the gene interaction modeling step, several ways for calculating the average time course of a cluster based on its elements are described.

A discrete, linear time-invariant state space system is taken as model for the gene interac-

tion network. It is fitted to the data using linear regression. The result is evaluated concerning system theoretical attributes as autonomy and stability. Furthermore, the robustness with respect to the discretization step width is tested. Finally, potentials and limits of the presented methods are discussed.

Contents

Thanks

This work would not be accomplished without the aid of so many people. I would like to thank all of them.

First of all I am deeply grateful to my dear wife Eva. Your help and your exhortation were an essential basis for this work. Thank you very much for just listening to these (in your opinion probably very strange) mathematical thoughts. Thank you for your support in finding appropriate English translations and phrases. Thank you for working so hard, doing all the housework, keeping the overview of all things to do, doing them and thus allowing me to stay as much time as possible at the computer, especially in the last half-year after the birth of our daughter Pia who also needed care and attention.

This leads to the next one, I would like to thank: My dear little Pia. Your smiles spent the energy I needed during the final rush. I hope you will forgive me for sitting in front of this strange but also a little bit exciting computer-thing around-the-clock instead of playing with you.

I am very grateful to my adviser Prof. Dr. Dieter Prätzel-Wolters who made it possible to write this thesis directly within a research project and for his scientific support during the last years.

Many thanks to Dr. Karsten Andresen from the Institute of Biotechnology and Drug Research in Kaiserslautern. Besides providing the measurements used in this work, he was my gateway to biology. Thanks a lot for the conveyed background knowledge in biology, the direct insight into the measurements procedures, the many papers he made available to me and all the fruitful discussions.

I am also grateful to Annette Krengel. She started as diploma candidate in the project and quickly became a valuable colleague and friend during our work.

Thanks to my employer, the Fraunhofer Institute for Industrial Mathematics Kaiserslautern, the department System Analysis, Prognosis and Control where I was allowed to work the last four years and its head Dr. Patrick Lang. Thanks to all colleagues who set up a pleasant working atmosphere, especially Dr. Jochen Broz, Thomas Halfmann, and Dr. Jan Hauth.

I also thank the Graduate School of department of Mathematics at the the University of Technology in Kaiserslautern for the scholarship.

Furthermore, I would like to thank all other partners working in the research project *Microarray Based Analysis of Transcriptome and Proteome for Solving Complex Problems*:

Prof. Dr. Ekkehard Neuhaus, Dr. Thorsten Möhlmann, and Dr. Michaela Traub from the Division of Plant Physiology of the University of Technology in Kaiserslautern.

Prof. Dr. Eckhard Friauf, Prof. Dr. Hans Gerd Nothwang, Dr. Heike Ehmann, and Michael Boesen from the Division of Animal Physiology of the University of Technology in Kaiserslautern.

Prof. Dr. Dr. med. Dieter Schrenk, Dr. Hans-Joachim Schmitz, Dr. Stefanie Knerr, Silke Germer, and Yvonne Fery from the Division of Food Chemistry and Environmental Toxicology of the University of Technology in Kaiserslautern.

Prof. Dr. Regine Hagenbeck and Dr. Patrick Maurer from the Center for Nanostructure Technology and Biomolecular Technology in Kaiserslautern.

Prof. Dr. Timm Anke and PD Dr. Eckhard Thines from the Institute of Biotechnology and Drug Research in Kaiserslautern.

I would like to thank Prof. Dr. Jürgen Franke for reviewing the thesis and the helpful talks. Thanks to Prof. Dr. Gerhard Pfister for being head of the examination board.

Many thanks to my parents Hildegard and Walter and my sister Marion who supported me during all the years but especially when the work stagnated, such that I kept my spirits up.

Thanks to Oliver Schmidt, Steffen Ilschner, and Stephan Wilz for proof-reading.

Finally, thanks to the Coca-Cola Company for their liquid containing the sugar and caffeine I needed during several nights [1].

[1] Dear Jan, its nearly four years ago, that you told me, no PhD thesis was ever written without drinking coffee. This work shall prove the opposite!

1 Introduction

If you try and take a cat apart to see how it works, the first thing you have on your hands is a non-working cat.

— Douglas Adams, British author and satirist (1952-2001)

1.1 History and Appetizer

Aristotle (384 BC – 322 BC), perhaps the most famous pioneer of biological science, described the analysis of growth and development of live in his work *On the Generation of Animals* [Ari]. He opened fertilized chicken eggs at several mature times for observing, when the visible organs were generated. However, Aristotle and many of his successors did empirical research based on macroscopic observations, and the results are often influenced by religious or spiritual beliefs. This held up to the Middle Ages.

During the scientific revolution the research abandoned supernatural argumentation and started to collect facts and involve mathematics. Many important inventions and developments paved the way to modern biology and especially genetics. The invention and enhancements of the microscope enables the view onto living cells. In 1676, the Dutch tradesman Antonie van Leeuwenhoek (1632 - 1723) observed microorganisms for the first time which established the field of *microbiology*.

About 200 years later, in 1865, the Augustinian Gregor Johann Mendel (1822 - 1884) published at two meetings of the Brünn Natural History Society his research results concerning systematical breeding experiments with pea plants, which laid the foundation for the biological field of *genetics*. He suggested the existence of genes, basic units carrying the traits from parents to offspring. His report also contained several mathematical formulas for the laws of heredity [Men66].

Thomas Hunt Morgan (1866 – 1945) was able to prove the existence of genes and that these are situated on inner cellular structures which were called *chromosomes*. In 1933 he was awarded the Nobel Prize in Physiology or Medicine for these results.

Several discoveries as gene mutations and the deoxyribonucleic acid (DNA) led to the *central dogma of molecular biology* articulated by the British molecular biologist Francis Crick (1916 – 2004) in 1958 [Cri58]:

"Once information has got into a protein it can't get out again."

This transfer as well as the involved elements will be described in detail in chapter 2.

Finally, the microarray technology was invented in the late eighties of the last century and a gene expression profiling using miniaturized *cDNA microarrays* was presented for the first time in 1995 by Mark Schena, Dari Schalon et al. [SSDB95].

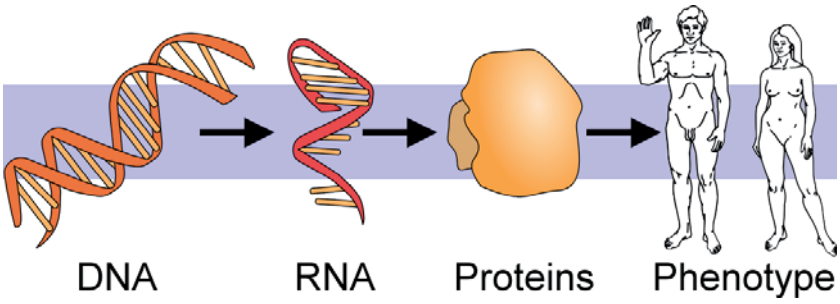


Figure 1.1: Genetic information flow

Although huge steps in biological science were made since Aristotle, this work describes the analysis of experiments which are very similar. However, due to the developments and inventions of the scientists mentioned above as well as many others the analysis could be done using more mathematics. Whereas the chicken mature experiment was mainly based on visible changes of the organism, this work will use subjective visual inspection only in the very first step. Thereafter microarray experiments were set up and evaluated using appropriate mathematical methods.

This skip from biology to mathematics should be used to introduce also some of the mathematicians, whose work were essential for the microarray analysis presented in this thesis, sorted by the usage of their methods during the analysis.

The Briton Sir Ronald Aylmer Fisher (1890-1962) was one of the most famous biologists and statisticians of the 20th century. He especially contributed to statistical design of experiments and analysis.

In 1979, the statistician Bradley Efron (born 1938) published the bootstrap technique for computer-based calculation of estimator accuracies [Efr79]. This method is essential for the time course interpolation of the microarray measurements which was done in this work.

This leads to the Romanian Isaac Jacob Schoenberg (1903-1990) who became famous for the development of interpolating splines [Sch46].

Last but not least George David Birkhoff (1884-1944) who formulated the modern dynamical system, but representing all mathematicians who contributed to systems and control theory which will be needed as final step in the analysis and modeling of the gene interaction network.

1.2 Task and Work flow

This work focuses on experiments made by the Institute of Biotechnology and Drug Research in Kaiserslautern for analyzing the genetic expression time courses during the growth of the fungus Magnaporthe grisea. All steps from experimental design up to the generation of a gene interaction network had to be mathematical well-founded.

However, two bottlenecks hampered the work:

When the project started in 2005 there was exactly one microarray data set published comparing dormant and germinated fungus spores. In fall 2007 the *Magnaporthe grisea Oryza sativa interaction database* (www.mgosdb.org) was set up to allow web-based submission and publishing of microarray data. [WV09]

So the complete experiments and the analysis had to be made from scratch. And even today there are only few suitable data sets freely available. The public repository *Gene Expression Omnibus* of the National Center for Biotechnology Information (www.ncbi.nlm.nih.gov/geo/) contains eleven microarray data sets in the beginning of 2010.

Secondly the budget of the project restricted the number of microarrays to be available. While many qualitatively impressing results are presented in literature, the own sight had to be lowered with respect to this boundary. Methods as for example Bonferroni-techniques for handling the statistical significance of gene family-wise test statements had to be neglected since they would result in an increase of the needed sample size and thus the needed microarrays.

Based on these guidelines, the experimental flow was as follows (cf. figure 1.2):

First of all the growth of the fungus was visually inspected for detecting phenotypical changes, which gave rise to the first time points for the measurements. Microarray experiments were made and the resulting data was statistically analyzed. In the time intervals exhibiting the most and highest changes in gene expression levels additional time knots were inserted. A second run of microarrays was used to hybridize all chosen time points of the fungus growth in a balanced and even manner.

The resulting data was normalized and the gene expression levels were statistically estimated. Thereafter the discrete time measurements were interpolated to receive a continuous gene expression time course. These time courses were clustered into large sets of simultaneously expressed genes before they are fitted by a mathematical model.

Anyhow, all these working stages and the mentioned keywords will be explained more precise and step by step in the following chapters.

1.3 Structure of the Thesis

Chapter 2 contains biological and technical basics which are essential for readers without appropriate background knowledge to understand the following analysis methods. A short

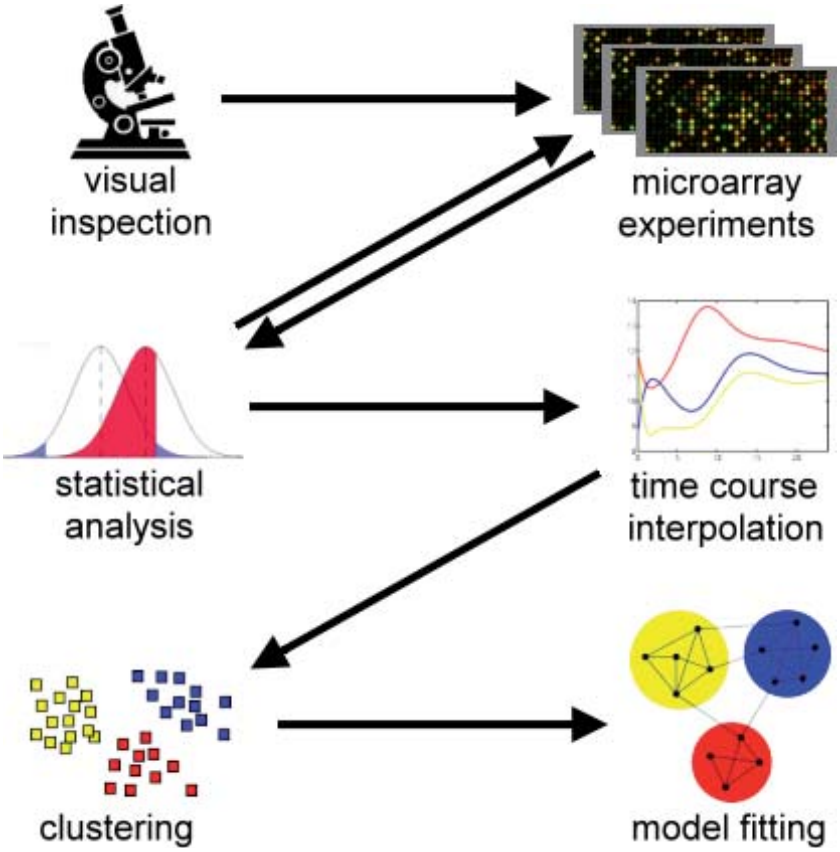


Figure 1.2: Work flow presented in this thesis

introduction into genetics is given and the functionality of microarray chips is presented. Especially sources of variances during the experiments are pointed out. Furthermore, the test organism Magnaporthe grisea, its growth and relevance in agriculture is described.

In chapter 3 all methods used for the design of experiments, the normalization, and data extraction of microarray measurements are presented. Light is shed on the key elements and main influences for experimental design before the finally used design is given. Thereafter several normalization steps for handling different error sources of microarray experiments are shown. The chapter is completed by the statistical analysis of the data using the non-parametric Fisher-Pitman-Test and the calculation of the minimal number of microarrays needed for a specific experiment.

Chapter 4 deals with the interpolation of the microarray measurements during time. Therefore, the interpolation points and its accuracies are calculated. For the latter one an exact variant of the bootstrap method is introduced. Based on these values a smoothing splines are fitted to the data resulting in continuous estimations of the expression time courses of each gene.

Due to the fact, that the calculation of a full-genome interaction network is not possible – Magnaporthe grisea has more than 15000 genes – the genes had to be clustered. This is done in chapter 5. Therefore, several appropriate distance measures of gene time courses as well as three common clustering methods are discussed. The resulting clusters were validated using quality indices. Finally, methods for estimating the overall time course of clusters are presented.

Chapter 6 shows the calculation of the cluster interaction network. Interestingly a linear model fits well to the data while more complex models as recurrent neural networks which take non-linear effects as saturation into account did not yield comparable results or did not converge at all. Neither classical neural network training algorithms as the backpropagation through time [RHW86], [Wer90] nor Bayesian particle filter methods [Hau08] did satisfying jobs fitting non-linear models to the data. Thus this chapter focuses a discrete linear time-invariant state space model, its fitting to the data and the evaluation of the result based on system theoretical properties.

In chapter 7 finally the complete procedure from design of experiments up to the gene interaction model is reviewed and summarized.

Please note also the extensive bibliography with many books and articles containing alternative approaches and possible extensions depending on the available data records.

This work is a mixture of many fields of mathematical application. Unfortunately each field has its own notation, which results in several overlaps in the usage of letters in the different chapters. Nevertheless, the common notation is kept, such that each mathematical chapter (i.e. the chapters 3-6) is written in its own private variable context. At the end of each of these chapters the results are summarized, applied to the given data example. Additionally, the resulting parameters, variables, and formulas which will be used in the following chapters are defined. This notation is kept unique.

2 Biological and Technical Background

The moment one give close attention to anything, even a blade of grass,
it becomes a mysterious, awesome, indescribably magnificent world in itself.

— Henry Miller, American writer (1891-1980)

2.1 Genes and DNA

It is not just fate that organisms look as they look and do as they do. The central statement of the biological field of genetics is that everything is controlled by a large amount of factors, inherited from the organisms parent or parents, called the genes. These bunches of biological information are situated as distinguishable units in each cell of each living being. The whole set of genes, called genome, contains every piece of information needed for a fully functional individual of a species. Everything said for the organism holds for its single cells, too. Most organic cells also do not do nothing - they interact. Whenever influences like nutrients, hormones, toxicities etc. reach them from neighbor cells or from the outer environment, the genes determine the cell responses.

There are many school books about the biochemical background of genes, their interaction and control of cells' behavior containing much more detailed information than it is possible to write here in this chapter. This work also does not take all mechanisms into account gene regulation exhibits - this would truly go beyond the scope of it. Thus, if desired the gently reader may refer to [Hen98] or [Bro93].

The genes are encoded into the deoxyribonucleic acid (abbr.: DNA), which is a polymer of nucleotides. Each of those nucleotides is a molecule of the sugar deoxyribose, a phosphate group and one of four different nucleobases: adenine (A), cytosine (C), guanine (G) and thymine (T). Sugar and phosphate build the backbone of the polymer, whereas the nucleobases, or better the sequence of these bases determine the genetic information. Between the bases a different number of hydrogen bonds can be formed and therefore for each base a complementary one exists: Adenine and thymine are coupled by two, cytosine and guanine by three hydrogen bonds (cf. figure 2.1). Thus, a DNA strand fits exactly to another, its inverse copy (lock and key principle).

In eukaryotic cells the gene encoding DNA (called sense) and its complementary DNA

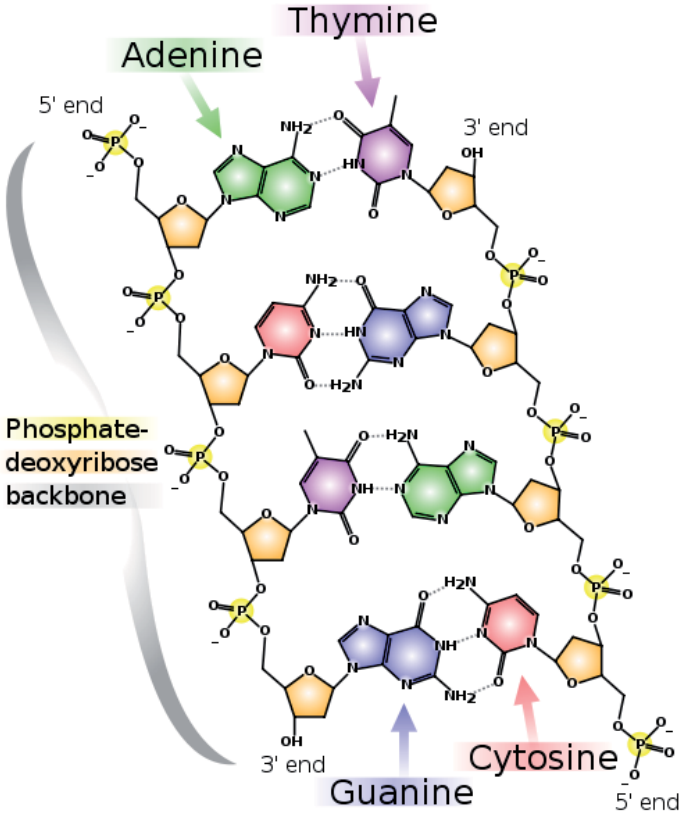


Figure 2.1: Piece of a DNA double strand

(called antisense) form a double-stranded compound twisted to a double helix and coiled around proteins forming structures called chromosomes as shown in figure 2.2. In the majority of cases, these chromosomes exist diploid, i.e. in a twin set, one set from each parent, and are situated in the cell nucleus.

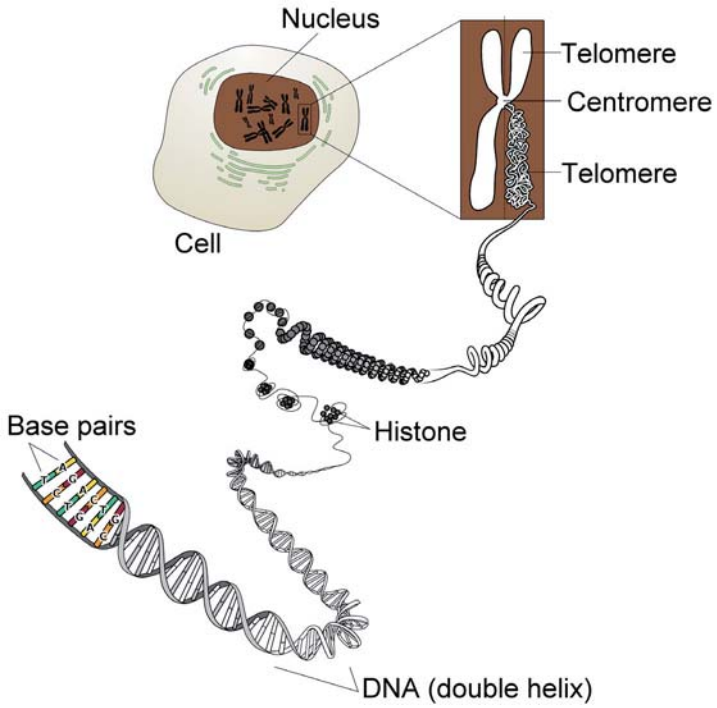


Figure 2.2: Structure of chromosomes

To give an idea of the size of DNA, imagine that the largest human DNA strand is composed of more than 200 million base pairs building a fiber of 8.4 cm length. The complete haploid (single, contrary to diploid) set of human chromosomes consist of approximately 3.2 billion base pairs, from which only a small amount encodes the genes. The human genome comprises about 25.000 genes, depending on the source [LLB+01], [Ste04], [Con04].

A single gene is encoded in sequences between some hundreds up to tens of thousands of bases. Most of the DNA, about 2.1 billions of base pairs, do not encode genetic information. This so-called extragenic DNA contains sequences needed for gene expression (cf. next section 2.2), for DNA stability and replication during cell divisions as the telomeres at the chromosome endings, or remainders of the evolutionary process.

2.2 Information Flow in Biological Cells

The basal regulation of the cells behavior is done by the so-called gene expression, where proteins for needed purposes are built from the genetic information. Whenever a special pro-

tein is needed as a response of environmental influences or simply for the upkeep of the cell metabolism, a gene corresponding to this protein has to be activated. Therefore, other proteins, known as transcription factors, bind to an extragenic part of the DNA called promoter which enables an enzyme called RNA polymerase to read the desired gene information following to the promoter region. A small part of the DNA in front of the RNA polymerase is uncoiled and an inverse version of the antisense strand is produced (cf. figure 2.3).

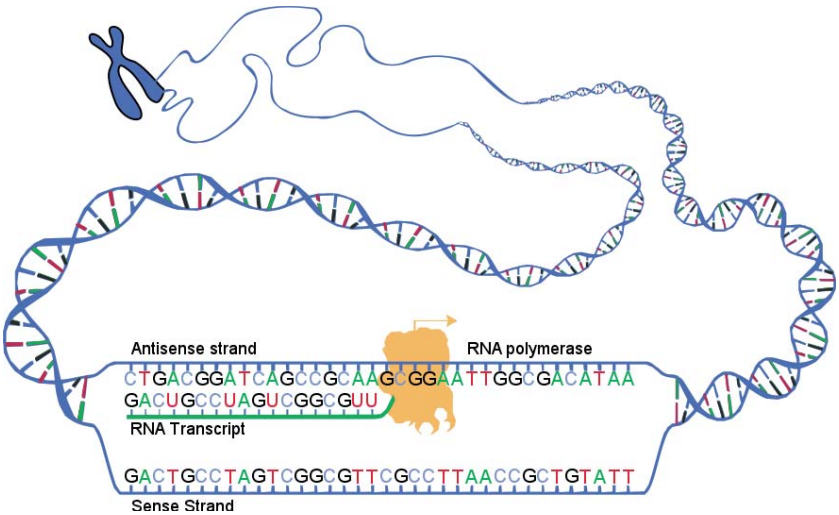


Figure 2.3: RNA polymerase and transcription

This inverse copy is again a nucleic acid, but unlike DNA it contains another base called uracil (U) as complement to adenine and the sugar ribose instead of deoxyribose, and therefore these are called ribonucleic acids (RNA). Besides this molecular biological difference the RNA coincides the sense sequence corresponding to the desired gene. When the RNA polymerase reaches the extragenic section behind the genetic information, the so-called transcription terminator, it releases the synthesized RNA strand and detaches itself from the DNA. This transfer, the so-called transcription, takes place in the nucleus. The RNA molecules produced by transcription can be divided into three main classes: ribosomal RNA (rRNA), transfer RNA (tRNA) and pre-messenger RNA (pre-mRNA). The first two types are relatively persistent and are needed for protein synthesis without encoding the protein structure itself as the more short-lived pre-mRNA does.

As the name indicates, the pre-mRNA has to be further processed. Between gene coding sections, called exons, there are still non coding parts called introns. The introns are removed and the remaining exons are combined to the mature mRNA by a catalytic structure called spliceosome. This process called splicing allows the coding of multiple different protein isoforms by one gene, if exons are extended or skipped or introns are kept. This phenomenon called alternative splicing and is regulated again by special proteins.

Once the mature mRNA is built, it is transported from the nucleus to special organelles

		2nd base			
		U	C	A	G
1st base	U	UUU Phenylalanine UUC Phenylalanine UUA Leucine UUG Leucine	UCU Serine UCC Serine UCA Serine UCG Serine	UAU Tyrosine UAC Tyrosine UAA Stop UAG Stop	UGU Cysteine UGC Cysteine UGA Stop UGG Tryptophan
	C	CUU Leucine CUC Leucine CUA Leucine CUG Leucine	CCU Proline CCC Proline CCA Proline CCG Proline	CAU Histidine CAC Histidine CAA Glutamine CAG Glutamine	CGU Arginine CGC Arginine CGA Arginine CGG Arginine
	A	AUU Isoleucine AUC Isoleucine AUA Isoleucine AUG Methionine*	ACU Threonine ACC Threonine ACA Threonine ACG Threonine	AAU Asparagine AAC Asparagine AAA Lysine AAG Lysine	AGU Serine AGC Serine AGA Arginine AGG Arginine
	G	GUU Valine GUC Valine GUA Valine GUG Valine	GCU Alanine GCC Alanine GCA Alanine GCG Alanine	GAU Aspartic acid GAC Aspartic acid GAA Glutamic acid GAG Glutamic acid	GGU Glycine GGC Glycine GGA Glycine GGG Glycine

Table 2.1: The genetic code

in the cytoplasm, the so-called ribosomes. Those ribosomes consist of rRNA molecules and proteins and are the main location for the protein biosynthesis called translation. Proteins are polymers of amino acids, which develop special functional shapes, depending on their sequence which is defined by the nucleobase sequence of the generating mRNA strand. Each base triplet called codon maps to exactly one of twenty possible amino acids or acts as start or stop codon for the translation. Table 2.1 shows this codon-protein mapping, the so-called genetic code.

Note that the mRNA sequence can be read in three different ways, depending on which base is used as start of a codon. These three possibilities are called reading frames. For preventing ambiguousness a special triplet AUG (marked with * in table 2.1) is perceived as start codon in addition to encoding the amino acid methionine. This means that the translation begins at one of the first AUG codons on the mRNA strand. Which start codon on the mRNA actually initializes the protein synthesis depends on the sequence of adjacent codons.

The colors of the amino acids in table 2.1 denote physical properties of the amino acids: The red marked amino acids are nonpolar (hydrophobic), which means they prefer contact to other nonpolar molecules and solvents. They are especially repelled by water. The other colors denote hydrophilic amino acids. The green amino acids are polar but electrical neutral, while the blue ones are basic and thus positive chargeable and the magenta ones are acidic and negative chargeable. The importance of the difference in physical behavior will be explained later on.

The tRNA is a short RNA sequence with two functional sites: The anticodon, an triplet of nucleobases which is the inverse of a codon on the mRNA and an acceptor arm which carries the amino acid corresponding to the codon. At the ribosome a mRNA codon and the adequate anticodon of a tRNA fit together and the tRNA releases its amino acid which is bound to the end of the in this way generated amino acid chain (cf. figure 2.4). Once translated the mRNA is released from the ribosome and might be read again or denature after a while. The degeneration ensures that a specific mRNA is only present, when it is needed.

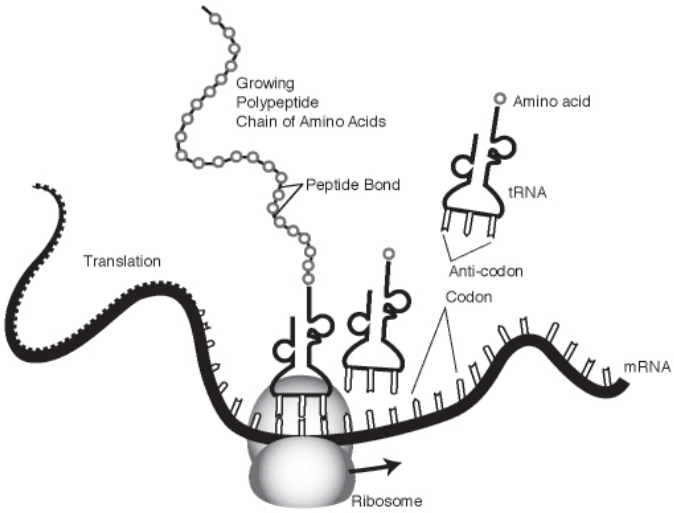


Figure 2.4: Translation at a ribosome

The electrical charges and the ability or inability of the amino acids to develop hydrogen bonds let the protein fold to a specific three-dimensional shape, depending on the order of amino acids. The resulting structure usually has a hydrophobic core and a hydrophilic surface, gaps and binding sites for building larger compounds with other proteins or molecules. These protein compounds finally are capable to do the job which was desired, when its gene was read. They might be enzymes catalyzing some biochemical reaction, transport proteins carrying important molecules through the cell membrane, regulatory proteins which again control metabolic reactions or might have several other functions. Finally, the protein degrades after a certain amount of time, differing by the protein type.

The genes whose DNA sequences are currently translated into proteins are called expressed genes.

But how does a cell know, which gene is needed to be expressed?
Any step of the information flow from DNA to proteins can be modulated by regulatory factors. Repressors inhibit the transcription of genes while activating transcription factors may initiate it. Environmental conditions might change the structure of regulatory proteins which

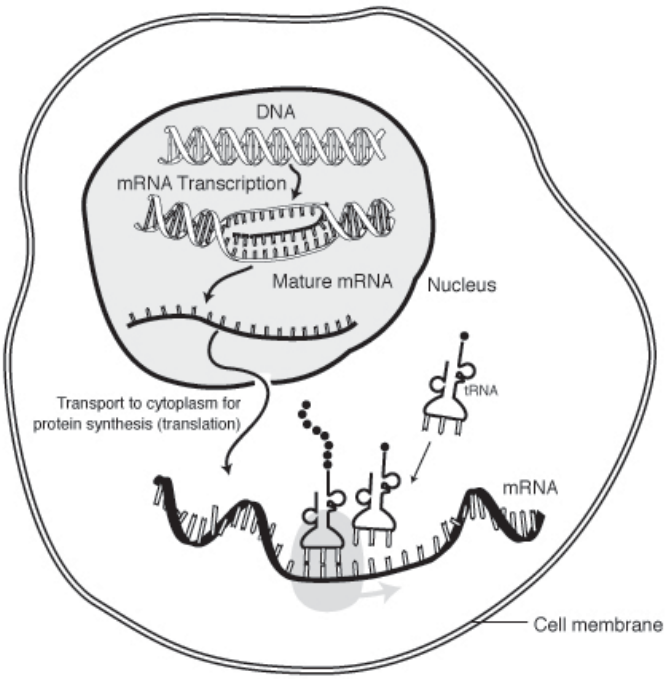


Figure 2.5: mRNA transcription and translation in a cell

causes the expression of genes which will generate an adequate respond to these influences. In this manner a special piece of genetic information is only read and translated to proteins when it is needed.

The amount of mRNA corresponding to a currently expressed gene shows saturation effects. Obviously, the concentration of mRNA in the cytoplasm is bound from below by zero, whereas a upper saturation is caused by two effects. Firstly the degradation rate of mRNA is concentration based, which causes together with a constant production a ordinary differential system of first order. Secondly produced proteins might act as inhibitors for the their own corresponding transcription.

As already hinted, in the majority of cases more than one special protein is needed for an appropriate reaction to environmental influences. A whole bunch of proteins work together in parallel or serial interaction pathways. Thus, genes are often expressed simultaneously in clusters.

Furthermore, some proteins act as regulatory factors for others resulting in a network of dependencies. This so-called gene interaction networks are expected as sparse, which means each gene is regulated only by a few others.

For scientific research of cell processes it is important to obtain information about the transcriptional activity of all involved genes as extensive as possible. Measuring the amount of mRNA in the cytoplasm and thus of the expression level of corresponding genes yield an overview for metabolism, cellular differentiation and cell signaling.

Especially time course experiments, where the gene expression levels are measured at several time steps during some treatment period, give conclusions concerning the gene to gene interaction and gene clustering. Such analysis allows a systematic manipulation of specific components of the physiological network.

2.3 Two-Channel DNA Microarrays

”Microarrays” is a collective name for molecular biological multiplex assay technologies, which allow parallel analysis of a large number of individual features of a small amount of biological sample materials. Therefore, an arrayed series of many thousands of diminutive spots of biological material as DNA or antibodies is imprinted on a glass slide, each spot dedicated for one feature.

In the last decade the usage of microarray techniques became a very popular method for high-throughput screenings. From 1998 to 2001 the technology became more an more integrated in everyday research and the publications concerning microarrays quadrupled each year. Thus, referring to single articles or books is nearly impossible, however [Bla03] and [HKL$^+$05] should be mentioned, since these books overview microarray techniques and analysis very well. Many other papers about functionality, pros and cons of microarrays are listed in the bibliography.

In the following let the term ”microarray” denote the special type of full-genome two-channel DNA microarrays used for gene expression profiling, described below.

These microarrays allow for the parallel measurement of the full transcriptional activity of cells or cell compounds and thus they are a powerful biomolecular tool for measuring the activity of thousands of genes up to the entire genome.

For that purpose gene corresponding DNA strands are spotted and fixed in a microscopic grid on a slide of size of an object holder. Each probe spot position can be mapped to one special gene of the organism which shall be examined. In addition to spots containing DNA strands of the organism to be analyzed there is a small amount of control spots, showing a special behavior which will be described below.

The typical microarray experiment takes course as follows (cf. figure 2.6):
First of all two cell cultures are generated and hold under different conditions or taken from different tissues, e.g. one healthy, the other one diseased. Each culture lives and grows by activating the genes appropriate for its own environmental conditions. The needed gene information is transcribed into messenger RNA and released into the cytoplasm where it is translated into proteins as explained in the previous section. This mRNA is extracted from the cells of each culture and purified. If the mRNA fulfills requirements in quality and quantity it is converted back to a more stable complementary DNA (cDNA) by a reverse transcription. During this procedure different fluorescent markers, typically the green Cyanine 3 (Cy3) and

the red Cyanine 5 (Cy5), are bound to the cDNA strands of the two samples, such that they can be distinguished by their emitted light. Thereafter the solutions of color-labeled cDNA of both cultures are mixed and incubated with the microarray chip where the cDNA can attach to the spots of its inverse DNA strands by base pairing. This procedure, called hybridization, runs under beneficial, as constant as possible conditions such that the technical variation is as minimal as possible, i.e. the percentage of bound cDNA is comparable between different experiments. After some time, usually several hours, an equilibrium is reached, when the ratio of bound cDNA of both samples coincides approximately with the ratio in the solution.

The control spots behave different. They are made such that genetic material binds depending on the embedded dye. Thus, none, one particular, or both color channels bind to them strongly, depending on the control spot type.

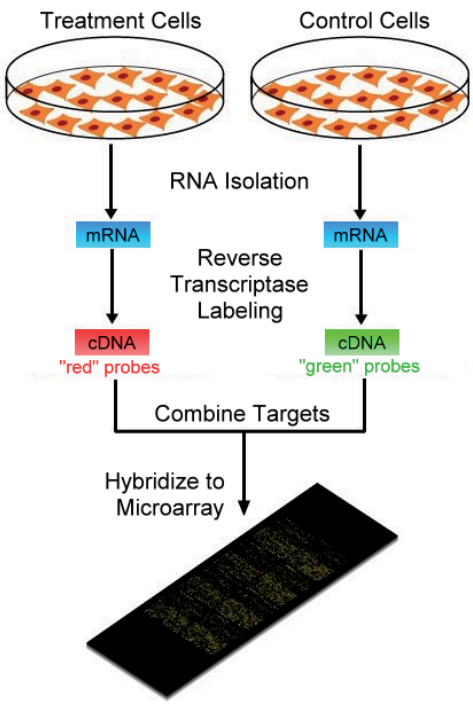


Figure 2.6: Microarray hybridization

After the hybridization step the solution containing the remaining unbound cDNA is washed off and the microarray chip is ready for scanning. Therefore, the microarray is irradiated by a laser exciting at the characteristic wavelength of one of the dyes. The light emitted by the concordant fluorescent markers is caught in a photomultiplier tube and quantified by the scanner. Afterwards the procedure is repeated using the characteristic wavelength of the other dye. That way at each spot of the microarray the intensity of excited light emission is measured. An example result of the scanning process is shown in figure 2.7, where green and red spots indicate the expression of the corresponding genes in the green respectively red labeled specimen and the yellow spots represent the expression in both channels.

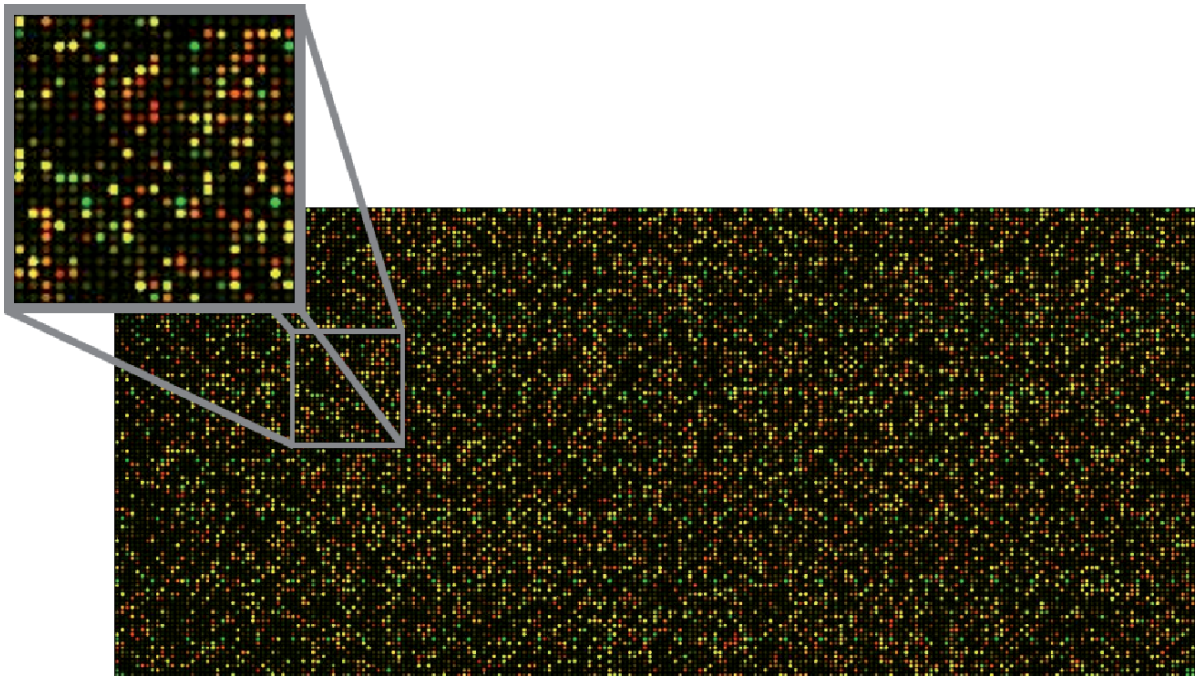


Figure 2.7: Scanning result of a microarray hybridization. Greener spots indicate a higher amount of bound Cy3-labeled cDNA, redder spots have more Cy5-labeled cDNA bound while yellow spots are close to an even amount of Cy3- and Cy5-labeled cDNA

The intensity values allow the calculation of the amount of bounded cDNA of each sample, which can be used for reconstructing the expression level of the genes, i.e. the amount of mRNA, in the cell cultures. The disparity between both intensities at one spot yields an up- or down-regulation of the associated gene in one sample compared to the other.

2.3.1 Multi- and Time Course Experiments

Often there is not only one special comparison between one test group and one control group desired, but a whole set of comparisons of different cell states, be they different tissues, organisms or treatments. From now on, let a set of microarrays for an analysis comparing more than two samples be called multi-experiment.

A special case of multi-experiments are so-called time course experiments.

This setup compares the changes in gene expression of a tissue under a constant or after an initial treatment during time. Growth processes and long-run stress response studies are common examples. Time course experiments are nothing else than a bunch of single microarrays comparing the gene expression states of the tissue at several times. The resolution of

time points is bounded from below due to some variance sources. First of all, biological individuals and even each single cell of an individual differ in their reaction time. Thus, the mRNA extracted from the tissue is no snap-shot of an exact particular time, but an average over a small time interval. Secondly, the extracting of the tissue and its mRNA might take some time. Especially this fact will be the restricting factor for the microarray time course experiments considered in this work as described in section 2.5.

2.4 Error Sources during Microarray Experiments

Unfortunately microarray measurements are very noisy. Besides the biological variance, the variation of expression levels between single individuals of the same kind, there are many technical sources of errors and variance.

Chip Dependent Errors

Each spot differs in the amount of genetic material plotted onto it during the chip fabrication. Furthermore, the binding reaction of the cDNA to the spots might have proceeded more fully to the equilibrium in one array than in another. Thus, the overall fluorescence of a designated spot might differ between the different chips although exactly the same probes are applied to them. Nevertheless, this chip dependent variation holds for both, the red and the green labeled cDNA on the chip. Therefore, the ratio between the color intensities is not changed by this type of error source.

Sample Dependent Errors

Even if the genetic material is extracted as accurate as possible from the tissue, the quantities of obtained mRNA of the samples for the the microarray experiment differ. The biased reverse transcription into cDNA as well as different build-in rates of the dye labels into the cDNA strands cause further variance. This results in differences in the ratio of dye labeled cDNA in the probes compared to the pristine amount of mRNA in the tissues.

Dye Dependent Errors

The two colors show different fluorescent behaviors. Due to frequency dependent differences in the emission responses to the excitation laser and measurements in the photomultiplier tube, the red dye (Cy5) tends to be brighter as the same amount of the green dye (Cy3), but depending on the overall intensity of the measured spot. The logarithmic ratio of the intensities of the two color channels plotted against the spot's overall brightness shows a characteristic

curvilinear shape (cf. figure 2.8). This systematic dependence of the ratio from the overall intensity is not reasonable by any biological process.

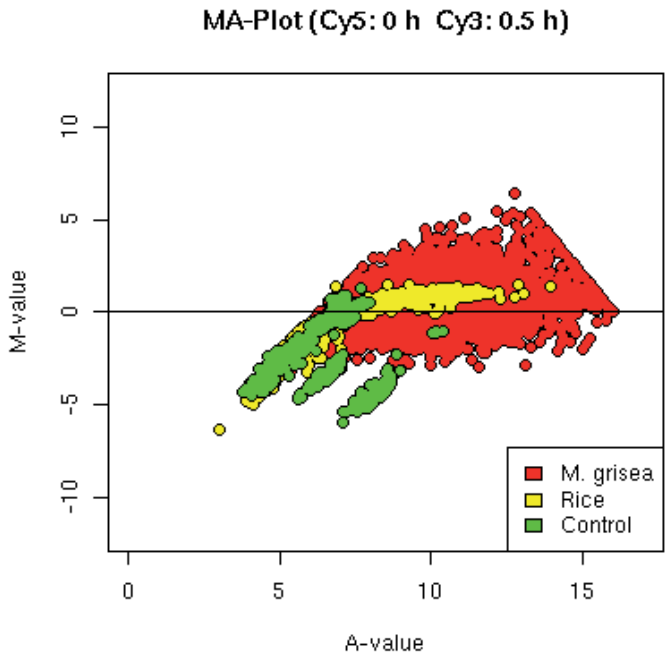


Figure 2.8: Typical curvilinear shape in plots comparing brightness and color ratio of microarray spots

Furthermore, environmental influences affect the dyes in different ways (e.g. ozone degrades Cy5, while it does not affect Cy3 in a comparable manner) which results in a systematic chip-wide bias.

Scanner Dependent Errors

Due to the technical properties of the laser scanning mechanism a microarray spot appears brighter if the area around it is bright. Additionally to the spot-specific intensity information the photomultiplier tube also catches light emitted by the labeled cDNA, which was not completely washed off and remains on the slide around the real spots.

Upper boundaries for measurement values of the photomultiplier are reached at very bright spots resulting in wrong ratios of measured dye intensities.

Uncertainties during the spot detection of the scanner cause further variance.

Work flow Dependent Errors

Slight differences in the handling of the test subjects and microarray slide might also result in noticeable technical variation. Organisms are sensitive to time of day, the way of handling (and thus sensitive to the practitioner) and so on. Unevenly exhaustive washing of the chips let the amount of remaining unbound cDNA differ within as well as in between chips. Finally, white noise during all working steps causes an additional variation in each single measurement.

2.5 Magnaporthe grisea

The organism, analyzed in this work, is the plant-pathogenic fungus Magnaporthe grisea, which is probably the most destructive pathogen of rice in the world and thus one of the worldwide most devastating threats to food production. Rice is the most important food crop, being the primary source of food for more than half of the world's population. The rice blast disease caused by Magnaporthe grisea is known as rice fever in China since 1637. It spread over more than 85 countries worldwide and destroys nowadays hundreds of millions of tons of rice grain each year – enough to feed about 60 million people. Solely in China 5.7 million hectares of rice were lost to the blast disease between 2001 and 2005. The importance of the fungus is even increased by the fact, that the rice production has to be increased significantly to ensure the feeding of the growing world population. Recent studies of the International Food Policy Research Institute indicate a need of an increase of 38% up to 2030. [WT09], [CSC+09], [PL09], [WV09]

Due to its importance to the world agriculture but also because of advantageous properties as model organism for host-parasite interactions Magnaporthe grisea became subject of many biological studies. [Ebb07]

The airborne spores of the fungus may infect all aboveground parts of rice plants and other cereals at any growth stage. The spread of the fungus in the host causes white to dark gray lesions at diseased leaves, stems or panicles from which the fungus sporulates again to infect new plants. While younger plants die, an infection of older rice plants causes the loss of the grain set.

Magnaporthe grisea has seven chromosomes which are sequenced, i.e. its DNA base sequence is identified, except for about 3%. These gaps cause that up to now still 159 so-called supercontigs exist (www.broadinstitute.org). These supercontigs are identified parts of chromosomal DNA which cannot be linked together. For understanding, if Magnaporthe would be fully sequenced, only seven supercontigs would remain, one for each chromosome. The genome of Magnaporthe grisea contains more than 11000 protein-encoding genes. [DTEF05]

The data used in this work was generated using Magnaporthe grisea grown in vitro on an appropriate culture medium under favorable conditions. For time 0 mRNA was taken from dormant spores, which were not applied to the culture medium. Additional times were 1/2, 1, 2, 4, 8, 12, 18, and 24 hours after application (post-inoculation, abbr.: p.i.) of the

Figure 2.9: Blast disease lesions on rice leaf, collar, node, and neck (l. to r.)

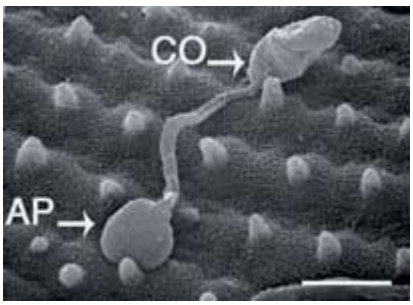


Figure 2.10: Scanning electron micrography of a Magnaporthe spore (conidium, CO) developing an appressorium (AP) on a rice leaf; scale bar 10μm

spores. Therefore, the fungi of the different growth stages were scrapped from the culture medium, shock-frosted and their mRNA was extracted for the further steps of the microarray experiments. The scrapping and frosting took about half an hour, which was the delimiting factor of time resolution for the time course experiment. The times 2, 4, 8, and 24 hours were taken due to visible physiological changes of the fungus, which are described below, while the selection of the other times will be explained in section 3.1.

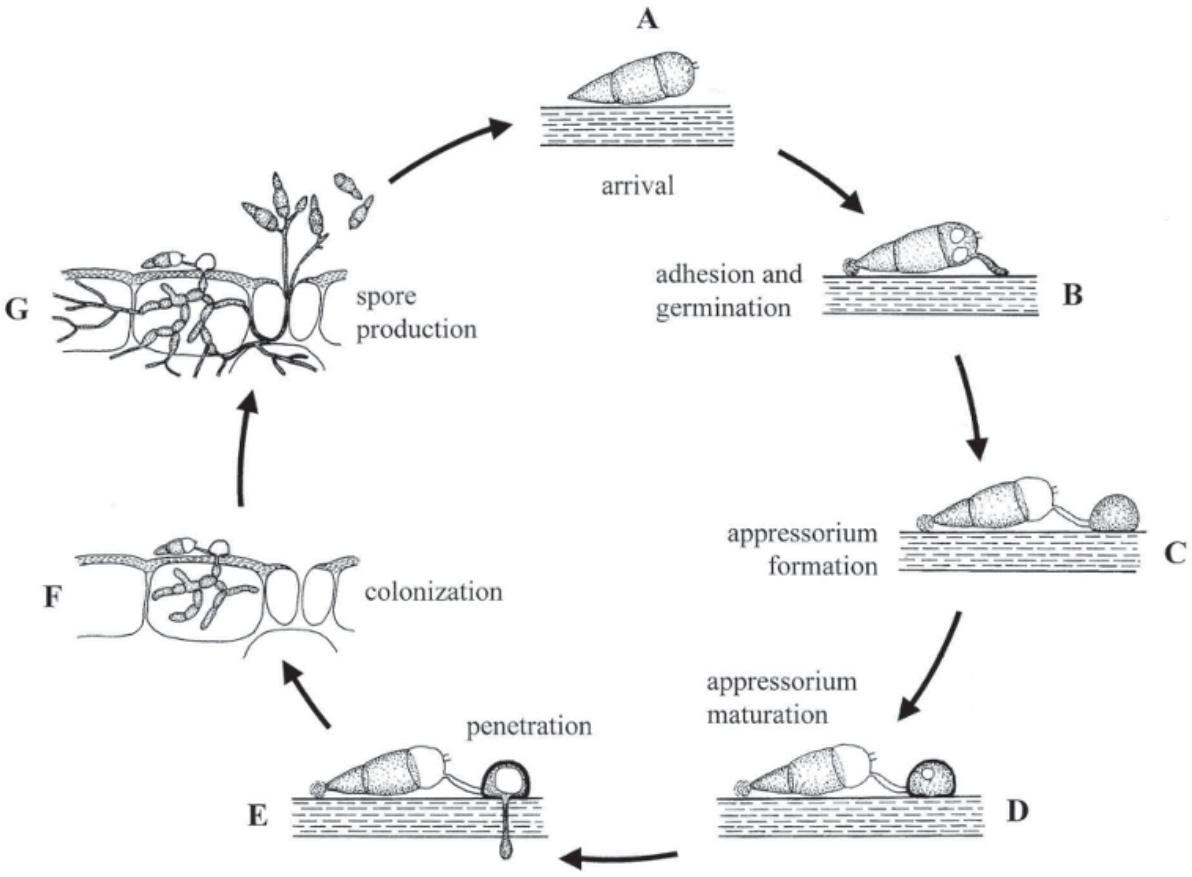


Figure 2.11: The infection cycle of Magnaporthe grisea

Immediately after the arrival of the Magnaporthe grisea conidium (spore) on a rice leaf, i.e. at 0 hours p.i., it releases a drop of mucilage, a gluey substance which sticks the spore to the leaf surface (fig. 2.11, A) At 2 hours p.i. the conidium germinates by generation a short germ-tube which forms a hook (fig. 2.11, B). By 4 hours p.i. an immature appressorium, a pressing organ at the end of the germ-tube, is formed (fig. 2.11, C). The appressorium matures at 8-24 hours p.i. by embedding a thick melanin layer in the inner appressorial cell wall for resisting the pressure. The turgor pressure is generated by glycerol which is built and transferred to the appressorium time-delayed with respect to the melanin production (fig. 2.11, D). At 30 hours p.i. the pressure is high enough to force a penetration peg into the rice leaf (fig. 2.11, E).

Thereafter the fungus grows in the host and 4-5 days p.i. new conidia are produced (fig. 2.11, G).

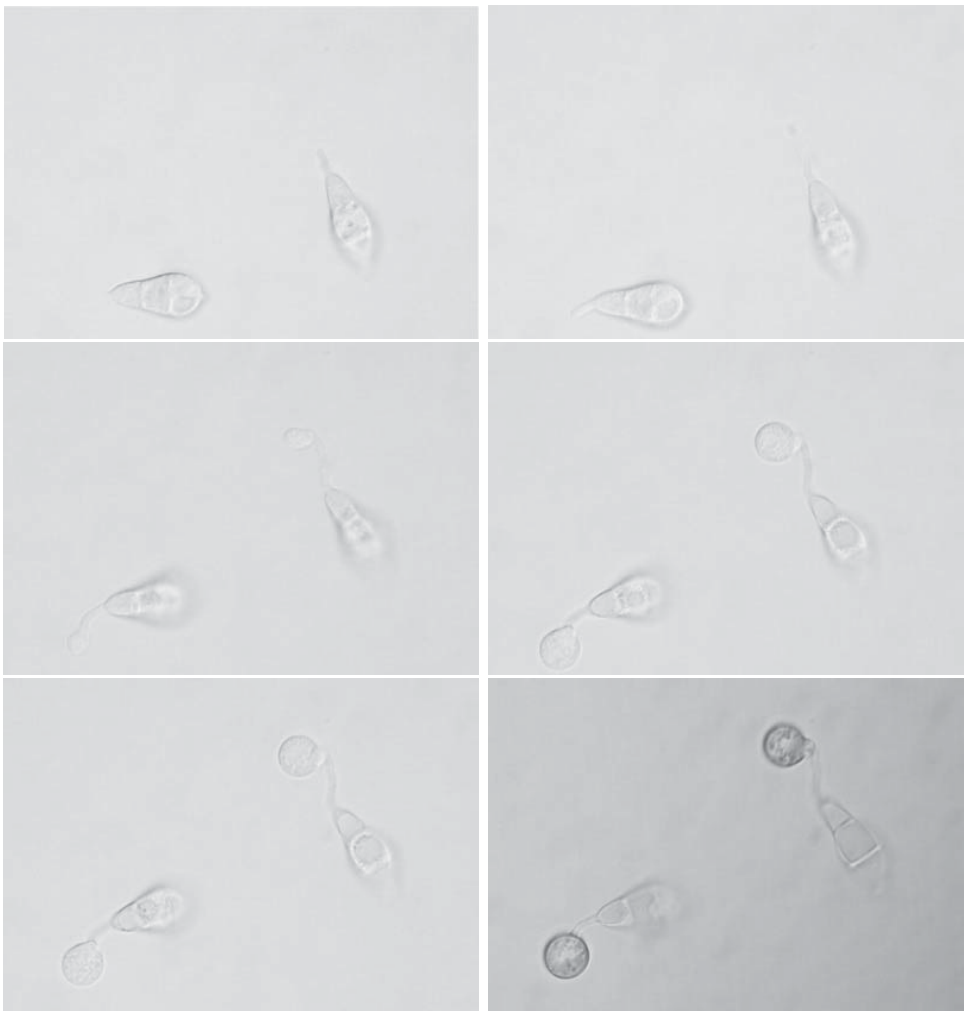


Figure 2.12: Growth of Magnaporthe grisea conidia; 1/2h p.i. (top left), 1h (top right), 2h (middle left), 4h (middle right), 9h (bottom left), 18h (bottom right)

The used microarray architecture was the Agilent Magnaporthe Gene Expression Microarray consisting of 15170 spots containing Magnaporthe grisea DNA, 6325 spots with DNA of the host plant rice and 1080 control spots.

A last comment concerning the nomenclature of Magnaporthe. While Magnaporthe oryzae is the scientifically correct name of the fungus, which prefers rice (scientific name: Oryza) as host, the old name Magnaporthe grisea is still used in many communities [CK02]. So this thesis refer to this old but more common name.

3 Preprocessing of Microarray Measurement Data

It is not enough to put thumbscrews on nature.
One has to understand when she testifies.

— Arthur Schopenhauer, German philosopher (1788-1860)

When making measurements, two main questions arise inevitably:
How to make them? and *How to use them?*

Rather than addressing the technical realization of the measurement procedure (this task should already be solved), the first question deals with the problem is, how to set up a good strategy for producing meaningful data. There is nothing worse than running many expensive experiments and collecting data which is not capable of generating significant results. A so-called design of experiments has to be made to ensure that the resulting data yields as much information as possible. Thus, in the first section of this chapter design strategies for the microarrays are presented.

Furthermore, the raw data drawn from experiments is as the name promises: raw. Many error sources as already seen in section 2.4 falsify the measurements. Errors have to be identified and removed or at least reduced, since a complete elimination is hardly ever possible. Section 3.2 presents methods for error reduction.

The remaining influence of errors give rise to the second question: how to use measurement data. For separating the "true data" from noise, the measurements have to be repeated. Then statistical methods allow the estimation of the desired values. Section 3.3 yields a statistical test appropriate for evaluating microarrays. Furthermore, this section closes the circle, since a design of experiments could not be done before defining the needs of the statistical methods. Here especially a formula for the number of needed measurement replications is presented.

3.1 Microarray Design of Experiments

Now knowing what microarrays are and that two different subjects are hybridized on them, the question arises how to select the sample pair for each chip for getting a cheap but statistically meaningful experiment. This selection is known as design of experiments.

In this field many articles and books present different design approaches. Many of them are designated as *optimal* - which might be correct for the special tasks they presented. A very good overview to existing designs is given in [Chu02] and [KVLB09].

The selection has to be driven by statistical reasoning which Blalock divides appositely in four factors: control, balance, randomization and replication. [Bla03]

"Control" denotes the structure of sample groups to be analyzed. Is there a dedicated control group from which differences to other samples have to be discovered, or are all the subjects equal and every comparison of note?

Considerations of all possible technical error sources are subsumed under the keyword "balance". The general guideline is to distribute sources of variation equally to all samples. Those variations might be time of day, speed of preparation and many others, the practitioner would not even think of (cf. section 2.4). In two-channel microarrays one of the most important design rules is the usage of so-called dye-swaps. Since the dye-labeling influences the measured data strongly, in addition to all normalization methods it is recommended to hybridize each sample equally often with both colors. It would complicate the analysis, if one treatment sample would have been hybridized solely with Cy5, another solely with Cy3. Then obviously changes in gene expression levels can not be distinguished from dye influences.

While balance considers technical error sources, "randomization" handles biological ones. Many biological factors (e.g. age, gender, weight) might have unknown influences to the experiment, causing a biased result, if these factors are not equally distributed to the treatment groups.

Finally, "replication" covers the number of repeated experiments and the number of individuals in one treatment group. The dye-swap mentioned already under balance shows the importance of repeated measurements, called replicate. Additionally to the reduction of technical variances replicates allow statistical estimation of variances in the first place. In this context one has to distinguish technical and biological replicates. Technical replicates (also known as duplicates) are copies of the extracted genetic material prepared for hybridization, while biological replicates are samples taken from different individuals of the same organism under the same treatment. Thus, the technical replicates enable the measurement of technical variances during the hybridization and scanning process, whereas biological replicates also take biological variance into account, but intermixing it with the technical variance. Having more than one individual in a treatment group is known as pooling. Interestingly the community, practitioners as well as statisticians, differs about the usefulness of pooling. Fact is, that pooling reduces the variance of the treatment group, since a pool of n individuals has a variance of

$$\sigma^2_{pool} = \frac{\sigma^2}{n} \tag{3.1}$$

where σ^2 denotes the variance of the individuals.

This variance reduction would be helpful for distinguishing data distributions from microarray experiments by many statistical tests. Expectedly pooling has also a drawback, since (3.1)

holds only for normally distributed data and approximately for distributions similar to it. Often gene expression levels in treatment groups are not approximately normally distributed but exhibit significant outliers. These might be caused by many unknown and uncontrolled error sources like diseases and unusually high stress responses. While individual hybridizations would easily allow for the detection of those outliers pooling blurs its influence distorting the average expression level of the treatment group. However, often the species to be examined already implicates the decision if pooling should be used. Obviously, in the case considered in this work, the fungus Magnaporthe grisea, a single spore would not yield enough mRNA for the microarray machinery and thus pooling is inevitable.

Before selecting one of the standard designs, these principles have to be regarded to determine an appropriate one. Using the words of Sir Ronald A. Fisher:

"To call in the statistician after the experiment is done may be no more than asking him to perform a postmortem examination: he may be able to say what the experiment died of."

For the design of multi-experiments there are two main standard designs known in literature: the common-reference and the loop design (also known as balanced design). Most of other designs are special cases or mixtures of those.

The Common-Reference Design

For this design a common control group is defined as reference. This might be the untreated organism, a dormant spore or a tissue of a specific age or at a specific time. cDNA from this control group is hybridized against the cDNA of treated organisms at the desired time points (cf. figure 3.1, left). This yields a large number of control data but only one measurement of a treated tissue per microarray chip. The common-reference design is expedient whenever no comparison between the treatment groups but only the direct comparison of treatment to control group is desired. So the statistical analysis of each comparison takes advantage of the strongly represented control data, yielding usable results with a low number of necessary chips. Even if a reference is not desired for examination and only the different treatments have to be compared to each other, the common-reference design is chosen frequently. In that case, each microarray has only one color channel of interest reducing the yield of data per chip, but the experiment is less sensitive to biases and variances, since each sample to be analyzed is handled in the same manner: Equally often hybridized in the same color channel and the same reference. A further benefit of the reference design is the easy expandability by adding additional experiments with new treatment groups.

The Loop Design

Loop designs need no definite control group. Each test group is considered as equal and thus hybridized equally often (cf. figure 3.1, middle). This yields a common data size for all statistical analyses and thus it is the preferable design whenever all test groups shall be compared to each other. Furthermore, this design allows the usage of both color channels of the microarray

for groups to be examined - which is not always the case in the common-reference design. The drawbacks are firstly that each treatment group is hybridized with different others, resulting in a less comparable situation than the common-reference design. Secondly, this kind of design is hardly expendable for additional treatment groups.

A variant of the loop design is the saturated design where each test group combination is hybridized equally often (cf. figure 3.1, right). If the saturated design demands that each combination is hybridized twice with interchanged colors (dye-swap), the number of needed microarrays for n test groups is at least $n(n-1)$. Due to the polynomial growth of this variant of the well-known handshake-problem this design results already for small numbers of test groups in large numbers of necessary microarrays.

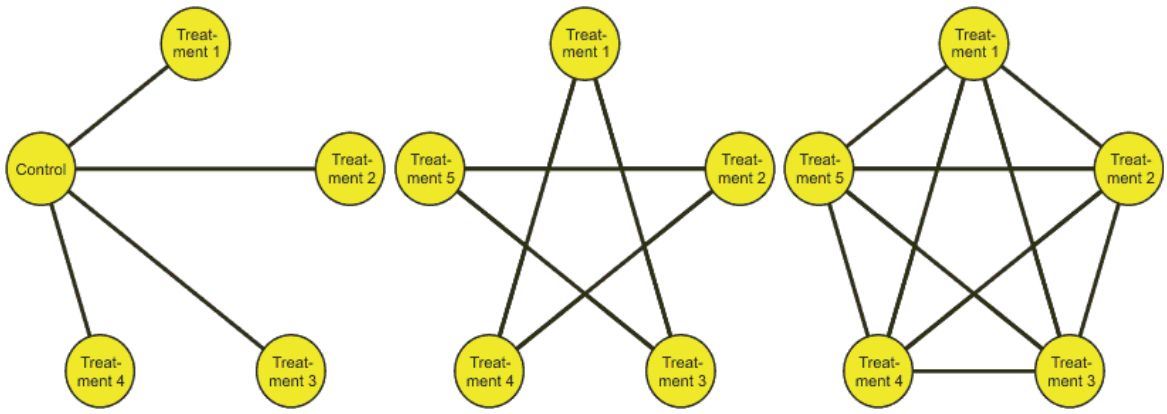


Figure 3.1: **Left:** Common-reference design. **Middle:** Loop design. **Right:** Saturated design

The Mixed Design

Even if no dedicated control group exists, often a mixture of both described design types is used for microarray experiments, profiting from the advantages of both methods.

The experiment in this work was made in a two-step design (cf. figure 3.2):
The dormant Magnaporthe grisea spore at time 0 was used as control group. Further spores applied to the culture medium and grown for different maturing times became the test groups. The first run of microarrays was made using the common-reference design, comparing a low number of test groups to the control. This yielded knowledge about the genetically most active time intervals. In these intervals additional times were added to the test groups for the second phase. The additional microarrays needed for the new test groups were used to balance the design. This resulted in a design with a higher weight on the control group due to the first phase, but less experiments as needed for a common-reference design with the same number of test group hybridizations.

The selection of times for the first phase were based on the visible physiological states of the fungus growth (cf. section 2.5), and differs obviously for other organisms or treatments.

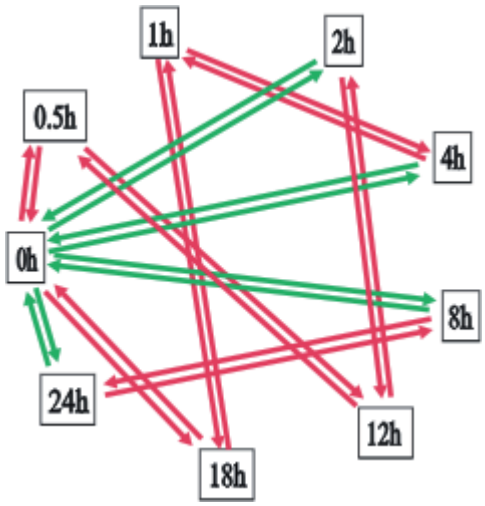


Figure 3.2: Two-step design of the microarray experiment used in this work; each arrow (1st phase: green, 2nd phase: red) represents an microarray experiment, its direction determines the color labeling

Besides the general concept of "who-with-whom" the question arises how many measurements have to be done to ensure a statistically meaningful result. The answer will be given in section 3.3. But since the minimal sample size depends on the used statistical method and there are several steps necessary before a statistical evaluation can be done.

3.2 Data Preprocessing and Normalization

Since Microarray experiments are quite expensive and especially for animal experiments the number of individuals should be kept as small as possible, the data preprocessing and analysis have to be very effective. Even an optimal design would not remove all sources of variances and thus technical variation has to be distinguished from the biological one and reduced by appropriate methods. Besides the Lowess-transformation method which will be presented below, there exist many techniques and variants for data normalization as e.g. the method of Workman et al. [WJJ+02] using smoothing B-splines on quantiles calculated from the complete experimental data. Other approaches are available in many articles and books as e.g. [YDLS01], [SS03], [BIAS03], [WBHW03], [Edw03], [SPT+04], [BHJ+04].

The main assumption for microarray normalization procedures is that the bulk of genes is transcribed at a relatively constant level which does not differ under the treatment, during the time course or between the compared tissues. Only a minority shows differences in their levels of gene expression.

A second often used assumption is the existence of so called housekeeping genes, which always belong to the constant transcribed genes. Their corresponding proteins are typically

involved in the general maintenance of the cells. Nevertheless, it is a matter of dispute if genes exist which possess the housekeeping property under all circumstances. Thus, the normalization methods presented in this section will only use the first assumption.

3.2.1 Microarray Background Subtraction

Now let us assume that a microarray hybridization was run and the fluorescence data of the hybridized chip was measured. The first step to reduce the influence of technical variation on the microarray measurements is the subtraction of the background. Due to different amount of labeled cDNA or an uneven distribution of the cDNA on the microarray slide, chips might show a spatial or overall bias in the two color channels (cf. figure 3.3).

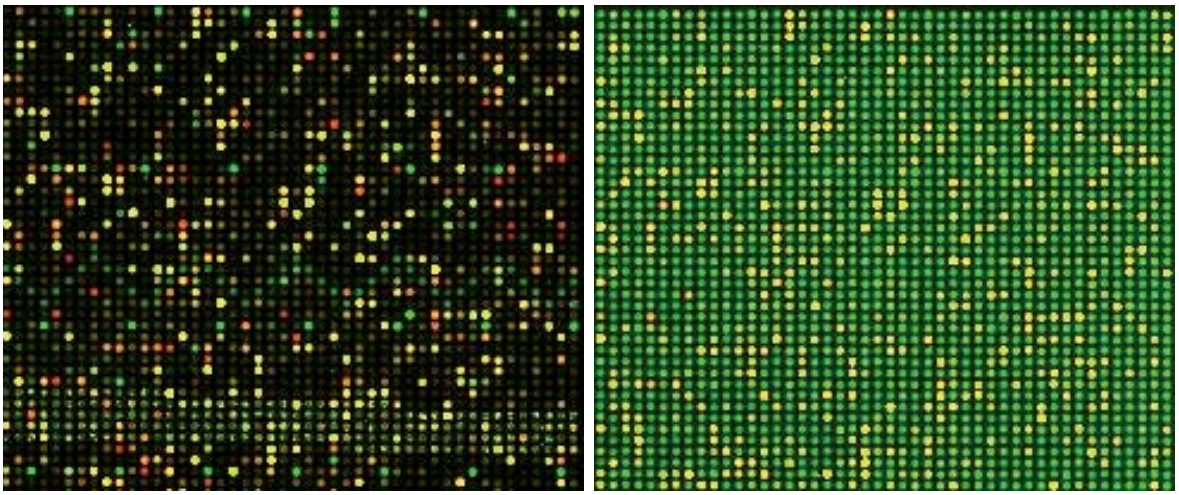


Figure 3.3: **Left:** Spatial bias - the lower part shows a band of locally greener (Cy3) background. **Right:** Systematic bias - the complete chip shows a much higher Cy3 concentration throughout

This bias can be reduced by subtracting the intensities of the areas around the spots from the measured spot brightnesses, separately for each color. In this connection literature distinguishes between three methods: overall, local or per-spot. But in fact, two are special cases of the third. The local background correction calculates the average background brightness in a window of predefined size around a spot. This average is subtracted from the brightness of that spot. The overall background correction uses the complete chip as window, whereas the per-spot correction sets the window size to a value small enough that it contains only the direct neighborhood of the current spot, but no other adjacent spots.

The average of the background is calculated via arithmetic mean, trimmed mean, or median.

Remark that the background subtraction causes that especially dark spots, not much brighter than the background become very noisy: While the difference of means is near to zero the variation of the two intensities add up resulting in a high relative variance. Thus, all spots

darker than a background-depending bound have to be removed for further analyses - their values are to uncertain.

Further description and examples of microarray background subtraction are available in [Bla03].

From now on assume that the background correction was done and thus the spatial errors are removed as far as possible which will allow to disregard the spatial information in the following chapters.

3.2.2 Between Chip Brightness Normalization

The background subtraction removed only the brightness differences caused by remaining unbound cDNA. Independent of that, microarrays differ in their spot brightness due to hybridization time and spot qualities. Dye-swaps reduce the color influences on the data of the test groups due to the balance principle, but they do not cancel it completely and thus the chip brightness has to be normalized. Because of the main assumption, that only a comparatively small amount of genes shows a different behavior in the experiments, each slide should have the same average brightness.

The control spots, which show very high or nearly no intensities, partly differing with respect to the two color channels, as described in section 2.3, have to be disregarded during the following normalization steps because they do not fulfill the assumption of equal distribution in both color channels (cf. figure 3.4).

So let $N \in \mathbb{N}$ denote the number of the remaining spots at a microarray chip, mappable to the genes of the corresponding organism. Further let R_i and G_i denote the background-corrected fluorescence intensity of the red (Cy5) respectively green dye (Cy3) at a given spot $i \in \{1,\ldots,N\}$ measured by the photo multiplier.

Now let

$$M_i = \log_2\left(\frac{R_i}{G_i}\right) = \log_2 R_i - \log_2 G_i \tag{3.2}$$

be the logarithmic ratio of the color channels at spot $i \in \{1,\ldots,N\}$, the so-called M-value, and

$$A_i = \log_2 \sqrt{R_i\, G_i} = \frac{1}{2}\left(\log_2 R_i + \log_2 G_i\right) \tag{3.3}$$

denote the logarithmic intensity average of spot $i \in \{1,\ldots,N\}$, the so called A-value.

Two typical MA-plots, plotting the M- and A-values of one microarray experiment, look as shown in figure 3.4.

Remark four obvious and very often occurring properties of microarray experiments:

1. The right plot of figure 3.4 shows a higher overall intensity (A-value 7.45 average, compared to 6.79 of the left one).

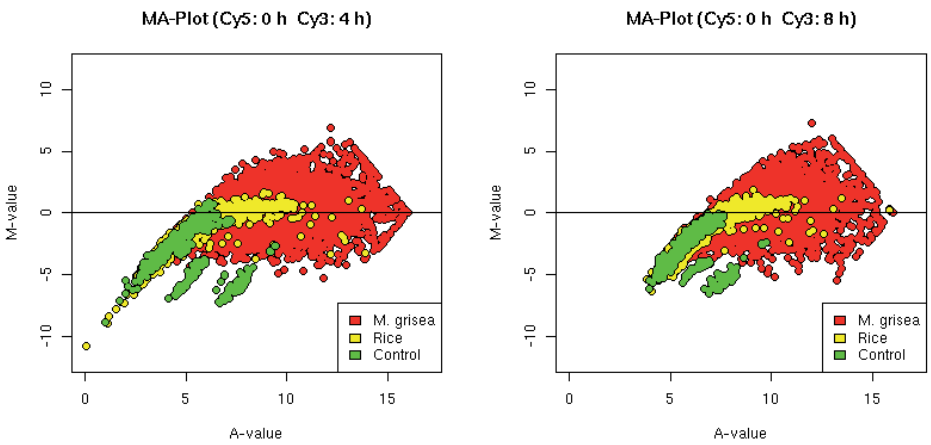


Figure 3.4: MA-Plots; magnaporthe conidia before (Cy5) and 4 resp. 8 hours after application to a culture medium (Cy3)

2. The scatter plot seems not to be evenly distributed around the log-ratio of 0, which means that one dye is stronger all in all (mean M-value -0.97 on the left plot, -1.12 on the right).

3. On the right side, the plots have a wedge-shaped boundary with 45° angles with respect to the line $M = 0$.

4. The plots have curvilinear shapes, showing that M-value at low intensities tends notedly towards the green color channel.

The differences in the color channels as well as the local artifacts of items 3 and 4 will be discussed in the next section. Prior to that, the global brightness differences between chips will be handled.

Due to the basic assumption, besides the in comparison few up- oder down-regulated genes there is no biological reason for a higher overall brightness in one color channel or in one duplicated microarray experiment. Differences have to be caused technically. Deviations in the general intensity of the fluorescence, a varying build-in rate of the dyes or differences in the amount of applied labeled cDNA of the two samples are possible reasons as argued in section 2.4. Also the brightness differences between microarrays are caused by chip errors and environmental influences as described in the place cited. Thus, these technical artifacts have to be removed by an adequate data normalization.

Therefore, let P be the permutation function on $\{1,\ldots,N\}$ such that the spot brightnesses are sorted in an ascending order:

$$A_{P(i)} \leq A_{P(i+1)} \quad \text{for all} \quad i \in \{1,\ldots,N-1\}\,. \tag{3.4}$$

The trimmed arithmetic mean value (TAM) of the set $A = \{A_i\}_{i \in \{1,\ldots,N\}}$ is given by the arithmetic mean, where the lowest and highest quantile of the scores are discarded. It is used instead of the standard arithmetic mean to discard outliers, be they artifacts or caused by strongly expressed genes.

Therefore let $q \in [0, 0.5)$ denote the size of the upper and lower quantile which shall be neglected and

$$IQR_q(\{1,\ldots,N\}) = \{i \in \{1,\ldots,N\} \quad | \quad i > qN \wedge i < (1-q)N\} \tag{3.5}$$

be the corresponding interquantile range. Further let

$$\tilde{N} = \left|IQR_q(\{1,\ldots,N\})\right| = N - \lceil qN \rceil - \lfloor qN \rfloor \tag{3.6}$$

denote the size of the interquantile range.

This yields the trimmed arithmetic mean

$$TAM_q(A) = \frac{1}{\tilde{N}} \sum_{i \in IQR} A_{P(i)} \tag{3.7}$$

The trimmed arithmetic mean can be used to normalize the brightness of the chip with respect to other microarrays hybridized with cDNA coming from the same experiment setup. Therefore, let the scaled A-value be given by

$$A_i^{shifted} = A_i - TAM_q(A) + avTAM_q \tag{3.8}$$

where $avTAM_q$ denotes the average of the trimmed means of all microarrays within the experiment set.

3.2.3 Within Chip Normalization by Lowess

Obviously, the point cloud in the MA-plot showing M_i vs. $A_i^{shifted}$ is the same as the original one, just moved along the abscissa. The ratio offset as well as curvilinear shape still exist. As expected this is an artifact, which can be proven easily by a so called dye-swap:
The probes are technically replicated and dye-labeled inversely to the original experiment. If the banana-shape would be caused by the gene expression, it should now bend into the other direction. But in fact the direction is again the same, which reasons that this shape is color-based (cf. figure 3.5).

Here again, the main assumption can be applied. The majority of spots has to be distributed equally around the the log-ratio of 0. This can be achieved by the point-by-point subtraction of a regression curve which fits to the central axis of the banana shape.

An appropriate regression curve can be calculated by the locally weighted polynomial regression method “Lowess” (LOcally WEighted Scatterplot Smoothing). This method calculates a least square regression polynomial of a low degree ∂ (almost always $\partial = 1$ or $\partial = 2$) in a local sliding window of the data. Higher degree polynomials tend to overfitting.

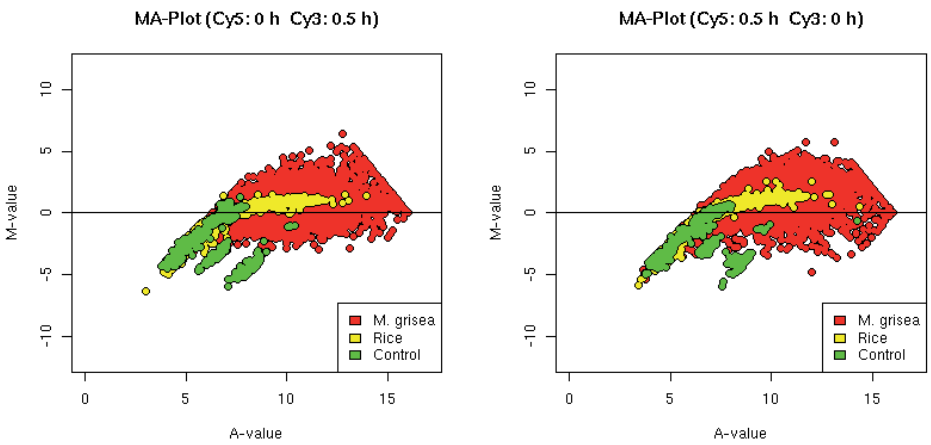


Figure 3.5: MA-Plots of a dye-swap: Magnaporthe grisea conidia before and 0.5 hours after application to the culture medium

For each spot in the scatter plot the $n = \lceil fN \rceil$ nearest neighbors along the A-axis are used for the estimation. Here, $f \in \left(\frac{\partial+1}{N}, 1\right)$ denotes the so called smoothing parameter. The lower bound is needed to ensure the minimal number of data points to calculate the regression polynomial. f is typically set to a value between 0.2 and 0.5, which lies surely in the interval of allowed values, since normally thousands of gene spots are situated on microarrays. [Bla03], [NIS09]

Let the sliding window

$$\mathbb{W}_n(k) \subset \{1, \ldots, N\} \tag{3.9}$$

be the index set of the n data spots, which are nearest to the actually considered spot k with respect to the A-axis.

In addition to the selection of a local window, the influence of the M-values of the contained data spots to the regression is weighted by a function of their A-value distance to the actually considered spot. Therefore, let

$$D_k = \max\left\{ \left| A_i^{shifted} - A_k^{shifted} \right| \;\middle|\; i \in \mathbb{W}_k \right\} \tag{3.10}$$

denote the maximal possible distance in the window.

The traditionally selected function is the tri-cube weight function is given by

$$w_k : \begin{cases} \mathbb{W}_n(k) & \longrightarrow [0,1] \\ i & \longmapsto \left[1 - \left(\frac{\left| A_i^{shifted} - A_k^{shifted} \right|}{D_k} \right)^3 \right]^3 \end{cases} \tag{3.11}$$

With this weightening the following error function has to be minimized for every spot k:

$$E_k(a_1,\ldots,a_\partial) = \sum_{i\in\mathbb{W}_n(k)} w_k(i)\left[M_i - p_k\left(A_i^{shifted};a_{k1},\ldots,a_{k\partial}\right)\right]^2 \tag{3.12}$$

where p is a polynomial of degree ∂ and $a_{k1},\ldots,a_{k\partial}$ its coefficients.

Now define the value of the global regression curve F at point $A_k^{shifted}$ by

$$F\left(A_k^{shifted}\right) = p_k\left(A_k^{shifted};a_{k1},\ldots,a_{k\partial}\right) \tag{3.13}$$

Doing this for each data spot yields the function values of the regression curve F at $A_i^{shifted}$ for all $i\in\{1,\ldots,N\}$, which allows the normalization of all data spots by

$$M_k^{lowess} = M_k - F\left(A_k^{shifted}\right) \quad \text{for all } k\in\{1,\ldots,N\} \tag{3.14}$$

and reduces the brightness-dependence on the intensity ratio (cf. figure 3.6).

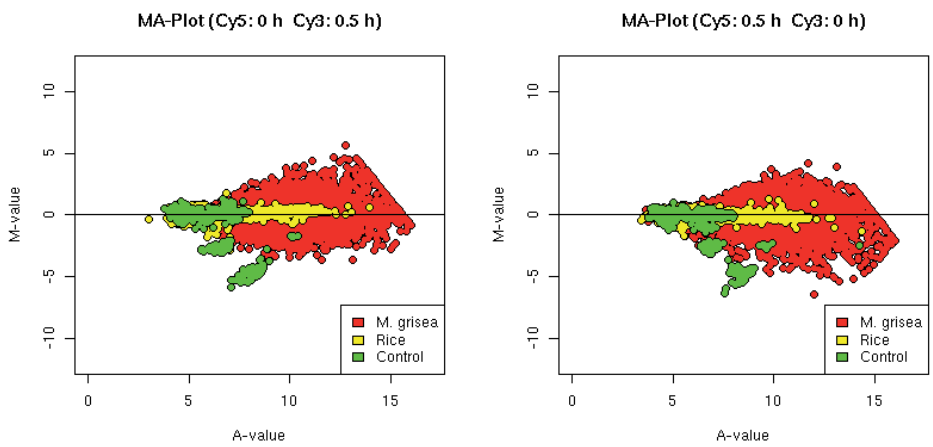


Figure 3.6: MA-Plot of the Lowess-transformed data of figure 3.5

The brightness-ratio-data is back-transformed to absolute values in the two color channels by

$$\begin{aligned} R_k^{norm} &= A_k^{shifted} + \frac{M_k^{lowess}}{2} \\ G_k^{norm} &= A_k^{shifted} - \frac{M_k^{lowess}}{2} \end{aligned} \tag{3.15}$$

keeping the logarithmic scale for further computations presented in the next sections.

Now, the dye dependence of the measured expression levels at one chip is reduced as far as possible. Thus, the dye information can be discarded and the measurements are treated equally, no matter which label they had (cf. section 3.4).

One technical artifact type remains: the wedge-shape on the right side of the MA-plots. This is due to the bounded maximal brightness detectable by the scanner. Along the straight boundary one color channel reached that maximum. In the half-plane $M > 0$, this is the red channel, for $M < 0$ the green channel is maximal. However, this error is not removable, since there is no possibility to determine the true value of a spot with maximal brightness in one or both channels. Thus, there are two ways to deal with this artifact: Either remove all data with at least one color channel exhibiting the maximal value or "ignore" the problem. In this work the latter solution is chosen. The limiting of a high expression level in one channel causes that the color ratio is less extreme than the underlying mRNA ratio. This means, taking this artificially disturbed data into account during the statistical analysis would only increase the number of false negatives, namely whenever the reduced ratio yields a non-significant difference in gene expression. In contrast, removing the data before applying the statistical test results in a too low sample size for several genes for detecting expression differences at all. Thus, the slight decrease of statistical power for genes exhibiting maximal expression levels has to be accepted.

3.3 Statistical Analysis and Design of Experiments (cont.)

Statistical testing methods are multitudinous and the selection of the one test to be used seems often to be some kind of educated guess as well as a little bit luck. A vast overview of the most common testing scenarios and suitable statistical tests is given in [BLB00] or [NIS09].

However, for the selection of an appropriate statistical testing procedure and later on for the design of the experiments, there are some crucial points which shall be taken into account:

- First, in spite of the data preprocessing, microarray data seems not to be normally distributed, neither in the linear nor in the logarithmic scale. Here again, the microarray community is not agreed and many microarray experiments are evaluated using parametric hypothesis tests like the two-sample t-test, which does not keep the desired significance level if the assumption of normal distribution is not fulfilled. Due to the very low number of measurements a test for normally distributed data is insignificant as well. Therefore, a nonparametric statistical test, which is independent of underlying distributions, has to be chosen for the detection of differentially expressed genes.

- Because of the high costs of microarrays, the size of the sample sets usable for statistical analysis is quite low, which allows the selection of computational highly expensive randomization tests.

- From the biological point of view a change in gene expression is not noteworthy until it rises beyond twice the level as normal or falls below half the level. This is a rule

of thumb. Other organisms and treatments might cause thresholds different from two. Nevertheless, all genes whose fold-change in expression do not exceed this specific factor will be neglected in the further statistical analysis.

- Finally, for a gene-wise hypothesis testing a α-risk of 5% holds for most needs, since microarrays are often used as a first whole-genome search for candidate genes, whose expression levels are then examined separately by other, more exact and much cheaper methods as the so-called *qualitative Real Time Polymerase Chain Reaction* (qRT-PCR). This second examination method is additionally recommended, because the high number of gene spots at a full-genome microarray either increases the number of false positives to an unacceptable high value or raises the needed significance level by a Bonferroni correction method. The later one again demands much more microarray experiments to ensure the detection of differentially expressed genes at all. Thus, from now on, lets only consider a hypothesis testing for each gene separately, being aware of the fact, that this would also yield many false positives under the tens of thousands genes for (nearly) sure. More about Bonferroni correction and adjustment of significance levels can be found in [Bla03].

3.3.1 Fisher-Pitman-Test

The nonparametric Fisher-Pitman Test detects differences in mean values of two independent sample sets fulfilling the demands in the outset. It is based on resampling of the data which shall be examined and thus closely related to bootstrapping methods like the one presented in section 4.1. The underlying theory was developed by Sir Ronald A. Fisher, who was already quoted in this chapter and the Australian mathematician Edwin J. G. Pitman, in the 1930s.

The test goes as follows:
Let $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$ be two independent samples, in our case microarray data sets for two different treatments or tissues which shall be compared. Let $\overline{X}$ respectively $\overline{Y}$ denote the corresponding arithmetic means.

Furthermore, let $D = \overline{X} - \overline{Y}$ be the difference of the mean values. For simplicity and without loss of generality, let $D \geq 0$, else exchange the sets X and Y.

Now assume, as null hypothesis H_0, both sets X and Y were drawn at once from the same distribution. Then the question arises, how probable is a split into sets of size m and n that that a mean value difference greater or equal as D arises. If only relatively few possibilities exist, the null hypothesis can be refuted and the sets X and Y are assumed to be samples from distributions with different means, i.e. the gene expression levels differ.

More precise, calculate the number of dichotomies of $X \cup Y$ into sets of size m and n which causes a difference of at least D and compare it to the overall number of possible dichotomies.

From the computational point of view, its faster, if the sum of one sample set $S_x = \sum_{i=1}^{m} x_i$ is used as test statistic instead of D. Both are obviously linked by the linear equation

$$D = \frac{m+n}{mn} S_x - \frac{1}{n} S \tag{3.16}$$

where S denotes the sum of all values in X and Y.

Thus, the probability P for choosing m elements out of $X \cup Y$ with a sum of at least S_x has to be calculated. Since the sample set sizes are quite small counting the number of these possibilities, call it k, is not computational involving at all. The probability for selecting one of those extreme dichotomies is consequently

$$P = \frac{k}{\binom{m+n}{m}} \tag{3.17}$$

If P is not greater than the desired α-risk the null hypothesis H_0, no existing difference in the tested gene between both treatment types, is rejected. The gene is detected as over-expressed in the first sample.

An algorithm for calculating the probability P is given in appendix B.

Because of the dichotomy (X,Y), which obviously fulfills the demands, k is always at least one and thus the probability is bound from below by

$$P \geq P_- = \frac{1}{\binom{m+n}{m}} \tag{3.18}$$

Hence there is a need for $P_- \leq \alpha$ to have a chance to detect anything at all, which yields a lower bound for the sample sizes m and n.

This leads to the minimal design of experiments as calculated in the next chapter.

3.3.2 Minimal Sample Sizes

In this section a way to calculate the minimal sample size for a microarray common reference design is presented. Treat a loop design as a common reference design with only one treatment group. Then the resulting minimal sample size holds obviously for all test groups in a loop or some mixed design.

For a common reference design, let the α-risk, a control population C and k treatment populations T_i, $i \in \{1,\ldots,k\}$ be given. Find the minimal number of microarrays experiments such that differential expressed genes are detectable under optimal, noise-free conditions, i.e.:

$$\binom{c+t_i}{t_i} \geq \frac{1}{\alpha} \tag{3.19}$$

where the control population is hybridized c times and the treatment population T_i is hybridized t_i times. Obviously, due to the constant α and c, $t = t_i$ is constant for all $i \in \{1,\ldots,k\}$, too.

Thus, the optimization problem is given by

$$\begin{aligned} \text{Minimize} \quad & c + kt \\ \text{subject to} \quad & c, t \in \mathbb{N} \\ & \binom{c+t}{t} \geq \frac{1}{\alpha} \end{aligned} \tag{3.20}$$

However, for α-risks about 1-5% the numbers are quite small, such that simply trying all possible combinations gives the solution even faster than highly involved algorithms.

For $\alpha = 1$, 2 and 5% table 3.1 shows the necessary number of hybridizations under optimal conditions. Nevertheless, it is recommended to hybridize each population equally often in both color channels to decrease the influence of the dyes further. This allows only even numbers for the optimization problem, which are given in parentheses whenever the optimization yields odd numbers. In this case c and t stay constant also for k larger than 8. Keep in mind that the numbers specify the minimal amount of chips to detect anything at all and are calculated for optimal conditions. However, in the majority of cases only some of the differential expressed genes are sought-after, with no special weighting. Therefore, these low numbers of experiments yield first statistical results which can be enhanced by hybridizing additional microarrays containing the relevant probes or by running more precise gene-specific methods as qRT-PCR.

3.4 Results

In this chapter the raw data taken from the microarrays was normalized and the error influences were reduced. MA-plots of all microarray data, both in raw format and normalized, are available in appendix A. The smoothing parameter f was set to 0.2.

As already said, the dye information can be neglected after this chapter. For further calculations collect the data of all measurement of each gene under a certain treatment in a set, no matter on which microarray and with which dye-label it was hybridized.

Therefore, put the values of R_g^{norm} respectively G_g^{norm} from (3.15) into the set $M_{g,t}$ if the red respectively green channel of the microarray was made from the genetic sample under treatment t.

In the following these sets will be denoted by

$$M_{g,t} = \left\{ m_{g,t}^{(1)}, \ldots, m_{g,t}^{(n_{g,t})} \right\} \tag{3.21}$$

where $n_{g,t} \in \mathbb{N}$ is the number of measurements of gene g under treatment t taken from a treatment set $T = \{t_1, \ldots, t_\tau\}$ ($\tau \in \mathbb{N}$). The genes are numbered serially $g \in \{1, \ldots, \Gamma\}$ with $\Gamma \in \mathbb{N}$.

Since this work is based on a microarray time course experiment of the growth of Magnaporthe grisea, the treatments are the post-inoculation (p.i.) times, i.e. the hours passed

	α-risk		
k	1%	2%	5%
1	$c = 5\,(6)$ $t = 5\,(4)$	$c = 4$ $t = 4$	$c = 3\,(4)$ $t = 3\,(4)$
2	$c = 6$ $t = 4$	$c = 4$ $t = 4$	$c = 4\,(6)$ $t = 3\,(2)$
3	$c = 7\,(6)$ $t = 3\,(4)$	$c = 5\,(4)$ $t = 3\,(4)$	$c = 3\,(6)$ $t = 3\,(2)$
4	$c = 8\,(6)$ $t = 3\,(4)$	$c = 6\,(10)$ $t = 3\,(2)$	$c = 6$ $t = 2$
5	$c = 7\,(14)$ $t = 3\,(2)$	$c = 5\,(10)$ $t = 3\,(2)$	$c = 6$ $t = 2$
6	$c = 8\,(14)$ $t = 3\,(2)$	$c = 10$ $t = 2$	$c = 6$ $t = 2$
7	$c = 7\,(14)$ $t = 3\,(2)$	$c = 10$ $t = 2$	$c = 6$ $t = 2$
8	$c = 14$ $t = 2$	$c = 10$ $t = 2$	$c = 6$ $t = 2$

Table 3.1: Necessary number of microarray hybridizations of control (c) and treated (t) samples

between application of the spores to the nutrient medium and the extraction of its mRNA. Thus, the treatment set or time set is given by

$$T = \{0, 0.5, 1, 2, 4, 8, 12, 18, 24\} \tag{3.22}$$

For the usage in indexed sums the elements of $t_i \in T$, $i \in \{1, \ldots, 9\}$, are sorted in ascending order, i.e. $t_i < t_{i+1}$.

The design of experiments was set up such that each p.i. period besides 0 was hybridized 4 times, i.e. $n_{g,t_i} = 4$ for all $i \in \{2, \ldots, 9\}$. The dormant spores were hybridized 12 times, i.e. $n_{g,t_1} = 12$.

The used microarray chips have 15170 data spots with Magnaporthe grisea genes, 6325 spots with rice genes, and 1080 control spots.

For significance analysis only data spots g were considered, which exhibit a fold change of at least 2 at any time t_i, $i \in \{2, \ldots, 9\}$, with respect to the dormant spore t_1, i.e.

$$\left|\log_2\left(\overline{M_{g,t_i}} - \overline{M_{g,t_1}}\right)\right| \geq 1 \tag{3.23}$$

where $\overline{M_{g,t}}$ denotes the arithmetic mean of set $M_{g,t}$.

The expression levels of all these data spots were compared using the Fisher-Pitman-Test with $\alpha = 5\%$. More precise, the microarray data of the growing fungus and the dormant

t_i	Differentially expressed genes					
	Magnaporthe		Rice		Control	
0.5	up:	1550	up:	1	up:	0
	down:	760	down:	6	down:	1
1	up:	1497	up:	23	up:	0
	down:	1048	down:	16	down:	2
2	up:	1672	up:	14	up:	0
	down:	1200	down:	82	down:	1
4	up:	1498	up:	18	up:	1
	down:	1653	down:	76	down:	3
8	up:	1193	up:	10	up:	0
	down:	1936	down:	23	down:	1
12	up:	2222	up:	10	up:	0
	down:	1865	down:	942	down:	3
18	up:	2018	up:	157	up:	0
	down:	1551	down:	10	down:	1
24	up:	1670	up:	45	up:	0
	down:	2067	down:	9	down:	1

Table 3.2: Significantly differentially expressed genes

spores was tested for differences. Therefore, M_{g,t_1} was compared to M_{g,t_i} for all $i \in \{2,\ldots,9\}$, which resulted in detecting differential gene expressions as given in table 3.2. A gene significantly differentially expressed at time t_i is said to be up-regulated, if $\overline{M_{g,t_i}} > \overline{M_{g,t_1}}$, and down-regulated, if $\overline{M_{g,t_i}} < \overline{M_{g,t_1}}$.

This results in the detection of a differential gene expression in 7174 Magnaporthe grisea genes, 1171 rice genes, and at 7 control spots.

For further calculations all genes with no detected differential expression levels at any time were discarded, which results in $\Gamma = 8352$.

4 Interpolation of Time Course Data

Forever is composed of nows.

— Emily Dickinson, American poet (1830-1886)

In this section the interpolation of time-discrete expression level measurements of a single gene g is presented. Therefore, let a time course microarray experiment be run and the data be normalized using the methods of section 3.2, which results in data sets $M_{g,t}$ from (3.21) with time t from the time set T of equation (3.22). Figure 4.1 shows the measurements of an exemplary gene time courses.

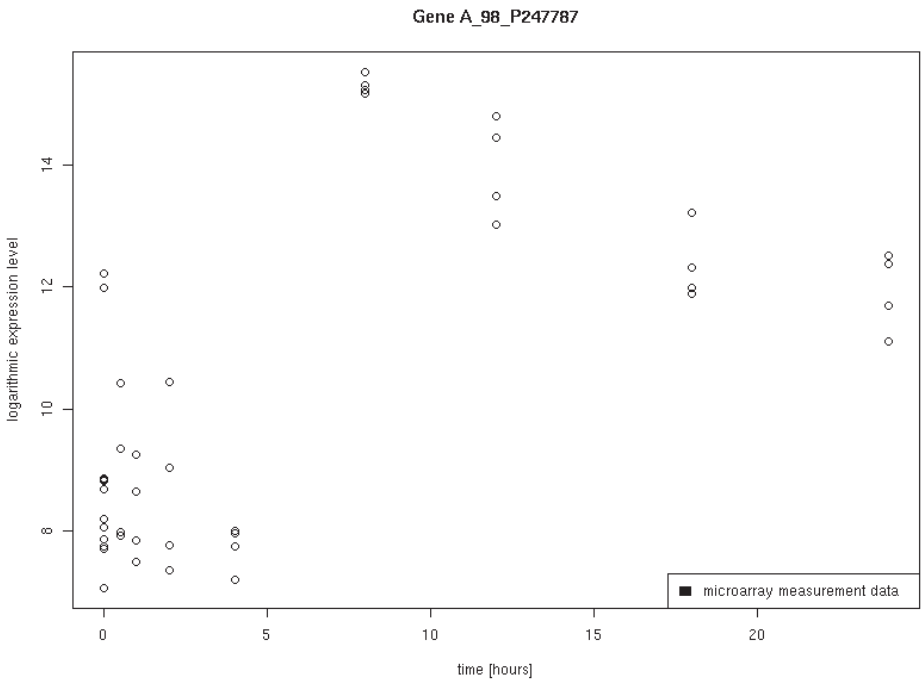


Figure 4.1: Measurements of an exemplary gene time course

Due to the low number of measurements at a specific time and the consequential sensitivity to measurement outliers the arithmetic mean is no meaningful estimator for the gene expression level. Instead the median is recommended. Let therefore

$$y_i = \text{med}\left(M_{g,t_i}\right) \tag{4.1}$$

denote the expression level at time t_i estimated by the median.

This results in a interpolation set

$$\{(t_1, y_1), \ldots, (t_\tau, y_\tau)\} \tag{4.2}$$

with $t_i < t_{i+1}$ for all $i \in \{1, \ldots, \tau - 1\}$.

There are many possibilities to interpolate these data points, starting with a simple connecting polygon up to more complex structures as wavelets. Since no further information of the true gene expression pathway is available, there is no right or even unique way for interpolation. However, in this chapter the concept of smoothing splines will be presented as interpolation method for microarray time course data and its advantages will be stated. Besides the usage of smoothing splines for time course interpolation presented below, they were also successfully used in several other tasks of microarray analysis as normalization [WJJ+02] or clustering [MCZL06].

In contrast to normal splines, smoothing splines do not force the interpolation curve exactly through the data points. Instead the curve only approaches the points depending on their quality.

This "quality" will be calculated in the first section of this chapter. While the common quality measure for the arithmetic mean of a set is its standard deviation, the median does not have a corresponding accuracy measure. Motivated by bootstrapping, a stochastic method able to estimate the accuracy of a median, an alternative method will be deduced. Due to its origin and properties the oxymoron *exact bootstrapping* is chosen as name for this method.

4.1 Accuracy Estimation by Bootstrapping

Besides the estimation of the gene expression level a measure of its accuracy is needed. While there exists a common and easily computable estimator for the standard deviation of the arithmetic mean, there is no such formula for an accuracy measure of the median. This section presents shortly the general bootstrap approach for accuracy estimations of statistics, which will be used afterwards for the median.

Bootstrapping is a statistical method based on a repeated calculation of statistics based on a sample to estimate this statistic of the underlying unknown population. Bradley Efron presented it for the first time in 1979. However, the basic idea behind the bootstrap goes back at least two centuries. While the high number of recalculations was obviously problematic at that time, the increased computing power nowadays allows the extensive usage method in few minutes.

A detailed introduction into bootstrapping methods is given in [ET93].

Now the accuracy of a statistic based on a sample set shall be estimated by bootstrapping. Let therefore $n \in \mathbb{N}$,

$$X^{(n)} = (x_1, \ldots, x_n) \in \mathbb{X}^n \tag{4.3}$$

be the sample of a population set $\mathbb{X}$, and

$$S : \mathbb{X}^n \to \mathbb{R} \tag{4.4}$$

the statistic, whose accuracy shall be estimated.

A corresponding bootstrap sample $B^{(n)} = (b_1, \ldots, b_n) \in \mathbb{X}^n$ is generated by sampling n times randomly from X with replacement. The statistic $S\left(B^{(n)}\right)$ applied to the bootstrap sample is the bootstrap replicate of S.

This procedure is repeated obtaining a set of bootstrap samples $B_1^{(n)}, \ldots, B_m^{(n)}$ of $m \in \mathbb{N}$ and its corresponding bootstrap replicates $S\left(B_i^{(n)}\right)$ for $i \in \{1, \ldots, m\}$. An usual choice of m is between 50 to 200. [ET93]

Now estimate the accuracy of the statistic by the standard deviation w of the bootstrap replicates:

$$w(S,m) = \sqrt{\frac{1}{m-1} \sum_{i=1}^{m} \left[S\left(B_i^{(n)}\right) - S\left(X^{(n)}\right) \right]^2} \tag{4.5}$$

This yields an estimation of root mean squared error (abbr.: rmse) $w(S,m)$ for any statistic S.

4.1.1 Exact Bootstrapping

The bootstrapping method is an obvious and straight forward approach to solve the problem concerning the accuracy of the median. However, one should not forget two things: Firstly microarray measurements often are only few. Secondly, the median depends only on one (in the case of an odd sample size) or two (if the sample size is even) central values of the sample. Both properties can be used to calculate the average exactly by considering each possible bootstrap replicate. [GPSB84]

While there are n^n possibilities for drawing n times out of n values with replacement, this number can be dramatically reduced if $\mathbb{X}$ is ordered and the statistic S is order-independent, i.e.

$$S\left(B^{(n)}\right) = S\left(\Pi\left(B^{(n)}\right)\right) \tag{4.6}$$

for any bootstrap sample $B^{(n)}$ and any permutation function $\Pi : \mathbb{X}^n \to \mathbb{X}^n$.

For example the accuracy of the median of a sample set of size 4 can be calculated exactly with 35 steps instead of the recommended lower bound of 50 bootstrap replicates, which will be shown below.

Lemma 4.1:
Let $\mathbb{X}$ be a non-empty set and $X^{(m)} = (x_1, x_2, \ldots, x_m) \in \mathbb{X}^m$. Then there exist

$$P_{n,m} = \sum_{a_n=1}^{m} \sum_{a_{n-1}=1}^{a_n} \cdots \sum_{a_1=1}^{a_2} 1 \tag{4.7}$$

different samples $B^{(n)} = (b_1, \ldots, b_n) \in \mathbb{X}^n$ of $X^{(m)}$, drawn with replacements, fulfilling the order condition for all $i \in \{1, \ldots, n\}$ and $k \in \{1, \ldots, m\}$

$$b_i = x_k \quad \Rightarrow \quad b_j \in \{x_1, \ldots, x_k\} \text{ for all } j \in \{1, \ldots, i\} \tag{4.8}$$

The lemma especially holds for the bootstrapping case $m = n$.

Proof. by induction over sample size n.
Basis ($n = 1$):
There are obviously

$$P_{1,m} = \sum_{a_1=1}^{m} 1 = m \tag{4.9}$$

different possibilities to chose one out of m values.
Inductive step ($n \to n+1$):
Let $b_{n+1} = x_{a_{n+1}}$ for an $a_{n+1} \in \{1, \ldots, m\}$. Because of (4.8) holds that $b_n, b_{n-1}, \ldots, b_1$ can only be chosen out of a_{n+1} possible values $(x_1, x_2, \ldots, x_{a_{n+1}})$. In that case, there exist

$$P_{n,a_{n+1}} = \sum_{a_n=1}^{a_{n+1}} \sum_{a_{n-1}=1}^{a_n} \cdots \sum_{a_1=1}^{a_2} 1 \tag{4.10}$$

possibilities fulfilling the order condition (4.8), due to the induction hypothesis.
Adding up all possibilities for a_{n+1} yields the desired formula

$$\begin{aligned} P_{n+1,m} &= \sum_{a_{n+1}=1}^{m} P_{n,a_{n+1}} \\ &= \sum_{a_{n+1}=1}^{m} \sum_{a_n=1}^{a_{n+1}} \cdots \sum_{a_1=1}^{a_2} 1 \end{aligned} \tag{4.11}$$

which finalizes the proof. □

While the lemma above shows the origin of the equation quite demonstratively and can be immediately used for deriving a corresponding algorithm for generating all these possibilities, it is is very unhandy for the calculation. However, on can observe that

- $P_{1,m} = \binom{m}{1}$ is nothing else than m
- $P_{2,m} = \binom{m+1}{2}$ is the sum of the first m natural numbers (m-th triangular number, 2-simplex number)

- $P_{3,m} = \binom{m+2}{3}$ is the sum of the first m triangular numbers (m-th tetrahedral number, 3-simplex number)
- $P_{4,m} = \binom{m+3}{4}$ is the sum of the first m tetrahedral numbers (m-th 4-simplex number)
- ...

This allows a faster calculation of $P_{n,m}$ as the following lemma shows.

Lemma 4.2: ***Number of Ordered Samples***
Let $P_{n,m}$ be defined as in (4.7), then the following holds

$$P_{n,m} = \binom{m+n-1}{n} \tag{4.12}$$

Proof. It is already known from the proof of lemma 4.1 that

$$P_{n+1,m} = \sum_{i=1}^{m} P_{n,i} \tag{4.13}$$

thus the statement of the lemma is proven, if

$$\binom{m+(n+1)-1}{n+1} = \sum_{i=1}^{m} \binom{i+n-1}{n} \tag{4.14}$$

holds.

This will be shown by induction over m:
Basis ($m = 1$):

$$\binom{1+(n+1)-1}{n+1} = 1 = \sum_{i=1}^{1} \binom{i+n-1}{n} \tag{4.15}$$

Inductive step ($m \to m+1$):
The right hand side of (4.14) becomes

$$\begin{aligned} \sum_{i=1}^{m+1} \binom{i+n-1}{n} &= \binom{m+n}{n} + \sum_{i=1}^{m} \binom{i+n-1}{n} \\ &= \binom{m+n}{n} + \binom{m+n}{n+1} \\ &= \binom{m+n+1}{n+1} \end{aligned} \tag{4.16}$$

where the penultimate equality is due to the induction hypothesis and the last one is Pascal's equality. □

Knowing the number of ordered possibilities of bootstrap samples, the question arises how many unordered versions of each of them exist.

More precise: Let $\mathbb{X}$, $X^{(m)}$ be given as in lemma 4.1. Let $B^{(m)} = (b_1, \ldots, b_m)$ be a bootstrap sample of $X^{(m)}$ fulfilling the order condition 4.8. Find $N\left(B^{(m)}\right)$ given by

$$N\left(B^{(m)}\right) = \left|\left\{\Pi\left(B^{(m)}\right) | \Pi \text{ is a permutation on } \mathbb{X}^m\right\}\right| \tag{4.17}$$

This can be easily calculated by an combinatorial consideration:

Lemma 4.3:
Let $X = (x_1, x_2, \ldots, x_m)$ and its bootstrap sample be $B^{(m)}$. Then

$$N\left(B^{(m)}\right) = \frac{m!}{\prod_{i=1}^{m} \nu_B(x_i)} \tag{4.18}$$

where $\nu_B(x)$ denotes the multiplicity of x in B.

Together with lemma 4.1 there is an *exact bootstrap* estimator for the accuracy of the statistic S by adding up the accuracy of each possible bootstrap replicate:

Theorem 4.1: ***Exact Bootstrapping***
Let $X^{(m)} = (x_1, x_2, \ldots, x_m) \in \mathbb{X}^m$. Then the accuracy of the statistic $S : \mathbb{X}^m \to \mathbb{R}$ can be calculated by

$$w\left(S, X^{(m)}\right) = \sum_{a_m=1}^{m} \sum_{a_{m-1}=1}^{a_m} \cdots \sum_{a_1=1}^{a_2} N\left((x_{a_1}, x_{a_2}, \ldots, x_{a_m})\right) \cdot w\left(S, X^{(m)}, (x_{a_1}, x_{a_2}, \ldots, x_{a_m})\right) \tag{4.19}$$

where

$$w\left(S, X^{(m)}, B^{(m)}\right) = \sqrt{\frac{1}{m-1} \sum_{i=1}^{m} \left[S\left(B^{(m)}\right) - S\left(X^{(m)}\right)\right]^2} \tag{4.20}$$

Proof. Construction using lemma 4.1 and lemma 4.3 □

While this result holds for any order-independent statistic the median allows for further reduction of calculation steps. Remark that the median depends only on the middle value (if the length of the data vector is odd) or the two middle values (if the length is even). The exact values of the other vector components do not matter, it is only important that one half is less, the others greater than the middle values. This fact can be used as the theorem below shows:

Theorem 4.2: ***Exact Bootstrapping for the Median of a Sample of Even-Numbered Size***
Let $m \in 2\mathbb{N}$, $X^{(m)} = (x_1, x_2, \ldots, x_m) \in \mathbb{R}^m$ sorted in ascending order and $v = \frac{m}{2} - 1$. Then the accuracy of the median can be calculated by

$$w\left(X^{(m)}\right) = \sqrt{\frac{1}{m^m - 1} \sum_{b=1}^{m} \sum_{c=b}^{m} P(b,c) \left(\operatorname{med}\left(X^{(m)}\right) - \frac{x_b + x_c}{2}\right)^2} \tag{4.21}$$

where

$$P(b,c) = \sum_{i=0}^{v} \sum_{j=0}^{v} (b-1)^{v-i} (m-c)^{v-j} \binom{m}{v-i} \binom{v+i+2}{v-j} \binom{i+j+2}{\left(1-\delta_{b,c}\right)(i+1)} \tag{4.22}$$

is the number of all different bootstrap samples of $X^{(m)}$ having b and c as middle values. (For convenience, in that equation let $0^0 = 1$.)

Proof. If (4.22) is proven, the equation (4.21) becomes quite obvious.

Therefore, assume firstly the indices b and c of the two middle values of a sample of $X^{(m)}$ with $b \leq c$. Then there are v indices of the sample less than or equal to b, call them $a_1, \ldots, a_v$, and v indices greater than or equal to c, call them $d_1, \ldots, d_v$ (since the data set was sorted).

Now have a look onto the vector $(a_1, \ldots, a_v, b)$:
Let $B = \{\beta | a_\beta = b\}$ and $\neg B = \{1, \ldots, v\} \setminus B$. The vector $(a_\beta)_{\beta \in \neg B}$ with $a_\beta \in \{1, \ldots, b-1\}$ can take

$$N_{|B|}^{\text{val}} = (b-1)^{v-|B|} \tag{4.23}$$

different values.

The same argument for $(c, d_1, \ldots, d_v)$ and $C = \{\gamma | d_\gamma = c\}$ results in

$$N_{|C|}^{\text{val}} = (n-c)^{v-|C|} \tag{4.24}$$

different possibilities for the vector $(d_\gamma)_{\gamma \in \neg C}$ with $d_\gamma \in \{c, \ldots, n\}$.

In the case of $b < c$ this yields four vectors whose components can be distributed to the $n = 2v+2$ components of the complete sample vector:

i) $v - |B|$ values less than b,

ii) $1 + |B|$ values equal to b,

iii) $1 + |C|$ values equal to c, and

iv) $v - |C|$ values greater than c.

Thus, there are

$$N^{\text{dist}} = \binom{2v+2}{v-|B|} \binom{v+|B|+2}{v-|C|} \binom{|B|+|C|+2}{1+|B|} \tag{4.25}$$

different possibilities for that distribution.

In the case $b = c$ there are three vectors

i) $v - |B|$ values less than b,

ii) $2 + |B| + |C|$ values equal to $b = c$,

iii) $v - |C|$ values greater than c.

and therefore

$$N^{\text{dist}} = \binom{2v+2}{v-|B|} \binom{v+|B|+2}{v-|C|} \tag{4.26}$$

different possibilities for that distribution.

The two equations (4.25) and (4.25) can be combined using the fact that $\binom{n}{0} = 1$ and the Kronecker symbol

$$N^{\text{dist}} = \binom{2v+2}{v-|B|} \binom{v+|B|+2}{v-|C|} \binom{|B|+|C|+2}{(1-\delta_{b,c})(1+|B|)} \tag{4.27}$$

The product of the number of possible values and its distribution of the sample vector yields the number of all possible sample vectors with

$$N_{|B|,|C|} = N^{\text{val}}_{|B|} N^{\text{val}}_{|C|} N^{\text{dist}} \tag{4.28}$$

Finally, for getting all sample possibilities with middle values b and c with $b \leq c$ all possible numbers for $i = |B|$ and $j = |C|$ have to be added up

$$P(b,c) = \sum_{i=0}^{v} \sum_{j=0}^{v} (b-1)^{v-i} (m-c)^{v-j} \binom{m}{v-i} \binom{v+i+2}{v-j} \binom{i+j+2}{\left(1-\delta_{b,c}\right)(i+1)} \tag{4.29}$$

which is exactly equation (4.22) in the theorem. □

Corollary 4.1: ***Exact Bootstrapping for the Median of a Sample of Odd-Numbered Size***
Let $m \in 2\mathbb{N}-1$, $X^{(m)} = (x_1, x_2, \ldots, x_m) \in \mathbb{R}^m$ and $v = \frac{m-1}{2}$. Then the accuracy of the median can be calculated by

$$w\left(X^{(m)}\right) = \sqrt{\frac{1}{m^m - 1} \sum_{b=1}^{m} P(b) \left(\text{med}\left(X^{(m)}\right) - x_b\right)^2} \tag{4.30}$$

where

$$P(b) = \sum_{i=0}^{v} \sum_{j=0}^{v} (b-1)^{v-i} (m-b)^{v-j} \binom{m}{v-i} \binom{v+i+1}{v-j} \tag{4.31}$$

is the number of all different bootstrap samples of $X^{(m)}$ having b as middle value.
(For convenience, in that equation let $0^0 = 1$.)

Proof. Similar to $b = c$ in theorem 4.2. □

Summarizing, the calculation of the accuracy of the median by theorem 4.1 needs the evaluation of $\binom{2n-1}{n}$ summands, where n is the size of the underlying vector $X^{(n)}$. The calculation by theorems 4.2 or corollary 4.1 has $\binom{n+1}{2}\frac{n^2}{4}$ summands for even n respectively $\frac{n(n+1)^2}{4}$ for odd n. For comparison, there are n^n possibilities for bootstrap samples of $X^{(n)}$, which have to be considered for an exact and straight forward calculation. These values for $n = \{1, 2, \ldots, 10\}$ are shown in table 4.1.

4.2 Smoothing Splines

Using either the bootstrap algorithm or one of the presented exact methods, for each time point t_i, $i \in \{1, \ldots, \tau\}$, finally two values are obtained: the gene expression level y_i, estimated by the data median, and its rmse, the accuracy w_i (cf. figure 4.2).

Now the question arises, how to connect the time discrete measurements y_i to get a continuous time course. Obviously, the trajectory should be closer to the y_i, which are more accurate, i.e. those which have a small corresponding w_i.

n	Summands to calculate		
	considering all possibilities	by thm. 4.1	by thm. 4.2 and cor. 4.1
1	1	1	1
2	4	3	3
3	27	10	12
4	256	35	40
5	3125	126	45
6	46656	462	189
7	823543	1716	112
8	16777216	6435	576
9	387420489	24310	225
10	10000000000	92378	1375

Table 4.1: Complexity of exact accuracy calculation

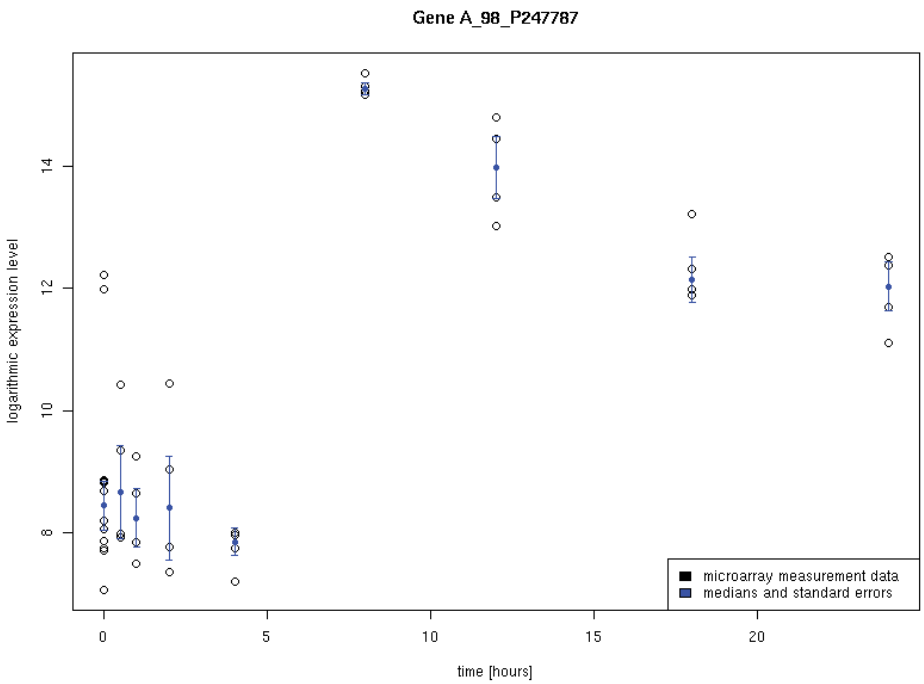


Figure 4.2: Gene time course of figure 4.1 with medians and root mean squared errors

In mathematical terms let the gene time course Θ follow the equation

$$\begin{aligned} \Theta : [t_1, t_\tau] &\to \mathbb{R} \\ y_i = \Theta(t_i) &+ \varepsilon_i \end{aligned} \tag{4.32}$$

where $i \in \{1, \ldots, n\}$ and the ε_i are random errors satisfying

$$\begin{aligned} E(\varepsilon_i) &= 0 \\ E(\varepsilon_i \varepsilon_j) &= \delta_{ij} w_i^2 \sigma^2 \end{aligned} \tag{4.33}$$

Here E denotes expectation, δ_{ij} is the Kronecker delta, being 1 if $i = j$ and 0 otherwise, and σ^2 is an unknown common variance factor.

Unfortunately there is no additional knowledge about the true pathway of the gene expression besides the few measurements, which would allow nearly every function type for Θ one can imagine. But in the following lets assume, that high oscillations or overshoots occur quite unlikely, which sounds reasonable from the biological point of view. Thus, the time course should be modeled by a curve fulfilling two conflicting objectives, being close to the data points, i.e. having a small squared interpolation error, weighted by the corresponding rmse w_i

$$E_w(\Theta) = \frac{1}{\tau} \sum_{i=1}^{\tau} \left(\frac{y_i - \Theta(t_i)}{w_i} \right)^2 \tag{4.34}$$

and being as smooth as possible, i.e. having a low roughness

$$R(\Theta) = \int_{t_1}^{t_\tau} \left[\Theta''(t) \right]^2 dx \tag{4.35}$$

which leads to the concept of smoothing spline interpolation.

A spline is a piecewise polynomial function with smooth junctions, more precise:

Definition 4.1: ***Splines***
Let $m, n \in \mathbb{N}$, $m, n \geq 2$. Let a set $\{t_1, \ldots, t_n\} \subset \mathbb{R}$ with $t_i < t_{i+1}$ for all $i \in \{1, \ldots, n-1\}$ be given. Let Π_δ denote the set of polynomials mapping $\mathbb{R}$ into $\mathbb{R}$ with a degree equal to or less than $\delta \in \mathbb{N}_0$.

Then S is called spline of order m (or degree $m-1$) with knot set $\{t_1, \ldots, t_n\}$ if the following holds

i) $S \in \mathscr{C}^{m-2}$, $S : \mathbb{R} \to \mathbb{R}$

ii) $S|_{[t_i, t_{i+1})} \in \Pi_{m-1}$ *for all* $i \in \{1, \ldots, n-1\}$.

iii) S has an $(m-1)$th derivative which is a step function having its jumps at the knots $t_1, \ldots, t_n$.

Let $\mathscr{S}^m(t_1, \ldots, t_n)$ denote the set of splines of order m with knots $t_1, \ldots, t_n$.

Lemma 4.4: ***Space of Splines***
Let $m,n \in \mathbb{N}$. Let a set $\{t_1,\ldots,t_n\} \subset \mathbb{R}$ with $t_i < t_{i+1}$ for all $i \in \{1,\ldots,n-1\}$ be given. Let the plus-function $(\ )_+ : \mathbb{R} \to [0,\infty)$ be defined by

$$(x)_+ = \begin{cases} x & x > 0 \\ 0 & \textit{otherwise} \end{cases} \tag{4.36}$$

Then the following two are equivalent

a) $S \in \mathscr{S}^m(t_1,\ldots,t_n)$

b) There are $a_0,\ldots,a_{m-1},b_1,\ldots,b_n \in \mathbb{R}$ such that

$$S(t) = \sum_{i=0}^{m-1} a_i t^i + \sum_{j=1}^{n} b_j \left(t - t_j\right)_+^{m-1} \tag{4.37}$$

Furthermore, $\mathscr{S}^m(t_1,\ldots,t_n)$ is a vector space over the reals with dimension $m+n$.

Proof. Obviously, (4.37) fulfills i) - iii) in the definition of splines, thus it only remains to prove the implication from b) to a).

Therefore, define intervals

$$\begin{array}{rcl} T_0 & = & (-\infty, t_1) \\ T_j & = & \left[t_j, t_{j+1}\right) \quad \text{for } j \in \{1,\ldots,n-1\} \\ T_n & = & \left[t_n, \infty\right) \end{array} \tag{4.38}$$

Note, that iii) implies together with i) that a spline R of order m coincides with a polynomial of degree $(m-1)$ or lower in the intervals T_0 respectively T_n.

Thus, a spline $S \in \mathscr{S}^m(t_1,\ldots,t_n)$ can be represented by its polynomial pieces

$$S_j(t) = \sum_{i=0}^{m-1} c_{j,i} t^i \quad \text{if } t \in T_j \tag{4.39}$$

with $c_{j,i} \in \mathbb{R}$ for all $i \in \{1,\ldots,m-1\}$ and $j \in \{0,\ldots,n\}$.

Furthermore, the spline pieces can also be defined recursively by using appropriate updates for the monomials:

$$\begin{array}{rcll} S_j(t) & = & S_{j-1} + \underbrace{\sum_{i=0}^{m-1} \beta_{j,i} \left(t - t_j\right)^{m-1}}_{\Delta S_j} & \text{if } t \in T_j \\ S_j(t) & = & S_{j-1} + \sum_{i=0}^{m-1} \beta_{j,i} \left(t - t_j\right)_+^{m-1} & \text{if } t \in T_j \end{array} \tag{4.40}$$

for $j \in \{1,\ldots,n\}$.

Due to i) the following holds

$$\lim_{t \nearrow t_j} S_{j-1}^{(i)}(t) = S_j^{(i)}(t_j) \tag{4.41}$$

for all $i \in \{0, \ldots, m-2\}$ and $j \in \{1, \ldots, n\}$, where $S^{(i)}$ denotes the i-th derivative of the function S.

Using (4.40) this becomes

$$\Delta S_j^{(i)}(t_j) = 0 \quad \text{for } i \in \{0, \ldots, m-2\} \tag{4.42}$$

which yields $\beta_{j,0} = \cdots = \beta_{j,m-2} = 0$ for all $j \in \{1, \ldots, n\}$.

Setting

$$\begin{array}{rcll} a_i & = & c_{0,i} & \text{for } i \in \{0, \ldots, m-1\} \\ b_j & = & \beta_{j,m-1} & \text{for } j \in \{1, \ldots, n\} \end{array} \tag{4.43}$$

results in the desired representation (4.37) which proves the implication from b) to a).

Using this representation shows immediately, that the set of splines $\mathscr{S}^m(t_1, \ldots, t_n)$ is closed with respect to addition and multiplication with real scalars, concluding that $\mathscr{S}^m(t_1, \ldots, t_n)$ is a vector space over the reals. The spanning function set

$$B = \left\{1, t, \ldots, t^{m-1}, (t-t_1)_+^{m-1}, \ldots, (t-t_n)_+^{m-1}\right\} \tag{4.44}$$

is obviously linearly independent, becoming a basis of the spline space which proves finally the dimension to be $m+n$. □

More important than the general concept of splines is are the so-called natural splines, which exhibit the minimizing property we desire for the time course approximation.

Definition 4.2: ***Natural Splines***
Let $m \in \mathbb{N}$. A spline $S \in \mathscr{S}^{2m}(t_1, \ldots, t_n)$ is called natural spline of order $2m$ if it satisfies the additional property

iv) $S|_{(-\infty,t_1)}, S|_{(t_n,\infty)} \in \Pi_{m-1}$

Let $\mathscr{N}^{2m}(t_1, \ldots, t_n)$ denote the set of natural splines of order $2m$ having knots at $t_1, \ldots, t_n$.

Remark 4.1:
The name *natural* is due to the fact, that a spline S fulfilling iv) has a natural boundary property:

$$S^{(j)}(t) = S^{(j)}(t) = 0 \quad \text{for all } t \in \mathbb{R} \setminus [t_1, t_n] \text{ and } j = m, \ldots, 2m-1 \tag{4.45}$$

Note that $\mathscr{N}^{2m}(t_1, \ldots, t_n)$ is a subspace of $\mathscr{S}^{2m}(t_1, \ldots, t_n)$ with dimension n. For proof please refer [Sch07].

Notation 4.1:
In the following let $[a,b] \subset \mathbb{R}$ be a finite interval and $W_2^m[a,b]$ denote the Sobolev space of order m:

$$W_2^m[a,b] = \left\{ w | w^{(j)} \text{ is absolutely continuous for all } j = 0,\ldots,m-1 \text{ and } w^{(m)} \in L_2[a,b] \right\} \tag{4.46}$$

With the basic definition and properties of natural splines, we have the tools to solve the minimization task at the outset of this chapter.

Theorem 4.3: ***Spline Interpolation (Schoenberg, 1964)***
Let $m \in \mathbb{N}$, $n \in \mathbb{N}$, $m < n$ and a set of pairs $\mathbb{Y} = \{(t_i, y_i) | i \in \{1,\ldots,n\}\} \subset [a,b] \times \mathbb{R}$ with ascending t_i, i.e. $a \leq t_1 < t_2 < \cdots < t_n \leq b$, be given. For a function $f \in W_2^m[a,b]$ the roughness is given by

$$R(f) = \int_a^b \left(f^{(m)}(x) \right)^2 dx \tag{4.47}$$

The minimization problem: Find $f \in W_2^m[a,b]$ fulfilling

$$\begin{aligned} f(t_i) &= y_i \qquad \text{for all } i \in \{1,\ldots,n\} \\ R(f) &= \text{minimum} \end{aligned} \tag{4.48}$$

has an unique solution $f(t) = S(t,\mu) \in \mathcal{N}^{2m}(t_1,\ldots,t_n)$.

Proof. See theorem 4.5 with $\mu \to 0$ and $w_1 = \cdots = w_n = 1$. □

Theorem 4.4: ***Smoothing Spline Interpolation (Schoenberg, 1964)***
Let m, n, $\mathbb{Y}$, and $R(f)$ be defined as in theorem 4.3. Let $\mu \in \mathbb{R}^+$. Further define the squared interpolation error

$$E(f) = \sum_{i=1}^{n} (y_i - f(t_i))^2 \tag{4.49}$$

Then the minimization problem: Find $f \in W_2^m[a,b]$ fulfilling

$$E(f) + \mu R(f) = \text{minimum} \tag{4.50}$$

has an unique solution $f(t) = S(t,\mu) \in \mathcal{N}^{2m}(t_1,\ldots,t_n)$.

Proof. See theorem 4.5 with $w_1 = \ldots, w_n = 1$. □

Remark 4.2:
The design parameter $\mu \in \mathbb{R}_0^+$ represents the relative weight of the two competing objectives, being close to the data knots and having a low roughness.

$\mu \to 0$ results in a normal interpolating spline corresponding to theorem 4.3 connecting exactly the data points, while $\mu \to \infty$ leads to linear regression.

Before the final theorem for the time course approximation is presented and proven, a technical lemma has to be stated:

Lemma 4.5: ***(Lyche and Schumaker, 1973)***
Let $x_1, \ldots, x_n$ be a basis of $\mathcal{N}^{2m}(t_1, \ldots, t_n)$.
Then there are coefficients $a_{0,j} \ldots, a_{m-1,j}, b_{1,j}, \ldots, b_{n,j} \in \mathbb{R}$ such that

$$x_j(t) = \sum_{i=0}^{m-1} a_{i,j}\, t^i + \sum_{i=1}^{n} b_{i,j}\ (t - t_i)_+^{2m-1} \tag{4.51}$$

If $S(t) = \sum_{i=1}^{n} \beta_i x_i(t)$ and $f \in W_2^m[a,b]$ then

$$\int_a^b f^{(m)}(t)\, S^{(m)}(t)\, dt = (-1)^m (2m-1)! \sum_{i=1}^{n} \left[f(t_i) \sum_{j=1}^{n} \beta_j b_{i,j} \right] \tag{4.52}$$

Proof. The proof is available in [Eub88]. □

Theorem 4.5: ***Weighted Smoothing Spline Interpolation***
Let m, n, $\mathbb{Y}$, and $R(f)$ be defined as in theorem 4.3. Let $\mu \in \mathbb{R}^+$. For weights $w_i \in \mathbb{R}^+$ for $i = 1, \ldots, n$ the weighted squared interpolation error is defined by

$$E_w(f) = \sum_{i=1}^{n} w_i (y_i - f(t_i))^2 \tag{4.53}$$

Then the minimization problem: Find $f \in W_2^m[a,b]$ fulfilling

$$E_w(f) + \mu R(f) = minimum \tag{4.54}$$

has an unique solution $f(t) = S(t, \mu) \in \mathcal{N}^{2m}(t_1, \ldots, t_n)$.

Proof. (Extension of the proof in [Eub88] for the unweighted case)
Define for $f_1, f_2 \in W_2^m[a,b]$ the functional

$$\Phi(f_1, f_2, \delta) = \frac{1}{2}\left[E_w(f_1 + \delta f_2) + \mu R(f_1 + \delta f_2)\right] \tag{4.55}$$

and

$$\Phi(f_1, f_2) = \left.\frac{d\Phi(f_1, f_2, \delta)}{d\delta}\right|_{\delta=0} \tag{4.56}$$

$2\Phi(f_1, f_2)$ is called the Gâteaux derivative of (4.54) at f_1 in direction f_2. Since the Gâteaux derivative has to vanish in an extremum as ordinary derivatives do

$$\Phi(f_1, f_2) = 0 \tag{4.57}$$

is a necessary condition for f_1 being a minimizer of (4.54).

With this knowledge take $f_1, f_2 \in W_2^m[a,b]$ and calculate

$$\Phi(f_1, f_2) = -\sum_{i=1}^{n} w_i f_2(t_i)[y_i - f_1(t_i)] + \mu \int_a^b f_1^{(m)}(t) f_2^{(m)}(t)\, dt \tag{4.58}$$

Thus, for being a minimizer f_1 has to fulfill

$$\sum_{i=1}^{n} w_i f_2(t_i)[y_i - f_1(t_i)] = \mu \int_a^b f_1^{(m)}(t) f_2^{(m)}(t)\, dt \tag{4.59}$$

for all $f_2 \in W_2^m[a,b]$.

Now let $x_1, \ldots, x_n$ be a basis of $\mathcal{N}^{2m}(t_1, \ldots, t_n)$ and $f_1 = \sum_{i=1}^{n} \beta_i x_i$, then using preliminary lemma 4.5 the equation (4.59) becomes

$$\sum_{i=1}^{n} w_i f_2(t_i)\left[y_i - \sum_{j=0}^{n} \beta_j x_j(t_i)\right] = \mu(-1)^m (2m-1)! \sum_{i=1}^{n}\left[f(t_i) \sum_{j=1}^{n} \beta_j b_{i,j}\right] \tag{4.60}$$

Since f_2 is arbitrary in $W_2^m[a,b]$ this transforms to

$$\sum_{j=1}^{n} \left[x_j(t_i) + \mu(-1)^m (2m-1)! w_i^{-1} b_{i,j}\right] \beta_j = y_i \tag{4.61}$$

which becomes

$$(X + \mu G)\beta = y \tag{4.62}$$

in matrix notation, where

$$\begin{aligned} X &= \{x_j(t_i)\}_{i,j=1}^{n} \\ G &= \{x_j(t_i)\}_{i,j=1}^{n} \\ \beta &= (\beta_1, \ldots, \beta_n)' \\ y &= (y_1, \ldots, y_n)' \end{aligned} \tag{4.63}$$

A system of equations $Ax = b$ has an unique solution if $Ax = 0$ has the unique solution $x = 0$. Thus, set $y = 0$ in (4.62) for a moment. Obviously, $\beta = 0$ solves the system, thus one has only to show its uniqueness. In the case $y = 0$, and $f_2 = f_1$ which is admissible since the minimizing $f_1 \in W_2^m[a,b]$ equation (4.58) becomes

$$\Phi(f_1, f_1) = \sum_{i=1}^{n} w_i f_1(t_i)^2 + \mu \int_a^b f_1^{(m)}(t)^2\, dt = 0 \tag{4.64}$$

This means $f_1^{(m)}(t) = 0$ almost everywhere and $f_1(t_i) = 0$ for $i = 1, \ldots, n$. Thus, f_1 has to be a polynomial with degree less than m which vanishes at $n > m$ points, which holds only for $f_1 \equiv 0$.

Since $x_1,\ldots,x_n$ is a basis of $\mathcal{N}^{2m}(t_1,\ldots,t_n)$, $f_1=\sum_{i=1}^{n}\beta_i x_i$ implies that $\beta_1=\cdots=\beta_n=0$ concluding that $\beta=0$ is the only solution of $(X+\mu G)\beta=0$.

We have proven the uniqueness of a natural spline fulfilling (4.59) which is a necessary condition for a minimizer of (4.54). For existence we show that $f_1=\sum_{i=1}^{n}\beta_i x_i$ with β_i got from (4.62) actually is the minimizer sought-after. Therefore, take again $f_2\in W_2^m[a,b]$ and calculate

$$\begin{aligned} E_w(f_2)+\mu R(f_2) &= \sum_{i=1}^{n} w_i\left(y_i-f_2(t_i)\right)^2+\mu\int_a^b f_2^{(m)}(t)^2\,dt \\ &= \sum_{i=1}^{n} w_i\left(y_i-[f_1+f_2-f_1](t_i)\right)^2+\mu\int_a^b [f_1+f_2-f_1]^{(m)}(t)^2\,dt \end{aligned} \tag{4.65}$$

Applying the binomial theorem yields

$$\begin{aligned} E_w(f_2)+\mu R(f_2) &= \sum_{i=1}^{n} w_i\left(y_i-f_1(t_i)\right)^2 \\ &\quad -2\sum_{i=1}^{n} w_i\left(y_i-f_1(t_i)\right)\left(f_2(t_i)-f_1(t_i)\right) \\ &\quad +\sum_{i=1}^{n} w_i\left(f_2(t_i)-f_1(t_i)\right)^2 \\ &\quad +\mu\int_a^b f_1^{(m)}(t)^2\,dt \\ &\quad +2\mu\int_a^b f_1^{(m)}(t)\,[f_2-f_1]^{(m)}(t)\,dt \\ &\quad +\mu\int_a^b [f_2-f_1]^{(m)}(t)^2\,dt \end{aligned} \tag{4.66}$$

Using the notation of Gâteaux derivative the equation becomes

$$\begin{aligned} E_w(f_2)+\mu R(f_2) &= \sum_{i=1}^{n} w_i\left(y_i-f_1(t_i)\right)^2+\mu\int_a^b f_1^{(m)}(t)^2\,dt \\ &\quad +2\Phi(f_1,f_2-f_1) \\ &\quad +\sum_{i=1}^{n} w_i\left(f_2(t_i)-f_1(t_i)\right)^2+\mu\int_a^b [f_2-f_1]^{(m)}(t)^2\,dt \\ &\geq \sum_{i=1}^{n} w_i\left(y_i-f_1(t_i)\right)^2+\mu\int_a^b f_1^{(m)}(t)^2\,dt \\ &= E_w(f_1)+\mu R(f_1) \end{aligned} \tag{4.67}$$

due to $\Phi(f_1,f_2-f_1)=0$ because f_1 fulfills (4.57) and $f_2-f_1\in W_2^m[a,b]$. This proves that f_1 is actually a minimizer.

Now one last step is missing. We have shown that a minimizer is unique in the space of natural splines. To finalize the proof the uniqueness has to be expanded to the space $W_2^m[a,b]$. Therefore, assume f_2 in (4.67) is a minimizer, too. Then the equation yields that

$$\sum_{i=1}^{n} w_i\left(f_2(t_i)-f_1(t_i)\right)^2=0 \tag{4.68}$$

and

$$\mu\int_a^b [f_2-f_1]^{(m)}(t)^2\,dt=0 \tag{4.69}$$

Equation (4.68) shows that $f_2 - f_1$ vanishes in n points while (4.69) yields that the m-derivative of $f_2 - f_1$ equals zero almost everywhere, i.e. $f_2 - f_1 \in \Pi_{m-1}$. This finally concludes to $f_2 - f_1 \equiv 0$, since $n > m$. □

Corollary 4.1: ***Cubic Spline Interpolation***
Let $n > 2$, $\mu \in \mathbb{R}^+$, $\mathbb{Y} = \{(t_i, y_i) \,|\, i \in \{1, \ldots, n\}\} \subset \mathbb{R}^2$ with $t_i < t_{i+1}$ for all $i \in \{1, \ldots, n-1\}$. Let $w_i \in \mathbb{R}^+$ for $i \in \{1, \ldots, n\}$. Then

$$E_w(f) + \mu R(f) = \sum_{i=1}^{n} \left(\frac{y_i - f(t_i)}{w_i} \right)^2 + \mu \int_{t_1}^{t_n} \left(f''(x) \right)^2 dx = minimum \tag{4.70}$$

has an unique solution $f(x) = S(x, \mu) \in \mathcal{N}^4$.

Now its proven that natural cubic splines, i.e. splines of degree 3 (or order 4), with natural ending conditions are the correct choice for the initial interpolation problem of microarray time course data (4.34) and (4.35).

An algorithm for the construction of the cubic smoothing splines minimizing (4.70) is available in [Spä73]. In addition the Matlab routine *csaps* is recommended, which solves exactly the minimization task.

4.3 Results

In this chapter smoothing splines were presented as method for interpolating gene expression time courses. Since microarray measurements often are few in number and susceptible to outliers the median

$$y_i^{(g)} = \mathrm{med}\left(M_{g,t_i}\right) \tag{4.71}$$

of the measurements of each gene g at each time t_i was taken instead of the arithmetic mean as representative for the data.

This gave rise to another quality measure as the standard deviation as default choice. The presented exact bootstrapping yields an accuracy estimation of the median in a efficient way of calculation. This accuracy measure of the median of gene g at time t_i is denoted by

$$w_i^{(g)} = w\left(M_{g,t_i}\right) \tag{4.72}$$

where the function w is given in theorem 4.2 and corollary 4.1.

In addition the exact bootstrapping for calculating the accuracy of any statistic S was given in theorem 4.1, if another statistic than the median is desired. However, the generality of that theorem reduces the efficiency, especially if larger data sets are considered. The gentle reader might improve the theorem with respect to her/his statistic in the same way as it was done for the median in this work.

After calculating a statistic and its accuracy these values were used for calculating a cubic smoothing spline. This type of splines only approximates the interpolation set, but reduces the curvature and overshoots of the resulting interpolation. The approximation quality of each point depends on its accuracy measure, the lower the closer to that point the spline is.

Therefore, a linear combination of the weighted squared approximation errors and the spline curvature is minimized. The error weights were given by the accuracy estimation of each data point.

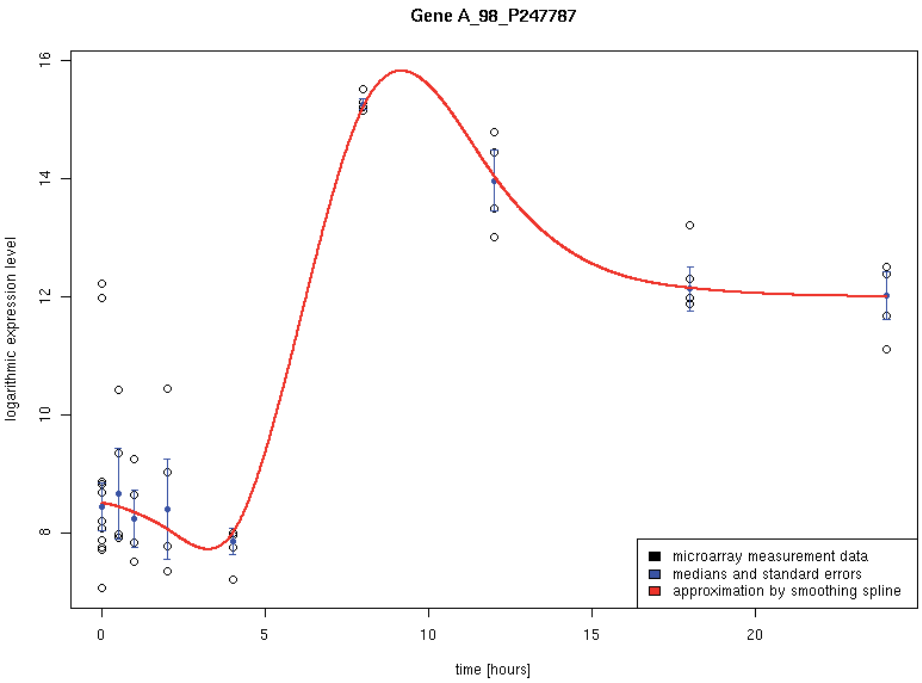


Figure 4.3: Gene time course of figure 4.1 with medians, rmse and corresponding smoothing spline ($\mu = 0.43$)

In the following chapters let the interpolating cubic smoothing spline $S_g : [t_1, t_\tau] \to \mathbb{R}$ of gene g be represented piecewise by polynomials:

$$S_g(t) = \sum_{i=1}^{\tau-1} \delta_i(t)\, a_{i,3}^{(g)} (t-t_i)^3 + a_{i,2}^{(g)} (t-t_i)^2 + a_{i,1}^{(g)} (t-t_i) + a_{i,0}^{(g)} \tag{4.73}$$

where

$$\begin{aligned} \delta_i(t) &= \begin{cases} 1 & t \in [t_i, t_{i+1}) \\ 0 & t \notin [t_i, t_{i+1}) \end{cases} \quad \text{for } i \in \{1, \ldots, \tau-2\} \\ \delta_{\tau-1}(t) &= \begin{cases} 1 & t \in [t_{\tau-1}, t_\tau] \\ 0 & t \notin [t_{\tau-1}, t_\tau] \end{cases} \end{aligned} \tag{4.74}$$

denote the characteristic functions of the according intervals.

5 Clustering of Time Course Data

Like and equal are two entirely different things.

— Madeleine L'Engle, American writer (1918-2007)

Recall the main goal of this work, to develop a dynamical system which reproduces the behavior and interaction of the gene expression level time courses. But as already mentioned in the biological introduction, genes accumulate to functional groups and thus their expression level dynamics coincide except for biological or technical noise. This precludes the determination of one special activating or inhibiting gene out the set of genes having similar time courses. Thus, a dynamical system can only handle clusters, whose generation is topic of this chapter.

At first an overview to clustering is given where its usage and its types are shortly introduced.

Thereafter a data standardization technique is presented. The standardization removes information of the data which is not desired as data feature for clustering. More precise, the time courses will be clustered with respect to their relative changes in expression level, no matter how large the absolute values are. Thus, the values will be standardized to a common level, but keeping their time course shape. While common methods as the standardization of random variables do not yield the desired behavior for the concrete case, an appropriate alternative is given.

The definition of "close" and "far" is one main setscrew in clustering. Thus, the next sections deal with different distance measures, which are applicable to the time course clustering task. First distance measures between single time courses are presented, while the subsequent one considers distance between sets of time courses, i.e. clusters. The latter one is also known as linkage method.

After this, three clustering methods are presented and its properties are discussed. For each technique also an algorithm is provided.

Finally, several measures for the quality of clusterings are given. Since no additional information about the true gene correlations is available, these measures use only the properties of the clusterings themselves. They validate clusters by their within densities and their in-between separation.

In the next sections assume that a microarray time course experiment has been run and the resulting measurement data has been normalized using the methods of section 3.2. Let the Fisher-Pitman-Test be applied to each time course data set and every gene be discarded if it

has not at least one time point where the measurements differ significantly from those at the starting time. Finally, let cubic smoothing splines (4.73) be calculated, interpolating the time course data sets.

5.1 A Short Introduction to Clustering

Clustering is a field of unsupervised learning for finding intrinsic structures in unlabeled data sets. Therefore, a set is divided in sub-classes containing similar elements, called clusters, having high dissimilarity to members of the other classes. Because there are many different types of clustering, we restrict to the so called exclusive clustering, which implies that after the separation each element of the original set has to belong to exactly one cluster.

In mathematical terms:

$$\begin{aligned} X &= \bigcup_{i=1}^{k} C_i \\ C_i \cap C_j &= \emptyset \qquad \text{for all } i,j \in \{1,\ldots,k\}, i \neq j \\ Q(C_1,\ldots,C_k) &= \textit{minimum} \end{aligned} \tag{5.1}$$

Here C_i is called i-th cluster and Q is the partition quality measure of the clustering $(C_1,\ldots,C_k)$.

There is no "best" or general rule for within-cluster similarity and in-between-clusters dissimilarity - but some are quite common and will be shortly presented in the next section 5.3. The same holds for the algorithms for finding clusters, there is also a vast amount of clustering techniques and related variants using different approaches for distinction and separation of similarity groups. Many of them were also successfully tested and compared on the basis of microarray data and gene expression levels [ESBB98], [YMB03], [MCZL06], [YK06].

Clustering methods can be separated in five categories:

1.) Partitioning Clustering
These methods generate different partitions and modify them using an iterative control strategy to optimize a predefined quality criterion. The K-Means Algorithm described in section 5.5.1 belongs to this category.

2.) Hierarchical Clustering
Hierarchical algorithms generate a decomposition tree, where the complete data set is used as root and is divided step by step into sub-classes until having singletons as leafs. Such algorithms are called agglomerative (or bottom-up) when they start with the singletons and combine them to larger sets, or divisive (or top-down) if they start with the complete data set and split it. After setting up the decomposition tree (completely or only partially), a quality prerequisite defines a cut through the tree yielding the desired clustering. An agglomerative hierarchical clustering method is given in section 5.5.2.

3.) Density-based Clustering
These clustering methods are based on local density criteria, identifying elements of the data set which are near to each other according to a given distant measure. The method DBSCAN which will be presented in section 5.5.3 is density-based.

4.) Grid-based Clustering
Grid-based algorithms use a partition of the complete data space, where the given data points are distributed. Thereafter the data space cells containing points are merged to clusters following a cluster quality rule.

5.) Model-based Clustering
These methods use a cluster model hypothesis, which implies that additional knowledge about the structure of the clusters in the data set is necessary. Having such cluster models, the clustering itself is "simply" an optimization searching for the best fitting distribution of the data. Neural network approaches as the well-known Self-Organizing-Map method (SOM) belong to this category.

5.2 Data Standardization

For biological analyses often the absolute gene expression level itself is not as important than its relative change caused by the treatment. Therefore, the ratio of expression levels between test and control group will be used for clustering. Time course experiments lack a dedicated control group which implies some degree of freedom in selecting the standardization.

A common method in statistics is known under the name of standardization itself. For distinguishing it from other data standardization techniques this work keeps with the nomenclature chosen in [Kre08]:

Definition 5.1: ***0-1-Standardization***
Let $X \in \mathbb{R}^n$ be a sample of real numbers. The corresponding 0-1-standardized sample X^* is given by

$$X_i^* = \sigma_X^{-1}\left(X_i - \mu_X\right) \tag{5.2}$$

where μ_X and σ_X denote the arithmetic sample mean respectively the empirical standard deviation

$$\begin{aligned} \mu_X &= n^{-1} \sum_{i=1}^{n} X_i \\ \sigma_X &= (n-1)^{-1} \sum_{i=1}^{n} \left(X_i - \mu_X\right)^2 \end{aligned} \tag{5.3}$$

The name 0-1-standardization is due to the fact that $\mu_{X^*} = 0$ and $\sigma_{X^*} = 1$.

It can be easily shown that the samples X and $Y = \alpha X$ have the same 0-1-standardized representation $Y^* = X^*$, independently of $\alpha \in \mathbb{R} \setminus \{0\}$, which seems to result exactly in the desired behavior for microarray data, putting the focus on the relative changes:

$$\frac{X_i}{X_j} = \frac{Y_i}{Y_j} \text{ for all } i, j \in \{1, \ldots, n\}, X_j \neq 0 \tag{5.4}$$

However, the shift by the mean value yields an undesired side effect. Two samples X, Y having a constant offset $a \in \mathbb{R} \setminus \{0\}$ in each component $Y_i = X_i + a$ also have the same 0-1-standardized value $X^* = Y^*$. In this case only the absolute but not the relative differences within the components of X and Y coincides:

$$X_i - X_j = Y_i - Y_j \text{ for all } i, j \in \{1, \ldots, n\} \tag{5.5}$$

Thus, the mean shift should be neglected, which results in a standardization called consistently 1-standardization.

Definition 5.2: ***1-Standardization***
Let $X \in \mathbb{R}^n$ be a sample of real numbers. The corresponding 1-standardized sample X^* is given by

$$X^* = s_X^{-1} X \tag{5.6}$$

where $s_X \in \mathbb{R} \setminus \{0\}$ denotes a statistic of X appropriate for the standardization (cf. remark 5.1).

Remark 5.1:
Microarray time course data as generated in this work give rise to two meaningful choices for the statistic s_X. The division by the value at the beginning of the experiment

$$s_X = X_1 \tag{5.7}$$

or the division by the mean value over all time points

$$s_X = \mu_X \tag{5.8}$$

Both choices yield the desired focus on the relative changes, i.e. $X^* = Y^*$, where $Y = \alpha X$, $\alpha \in \mathbb{R} \setminus \{0\}$, and allow an easy interpretation of the standardized sample values. (5.7), which is obviously only applicable if $X_1 \neq 0$, results in a sample which components directly show the ratio to the first value of the original data sample. That means for microarray time course data the standardized value at time t_n denotes exactly the factor of up- or down-regulation with respect to the gene expression level at the beginning. The drawback is the strong dependency of the standardized sample on the first value of the original one, which causes a higher sensitivity to outliers and errors in the measurements at the first time point.

This sensitivity can be reduced by taking the sample mean for standardization (5.8), but this in turn means that the components of the standardized sample give the change in gene expression level with respect to the mean over time, which might be less useful for interpretation.

Since the microarray experiment in this work has a sufficiently large number of replicates at the starting time, the 1-standardization using (5.7) will be used as data standardization method.

Therefore, remember that the data is in logarithmic scale which transforms the quotient to a difference resulting in the standardized gene expression time course using the notations of section 4.3:

$$y_i^{(g)*} = y_i^{(g)} - y_1^{(g)} \text{ for all } i \in \{1, \ldots, \tau\} \tag{5.9}$$

This offset of the expression levels also results in a shift of the splines, which can be easily done due to its piecewise polynomial representation. The standardized spline corresponding to gene g is given by

$$S_g^*(t) = \sum_{i=1}^{\tau-1} \delta_i(t)\, a_{i,3}^{(g)} (t-t_i)^3 + a_{i,2}^{(g)} (t-t_i)^2 + a_{i,1}^{(g)} (t-t_i) + a_{i,0}^{(g)*} \tag{5.10}$$

where

$$a_{i,0}^{(g)*} = a_{i,0}^{(g)} - y_1^{(g)} \text{ for all } i \in \{1, \ldots, \tau-1\} \tag{5.11}$$

5.3 Distance Measures

For any clustering method a measure of dissimilarity of data points is essential. In our case, having quantitative data, dissimilarity coincides with distance. However, even the distance of clusters is not uniquely defined. Many different dissimilarity measures, especially those applicable to microarray time course data are presented in [Kre08].

For completeness, two of the most frequently used measures as well as one task-specific one are stated below.

5.3.1 p-Minkowski Metric

Definition 5.3: ***p-Minkowski metric***
Let $(\mathbb{R}^n, \| \|_p)$ be the normed vector space where the norm is defined by

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}} \tag{5.12}$$

with $x \in \mathbb{R}^n$, $p \in \mathbb{R}$ and $p \geq 1$.

The induced metric is called p-Minkowski metric:

$$d(x,y) = \|x-y\|_p = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{5.13}$$

where $x, y \in \mathbb{R}^n$.

The special cases Manhattan metric ($p = 1$), Euclidean distance ($p = 2$), and Chebyshev metric ($p \to \infty$) are well-known and quite obvious and intuitive choices for a distance measure between two points in the real vector space.

Using the notation of the microarray data set the distance between gene g_1 and gene g_2 becomes

$$d_p(g_1, g_2) = \left(\sum_{i=1}^{\tau} \left| y_i^{(g_1)*} - y_i^{(g_2)*} \right|^p \right)^{\frac{1}{p}} \tag{5.14}$$

Please note, that the p-Minkowski metric uses only the average gene expression $y_i^{(g)*}$, neglecting its accuracy $w_i^{(g)}$. The advantage of the combination with the median as average is its insensitivity to outliers. The drawback is the loss of information, which might be useful for the clustering.

5.3.2 Pearson's Product-Momentum Correlation Coefficient

Another approach to measure the distance between data vectors is the statistical concept of correlation. The correlation measures the degree of linear dependency of two samples. In the following the idea is explained by means of Pearson's product-momentum correlation coefficient.

But before defining the correlation coefficient, one should give credit where credit is due. In 1885, the English polymath Sir Francis Galton developed the statistical concept of correlation to which Karl Pearson assigned his index eleven years later. The issue of correlation itself was already broached by Carl Friedrich Gauss in 1823, the French astronomer Auguste Bravais in 1846, and also Galton's half-cousin, the founder of the evolutionary theory Charles Darwin in 1868 [RN88].

However, the index itself became generally known as Pearson's product-momentum correlation coefficient or shortly Pearson's r.

Definition 5.4: ***Pearson's product-momentum correlation coefficient***
Let two samples $X, Y \in \mathbb{R}^n$ be given. Its product-momentum correlation coefficient is given by dividing the sample covariance by the product of the standard deviations:

$$r(X, Y) = \frac{\sum_{i=1}^{n} (X_i - \mu_X)(Y_i - \mu_Y)}{\sqrt{\sum_{i=1}^{n} (X_i - \mu_X)^2} \sqrt{\sum_{i=1}^{n} (Y_i - \mu_Y)^2}} \tag{5.15}$$

where X_i and Y_i denote the i-th component of the vectors X respectively Y while μ_X and μ_Y are the corresponding arithmetic means (cf. equation 5.3).

For any samples X and Y Pearson's r lies in the interval $[-1, 1]$. In the case of maximal positive correlation ($r = 1$) the samples show a perfectly linear, positive relationship, i.e.

$$X = \alpha Y \tag{5.16}$$

for an $\alpha \in \mathbb{R}^+$, while $r = -1$ denotes a perfectly linear, negative relationship, i.e. (5.16) with $\alpha \in \mathbb{R}^-$. $r = 0$ signifies that no linear relationship exist between the samples at all (cf. figure 5.1).

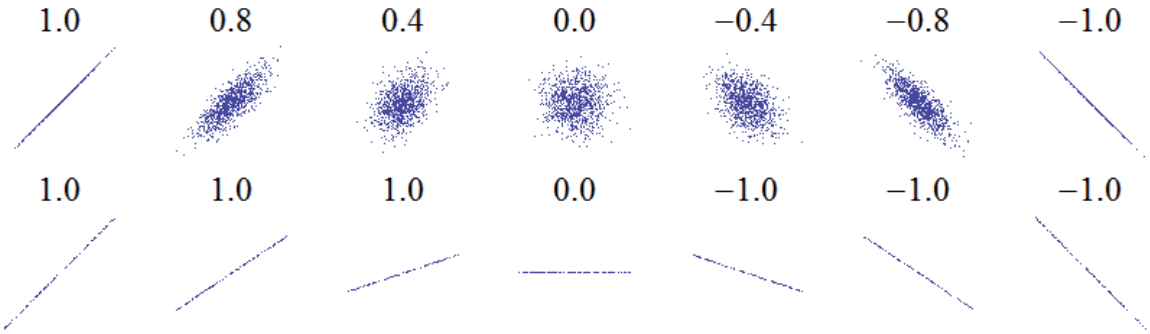


Figure 5.1: Correlation of data sets

For the search for common gene expression profiles, it does not matter if the linear relationship between genes is positive or negative. The knowledge of the factor $\alpha \in \mathbb{R}$ suffices to reconstruct the genes from the cluster representative. (Please note that $\alpha = 0$ is not possible due to the removal of non-significant gene time courses.) Thus, an appropriate distance measure for the genes g_1 and g_2 using Pearson's r is

$$\begin{aligned} d_{Pearson}(g_1, g_2) &= 1 - \quad |r(g_1, g_2)| \\ &= 1 - \frac{\left| \sum_{i=1}^{n} \left(y_i^{(g_1)*} - \mu_{y(g_1)*} \right) \left(y_i^{(g_2)*} - \mu_{y(g_2)*} \right) \right|}{\sqrt{\sum_{i=1}^{n} \left(y_i^{(g_1)*} - \mu_{y(g_1)*} \right)^2} \sqrt{\sum_{i=1}^{n} \left(y_i^{(g_2)*} - \mu_{y(g_2)*} \right)^2}} \end{aligned} \tag{5.17}$$

where

$$\mu_{y(g)*} = \frac{1}{\tau} \sum_{i=1}^{\tau} y_i^{(g)*} \tag{5.18}$$

is the arithmetic mean of the standardized expression levels at times $t_1, \ldots, t_\tau$ of gene g.

5.3.3 L^p-Metric

Since they are simply working on real vectors, the p-Minkowski metric as well as Pearson's r neglect the fact that a time course measurement contains additional information:

1. The order of the values:
   Both methods ignore that the time course data is not only a vector, but its components are ordered with respect to time.

2. The intervals between the time points of measurements:
   All components are treated equally. However, if the time points T are not distributed

in an equidistant manner, some components have higher influences on the interpolated time course trajectory.

A distance measure using the spline interpolation would include both items into the clustering.

Definition 5.5: ***L^p-metric***
Let $L^p = L^p([t_1,t_n],\mathbb{R})$ be the space of measurable and p-th power absolute integrable functions mapping from $[t_1,t_n]$ into $\mathbb{R}$. The corresponding norm of $f \in L^p$ is given by

$$\|f\|_p = \left(\int_{[t_1,t_n]} |f(t)|^p \, d\mu(t)\right)^{\frac{1}{p}} \tag{5.19}$$

where μ denotes the Lebesgue-measure.

The metric induced by $\| \, \|_p$ is given by

$$d_{L^p}(f_1,f_2) = \|f_1 - f_2\|_p \tag{5.20}$$

The most common choice is $p = 2$, which allows the transformation of the integral into a finite sum if applied to splines by using its piecewise polynomial structure.

Lemma 5.1: ***Spline metric***
Let $S_g, S_h \in \mathcal{N}^4$ be two natural splines according to notation 4.73. Its L^2-distance is given by

$$\begin{aligned} d_{\mathcal{N}^4}(g,h) &= d_{L^2}(S_g,S_h) \\ &= \sqrt{\sum_{j=0}^{3}\sum_{k=0}^{3} \frac{\left(a_{i,j}^{(g)} - a_{i,j}^{(h)}\right)\left(a_{i,k}^{(g)} - a_{i,k}^{(h)}\right)}{j+k+1} \sum_{i=1}^{\tau-1} (t_{i+1}-t_i)^{j+k}} \end{aligned} \tag{5.21}$$

Proof.

$$\begin{aligned} (d_{L^2}(S_g,S_h))^2 &= \int_{t_1}^{t_\tau} \left|S^{(g)}(t) - S^{(h)}(t)\right|^2 dt \\ &= \sum_{i=1}^{\tau-1} \int_{t_i}^{t_{i+1}} \left(\sum_{j=0}^{3} \left(a_{i,j}^{(g)} - a_{i,j}^{(h)}\right)(t-t_i)^j\right)^2 dt \\ &= \sum_{i=1}^{\tau-1} \int_{t_i}^{t_{i+1}} \sum_{j=0}^{3}\sum_{k=0}^{3} \left(a_{i,j}^{(g)} - a_{i,j}^{(h)}\right)\left(a_{i,k}^{(g)} - a_{i,k}^{(h)}\right)(t-t_i)^{j+k} dt \\ &= \sum_{j=0}^{3}\sum_{k=0}^{3} \frac{\left(a_{i,j}^{(g)} - a_{i,j}^{(h)}\right)\left(a_{i,k}^{(g)} - a_{i,k}^{(h)}\right)}{j+k+1} \sum_{i=1}^{\tau-1} (t_{i+1}-t_i)^{j+k} \end{aligned} \tag{5.22}$$

□

5.4 Linkage Methods

No matter which distance measure is selected, it just provides the distance between single genes, not between clusters. This cluster distance is known under the nomenclature *linkage* and is needed by some clustering methods as the hierarchical clustering (see section 5.5.2).

A few common choices for linkage are presented in this work, for more methods please refer [Kre08].

5.4.1 Complete and Single Linkage

These methods are the easiest ways to derive a cluster distance from the distances of its elements. Therefore, the extrema of distances between the elements of the two sets are taken to define the distance of the sets themselves. More precise:

Definition 5.6: ***Complete Linkage***
Let V be a vector space, $X_1, X_2 \subset V$ be two finite and non-empty sets and let $d : V^2 \to \mathbb{R}_0^+$ be a metric on V. Then the complete linkage distance of X_1 and X_2 is defined as

$$d_{CL}(X_1, X_2) = \min_{x_1 \in X_1, x_2 \in X_2} d(x_1, x_2) \tag{5.23}$$

The complete linkage is the probable most restricting meaningful linkage method, because the worst of the distances between all possible element combinations is taken for the cluster. Thus, an iterative merging of clusters with low distances - as it is done in agglomerative hierarchical clustering - results in clusters, which are quite spherical with respect to d.

Definition 5.7: ***Single Linkage***
Let V be a vector space, $X_1, X_2 \subset V$ be two finite and non-empty sets and let $d : V^2 \to \mathbb{R}_0^+$ be a metric on V. Then the single linkage distance of X_1 and X_2 is defined as

$$d_{SL}(X_1, X_2) = \max_{x_1 \in X_1, x_2 \in X_2} d(x_1, x_2) \tag{5.24}$$

While the complete linkage the is most restrictive, the single linkage is the loosest linkage method. Here the best combination of elements of the two clusters yields the cluster distance. Thus, the iterative merging might cause arbitrary widespread clusters as long as it contains a sufficiently dense chain between the elements.

Both methods, complete and single linkage, are easily computable. Once the distances between the individual elements are known, every cluster distance can be calculated simply by applying the minimum or maximum to all distances between the cluster elements. Alternatively, the cluster distances are also easily obtainable by recursion, which will be useful for an iterative merging:

Lemma 5.2: ***Update Formulas for Single Linkage and Complete Linkage***
Let V be a vector space, $X_1, X_2, X_3 \subset V$ be finite and non-empty sets and let $d : V^2 \to \mathbb{R}_0^+$ be again a metric on V. Then the following holds

$$\begin{aligned} d_{CL}(X_1, X_2 \cup X_3) &= \max(d_{CL}(X_1, X_2), d_{CL}(X_1, X_3)) \\ d_{SL}(X_1, X_2 \cup X_3) &= \min(d_{SL}(X_1, X_2), d_{SL}(X_1, X_3)) \end{aligned} \tag{5.25}$$

5.4.2 Average and Centroid Linkage

Between the very loose single linkage and the very tight complete linkage there are various suggestions to calculate distances of clusters.

Two very intuitive approaches are the average and the centroid linkage.

The average linkage is the arithmetic mean of all possible distances between elements of the two clusters:

Definition 5.8: ***Average Linkage***
Let V be a vector space, $X_1, X_2 \subset V$ be two finite and non-empty sets and let $d : V^2 \to \mathbb{R}_0^+$ be a metric on V. Then the average linkage distance of X_1 and X_2 is defined as

$$d_{AL}(X_1, X_2) = \frac{1}{|X_1|\,|X_2|} \sum_{x_1 \in X_1} \sum_{x_2 \in X_2} d(x_1, x_2) \tag{5.26}$$

Lemma 5.3: ***Update Formula for Average Linkage***
Let V be a vector space, $X_1, X_2, X_3 \subset V$ be finite and non-empty sets and let $d : V^2 \to \mathbb{R}_0^+$ be again a metric on V. The average linkage can be calculated recursively by

$$d_{AL}(X_1, X_2 \cup X_3) = \frac{|X_2|}{|X_2| + |X_3|} d_{AL}(X_1, X_2) + \frac{|X_3|}{|X_2| + |X_3|} d_{AL}(X_1, X_3) \tag{5.27}$$

Proof. see [Kre08]. □

Alternatively the distance between the means of the cluster can be used, which is known as centroid linkage.

Definition 5.9: ***Centroid Linkage***
Let $X_1, X_2 \subset V$ be two finite and non-empty sets and let $d : V^2 \to \mathbb{R}_0^+$ be a metric on V. The centroids are given by the arithmetic means of the elements μ_{X_1} and μ_{X_1}. Then the centroid linkage distance of X_1 and X_2 is defined as

$$d_{CenL}(X_1, X_2) = d(\mu_{X_1}, \mu_{X_2}) \tag{5.28}$$

For the centroid linkage the cluster centroid has to be compatible to the distance measure. This is not always the case as it can easily be seen in the following example.

Example 5.1:
Consider the two vectors

$$a = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \quad b = \begin{pmatrix} -2 \\ -0.9 \\ -1 \end{pmatrix} \tag{5.29}$$

Their distance based on Pearson's r is

$$\begin{aligned} d_{Pearson}(a, b) &= 1 - |r(a, b)| \\ &= 1 - |-0.997| \\ &= 0.003 \end{aligned} \tag{5.30}$$

The centroid yields distances

$$\begin{aligned} d_{Pearson}\left(a, \tfrac{a+b}{2}\right) &= 0.5 \\ d_{Pearson}\left(b, \tfrac{a+b}{2}\right) &= 0.43 \end{aligned} \tag{5.31}$$

which means that the sets $X_1 = \{a\}$ and $X_2 = \{a,b\}$ have a centroid linkage distance of 0.5, although the elements themselves are very close to each other having a distance of at most 0.003.

The recommended distance measure for the centroid linkage is the Euclidean distance. However, the centroid might also be generalized to be compatible to other distance measures.

Lemma 5.4: ***Update Formula for Centroid Linkage***
Let $X_1, X_2, X_3 \subset \mathbb{R}^n$ be finite and non-empty sets and let d be the Euclidean distance on $\mathbb{R}^n$. The centroid linkage can be calculated recursively by

$$\begin{aligned} d_{CenL}(X_1, X_2 \cup X_3)^2 = & \quad \frac{|X_2|}{|X_2|+|X_3|} \quad d_{CenL}(X_1,X_2)^2 \\ & + \quad \frac{|X_3|}{|X_2|+|X_3|} \quad d_{CenL}(X_1,X_3)^2 \\ & - \quad \frac{|X_2|\cdot|X_3|}{(|X_2|+|X_3|)^2} \quad d_{CenL}(X_2,X_3)^2 \end{aligned} \tag{5.32}$$

Proof. see [Kre08]. □

5.5 Clustering Methods

Out of the vast variety of clustering methods, three common ones are presented below. For further clustering techniques please refer to the citations in this chapter or the bibliography at the end of this work.

5.5.1 K-Means

For the first time the term k-means appeared in 1967 in a work of MacQueen, while the idea itself as well as a corresponding algorithm existed already for about ten years. [Mac67]

The goal of k-means is the generation of at most $k \in \mathbb{N}$ clusters which have minimal within-cluster variance. More precise:
Let $X \subset \mathbb{R}^N$, $n = |X|$ with $0 < n < \infty$ and let $\| \|_2$ denote the Euclidean norm.
Find clusters $C_1, C_2, \ldots, C_k \subset X$, with $k \leq n$ such that the following holds:

$$\begin{aligned} C_i \cap C_j &= \emptyset \quad \text{for all } i \neq j \\ \bigcup_{i=1}^{k} C_i &= X \\ E(C_1, C_2, \ldots, C_k) &= \sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|_2^2 \\ &= \textit{minimum} \end{aligned} \tag{5.33}$$

where $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$ is the center of mass of cluster C_i and an empty sum, i.e. if there exist an $C_i = \emptyset$, is set to 0.

A heuristic hill-climbing algorithm for calculating a local minimum of optimization task was invented by Stuart Lloyd in 1957. Even today, more than half a century later, it is one of the most popular k-means clustering algorithms [ORSS06].

Algorithm 5.1: Lloyd's Algorithm for K-Means Clustering
Let $X \subset \mathbb{R}^N$ be the nonempty and finite set, which shall be clustered.

1) Choose maximal number of clusters $k \in \mathbb{N}$

2) Choose k initial cluster centers $c_1, \ldots, c_k \in \mathbb{R}^N$. (see remark 5.2)

3) For $i \in \{1, \ldots, k\}$,
   set $C_i := \emptyset$.

4) For all $x \in X$,
   $C_i := C_i \cup \{x\}$, where $i = \underset{j \in \{1,\ldots,k\}}{\arg\min} \left\| x - c_j \right\|_2$.
   (If i is not unique, take the lowest i yielding the minimum.)

5) For $i \in \{1, \ldots, k\}$,
   if $C_i \neq \emptyset$ set $c_i := \frac{1}{|C_i|} \sum\limits_{x \in C_i} x$.

6) If the exit condition is not fulfilled goto Step 3. (see remark 5.2)

Remark 5.2:

- Being a hill-climbing method, the Lloyd algorithm terminates in a local minimum depending on the initial seeding. Thus, the complete algorithm should be run more than once with different initializations. The clustering with lowest local minimum of the error measure $E(C_1, C_2, \ldots, C_k)$ is taken as final k-means clustering result.

- There is a variety of suggestions for selecting the initial cluster centers available in literature. In the following the seeding procedure proposed in [ORSS06] is presented. Exchange step 2 of Lloyds algorithm by

  2a) As first two seed centers $\hat{c}_1$ and $\hat{c}_2$ choose randomly two elements $x, y \in X$ with probability proportional to $\|x - y\|_2^2$.

  2b) For $i \in \{2, \ldots, k-1\}$
  take as next seed center $\hat{c}_{i+1}$ randomly an element $x \in X$ with probability proportional to $\min\limits_{j \in \{1,\ldots,i\}} \left\| x - \hat{c}_j \right\|_2^2$.

  2c) For $i \in \{1, \ldots, k\}$
  let $d_i = \min\limits_{j \neq i} \left\| \hat{c}_i - \hat{c}_j \right\|_2$ and

$$B_i = X \cap \left\{ x \in \mathbb{R}^n \mid \|x - \hat{c}_i\|_2 < \frac{d_i}{3} \right\} \tag{5.34}$$

The final centers for the initialization of Lloyds algorithm are given as center of mass of the balls B_i (called Ball-k-means):

$$c_i = \frac{1}{|B_i|} \sum_{x \in B_i} x \tag{5.35}$$

- Let E_i denote the error measure of the clustering in the i-th iteration of Lloyds algorithm. Then the following holds:

$$E_{i+1} \leq E_i \text{ for all } i \in \mathbb{N} \tag{5.36}$$

and

$$E_{i+1} = E_i \quad \Rightarrow \quad E_{i+j} = E_i \quad \text{for all } j \in \mathbb{N} \tag{5.37}$$

This yields two facts: Lloyds algorithm does not cycle and terminates after a finite number of steps.

The Lloyd algorithm is relatively efficient having a complexity of $\mathcal{O}(kmn)$, where k is the number of clusters as above, $n = |X|$ and m is the number of iteration steps of the algorithm, which is pretty low compared to n as experience has shown [HS05].

However, the k-means clustering has two notable flaws: Firstly the number of clusters has to be predefined. If it is unknown the complete algorithm has to be run with different numbers, reducing its effectiveness significantly. Secondly the assignment of a data point to the nearest center is equivalent to the usage of a Voronoi diagram. Each cluster is contained in the Voronoi cells of its center, which means that all clusters are convex. Thus, non-convex cluster shapes cannot be detected and outliers are assigned to the cluster of a boundary Voronoi cell - ignoring the distance to the corresponding center.

Closing this section it shall be mentioned that many variants of k-means are available in literature, as e.g. the k-medoids clustering method which uses elements from the data set X instead of the arithmetic centers of mass as cluster centers.

5.5.2 Hierarchical Clustering

Most hierarchical clustering methods are agglomerative, which means they start with atomic clusters merging to larger ones until a quality criterion is met.

Therefore, let (V,d) be a metric space and $X \subset V$ with $n = |X|$, $0 < n < \infty$, the data set to be clustered. Furthermore, let d_L be a linkage method (cf. 5.4) compatible to d.

Then the hierarchical clustering algorithm is as follows:

Algorithm 5.2: Hierarchical clustering algorithm

1) Let $\Xi = \{X_1, \ldots, X_n\}$ the set of all atomic clusters of X, i.e.:

$$\begin{aligned} |X_i| &= 1 \\ S &= \bigcup_{i=1}^{k} X_i \end{aligned} \tag{5.38}$$

2) Find $i, j \in \{1, \ldots, n\}$, $i \neq j$, such that

$$d_L\left(X_i, X_j\right) \leq d_L\left(X_k, X_l\right) \text{ for all } k, l \in \{1, \ldots, n\}, k \neq l \tag{5.39}$$

3) Update Ξ by

$$\Xi := \Xi \setminus \left\{X_i, X_j\right\} \cup \left\{X_i \cup X_j\right\} \tag{5.40}$$

4) If the exit condition is not fulfilled goto Step 2. (see remark 5.3)

Remark 5.3:

- The hierarchical clustering needs no predefined number of clusters, which has an important advantage. Instead of stopping the algorithm when a quality criterion is met, it can be run until $\Xi = \{X\}$. If the values of i, j and $d_L\left(X_i, X_j\right)$ are stored in each iteration, this yields a hierarchy tree, called dendrogram (cf. figure 5.2). Once this hierarchy is calculated, the number of clusters or the quality criterion can be selected without any recalculation.

- Note that the hierarchical clustering has no undo. Once clusters are merged, they cannot be separated anymore, even if it seems advisable in later iterations. This causes a lack of robustness.

- The only step which is expensive with respect to calculation time is the initialization. For the first iteration step the distances between each two elements of the data set S have to be computed. The results are stored in a triangular distance matrix $D \in \mathbb{R}^{n \times n}$, where $D_{ij} = d_L\left(X_i, X_j\right)$ where $1 \leq i < j \leq n$. Using D, step 2 of the algorithm becomes

  2a) $\{i, j\} = \arg\min_{i<j} D_{ij}$
  (If $\{i, j\}$ is not unique, take the smallest i and j yielding the minimum.)

  2b) Update D by deleting the j-th row and column.

  2c) Update the i-th row and column of D using the appropriate recursive linkage update (cf. lemma 5.2, 5.3 and 5.4)

- The calculation of the distance matrix has an complexity of $\mathcal{O}\left(n^2\right)$, which is usually significantly more than the complexity of the k-means.

5.5.3 Density-Based Spatial Clustering of Applications with Noise

This density-based method, abbreviated as DBSCAN, was developed by Martin Ester et al. in 1996. [EKSX96] It uses a slightly different notion of clusters: A cluster is a maximal

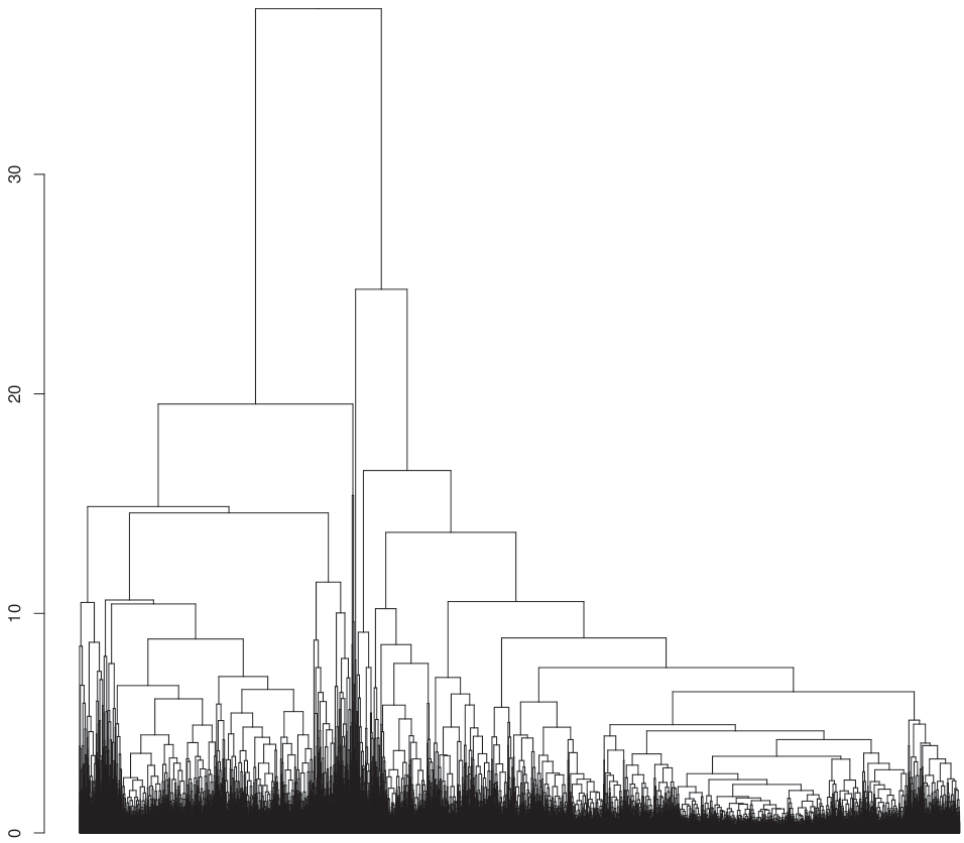


Figure 5.2: Dendrogram of the hierarchical clustering with complete linkage (abscissa: genes, no labeling for clarity reasons; ordinate: cluster dissimilarity)

(ε,μ)-density-connected set, which will be defined in the following. This will allow the detection of arbitrary shaped clusters.

Let again (V,d) be a metric space and $X \subset V$ be the set to be clustered, $n = |X|$ with $0 < n < \infty$.

Definition 5.10: ***Direct (ε,μ)-Density Reachability***
Let $\varepsilon \in \mathbb{R}^+$ and $\mu \in \mathbb{N}$. A point $y \in X$ is directly (ε,μ)-density reachable from $x \in X$ if

(1) $y \in B_\varepsilon(x)$

(2) x is a core point, i.e. $|B_\varepsilon(x) \cap X| \geq \mu$

where $B_\varepsilon(x) = \{v \in V | d(v,x) < \varepsilon\}$ is the open ε-neighborhood of x.

Definition 5.11: ***(ε,μ)-Density Reachability***
A point $y \in X$ is (ε,μ)-density reachable from $x \in X$ if there exists a chain

$$x = p_0, p_2, \ldots, p_k = y \tag{5.41}$$

with $k \in \mathbb{N}$, such that p_i is directly (ε,μ)-density reachable from p_{i-1} for all $i \in \{1,\ldots,k\}$.

Let

$$R_{\varepsilon,\mu}(x) = \{y \in X | y \text{ is } (\varepsilon,\mu)\text{-density reachable from } x\} \tag{5.42}$$

Definition 5.12: ***(ε,μ)-Density Connectivity***
A point $y \in X$ is (ε,μ)-density connected to $x \in X$ if there exists a $z \in X$ such that $x, y \in R_{\varepsilon,\mu}(z)$.

Please note that (ε,μ)-density reachability is not symmetric, because of the core-condition (2) in definition 5.10. This condition will ensure that density connected sets have a certain data concentration and thin connecting paths, so-called single links, are ignored as figure 5.3 shows.

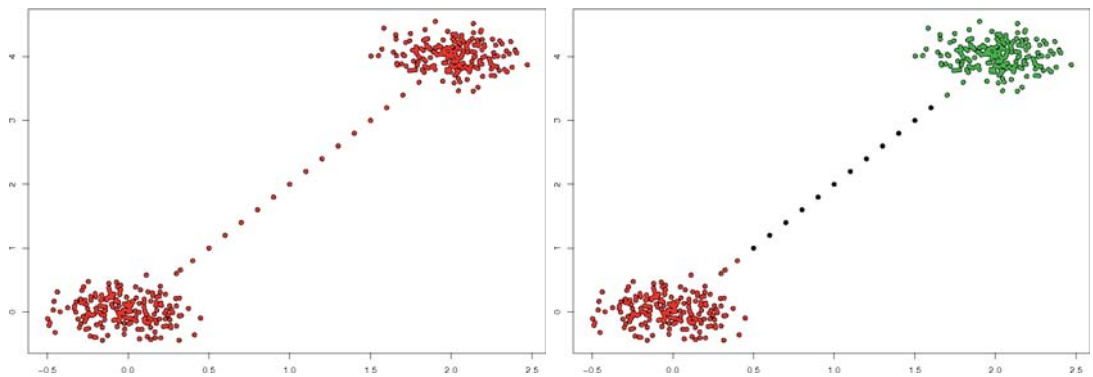


Figure 5.3: DBSCAN for $\mu = 2$ (left) and $\mu = 5$ (right, solid black dots denote noise); linkage line added artificially

Using the notion of (ε,μ)-density connectivity, the terms *cluster* and *noise* can be defined:

Definition 5.13: (ε,μ)***-Cluster***
A subset $C \subset X$ is called cluster if

(1) For all $x,y \in X$ holds:
If $x \in C$ and $y \in R_{\varepsilon,\mu}(x)$, then $y \in C$. (Maximality)

(2) For all $x,y \in C$ holds:
x is (ε,μ)-density connected to y. (Connectivity)

Due to the definition of (ε,μ)-density reachability each cluster contains at least one core point and thus at least μ data points. So there might remain points which neither have μ or more ε-neighbors nor have any ε-neighbor which is a core point itself. That means these points are not (ε,μ)-density reachable from any point and thus do not belong to any cluster, which gives rise to an additional subset of X collecting all those points:

Definition 5.14: (ε,μ)***-Noise***
Let $C_1,C_2,\ldots,C_m \subset X$ be all (ε,μ)-clusters of X. Then

$$N = X \setminus \left(\bigcup_{i=1}^{m} C_i \right) \tag{5.43}$$

is called (ε,μ)-noise of X.

Having these definitions it is not immediately obvious that the (ε,μ)-clusters of X are well-defined and even though how to generate them. Both is shown by the following lemma.

Lemma 5.5: ***Generation of*** (ε,μ)***-Clusters***
Let $C \subset X$ be a (ε,μ)-cluster and $x \in C$ a core point. Then

$$C = R_{\varepsilon,\mu}(x) \tag{5.44}$$

Proof. The inclusion $R_{\varepsilon,\mu}(x) \subset C$ holds due to the maximality of (ε,μ)-clusters.
Now let $y \in C$. Using the connectivity property of (ε,μ)-clusters there exists a $z \in C$ such that $x,y \in R_{\varepsilon,\mu}(z)$. That means there are chains

$$z = p_0, p_1, \ldots, p_k = x \tag{5.45}$$

and

$$z = q_0, q_1, \ldots, q_m = y \tag{5.46}$$

where $k,m \in \mathbb{N}$, $p_i, q_j \in X$ with p_i is directly (ε,μ)-density reachable from p_{i-1} and q_j is directly (ε,μ)-density reachable from q_{j-1} for $i \in \{1,\ldots,k\}$ and $j \in \{1,\ldots,m\}$. The directly (ε,μ)-density reachability is symmetric if both points are core points. Thus, p_{i-1} is also directly (ε,μ)-density reachable from p_i, for all $i \in \{1,\ldots,k\}$, since $p_k = x$ is a core point, too. That means that the following is a chain of directly (ε,μ)-density reachable points:

$$x = p_k, p_{k-1}, \ldots, p_1, p_0 = z = q_0, q_1, \ldots, q_m = y \tag{5.47}$$

Thus, $y \in R_{\varepsilon,\mu}(x)$, which yields the inclusion $C \subset R_{\varepsilon,\mu}(x)$, finishing the proof. □

Please note that the proof also has shown that (ε, μ)-density reachability is transitive and additionally symmetric if both points are core points.

The lemma shows that a cluster is uniquely defined by any of its core points, which will be used by the clustering algorithm.

Algorithm 5.3: DBSCAN
Let X be the non-empty, finite set to be clustered. Let $\varepsilon \in \mathbb{R}^+$ and $\mu \in \mathbb{N}$.

1) $D = \emptyset$ (set of visited points)
   $m = 0$ (number of clusters)

2) Take $x \in X \setminus D$

3) If $|B_\varepsilon(x) \cap X| \geq \mu$
   $m := m + 1$
   $C_m := R_{\varepsilon,\mu}(x) \setminus D$
   $D := D \cup C_m$
   else
   $D := D \cup \{x\}$

4) If $D \neq X$ goto step 2.

5) $N = X \setminus \left(\bigcup_{i=1}^{m} C_i \right)$

Remark 5.4:

- The calculation of the set $R_{\varepsilon,\mu}(x)$ goes as follows:
  a) $R_x := \{x\}$ (set of found reachable points)
     $D_x := \emptyset$ (set of visited points)
  b) Take $y \in R_x \setminus D_x$.
     $D_x := D_x \cup \{y\}$
     If $|B_\varepsilon(y) \cap X| \geq \mu$
     $R_x := R_x \cup (B_\varepsilon(y) \cap X)$
  c) If $R_x \neq D_x$ goto 2.
  d) $R_{\varepsilon,\mu}(x) = R_x$

- Note that it might happen, that two different clusters have common points, which are border points (i.e. no core points) of both clusters. In this case the points are assigned to the cluster with the lower index. Besides this rare case, the algorithm is deterministic.

- The selection of the two parameters ε and μ is the crucial point for this method. Zhou et al. propose $\mu = 2d$, where d denotes the dimension of the data space. [SAW+00]

Ester et al. propose a visual inspection method for choosing ε by plotting the values of the distances between each point and its μ-nearest neighbor and selecting manually a change point in the value series.

- The average runtime of DBSCAN is $\mathcal{O}(n \log n)$. [EKSX96], [SB05]

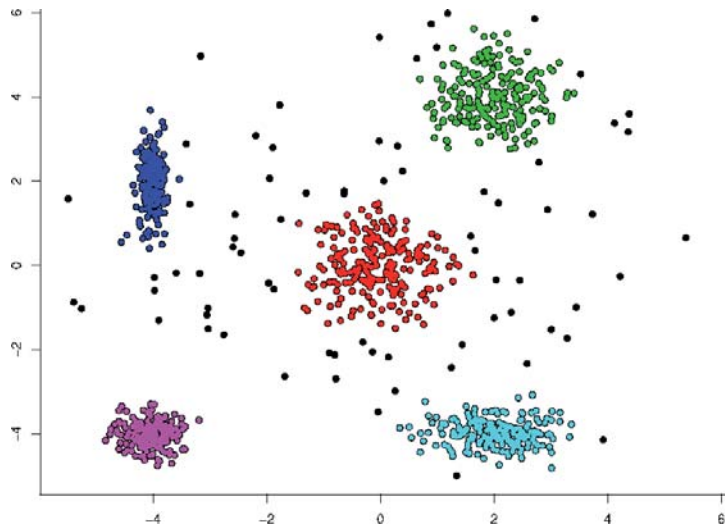


Figure 5.4: Clustering using DBSCAN, $\varepsilon = 0.4$, $\mu = 4$ (solid black dots denote noise)

The advantage of DBSCAN to find arbitrary shaped clusters also causes problems for the task of clustering gene time courses. For example, an expansion of a cluster along one dimension would give hints about the differentiation at one time point of otherwise parallel expressed genes. Furthermore, the freedom to exclude noise genes from the clustering is very helpful in detecting gene pathways and functional groups. The risk of genes which were accidentally assigned to a cluster, "just because no better one was available", is reduced.

However, the parameters have to be chosen carefully to ensure, that the clusters are not to widespread along one or more dimensions. Finally, the genes put into the noise set stay separated from the others, which would increase the dimension and thus the complexity of the model which shall be calculated in the next chapter.

Therefore, for the special task of generating a gene interaction network, these methods will not be further examined.

5.6 Quality Measures and Cluster Validation

Most clustering algorithms need a decision how many clusters shall be taken (e.g. k-Means) or where the cluster tree shall be cut (e.g. hierarchical clustering). The determination of the

correct number of clusters is one of the crucial points, as important as the selection of the method itself.

The estimation of a good cluster number can be aided by using validity indices.

There are two types of cluster validity indices, internal and external ones. While the latter ones need additional knowledge of the data and a prior clustering for comparison, the internal indices are based solely on the cluster structures. In this work it is assumed that no prior knowledge about gene clusters, e.g. metabolic pathways, is given, thus only internal validity indices are applicable.

A clustering has to meet two quality requirements. Firstly a cluster should be in some way dense and well-separated from others. Secondly the clustering itself should be robust to small changes in the data set. Both properties are quite imprecise providing sufficient room for subjective interpretation. In this section mathematically formulated quality measures are presented. These are taken from [Kre08] where several more are available.

5.6.1 Separation Indices

These indices always compare within-cluster and in-between-cluster structures, but they do it in quite different ways. In the following two indices are presented exemplary.

Therefore, let $\Xi = \{C_1, \ldots, C_k\}$ be a clustering of the set X with $n = |X|$, $0 < n < \infty$.

Definition 5.15: ***Dunn Index***
The Dunn index is the ratio between the smallest inter-cluster and the largest intra-cluster distance, given by

$$\Delta_{Dunn} = \frac{\min\limits_{i \neq j} d_{SL}\left(C_i, C_j\right)}{\max\limits_{i \in \{1,\ldots,k\}} D\left(C_i\right)} \tag{5.48}$$

where d_{SL} denotes the single linkage distance and $D\left(C_i\right) = \max\limits_{x,y \in C_i} d\left(x,y\right)$ is the diameter of the cluster C_i with respect to the metric d.

The Dunn Index takes values from the interval $[0, \infty]$, the larger the better is the clustering. However, please note that it has two drawbacks: Due to the the diameter in the denominator, the Dunn index prefers clusters which are spherical with respect to d. Furthermore, the definition yields a singularity when the clustering is atomic, i.e. each data point builds is own cluster, which is obviously not desired.

Definition 5.16: ***Connectivity Index***
Let $X = \{x_1, x_2, \ldots, x_n\}$ and $\nu_j\left(x_i\right)$ denote the j-th nearest neighbor of x_i in X. The connectivity index is given by

$$\Delta_{Con} = \sum_{i=1}^{n} \sum_{j=1}^{p} \left[1 - \delta_C\left(x_i, \nu_j\left(x_i\right)\right)\right] \frac{1}{j} \tag{5.49}$$

where

$$\delta_C(x,y) = \begin{cases} 1 & \textit{if there exist an } i \in \{1,\ldots,k\} \textit{ such that } x,y \in C_i \\ 0 & \textit{otherwise} \end{cases} \tag{5.50}$$

The parameter $p \in \mathbb{N}$ denotes the number of nearest neighbors taken into account.

The connectivity index mapping into $[0,\infty)$ checks if each data point and its nearest neighbors are put to the same clusters and increases if that is not the case. Thus, the lower the index is, the better is the clustering.

5.6.2 Robustness Indices

These indices measure the robustness of the clustering with respect to deletion of experimental conditions. One robustness index, the Adjusted Figure of Merit is presented below.

The adjusted figure of merit was constructed by Yeung et al. in 2000 [YHR00]. For its calculation a clustering is run after one experimental condition (in this work: a time point) was removed from the data, i.e.: Let $X = \{x_1,\ldots,x_n\} \subset \mathbb{R}^d$ be the data set to be clustered. The e-th component of a $x \in X$ contains the value of this data point under the e-th experimental condition, call it $x(e)$. Now let $X^e = \left\{x_1^e,\ldots,x_n^e\right\} \subset \mathbb{R}^{d-1}$ where x_i^e coincides x_i with its e-th component deleted. Furthermore, let $\Xi^e = \left\{C_1^e,\ldots,C_k^e\right\}$ be a clustering of X^e.

Definition 5.17: ***Figure of Merit***
Let $\mathbb{I}_m^e$ be the index set of cluster C_m^e, i.e.:

$$i \in \mathbb{I}_m^e \Leftrightarrow x_i^e \in C_m^e \tag{5.51}$$

Then the figure of merit is defined to be

$$FOM(e,k) = \sqrt{\frac{1}{n}\sum_{m=1}^{k}\sum_{i\in\mathbb{I}_m^e}\left[x_i(e)-\mu_m(e)\right]^2} \tag{5.52}$$

where

$$\mu_m(e) = \frac{1}{|C_m^e|}\sum_{i\in\mathbb{I}_m^e} x_i(e) \tag{5.53}$$

denotes the arithmetic mean of the deleted e-th experimental condition of all data points of the cluster C_m^e.

The aggregate figure of merit is given by

$$FOM(k) = \sum_{e=1}^{d} FOM(e,k) \tag{5.54}$$

Then the aggregate figure of merit is an estimator of the robustness and thus for the predictive power of the clustering algorithm. However, due to the sums of squared errors in the formula, an increase of the number of clusters k will often result in a decrease of the FOM, which demands a adjustment. Therefore, Yeung et al. defined the adjusted figure of merit in a way ,which amends the index:

Definition 5.18: ***Adjusted Figure of Merit***
The adjusted figure of merit is given by

$$adjFOM(k) = \frac{FOM(k)}{\sqrt{\frac{n-k}{n}}} \tag{5.55}$$

This index maps into $[0,\infty)$. The lower the adjusted figure of merit is, the higher is the predictive power of the clustering algorithm with respect to the given data set.

5.7 Time Course of Clusters

After achieving a clustering $\Xi = \{C_1,\ldots,C_k\}$ the question remains, how to define their time course behavior based on their corresponding elements.

5.7.1 Cluster Medoid

A first possibility is the selection of one element as representative of the complete cluster. Obviously, this element should exhibit the lowest dissimilarity to the elements of C. More precise, given a cluster C take a representative $\hat{g} \in C$ such that

$$E_C\hat{g} \leq E_C g \text{ for all } g \in C \tag{5.56}$$

where E_C denotes an error function with respect to C. A common choice for this error function is a least squared sum based on the distance measure d used for the clustering itself:

$$E_C g = \sum_{c \in C} d(g,c)^2 \tag{5.57}$$

Then the element $\hat{g}$ is called medoid of C with respect to E. This method has the advantage, that the time course of the representative is already available. However, please note, that the medoid is not necessarily unique, which can be easily seen considering a cluster consisting of two disjoint elements.

5.7.2 Cluster Centroid

This method is related to the medoids, but does not force to choose the representative from the cluster set itself. Therefore, assume that C is a subset of a complete metric space X with metric d. The centroid $\hat{g} \in X$ shall fulfill

$$E_C\hat{g} \leq E_C g \text{ for all } g \in X \tag{5.58}$$

where again E_C denotes an error function with respect to C. Also for the centroid a common choice is the least squared sum based on the distance measure d used for the clustering itself:

$$E_C g = \sum_{c \in C} d(g,c)^2 \tag{5.59}$$

If d is the Euclidean distance the centroid becomes the arithmetic mean of the cluster elements. Please note again, that the centroid is not necessarily unique. As example consider the distance measure based on Pearson's r (5.17) and a one-elemental cluster $C = \{g\}$. Then $\hat{g} = \alpha g$ fulfills (5.58) for any $\alpha \in \mathbb{R} \setminus \{0\}$ (cf. (5.16)). Furthermore, the centroid has to be meaningful in the sense, that its time course has to be determinable.

5.7.3 Cluster Smoothing Spline

The concept of smoothing splines gives an additional opportunity for calculating a cluster time course. Having a finite number of genes in the cluster C their corresponding standardized measurements may be collected and treated as measurements of one single "meta gene" $\hat{g}$.

$$M_{\hat{g},t} := \bigcup_{g \in C} M^*_{g,t} \tag{5.60}$$

where

$$M^*_{g,t} = \left\{ s_m^{-1} m \,|\, m \in M_{g,t} \right\} \tag{5.61}$$

denotes the data standardized with respect to the statistic s_m^{-1} (cf. definition 5.2).

This data can be interpolated using median, bootstrapping and a smoothing spline as it was done for each single gene. However, if the number of genes in a cluster is large the presented exact bootstrapping method becomes too expensive and the conventional method has to be used.

5.8 Results

A detailed analysis of the quality of clustering algorithms, distance measures and internal validity indices on the basis of the considered Magnaporthe time course data is available in [Kre08]. Thus, in this section these results will be shortly summarized.

The validity indices do not identify clearly a correct number of clusters. Therefore, a trade-off between the decrease of clustering errors and the increase of expected model complexity, due to a growing number of clusters, has to be made.

Figures 5.5 and 5.6 shows a large gain in predictive power by additional clusters up to about 100 clusters. Then the slope of the indices approaches to zero. The hierarchical clustering is run using all described linkage methods: single linkage (SL), average linkage (AL), complete linkage (CL) and centroid linkage (CenL). The initialization of k-means and k-medoids was made using the results of the hierarchical clustering. These methods provide the best separation as well the most robust results. The same holds for clusterings using the spline distance as it is shown in appendix C.

Thus, for further computation the clustering result of the setup given in table 5.1 is taken. The resulting clusters are plotted in Appendix D.

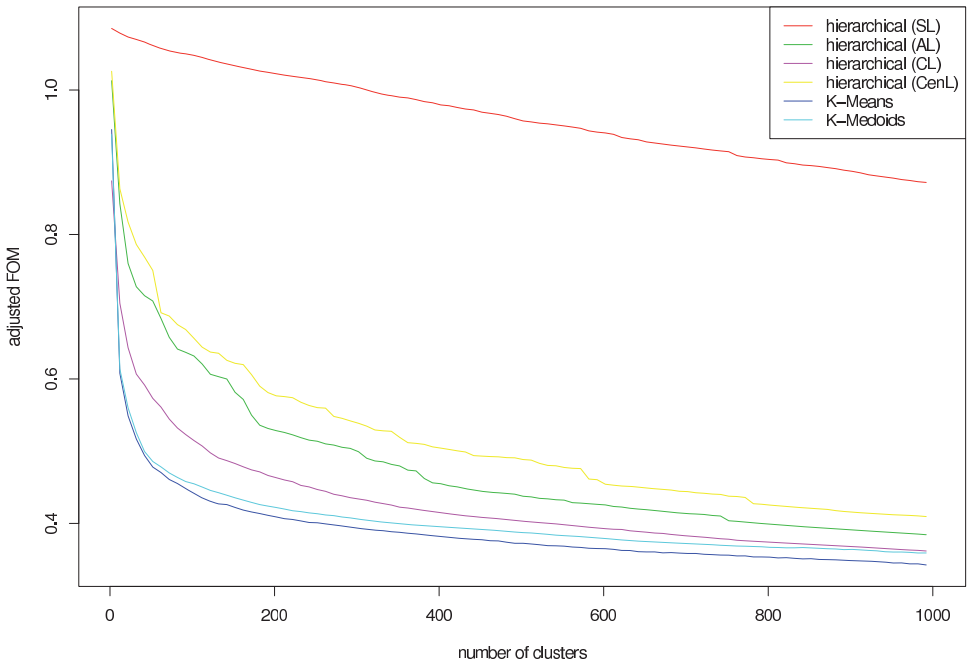


Figure 5.5: Adjusted Figure of Merit of hierarchical and k-means clustering using the Euclidean distance

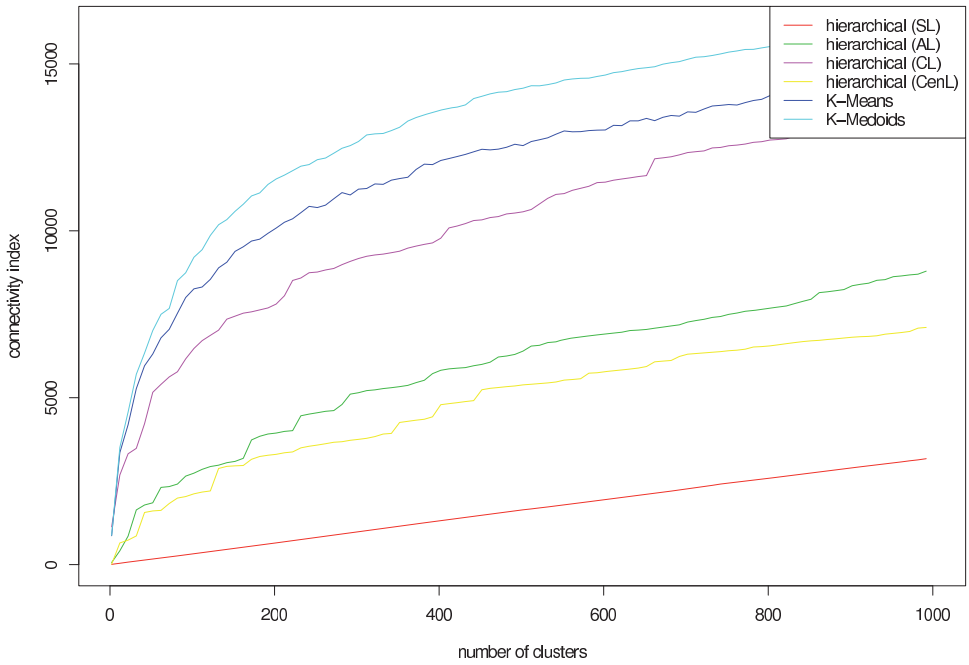


Figure 5.6: Connectivity index of hierarchical and k-means clustering using the Euclidean distance

Algorithm	k-means
Initialization	hierarchical clustering with complete linkage
Distance measure	euclidean distance
Cluster number	65

Table 5.1: Setup of the clustering taken for further calculation

For the following chapter, let $\Xi = \{C_1, \ldots, C_\xi\}$, $\xi \in \mathbb{N}$, with

$$\begin{aligned} X &= \bigcup_{i=1}^{\xi} C_i \\ C_i \cap C_j &= \emptyset \qquad \text{for all } i, j \in \{1, \ldots, \xi\}, i \neq j \end{aligned} \tag{5.62}$$

denote the clustering of the gene set X.

Let $\hat{g}_i$, $i \in \{1, \ldots, \xi\}$, be the representative of cluster C_i and its time course $S_{\hat{g}_i} : [t_1, t_\tau] \to \mathbb{R}$ shall be denoted by

$$S_{\hat{g}_i}(t) = \sum_{j=1}^{\tau-1} \delta_j(t)\, a_{j,3}^{(\hat{g}_i)} (t - t_j)^3 + a_{j,2}^{(\hat{g}_i)} (t - t_j)^2 + a_{j,1}^{(\hat{g}_i)} (t - t_j) + a_{j,0}^{(\hat{g}_i)} \tag{5.63}$$

with $\delta_j(t)$ as in (4.74).

6 Calculation of a Gene Interaction Network

Frustra fit per plura quod potest fieri per pauciora.
(It is futile to do with more things that which can be done with fewer.)

— William of Ockham, Franciscan monk (ca. 1285-1349)

In the last chapter genes of similar behavior were detected and bundled to clusters. Now the interaction of genes of different behavior respectively their clusters will be regarded.

Literature provides many differently complex models covering a full bandwidth (without this list being claimed to be exhaustive):

- Boolean or continuous with respect to the data
- Discrete or continuous in time
- Deterministic or stochastic
- Linear, piecewise linear or non-linear
- Regression, Neural Networks, Genetic Algorithms or Bayesian Methods

e.g. [D'h00], [JC04], [GHL05], [HMC+01], [JGHP03]

Knowing effects of the gene expression processes it seems obvious to chose a non-linear structure having the ability to model properties as e.g. saturation of the inner cellular mRNA concentration. However, as already mentioned in the introduction these non-linear approaches did not yield comparable results as the ßimplellinear approach does. The standard linear and time-invariant state space system turned out to approximate the data very well which will be shown in the following.

Therefore, let a clustering $\Xi = \left\{C_1, \ldots, C_\xi\right\}$ be calculated and the cluster time courses $S_{\hat{g}_i}$ for $i \in \{1, \ldots, \xi\}$ be given as in (5.63).

Accumulate these time courses as components to a vector-valued spline function

$$S : \begin{cases} [t_1, t_\tau] & \longrightarrow & \mathbb{R}^\xi \\ t & \longmapsto & \begin{pmatrix} S_{\hat{g}_1}(t) \\ \vdots \\ S_{\hat{g}_\xi}(t) \end{pmatrix} \end{cases} \tag{6.1}$$

The task of this chapter will be the calculation of a linear and time-invariant state space model, which generates S as trajectory vector.

6.1 Discrete Linear Time-Invariant State Space Model

In this section a discrete linear and time-invariant model for the interaction network will be generated. This network will have the standard state space representation which is defined as follows:

Definition 6.1: ***Discrete Linear Time-Invariant State Space Model***
Let $n, p, q, K \in \mathbb{N}$. A discrete linear and time-invariant dynamical system in state space form is given by

$$\left.\begin{array}{rcl} x_{k+1} & = & Ax_k + Bu_k \\ y_k & = & Cx_k + Du_k \end{array}\right\} \text{for } k \in \{1, \ldots, K\} \tag{6.2}$$

mapping an input sequence

$$u : \left\{ \begin{array}{ccc} \{1, \ldots, K\} & \longrightarrow & \mathbb{R}^p \\ k & \longmapsto & u_k \end{array} \right. \tag{6.3}$$

to an output sequence

$$y : \left\{ \begin{array}{ccc} \{1, \ldots, K\} & \longrightarrow & \mathbb{R}^q \\ k & \longmapsto & y_k \end{array} \right. \tag{6.4}$$

The matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times p}$, $C \in \mathbb{R}^{q \times n}$, $D \in \mathbb{R}^{q \times p}$ are called state matrix, input matrix, output matrix, and feed-through matrix. The sequence

$$x : \left\{ \begin{array}{ccc} \{1, \ldots, K+1\} & \longrightarrow & \mathbb{R}^n \\ k & \longmapsto & x_k \end{array} \right. \tag{6.5}$$

is called state sequence of the system.

Using the notation of the definition, the modeling task becomes:
Find matrices A, B, C, D of appropriate sizes, an initial state x_1, and an input sequence u such that the output y approximates a discretization of the vector-valued spline function S given in 6.1.

The values for these discretized trajectories are calculated by dividing the time interval $[t_1, t_\tau]$ equidistantly with step size h:

$$h = \frac{t_\tau - t_1}{K - 1} \tag{6.6}$$

Then the sequence s to be approximated by the discrete state space model by

$$s_k = S(t_1 + (k-1)h) \text{ for } k \in \{1, \ldots, K\} \tag{6.7}$$

Before the calculation of the state space system is done, the model is refined by the following additional assumptions:

1. The state of the fungus is given by the gene expression levels. Here, the representatives of the clusters have to be taken due to the indistinguishableness of the members in each cluster. Beside the cluster expression level, no further internal states will be assumed. Thus, $q = n = \xi$, the output matrix C becomes the identity, and the initial state x_1 is given by $s_1 = S(t_1)$.

2. An external stimulus of the fungus does not cause an immediate reaction. This will be modeled by claiming, that the delay has to be at least one time step. Therefore, the dynamical system has to be strictly causal, i.e. if two input sequences applied to the system coincide up to a time $k \in \mathbb{N}$ then the corresponding outputs coincide up to time $k+1$, independent of the values of the inputs at that time. Thus, the feed-through matrix D has to vanish.

3. The only external stimulus of the fungus is the initial application of the spores to the culture medium, where the fungus grows without any disturbances until it is taken for the mRNA extraction. This results in the assumption of a constant stimulus due to the nutrient supply, which will be modeled by a constant input $\hat{u} \in \mathbb{R}^p$. By defining $b = B\hat{u}$ the dynamical equation can be transformed as follows

$$\begin{aligned} x_{k+1} &= Ax_k + B\hat{u} \\ &= Ax_k + b \cdot 1 \end{aligned} \tag{6.8}$$

   Thus, the assumption of a constant input allows for the simplification $p = 1$ and $u_k = 1$, for all $k \in \{1, \ldots, K\}$. Therefore, the input matrix will be replaced by the column vector $b \in \mathbb{R}^\xi$ and the input function is scalar and constant: $u \equiv 1$.

Taken all this together results in the following model:

$$\left.\begin{aligned} x_{k+1} &= Ax_k + b \cdot 1 + \varepsilon_k \\ y_k &= x_k \end{aligned}\right\} \text{ for } k \in \{1, \ldots, K\} \tag{6.9}$$

where ε_k denotes the approximation error of the system in time step k.

The dynamical equation can be compressed further to

$$x_{k+1} = [\, A \mid b \,]\hat{x}_k + \varepsilon_k \text{ for } k \in \{1, \ldots, K\} \tag{6.10}$$

with $\hat{x}_k = \begin{pmatrix} x_k \\ 1 \end{pmatrix}$ for all $k \in \{1, \ldots, K\}$.

Because of the measurement errors the matrix has to be estimated by regression, which means: Find the matrix $[\, A \mid b \,]$ such that the sequence of approximation errors ε is minimal in the least square sense

$$E(\varepsilon) = \frac{1}{K} \sum_{k=1}^{K} (\|\varepsilon_k\|_2)^2 \rightarrow \text{minimum} \tag{6.11}$$

Therefore, transform (6.10) into the matrix notation

$$Y = [\, A \mid b \,] X + \mathscr{E} \tag{6.12}$$

where the states are collected in $X, Y \in \mathbb{R}^{n \times K}$, given by

$$\begin{array}{rcl} X & = & [\, \hat{x}_K \quad \mid \ldots \mid \quad \hat{x}_1 \,] \\ Y & = & [\, x_{K+1} \quad \mid \ldots \mid \quad x_2 \,] \end{array} \tag{6.13}$$

and the error matrix $\mathscr{E}$ contains the approximation error ε

$$\mathscr{E} = [\, \varepsilon_K \mid \ldots \mid \varepsilon_1 \,] \tag{6.14}$$

Each single row of equation (6.12) is a multiple linear regression task:

$$\left[\, \hat{x}_K^{(i)} \mid \ldots \mid \hat{x}_1^{(i)} \,\right] = [\, A_{i\cdot} \mid b_i \,] X + \left[\, \varepsilon_K^{(i)} \mid \ldots \mid \varepsilon_1^{(i)} \,\right] \tag{6.15}$$

The sum of error squares can be expressed by

$$\left\| \left[\, \varepsilon_K^{(i)} \mid \ldots \mid \varepsilon_1^{(i)} \,\right] \right\|_2^2 = \left(\left[\, \hat{x}_K^{(i)} \mid \ldots \mid \hat{x}_1^{(i)} \,\right] - [\, A_{i\cdot} \mid b_i \,] X \right) \left(\left[\, \hat{x}_K^{(i)} \mid \ldots \mid \hat{x}_1^{(i)} \,\right] - [\, A_{i\cdot} \mid b_i \,] X \right)^T \tag{6.16}$$

The solution of the minimizing problem is achieved by setting the derivative with respect to $[\, A_{i\cdot} \mid b_i \,]$ to zero, which results in:

$$[\, A_{i\cdot} \mid b_i \,] = \left[\, \hat{x}_K^{(i)} \mid \ldots \mid \hat{x}_1^{(i)} \,\right] X^T \left(X X^T \right)^{-1} \tag{6.17}$$

Further explanation will be skipped here, since several textbooks deal with multiple linear regression. Sachs et al. [SH09] presents the geometric interpretation as well as a complete algorithm for the calculation of the solution. However, the matrix division of Matlab does exactly the desired job, thus in following the regression will be done by just using the right matrix division (/) of Matlab. Therefore, the solution $[\, A \mid b \,]$ of the regression can calculated by

$$[\, A \mid b \,] = Y / X \tag{6.18}$$

For discussing the resulting dynamical system, several notions have to be defined:

Definition 6.2: ***Autonomy***
A standard state space system given as in (6.2) autonomous if the system output y is independent of the input u.

Remark 6.1:
The system (6.2) is autonomous if and only if the following holds:

Let an initial system state $x_0 \in \mathbb{R}^n$ and two input sequences $u^{(1)}$ and $u^{(2)}$ be given. Let $y^{(i)}$ denote the output corresponding to the initial state x_0 and the input $u^{(i)}$, for $i \in \{1, 2\}$. Then

$$y^{(1)} \equiv y^{(2)} \tag{6.19}$$

This means autonomy is equivalent to

$$\begin{aligned} B &= 0 \\ D &= 0 \end{aligned} \tag{6.20}$$

Definition 6.3: ***Stability***
Consider the state space system (6.2).

Let two initial system states $x_0^{(1)}, x_0^{(2)} \in \mathbb{R}^n$ and an input sequences u be given. Let $x^{(i)}$ denote the system state sequence corresponding to the initial state $x_0^{(i)}$ and the input u (for $i \in \{1,2\}$). Then the state space system is called stable if there exist an $M \in \mathbb{R}^+$ such that

$$\left\| x_k^{(1)} - x_k^{(2)} \right\| < M \text{ for all } k \in \mathbb{N} \tag{6.21}$$

The system is called asymptotically stable if additionally

$$\lim_{t \to \infty} \left\| x^{(1)} - x^{(2)} \right\| = 0 \tag{6.22}$$

Lemma 6.1: ***Stability of a Discrete Linear Time-Invariant State Space Model***
An autonomous system represented by

$$x_{k+1} = Ax_k \tag{6.23}$$

is stable if and only if all eigenvalues of A have modulus less than or equal to 1 and each eigenvalue λ with $|\lambda| = 1$ is semi-simple, i.e. its algebraic multiplicity equals its geometric multiplicity.

It is asymptotically stable if and only if all eigenvalues of A have modulus less than 1.

Proof. [Zer02] □

6.2 Results

The quality of the results depends strongly on the step size - as it also might be expected. Table (6.1) shows the main properties of resulting state space systems. The approximation of all cluster trajectories by the state space system generated with step size 1/40h is given in Appendix E.

Some interesting aspects of the results will be discussed below.

6.2.1 Autonomy

The first noteworthy fact is the autonomy of the resulting system. The matrix $[\,A \mid b\,]$ calculated by (6.18) has a zero column in the last position and thus b as well as the influence of the input function chosen in the beginning of the last section vanishes. This fact is quite interesting, since it yields no control possibility for the system behavior. The genes follow their trajectories without any external trigger as if the spore contains all energy needed for the gene activities during the fungus maturation. This matter should be followed up but not be overstated, since it remains a result of this concrete modeling approach.

step size (in hours) h	number of nodes K	system dimension n	auto-nomous?	largest eigenvalue λ_{max}	stable?	error $E(\varepsilon)$
1	25	21	X	1.0001		35.701
1/2	49	25	X	1.0000 + 0.0065i	X	55.026
1/3	73	27	X	1.0097		22.225
1/4	97	29	X	1.0083		55.720
1/5	121	30	X	1.0039		13.237
1/6	145	31	X	1.0036		28.867
1/7	169	31	X	1.0000		66.316
1/8	193	31	X	1.0000 + 0.0027i		17.568
1/9	217	31	X	1.0025		103.25
1/10	241	31	X	1.0000		23.522
1/20	481	31	X	1.0013		0.9470
1/30	721	31	X	1.0016		35.399
1/40	961	31	X	1.0000 + 0.0012i		**0.2614**
1/50	1201	31	X	1.0004		19.023
1/60	1441	31	X	1.0000 + 0.0002i		38.723
1/70	1681	30		1.0000 + 0.0003i		$3.11 \cdot 10^{15}$
1/80	1921	31	X	1.0000		51.287
1/90	2161	31	X	1.0003		203.55
1/100	2401	31	X	1.0002		6.3580
1/200	4801	31	X	1.0001 + 0.0012i		2363.5
1/300	7201	31	X	$1.0000 + 4.8 \cdot 10^{-8}$i	X	4.2784
1/400	9601	31	X	1.0000		327.02
1/500	12001	31	X	1.0000		50.873
1/600	14401	29	X	$1.0000 + 2.7 \cdot 10^{-8}$i		37.874
1/700	16801	30	X	$1.0000 + 4.8 \cdot 10^{-8}$i	X	251.50
1/800	19201	31	X	1.0000		3.2995
1/900	21601	30	X	1.0000		99.985
1/1000	24001	31	X	1.0000		1.8427

Table 6.1: Some results of the linear regression

6.2.2 Additional Dimension Reduction

The resulting system matrix $A \in \mathbb{R}^{65\times 65}$ is singular. Well approximating systems in table (6.1), i.e. those having an error $E(\varepsilon)$ less than 10, exhibit a system matrix with rank 31. Even those systems which approximate worse do not exceed that value. This yields a further reduction of system dimension. Therefore, assume without loss of generality that the last row of the resulting system matrix A is a linear combination the first r rows ($r \in \mathbb{N}$):

$$A_{n\cdot} = \sum_{i=1}^{r} \alpha_i A_{i\cdot} \tag{6.24}$$

Denoting the i-th component of the state vector at time k by $x_k^{(i)}$ yields

$$\begin{aligned} x_k^{(n)} &= A_{n\cdot} x_{k-1} \\ &= \left(\textstyle\sum_{i=1}^{r} \alpha_i A_{i\cdot}\right) x_{k-1} \\ &= \textstyle\sum_{i=1}^{r} \alpha_i \left(A_{i\cdot} x_{k-1}\right) \\ &= \textstyle\sum_{i=1}^{r} \alpha_i x_k^{(i)} \end{aligned} \tag{6.25}$$

Additionally, define $\alpha_n = -1$ and $\alpha_i = 0$ for all $i \in \{r+1, \ldots, n-1\}$. This results in

$$\begin{aligned} x_{k+1}^{(j)} &= A_{j\cdot} x_k \\ &= \textstyle\sum_{j=1}^{n} A_{ji} x_k^{(i)} \\ &= A_{jn} x_k^{(1)} + \textstyle\sum_{i=1}^{n-1} A_{ji} x_k^{(i)} \\ &= A_{jn} \left(\textstyle\sum_{i=1}^{r} \alpha_i x_k^{(i)}\right) + \textstyle\sum_{i=2}^{n-1} A_{ji} x_k^{(i)} \\ &= \textstyle\sum_{i=1}^{n} \left(A_{jn}\alpha_i + A_{ji}\right) x_k^{(i)} \end{aligned} \tag{6.26}$$

for all $j \in \{1, \ldots, n\}$. Using the matrix notation and the row vector $\alpha^T = (\alpha_1, \ldots, \alpha_n)$ the equation becomes

$$x_{k+1} = \underbrace{\left(A + A_{\cdot n}\alpha^T\right)}_{\hat{A}} x_k \tag{6.27}$$

That means there exists an equivalent dynamical equation which has only zeros in the last column, i.e. the influence of the linear dependent state component is removed. Which in turn allows the reduction of the dimension of the system matrix $\hat{A}$ by reintroducing the output matrix C. Therefore, let $\hat{A}_{\neg n} \in \mathbb{R}^{(n-1)\times(n-1)}$ be the matrix $\hat{A}$ where the n-th row and column was deleted and $\alpha_{\neg n}^T \in \mathbb{R}^{n-1}$ be the vector α^T where the n-th value was removed. Furthermore, let

$$C = \begin{pmatrix} I \\ \alpha_{\neg n}^T \end{pmatrix} \in \mathbb{R}^{n\times(n-1)} \tag{6.28}$$

This yields a state space system which generates the same trajectories as $x_{k+1} = Ax_k$ but having one dimension less:

$$\begin{aligned} \eta_{k+1} &= \hat{A}_{\neg n} \quad \eta_k \\ x_k &= C \quad \eta_k \end{aligned} \tag{6.29}$$

step size (in hours)	Approximation error $E(\varepsilon)$ after deleting cluster no. 1	2	3	4	5
1/10	10.942	23.063	9.587	35.274	23.114
1/20	0.944	242.18	55.516	6.757	0.938
1/30	35.146	34.814	108.92	15.446	34.795
1/40	0.259	0.258	10.393	18.347	0.257
1/50	18.873	4.243	32.347	2.851	18.714
1/60	38.385	38.156	1.102	0.108	38.097
1/70	$2.87 \cdot 10^{15}$	$2.76 \cdot 10^{15}$	$3.35 \cdot 10^{15}$	$3.48 \cdot 10^{15}$	$3.11 \cdot 10^{15}$
1/80	39.496	50.406	2.378	261.27	50.428
1/90	184.62	82.104	3.9	44.254	86.341
1/100	6.323	54.859	200.9	24.568	6.256

Table 6.2: Results of linear regression after deleting clusters

Applying this method to all linear dependent rows of the calculated system matrix A the system dimension can be boiled down to 31. This is especially noteworthy since the data was not easily separable and the correct number of clusters was not clearly determinable. Thus, selecting a too large number of clusters might be emended by this method.

6.2.3 Robustness with Respect to the Step Size

The step sizes yielding a good approximation are not completely data inherent, which can be seen if the same regression procedure is done after removing one cluster. In table 6.2 the approximation errors after deleting the first five clusters are exemplified in detail.

Several step sizes do not yield good approximations after deleting any cluster (especially 1/70 hours did not converge in any calculation), but some step sizes which did not cause good approximations in the original setting do fine after removing a cluster (e.g. 1/60 after removing cluster 4).

6.2.4 System Stability

A last comment should be given to the stability of the solutions. Only a few number of step sizes result in stable systems, i.e. system having a system matrix with eigenvalues inside the unit disc. Even fewer ones additionally exhibit an acceptable approximation, e.g. step size 12 seconds (=1/300 hours). Most of the results have one up to four instable eigenvalues. However, the absolute values of the largest eigenvalues λ_{max} of good approximating systems differ marginally from 1. $|\lambda_{max}|$ of the best approximation found (step size 90 seconds = 1/40 hours) is $1 + 2.0 \cdot 10^{-6}$.

7 Summary and Outlook

In nature we do not find words, but only the initial letters of words,
and if we then attempt to read them we find that the new so-called words are again merely the initial letters of other words.

— Georg Christoph Lichtenberg, German scientist and satirist (1742-1799)

This work described a complete experimental work flow for analyzing the first 24 hours of growth of Magnaporthe grisea starting from the design of experiments up to the resulting mathematical model of the gene interaction. In the following, each step will be shortly reviewed by recalling all assumptions and decisions made and summarizing the results. Additionally, pegs for further or refining work are discussed.

7.1 Design of Experiments

A two-step design was taken for the microarray measurements. In the first phase the gene expression levels of differently matured Magnaporthe grisea conidia were compared to the expression levels of dormant spores and significant differences were detected using the Fisher-Pitman-Test. In the time intervals, where the most changes in differently expressed genes occurred, additional time knots were inserted and the design was balanced with respect to all mature times.

Decisions and Assumptions

- The selection of the time knots of the first experiment phase was driven by visual inspection of the fungus growth. The knots of the second phase were taken such that they divide the corresponding intervals in an approximately equidistant manner. The final selection of the mature times was up to the biologists and their expertise, where more changes might occur. Obviously, this choice might be wrong and better time points might exist. However, assuming that only few genes exhibit a short-time expression, most genes with expression changes should be detected.

- The generation of mRNA for the microarray experiments took about half an hour. Thus, in fact each measurement at a specific mature time is an average behavior in this time

period, which means that fast regulating genes might disappear by blurring over time. Here again, long-term expressions in most of the relevant genes had to be assumed.

- The number of replicates of each mature time was set to 4 such that the Fisher-Pitman-Test was able to detect at least the most significant expression differences and the balance requirement was met, i.e. dye-swaps were hybridized equally often.

Results

The design of experiments is easily expandable, as the second phase of the two-step design has shown. Thus, experiments for new time points as well as additional microarrays for enhancing the data at existing time points can be embedded into this design. Furthermore, the minimal sample size for control and treatment groups needed for statistical analysis of time course experiments was calculated.

Possible further Work

Additional measurement data is the main requirement for all enhancements of the system analysis and modeling. An increase of the data for each time point allows for other statistical methods as e.g. a Bonferroni correction. New mature time points yield additional knots for the spline smoothing. Furthermore, measurements under different environmental conditions may result in differences of the expression level of genes which are indistinguishable up to now. This new information for the clustering procedure will increase the number of differentiated clusters.

7.2 Data Processing

The microarray data was normalized using within and in-between chip normalization techniques as e.g. the Lowess method. The resulting corrected data were statistically evaluated using the Fisher-Pitman-Test.

Decisions and Assumptions

- The normalization methods are well-known and the needed parameters were set to common values taken from literature.

- The Fisher-Pitman-Test was chosen since no assumption of normal data distribution was made. Four replicates and thus four data values were too few for a reliable well-founded decision of the distribution type. Furthermore, microarrays are known for their liability to exhibit outliers. Thus, a non-parametric hypothesis test was required. The

low number of values allowed a computational expensive, but more precise test which gave rise to the Fisher-Pitman-Test.

- An alpha risk of 5% was taken as sufficient and no Bonferroni-correction was applied due to the limited microarray resources. Thus, it should be kept in mind that the significance of each gene detected by the test should be verified using cheaper methods as the qRT-PCR before further analysis or biological work is done based on this single gene.

- In addition to the significance of the distribution differences obtained from the hypothesis test, a fold-change of at least factor 2 was taken as condition for a relevant change in expression.

Results

Using the Fisher-Pitman-Test as well as the fold-change condition yielded the following: Nearly one half of all Magnaporthe grisea genes (7174 out of 15170), about one sixth of the rice genes (1171 out of 6325), and 7 out of 1080 control genes are detected as expressed differently at least at one time point compared to time 0.

Possible further Work

As already mentioned, additional measurements are needed for further improvements of the statistical analysis. In this work, the control of false positives was mainly addressed, since for first screenings this error category is even more important than normally. Thus, enhancements should focus the decrease of false negatives, i.e. the increase of the statistical power.

7.3 Estimation of Gene Expression Time Courses

The normalized gene expression data were interpolated to generate continuous time courses. This was done by using cubic smoothing splines. The data points and their accuracy weights needed for this type of splines were derived from the replicated data using the median and an accuracy measure calculated via the exact bootstrap method, which is computationally cheaper than the standard bootstrap approach.

Decisions and Assumptions

- Since no normal distribution was assumed and outliers are suspected, the median was chosen instead of the arithmetic mean for values to be interpolated. This also gave rise to the bootstrap method for calculating the estimation accuracy, since the standard deviation is no adequate quality measure for the median.

- There are many possible interpolation methods for the data. Since no further knowledge about the shape of the gene expression time courses existed, the common approach using cubic splines was chosen. The fact that the data points are only estimates led to the concept of smoothing splines allowing interpolation weights based on the estimation accuracy. Additionally, standard splines interpolating exactly the data points generate more overshootings than smoothing splines. Not knowing the true time course, the assumption of more steady trajectories without artificially amplified peaks was made.

Results

In addition to yielding the exact accuracy of the median, the exact bootstrap technique is computationally less expensive than the standard bootstrap for normal microarray settings. The smoothing spline interpolation generates a well-founded and easily interpretable image of the gene time courses. Due to the polynomial character of the splines, the usage in further calculation steps, e.g. for the L^p-metric, is exceptionally simple.

Possible further Work

The exact bootstrap technique is given for any statistic. However, if another one is chosen, it should be checked, whether the algorithm for general statistics can be improved as it is done for the median in this work.

The interpolation quality of the smoothing splines increases if further time knots are added. Especially time course peaks which might be flattened due to the smoothing would be higher, if they are supported two or more adjacent knots.

7.4 Data Clustering and Modeling of the Interaction Network

The genes are clustered with respect to expression time course similarities. The resulting clusters were taken to be approximated by the dynamical system. That means, in fact, the gene interaction network is only a *cluster interaction network*. Based on the data and no additional information available, there was no possibility to distinguish parallel expressed genes as regulatory factors or other genes.

Decisions and Assumptions

- Based on the cluster validation indices, the K-Means/K-Medoid clustering method performed best. The distance measures did not influence the results significantly, thus, the

Euclidean distance was chosen. The optimal number of clusters was not clearly derivable from the indices, but a number below 100 could be assumed. Finally, it was set to 65.

- The discrete linear and time-invariant state space system for modeling the interaction network was taken.
- No hidden states, a constant input, and strict causality were assumed due to biological and simplicity reasons.

Results

The clustering yielded first impressions of gene correlations and common expression pathways which entailed useful information for further biological experiments. However, the genes were not clearly separable as the validity indices had shown.

The discrete linear time-invariant state space system approximated the cluster time courses very well. Three properties of the resulting state space systems which yielded good approximations were robust with respect to the step size:

1. The systems are autonomous. The states follow the desired trajectories without any additional input. The dormant spores already carry all energy and information for their normal growth and maturation. Even if this interpretation is comprehensible, it should be remembered that it is a result of the modeling approach.
2. The system matrices have rank 31. This especially qualifies the selection of the number of clusters.
3. Only few systems are stable, but the others exhibit a spectral radius of nearly 1. More precise, the modulus of the largest eigenvalue does not exceed 1 by more than a magnitude of 10^{-4}.

The system matrix of the state space system shows the dynamical interaction of the system states among each other. Since these states model the clusters, it also provides hints of activating or inhibiting regulatory interaction between cluster elements.

Possible further Work

Many genes cluster in common pathways which result in comparable time courses and therefore an indistinguishableness between them. Thus, there is no possibility to model the complex gene interaction but only the interaction of clusters. Gene expression measurements under different environmental conditions might separate several clusters. Especially treatments as e.g. drug stress, which are more severe influences than the simple growth might expose non-linear properties of the gene interaction. Then the more complex models as neural networks should be fetched again.

7.5 Epilogue

The task motivating this work was the understanding of relations between genes and their relevance during the mature process of Magnaporthe grisea.

Due to the number of genes and the complexity of their interaction, the results are far away from a complete gene interaction network. However, each analysis and modeling presented in this work yields additional knowledge about the genetic activity of Magnaporthe grisea during the maturation.

Thus, we did one step – one mathematical step – of a long journey to understand the gene activity of Magnaporthe grisea. Hopefully, there will be others running further experiments, measurements, and analysis settings to reveal the functionality of the fungus and stop the threat by the rice blast disease.

A Appendix: MA-Plots

In the following the MA-plots of all microarrays made for the time course experiment are given.

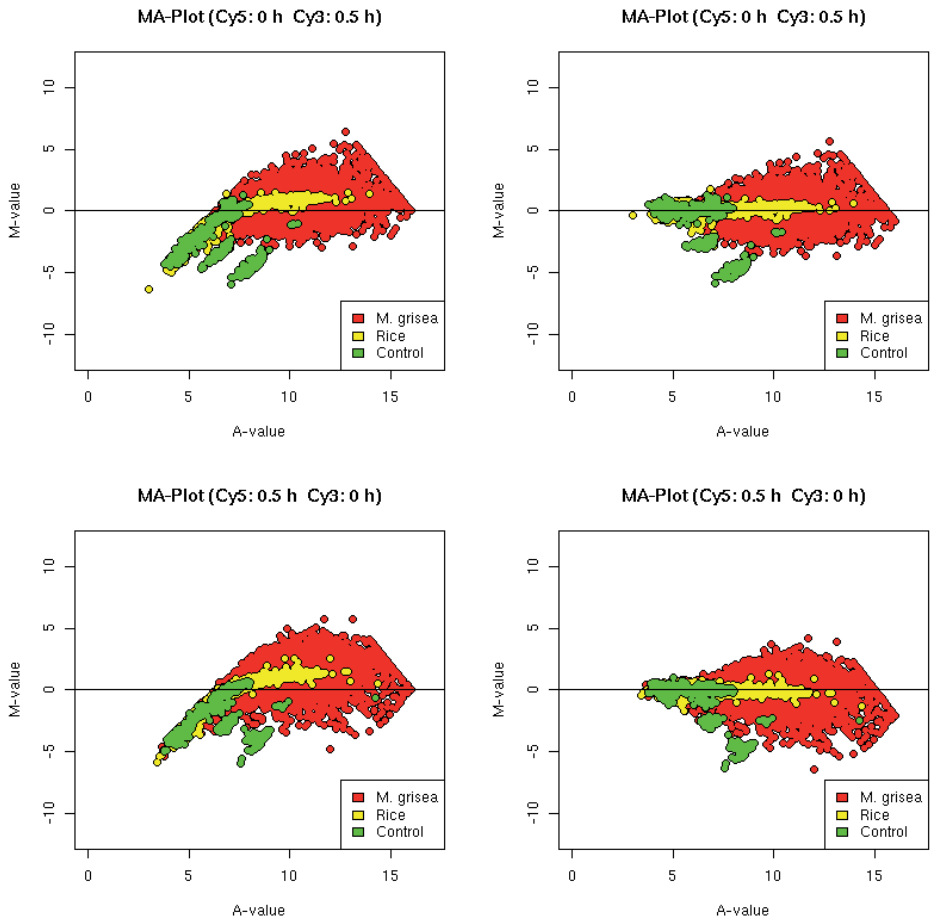


Figure A.1: MA-plots of raw microarray data (left) and its normalization (right)

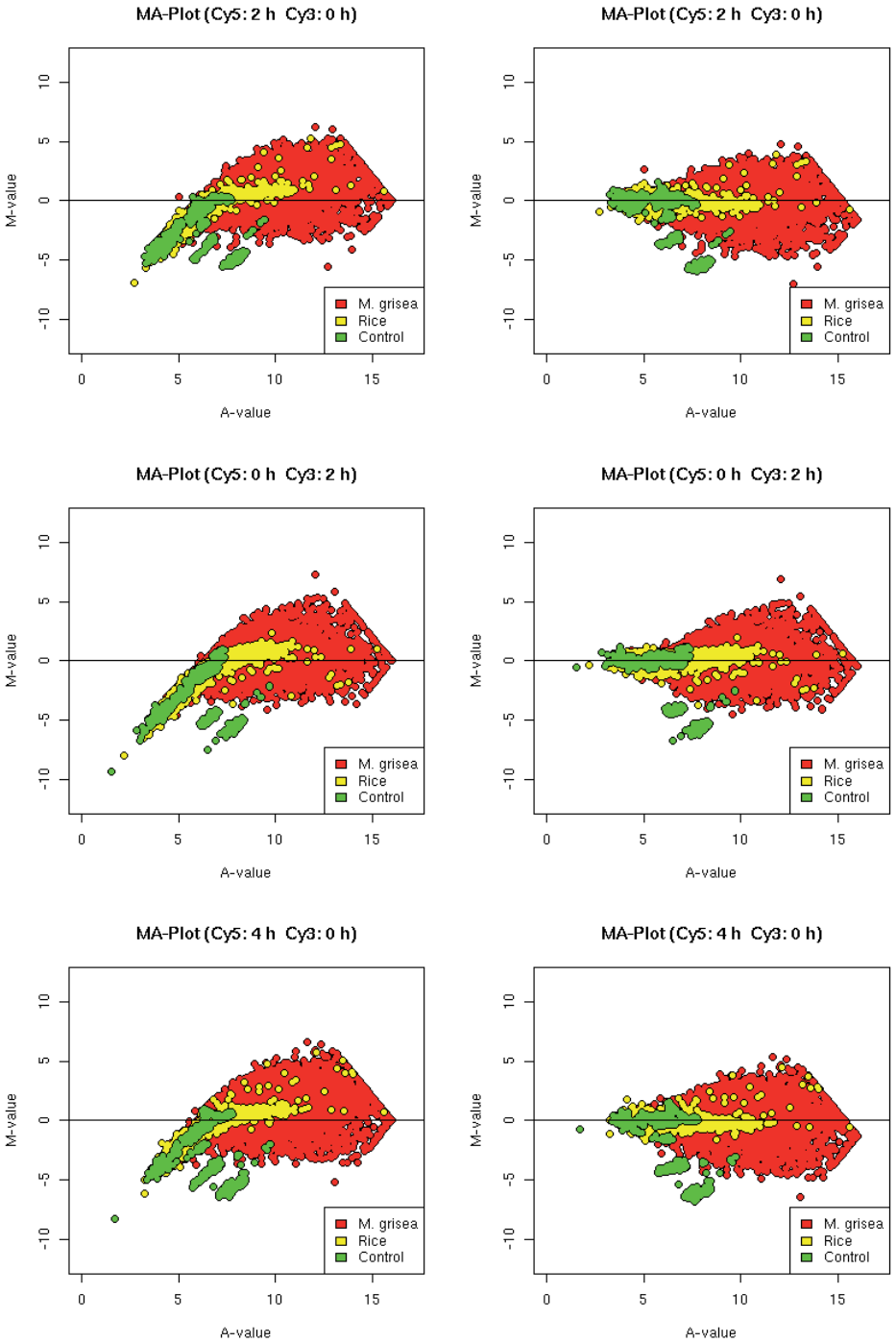


Figure A.2: MA-plots of raw microarray data (left) and its normalization (right)

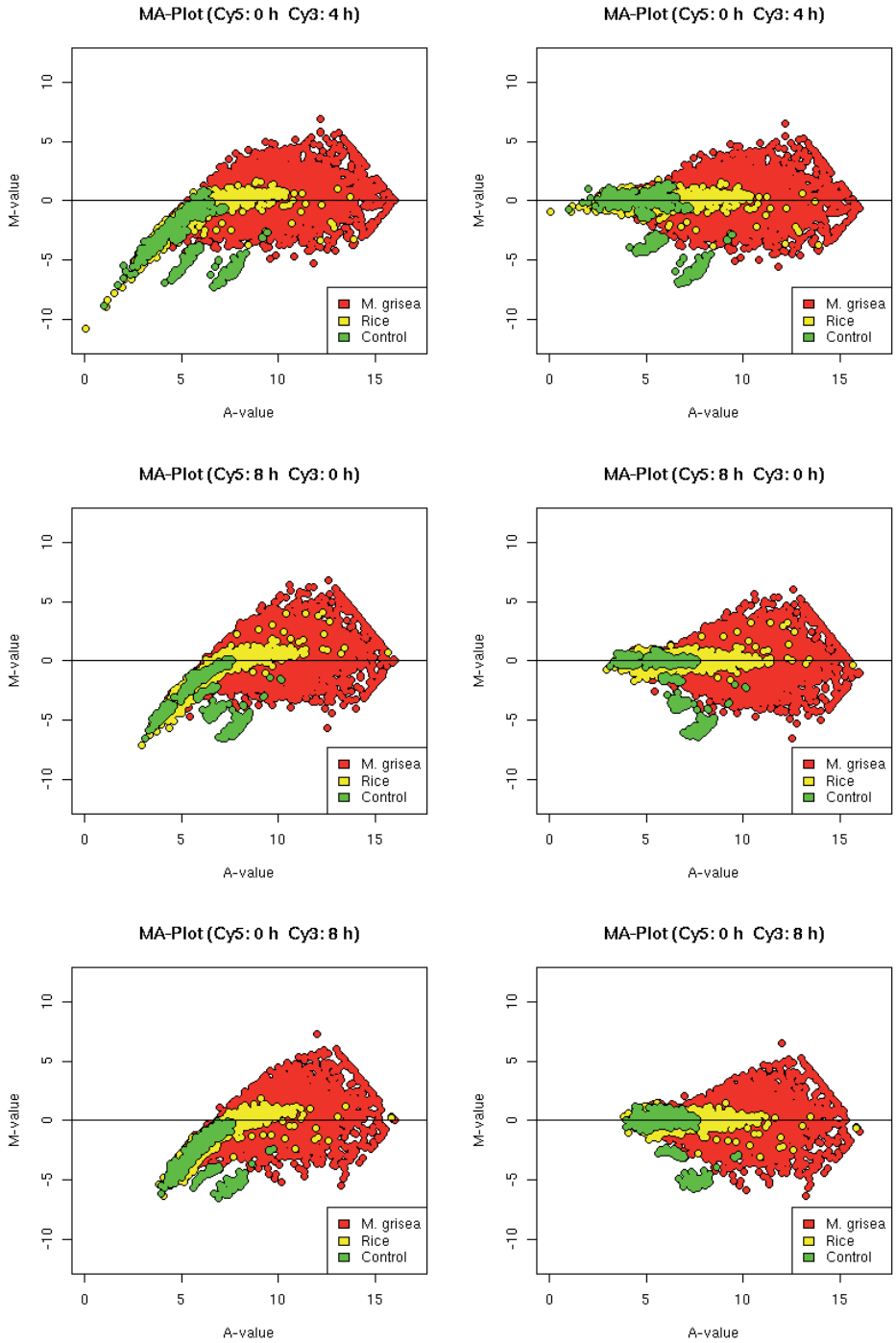


Figure A.3: MA-plots of raw microarray data (left) and its normalization (right)

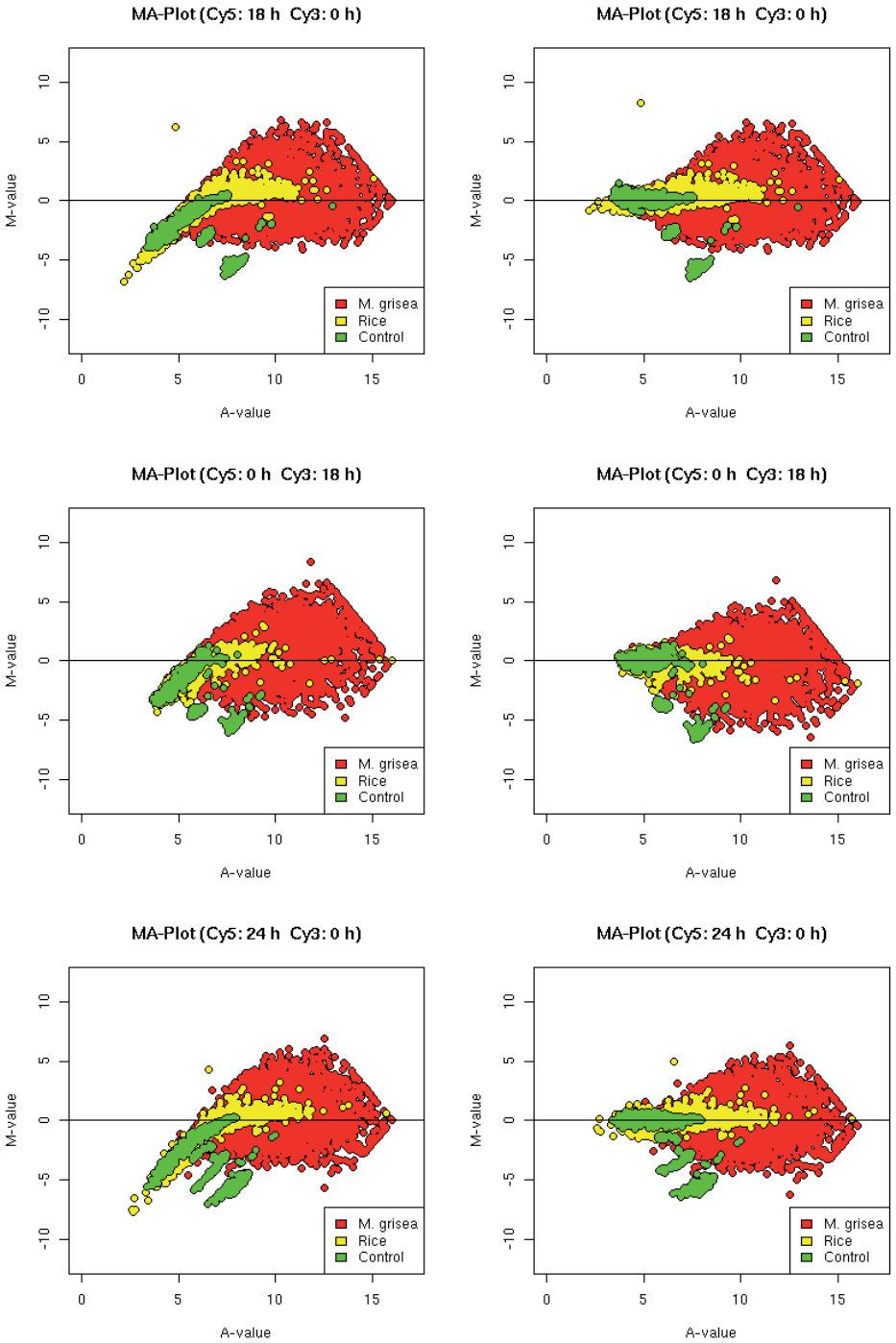


Figure A.4: MA-plots of raw microarray data (left) and its normalization (right)

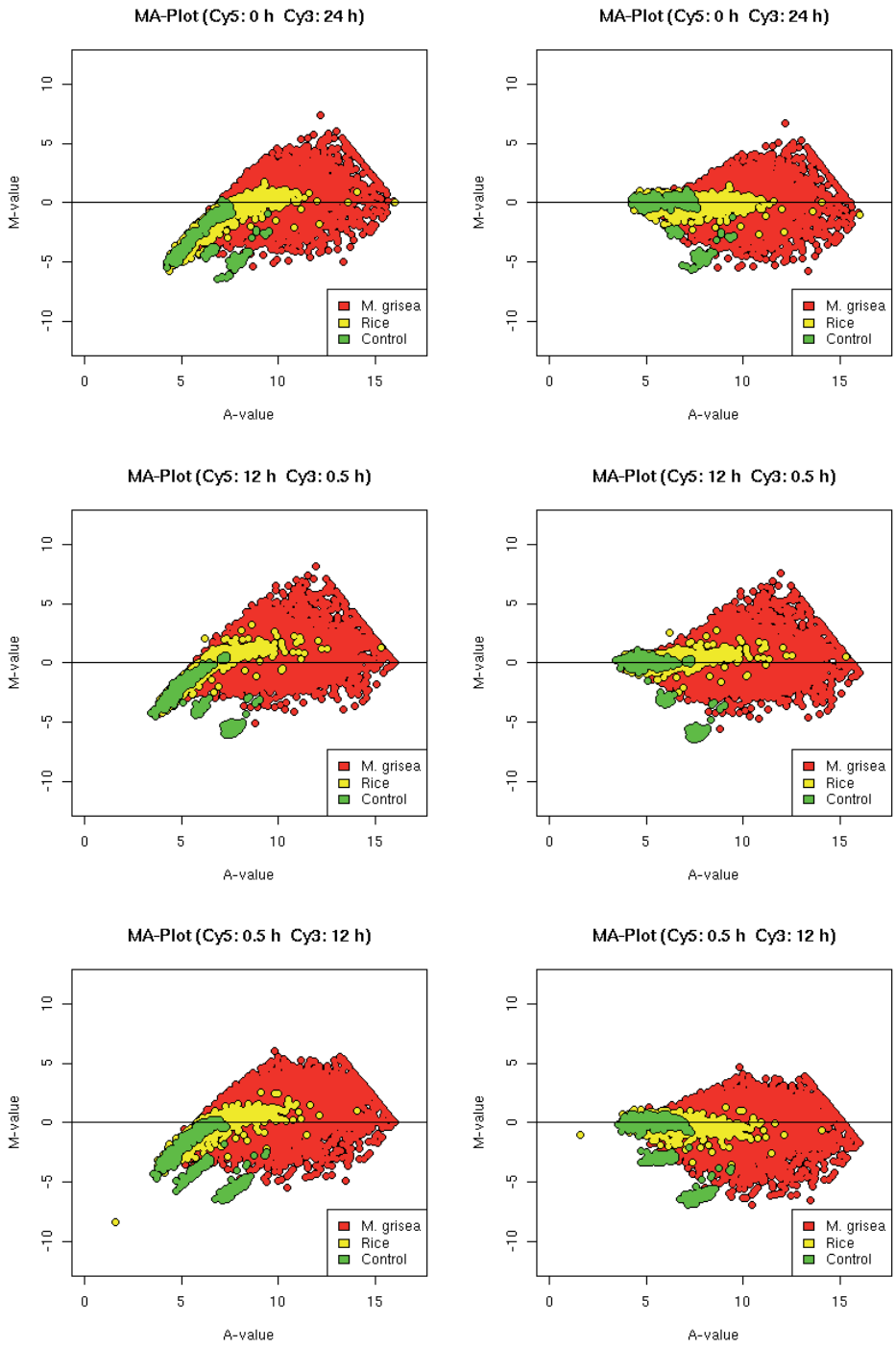


Figure A.5: MA-plots of raw microarray data (left) and its normalization (right)

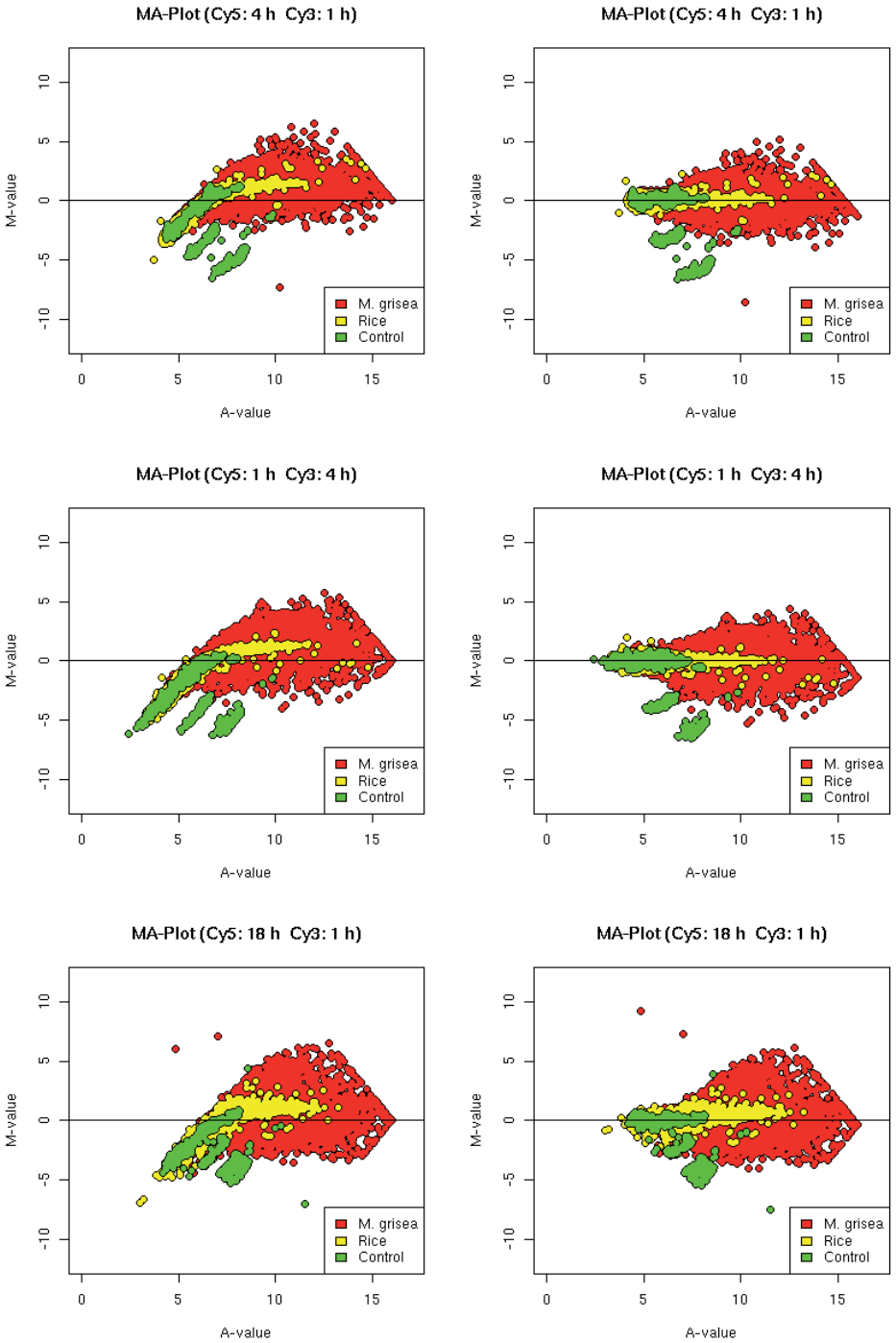


Figure A.6: MA-plots of raw microarray data (left) and its normalization (right)

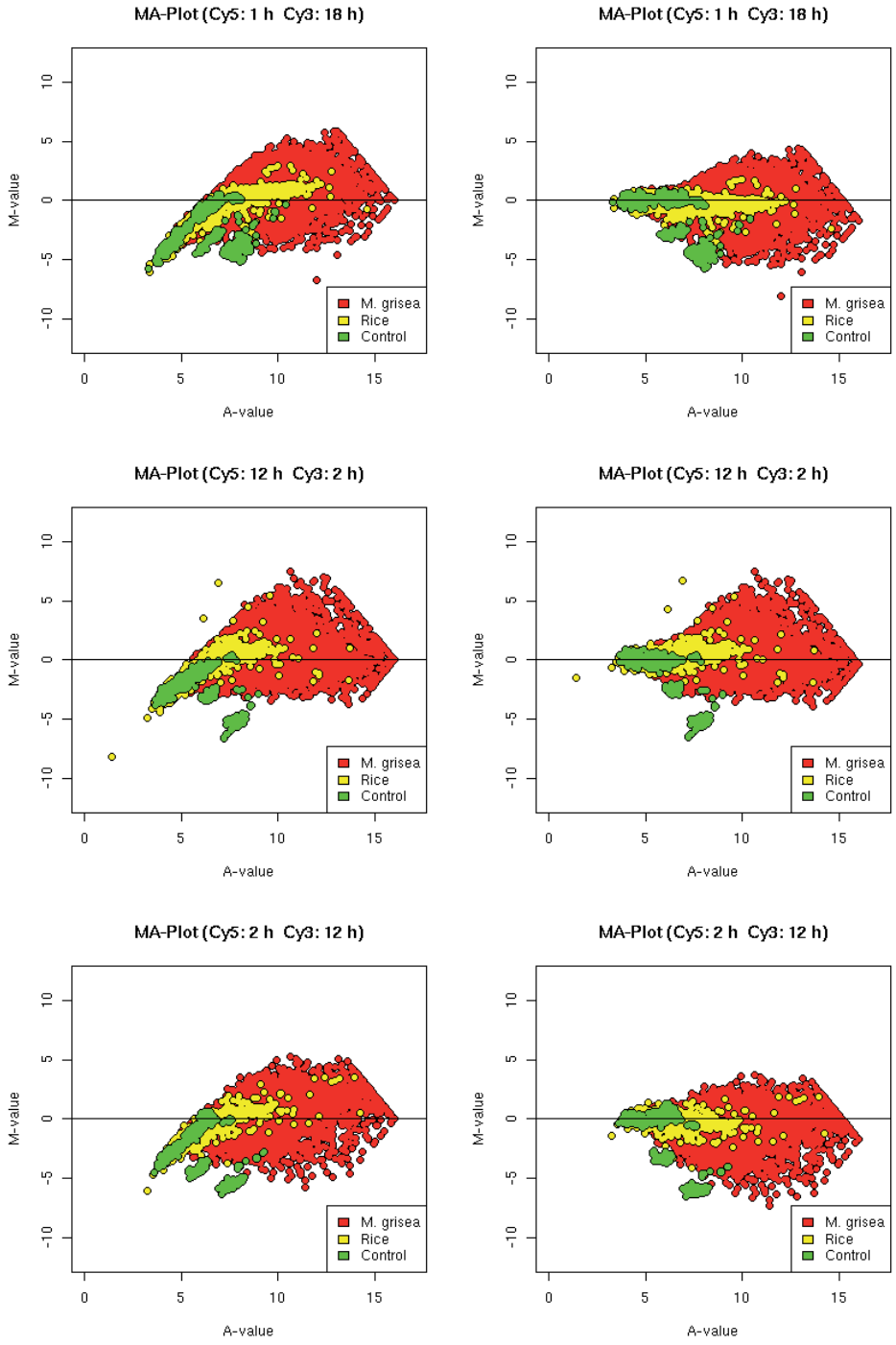


Figure A.7: MA-plots of raw microarray data (left) and its normalization (right)

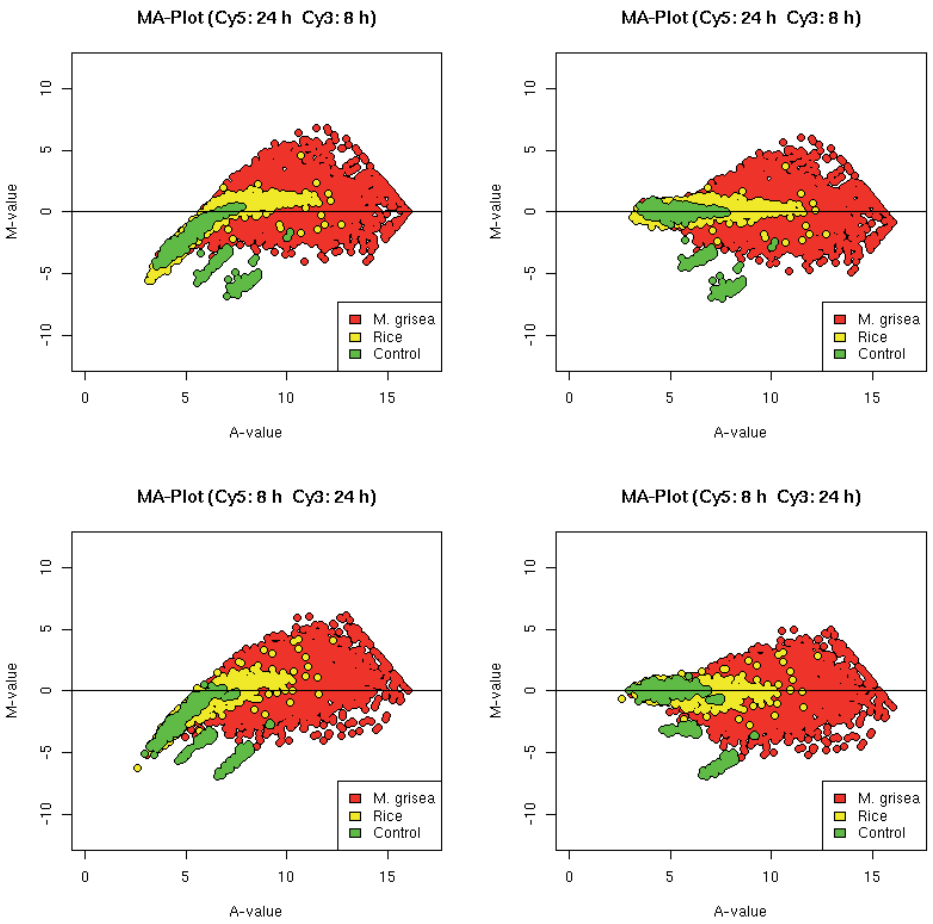


Figure A.8: MA-plots of raw microarray data (left) and its normalization (right)

B Appendix: An Algorithm for the Fisher-Pitman-Test

In the following an algorithm for calculating the Fisher-Pitman-Test for two non-empty real-valued data sets *set1* and *set2* is given using the the statistical programming language R.

```
fisher.pitman.test <- function(set1,set2) {
  if (length(set1) > length(set2)) {
    tmp <- set1
    set1 <- set2
    set2 <- tmp
  }
  if (median(set2) < median(set1)) {
    set1 <- - set1
    set2 <- - set2
  }
  set1 <- sort(set1,decreasing=TRUE)
  set2 <- sort(set2,decreasing=TRUE)
  nr.of.combs <- 1
  for (i in 1:length(set1)) {
    new.combs <- build.swap.set(set1,set2,i)
    if (new.combs == 0) {
      break
    } else {
      nr.of.combs <- nr.of.combs + new.combs
    }
  }
  P <- 1 - nr.of.combs/choose(length(set1)+length(set2),length(set1))
  P
}
```

The first two conditional statements ensure that *set1* is not longer than *set2* and contains the higher values (at least the higher median). This is done to reduce the number of needed steps during the search for extreme dichotomies (c.f. section 3.3.1).

These dichotomies are counted systematically by substituting elements of *set1* by elements from *set2* which is done by the function *build.swap.set*. In the first run of the for-loop only one element of *set1* is exchanged, in the next run two, and so on. If no substitution of $i \in \mathbb{N}$ elements yields an extreme dichotomy, there is also none with a higher number of substitutes which can easily be seen as follows:

Lemma B.1:
Let $m,n \in \mathbb{N}$ with $m < n$, $X = \{x_1,\dots,x_m\} \subset \mathbb{R}$, and $Y = \{y_1,\dots,y_n\} \subset \mathbb{R}$. Let further $S_x = \sum_{j=1}^{m} x_i$ and $\mathbb{P}_c$ denote the set of all permutations of the $\{1,\dots,c\}$.
Assume that there exists an $i \in \mathbb{N}$ with $i < m$ such that for any permutations $P_x \in \mathbb{P}_m$ and $P_y \in \mathbb{P}_n$ holds

$$\sum_{j=1}^{i} y_{P_y(j)} + \sum_{j=i+1}^{m} x_{P_x(j)} < S_x \textit{ for all } P_x \in \mathbb{P}_m, P_y \in \mathbb{P}_n \tag{B.1}$$

Then this equation holds also for any $k \in \mathbb{N}$ with $i < k \leq m$.

Proof. Without loss of generality let the sets X and Y be sorted in descending order, i.e. $x_a \geq x_{a+1}$ and $y_b \geq y_{b+1}$ for all $a \in \{1,\dots,m-1\}$ and $b \in \{1,\dots,n-1\}$.

It has to be proven, that even substituting the lowest k elements of X by the largest elements of Y does not improve the sum, more precise:

$$\sum_{j=1}^{k} y_j + \sum_{j=k+1}^{m} x_{m-j+1} < S_x \tag{B.2}$$

or equally

$$\sum_{j=1}^{k} y_j < \sum_{j=1}^{k} x_{m-j+1} \tag{B.3}$$

Using B.1 it holds especially that

$$\sum_{j=1}^{i} y_j + \sum_{j=i+1}^{m} x_{m-j+1} < S_x \tag{B.4}$$

which again reduces to

$$\sum_{j=1}^{i} y_j < \sum_{j=1}^{i} x_{m-j+1} \tag{B.5}$$

Since the sets are ordered this can be transformed to

$$iy_i < ix_{m-i+1} \tag{B.6}$$

Dividing by i and using again the order of the sets it holds that

$$y_j < x_{m-j+1} \text{ for all } j \in \mathbb{N} \text{ with } i < j \leq k \tag{B.7}$$

Together with (B.5) this proves (B.3) □

```
build.swap.set <- function(set1,set2,set.length,sel.elements=numeric(0)) {
  len <- length(sel.elements)
  if (len == set.length) {
    ret <- get.higher.combinations(set1,set2[sel.elements])
  } else {
    if (len > 0) {
      last.el <- sel.elements[len]
    } else {
      last.el <- 0
    }
    ret <- 0
    for (i in (last.el+1):(length(set2)-set.length+len+1)) {
      combs <- build.swap.set(set1,set2,set.length,c(sel.elements,i))
      if (combs == 0) {
        break
      } else {
        ret <- ret + combs
      }
    }
  }
  ret
}
```

build.swap.set counts the number of possible substitutions of the length given by the parameter *set.length* resulting in a higher sum. Therefore, it generates recursively a substitution. This generation is made systematically, which means the highest values of *set2* and the lowest elements of *set1* are chosen first. Thus, if the current substitution does not yield a higher sum, the later ones would not yield any either. So, the search can be stopped. This can be proven by simply applying lemma B.1 to the appropriate subsets.

```
get.higher.combinations <- function(set1, subset2, sum.diff=0) {
  pivot <- subset2[1]
  if (length(subset2) == 1) {
    ret <- length(set1[set1 <= pivot+sum.diff])
  } else if (length(set1) < length(subset2)) {
    ret <- 0
  } else {
    ret <- 0
    for (i in length(set1):2) {
      gain <- pivot - set1[i] + sum.diff
```

```
      if (gain < 0) {
        break
      }
      higher <- get.higher.combinations(set1[1:(i-1)],
                                        subset2[2:length(subset2)],gain)
      ret <- ret + higher
    }
  }
  ret
}
```

The input of the function *get.higher.combinations* is the complete *set1* and a *subset2* which shall be substitute the same number of elements from *set1*. This function counts the possible choices of subsets of *set1*, having the same number of elements as *subset2* and a lower sum than *subset2*. Therefore, the highest element of *subset2*, call it E, substitutes the lowest element of *set1*, call it e. The recursive call get.higher.combinations($set1 \backslash E$, $subset2 \backslash e$, $E - e$) searches for substitutions in the remaining elements, which cause a loss of at most $E - e$ which ensures that the overall sum of the substitution is not lower than the sum of *set1*. After that run, the highest element of *subset2* substitutes the secondary lowest element of *set1*, and so on.

C Appendix: Validation of Clusterings using the Spline Distance

In the following the connectivity index as well as the Adjusted Figure of Merit for the clustering results based on the spline distance are shown.

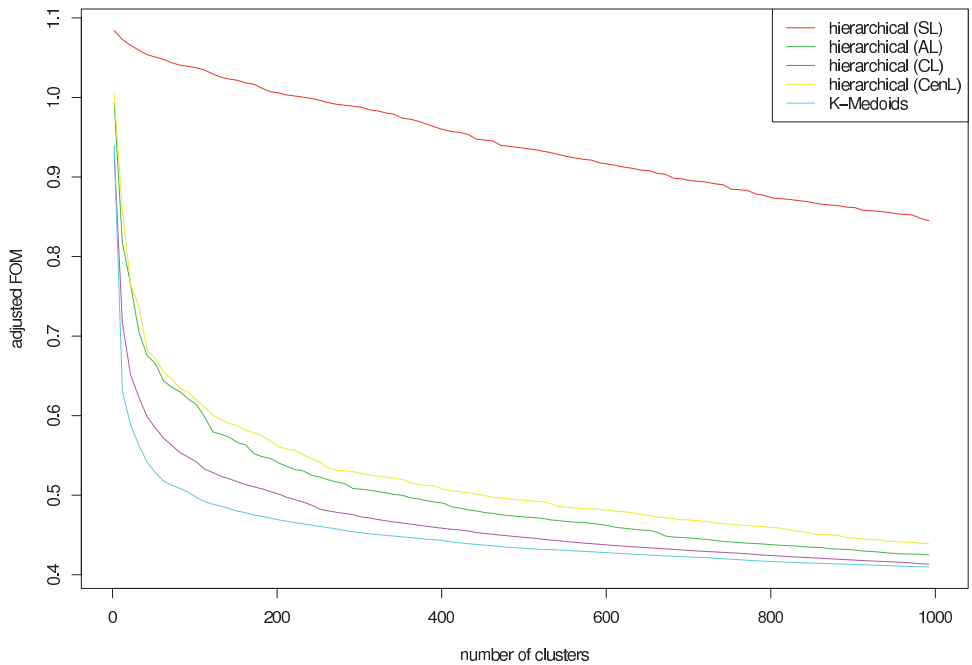


Figure C.1: Adjusted Figure of Merit of hierarchical and k-means clustering using the spline distance

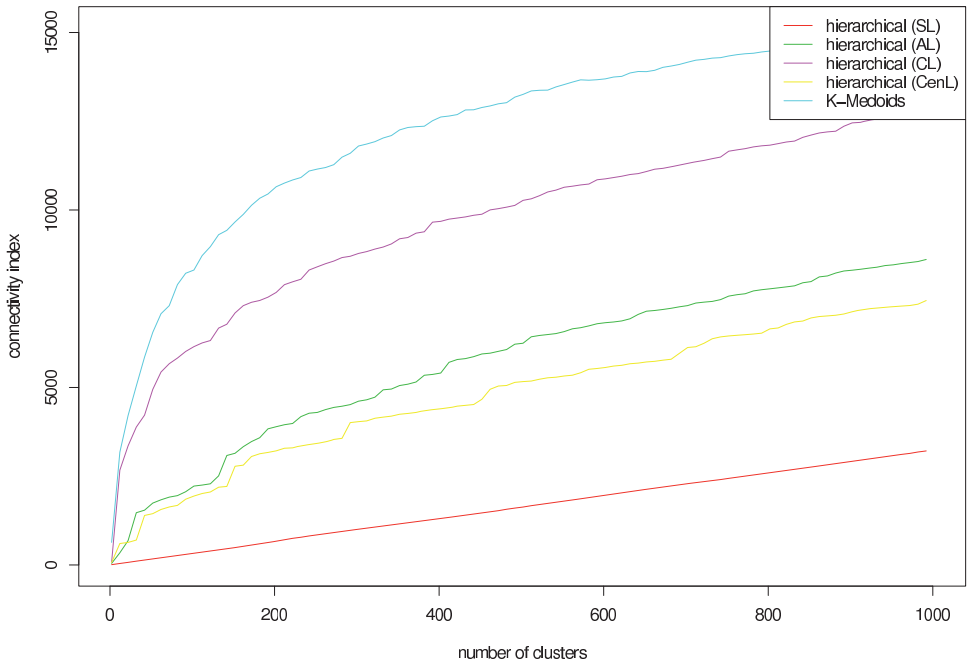


Figure C.2: Connectivity index of hierarchical and k-means clustering using the spline distance

D Appendix: Clustering Results

In the following the plots of all clusters and their corresponding 1-standardized and logarithmic gene trajectories are given.

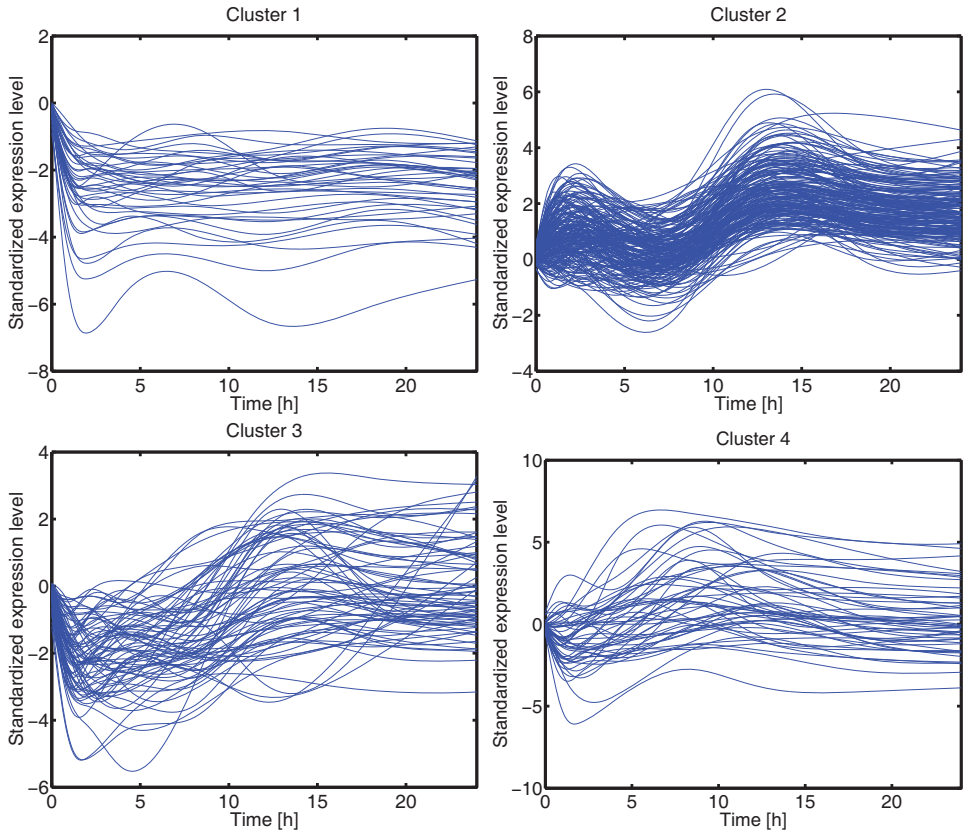


Figure D.1: Gene trajectories of clusters 1-4

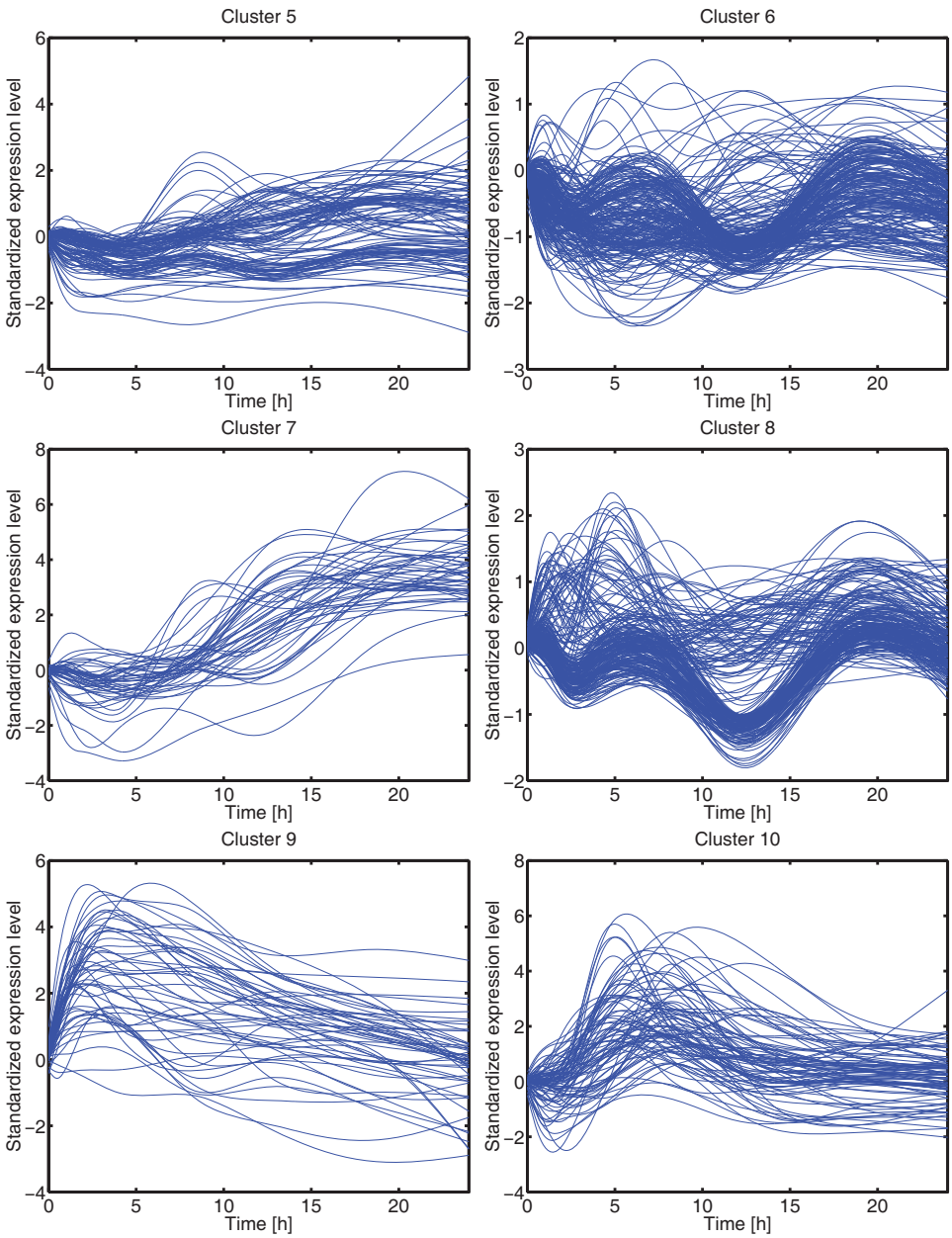


Figure D.2: Gene trajectories of clusters 5-10

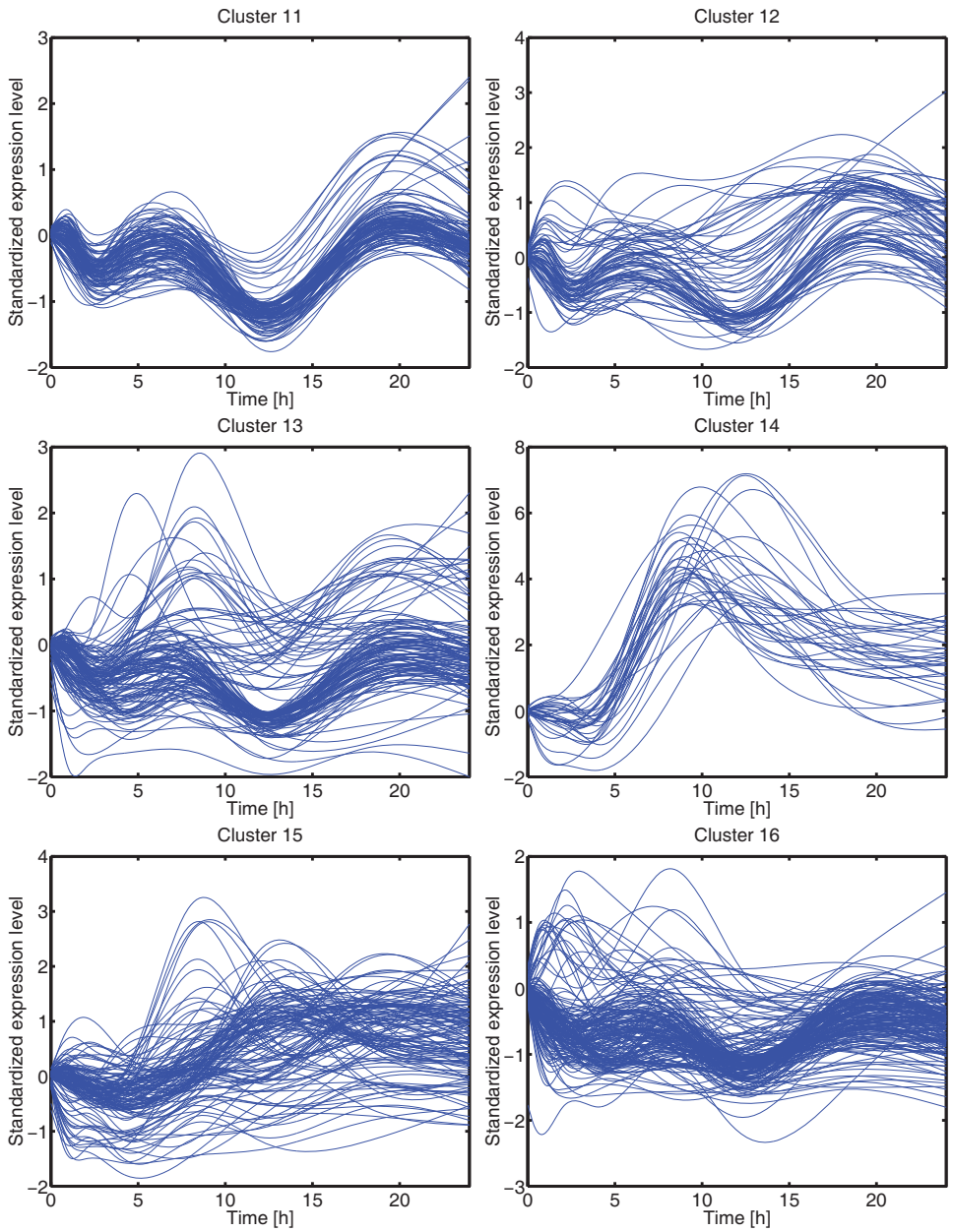


Figure D.3: Gene trajectories of clusters 11-16

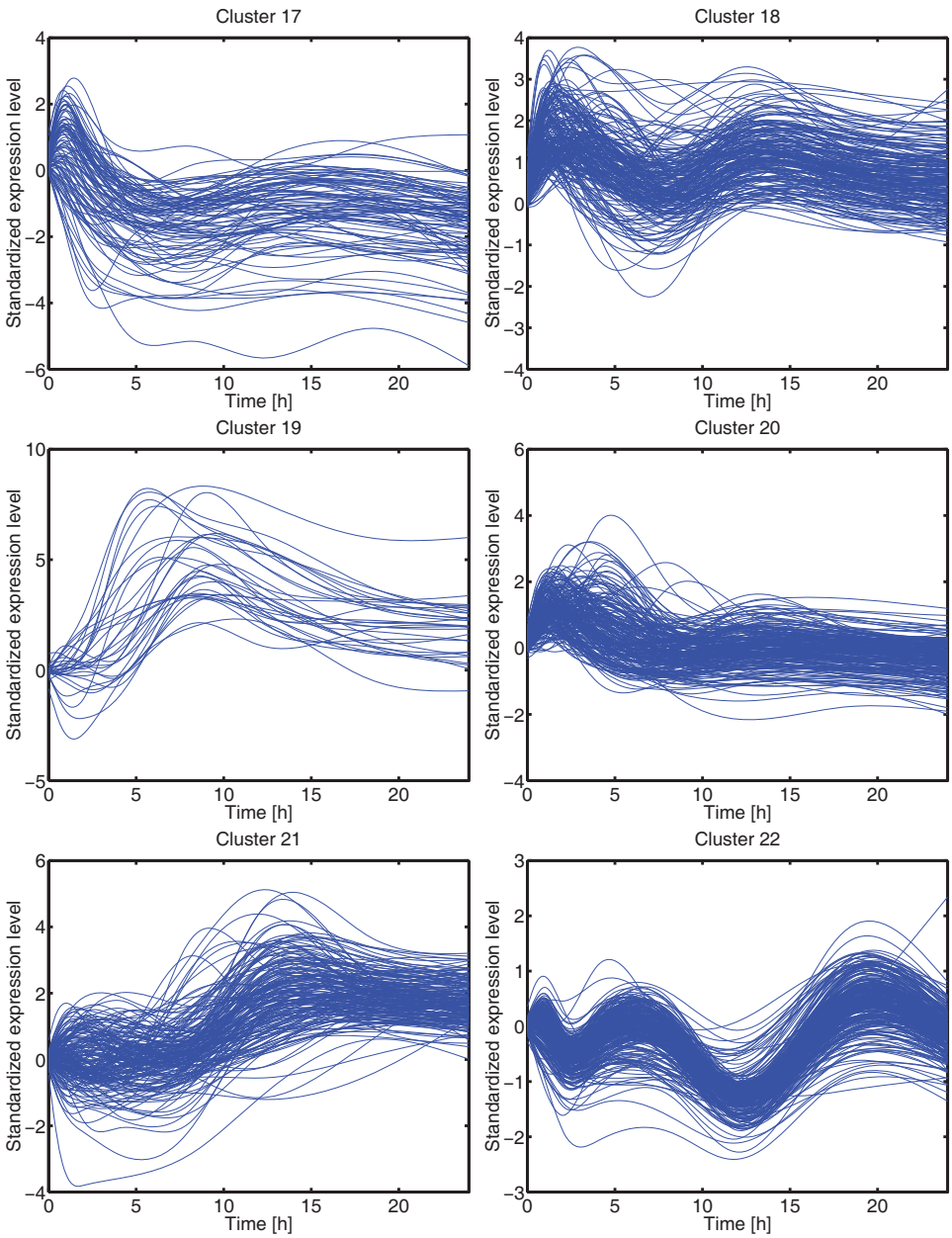


Figure D.4: Gene trajectories of clusters 17-22

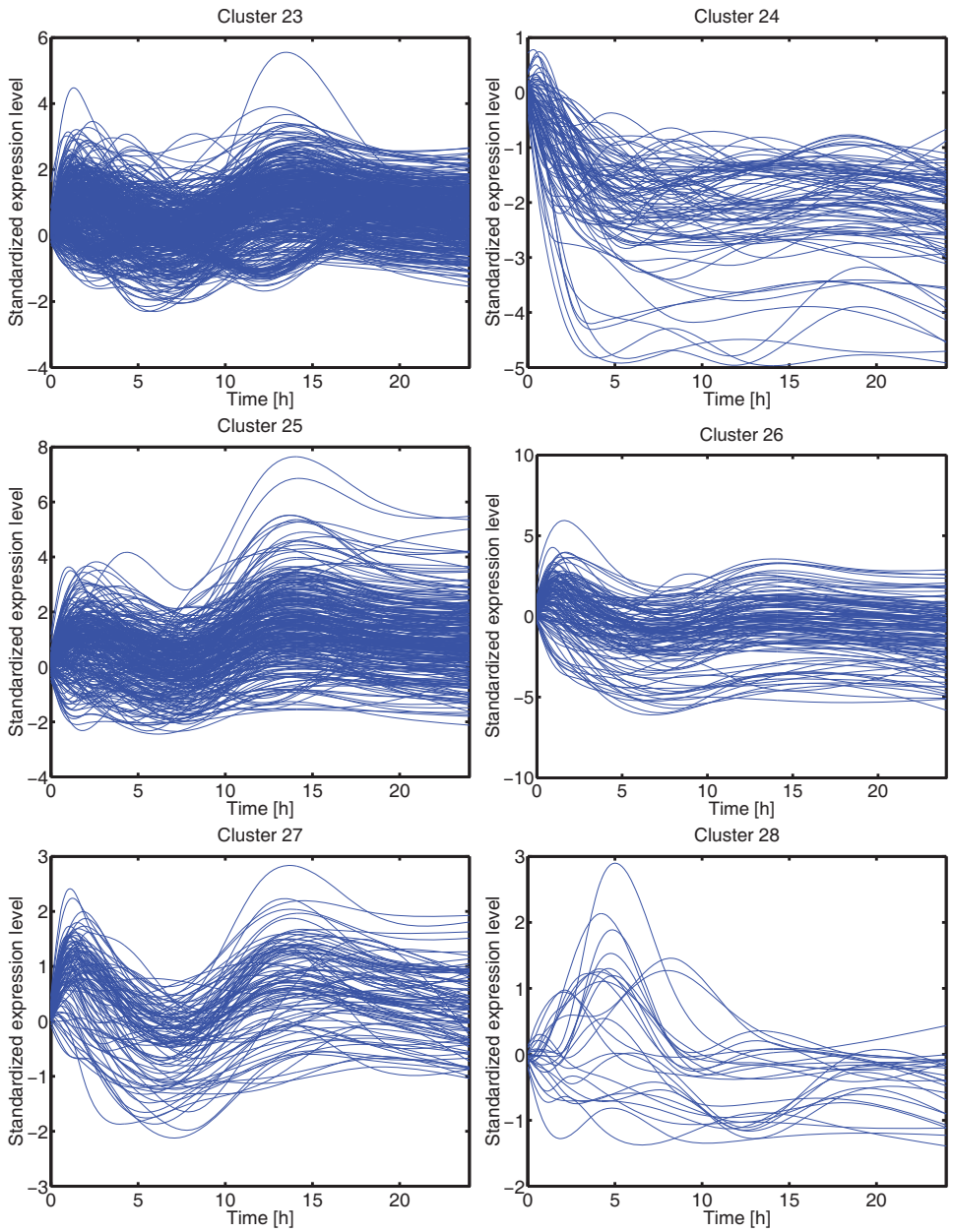


Figure D.5: Gene trajectories of clusters 23-28

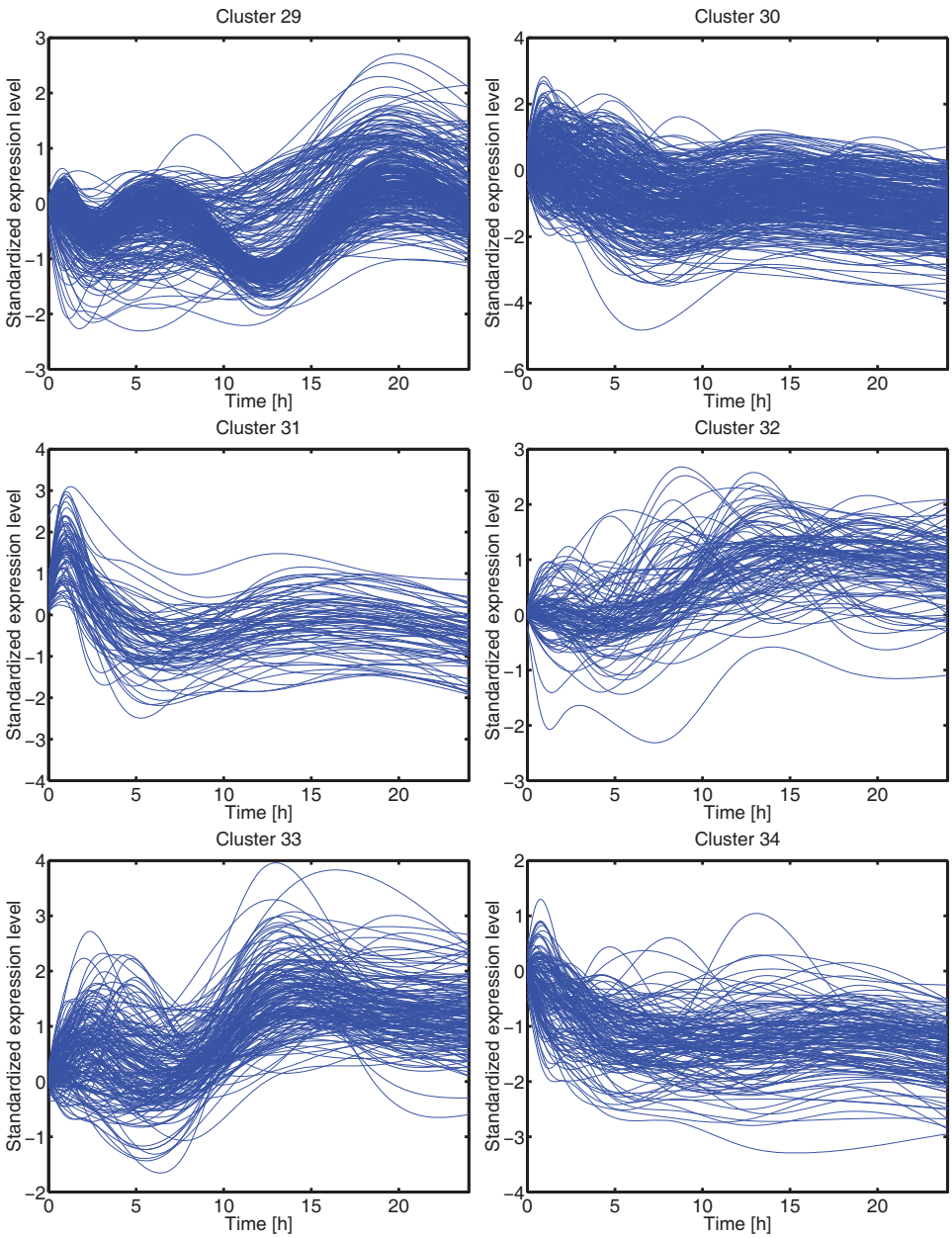


Figure D.6: Gene trajectories of clusters 29-34

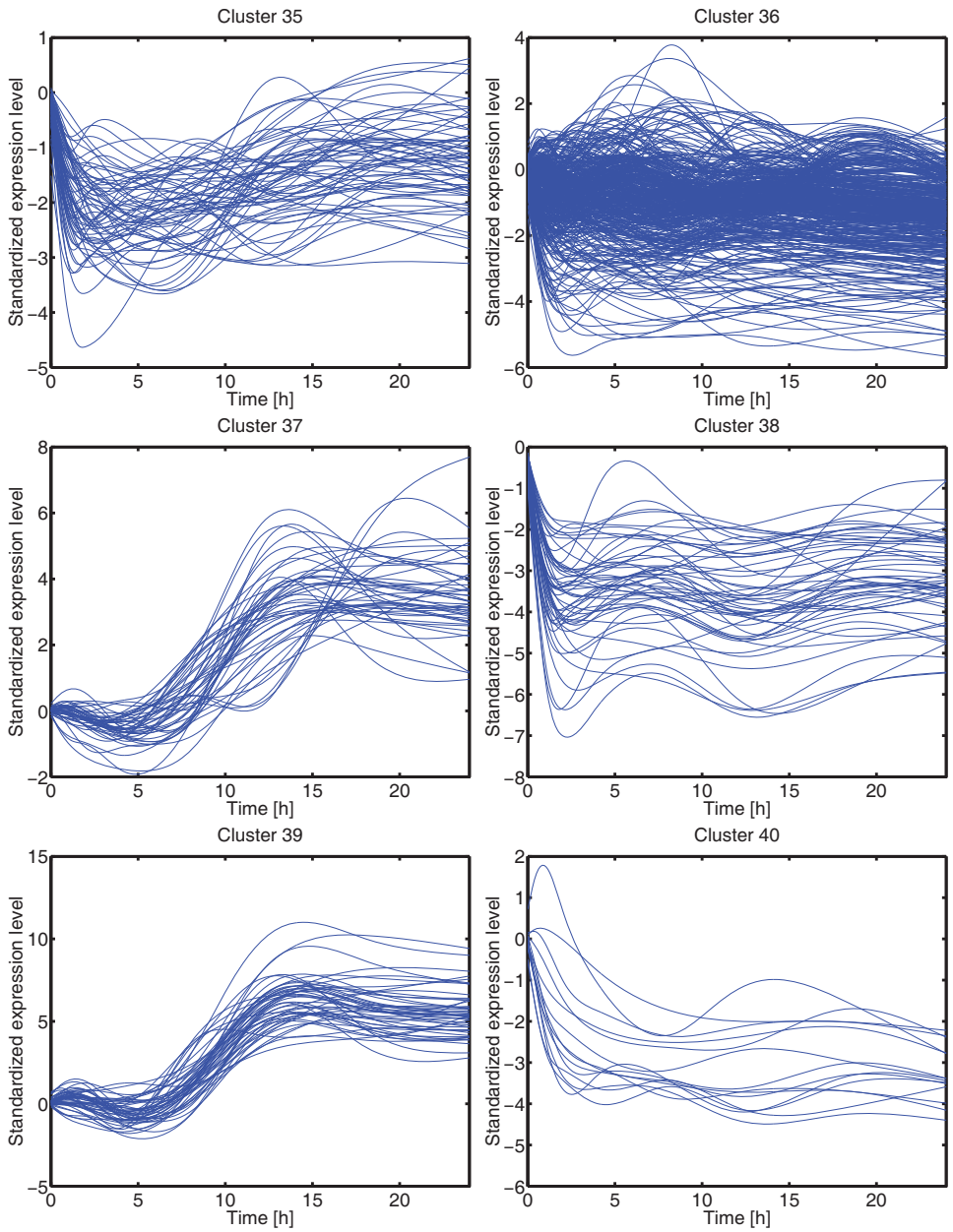


Figure D.7: Gene trajectories of clusters 35-40

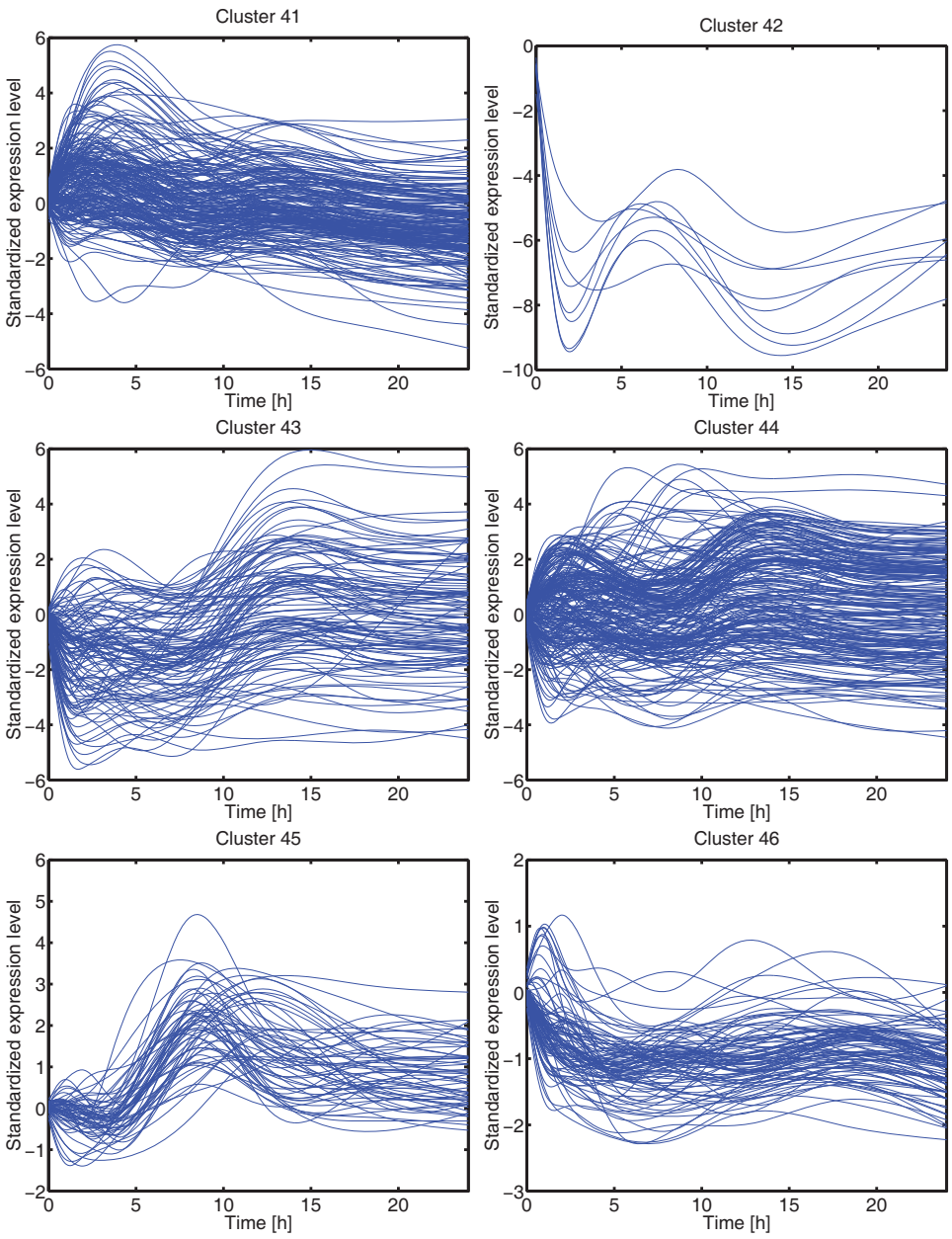


Figure D.8: Gene trajectories of clusters 41-46

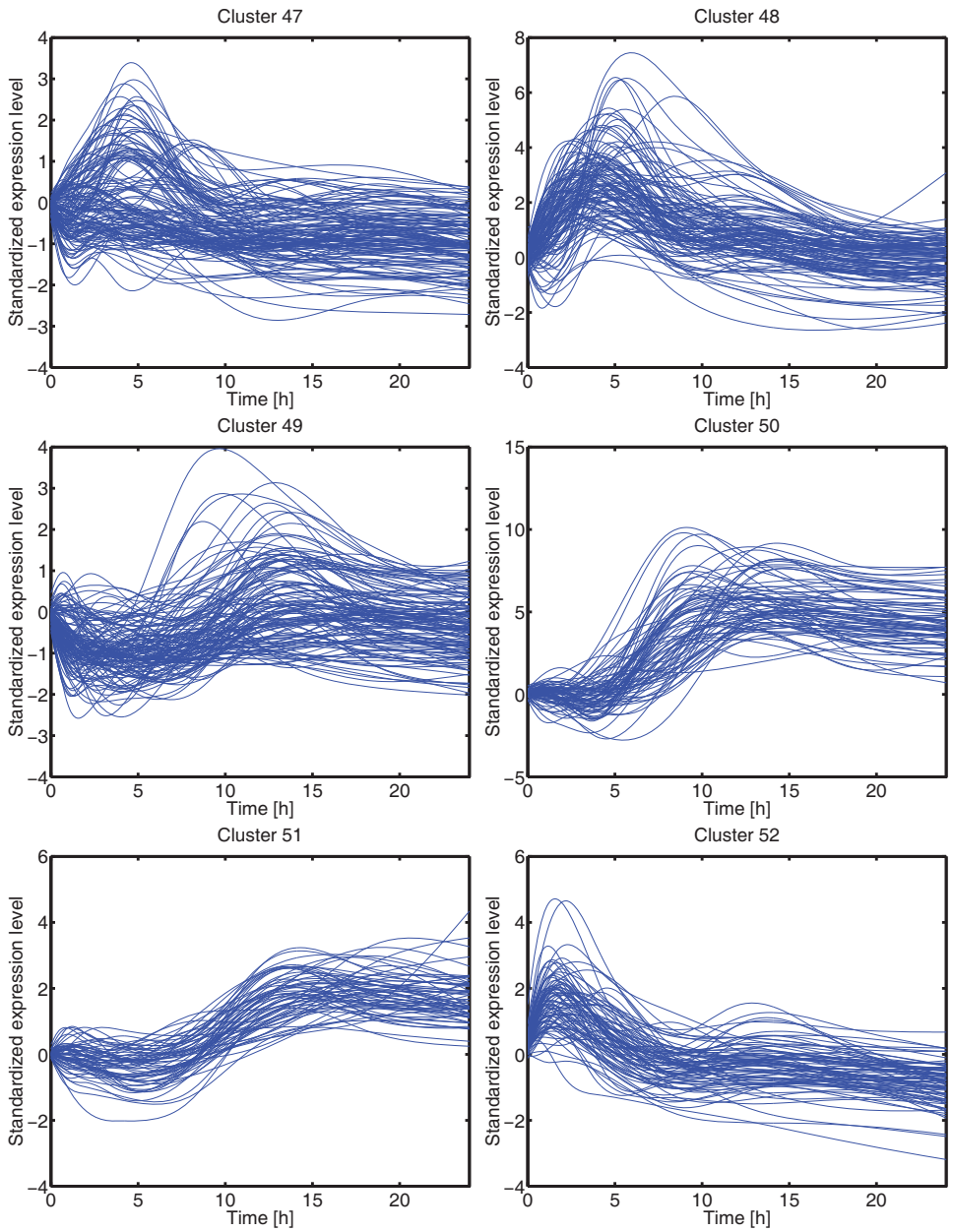


Figure D.9: Gene trajectories of clusters 47-52

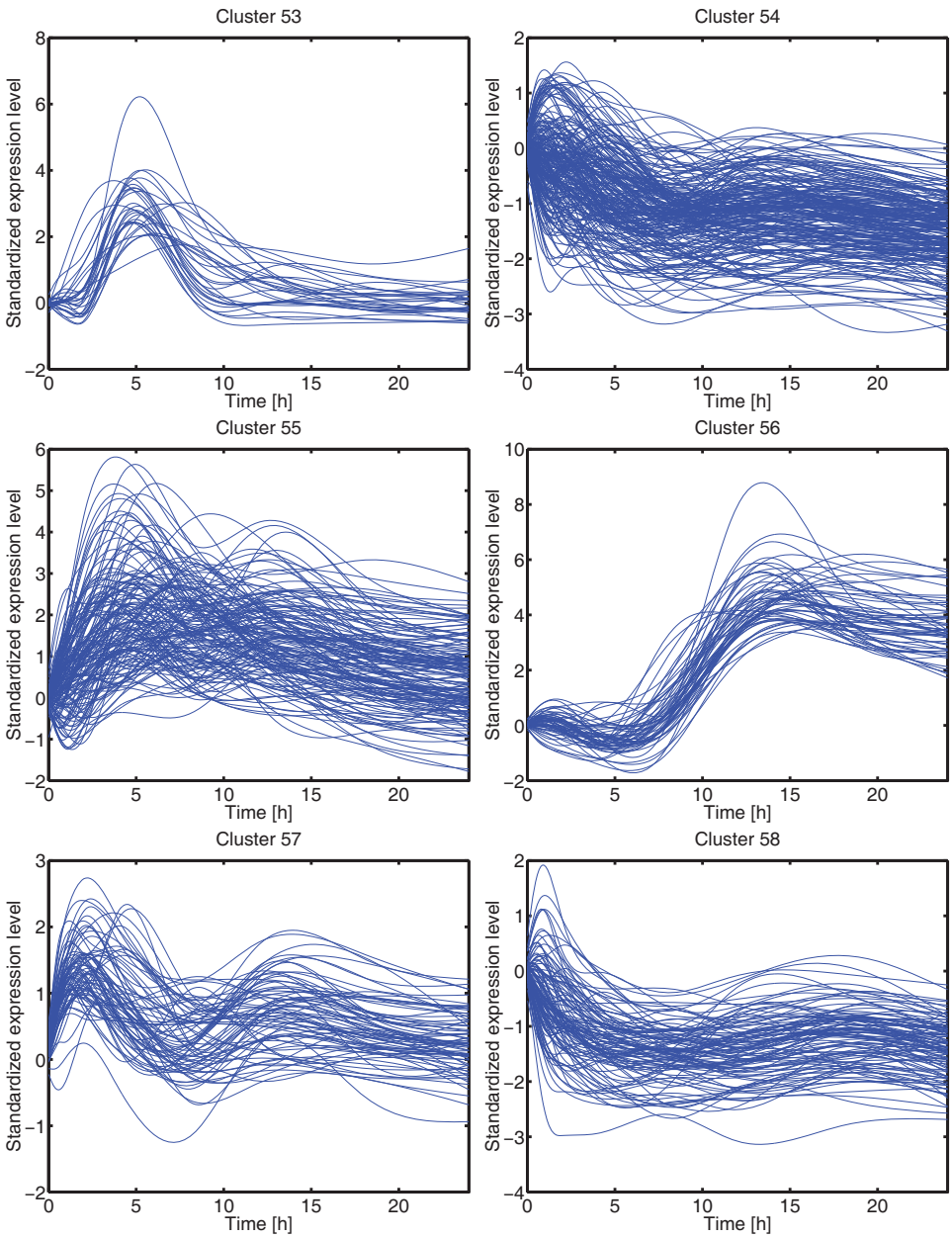


Figure D.10: Gene trajectories of clusters 53-58

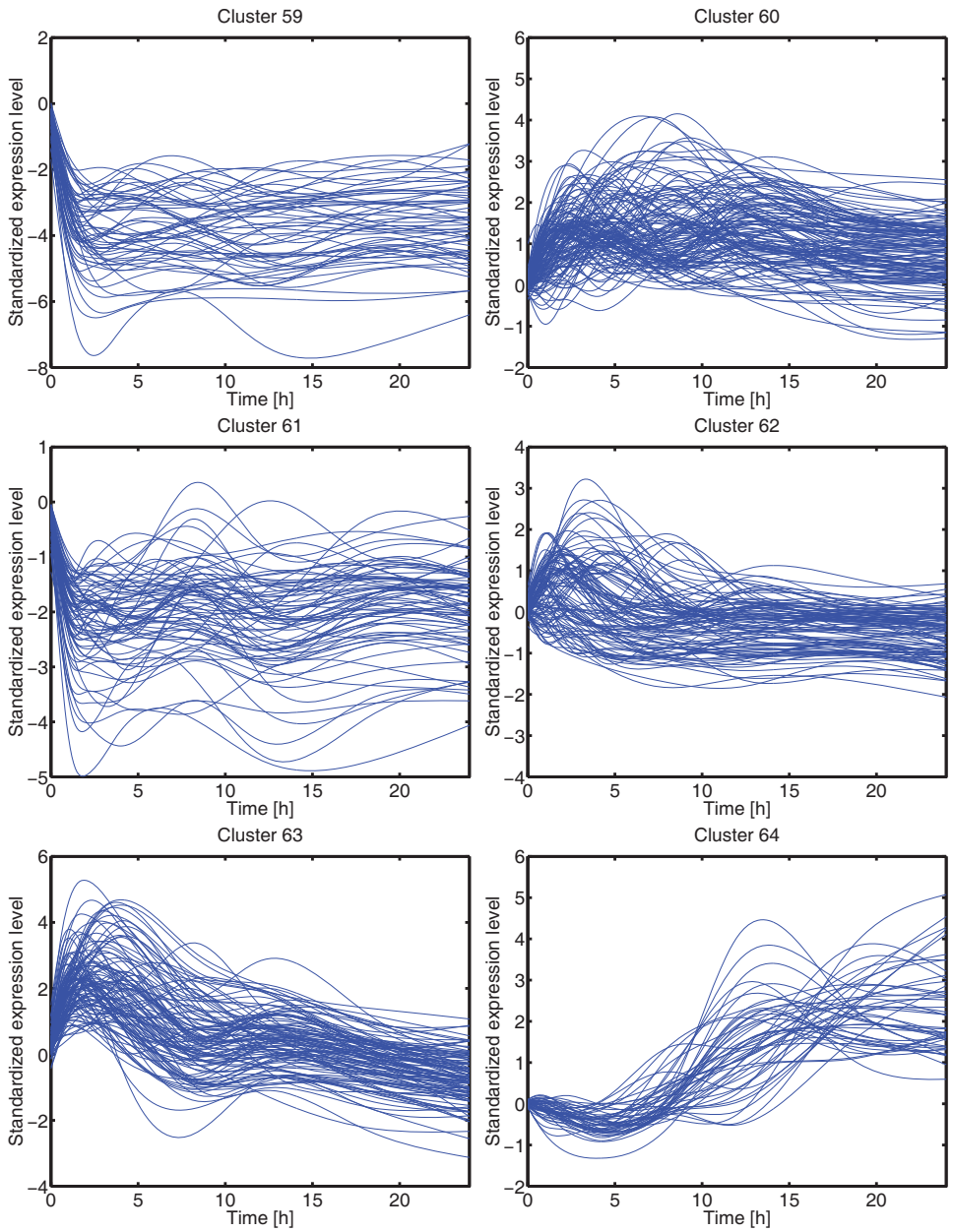


Figure D.11: Gene trajectories of clusters 59-64

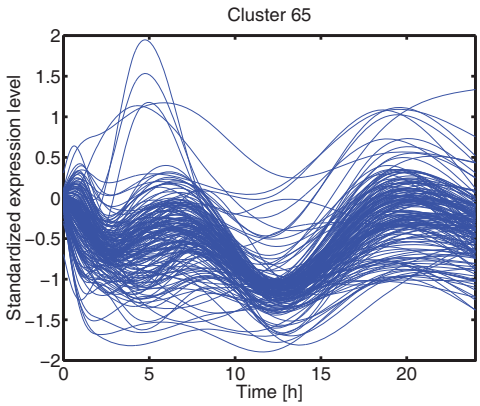


Figure D.12: Gene trajectories of cluster 65

E Appendix: Approximation of the Clusters

In the following the plots of all cluster time courses (solid blue) and their approximation by the standard state space system (dash-dotted red) are given.

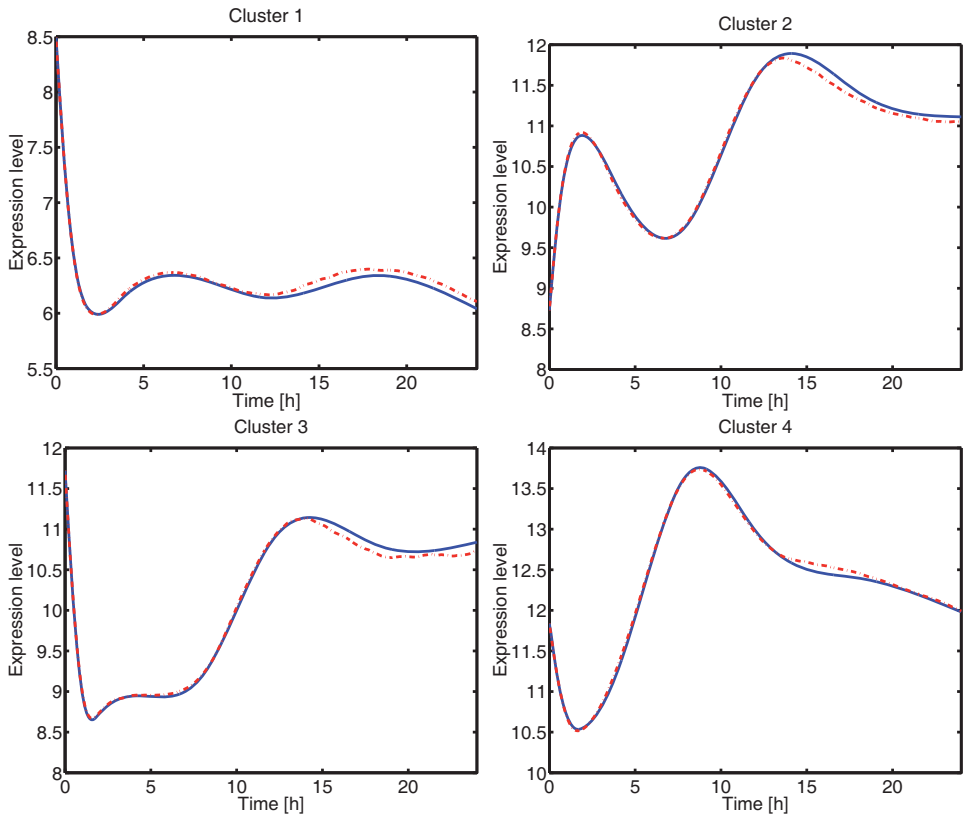


Figure E.1: Model trajectories of cluster representatives 1-4

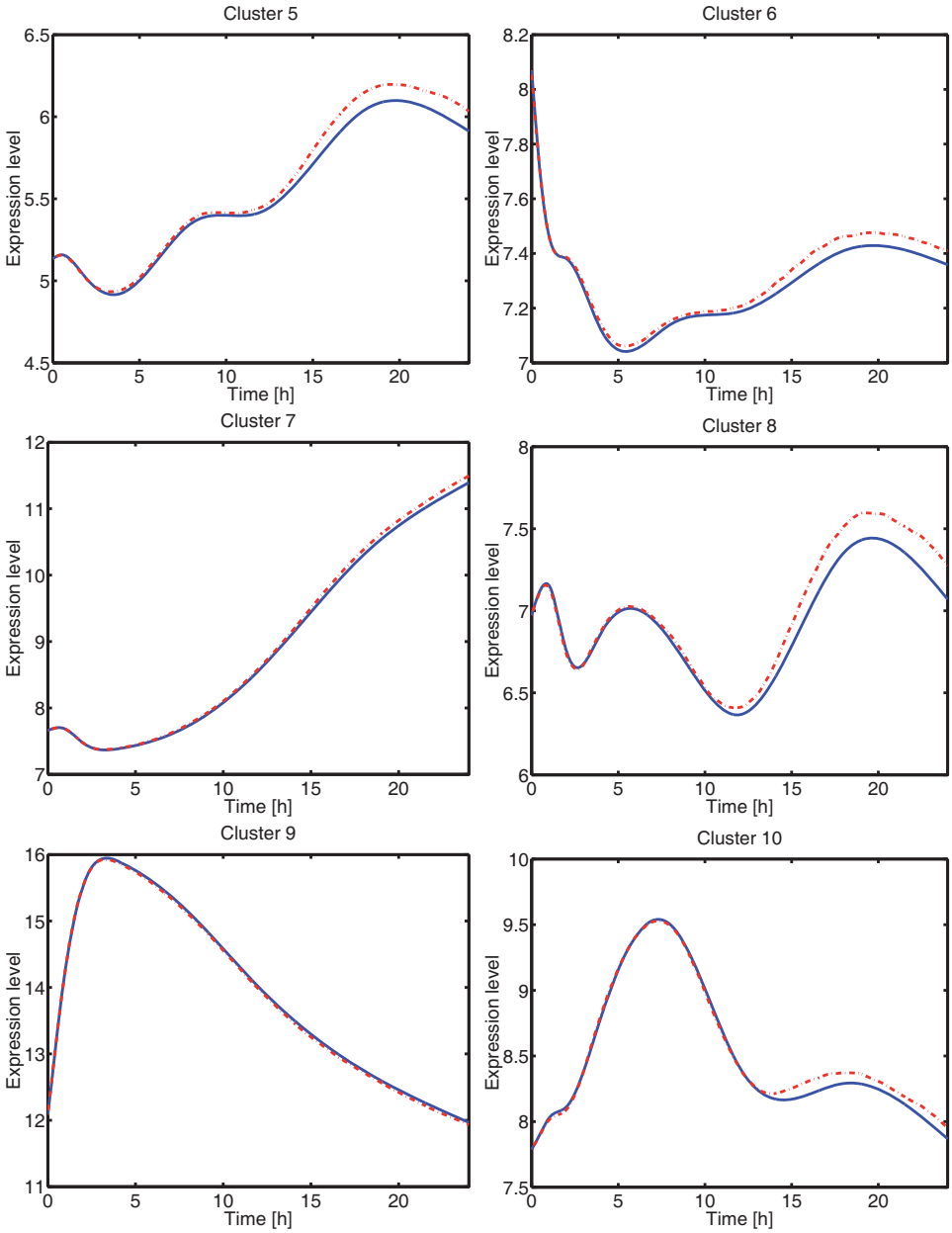


Figure E.2: Model trajectories of cluster representatives 5-10

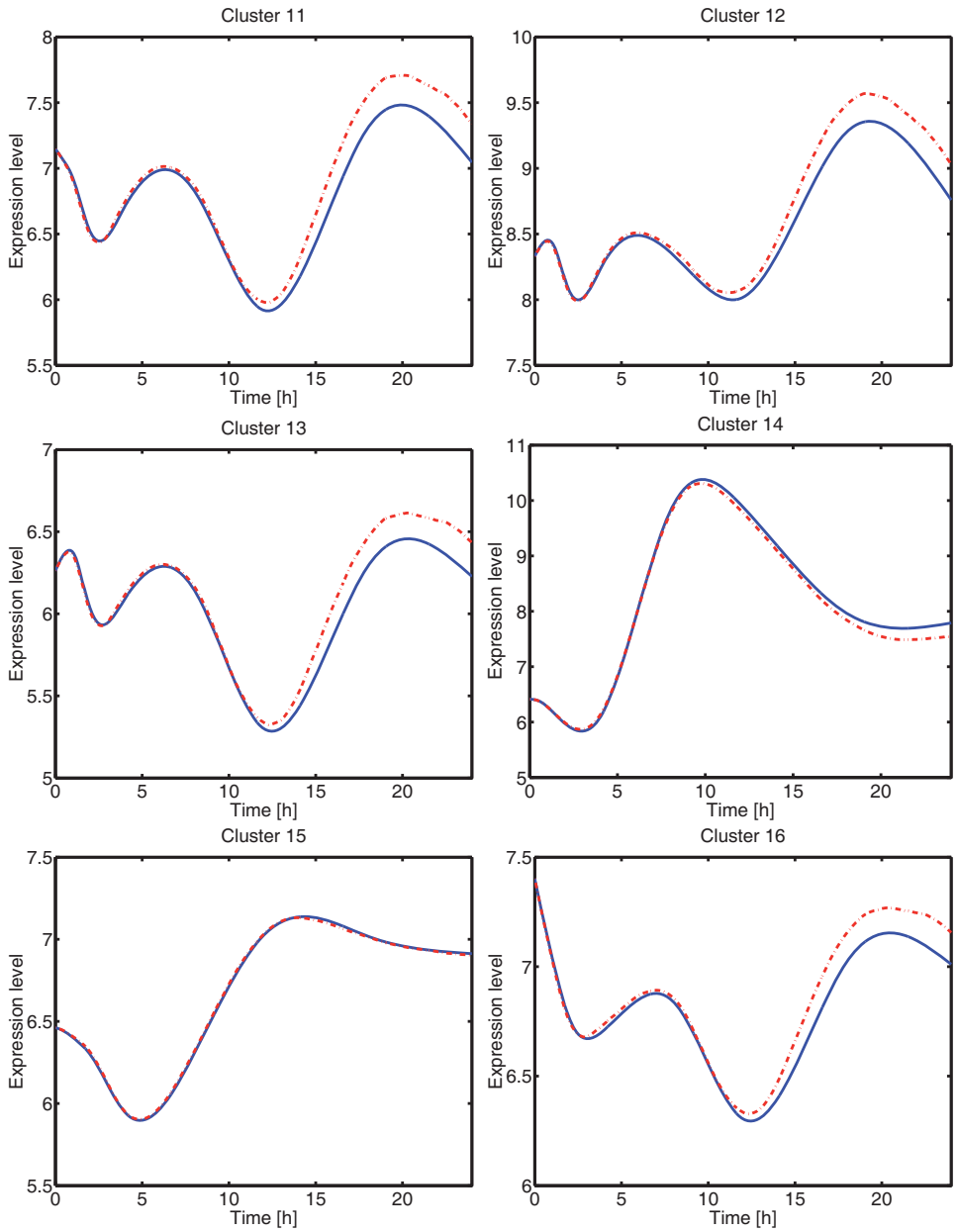


Figure E.3: Model trajectories of cluster representatives 11-16

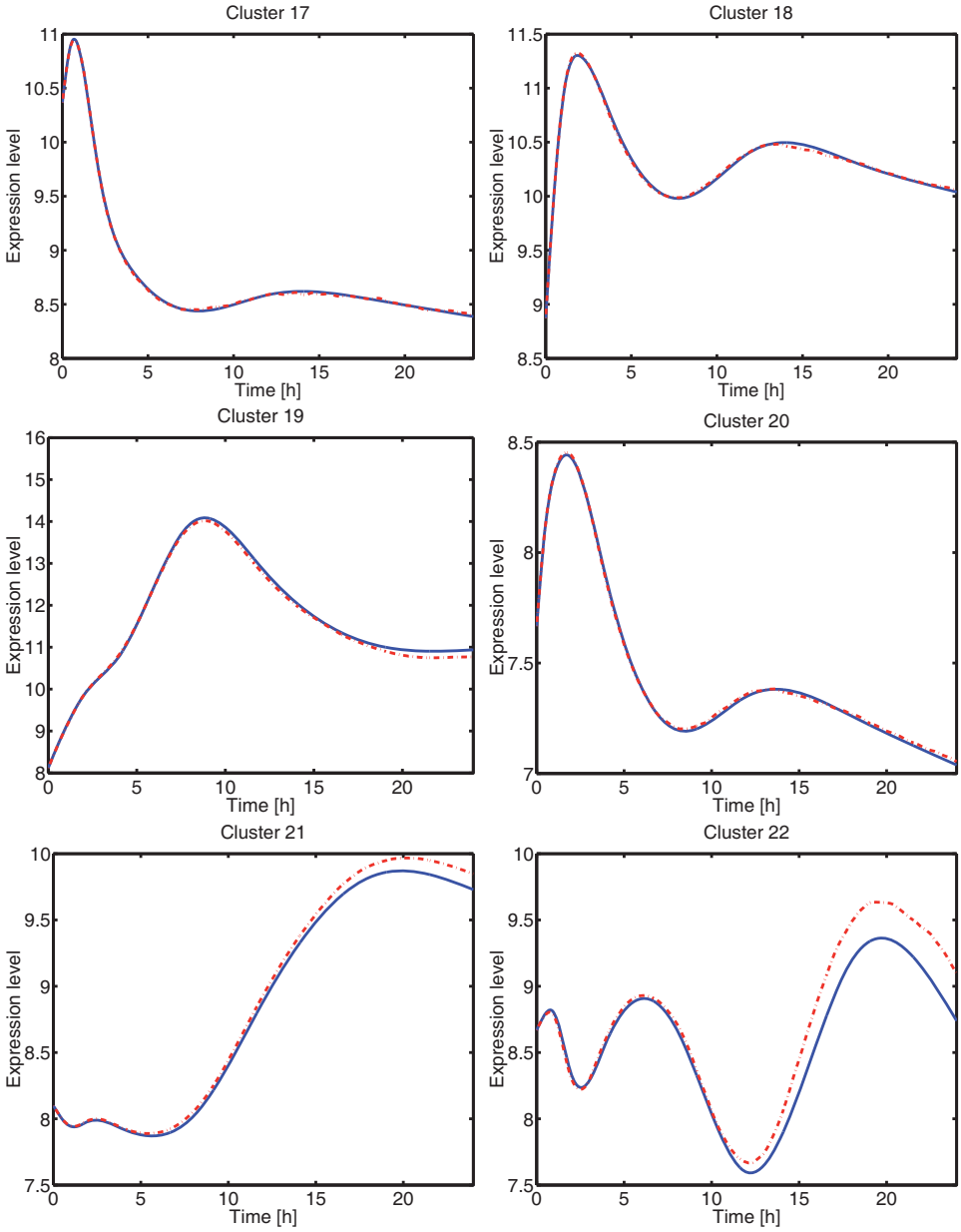


Figure E.4: Model trajectories of cluster representatives 17-22

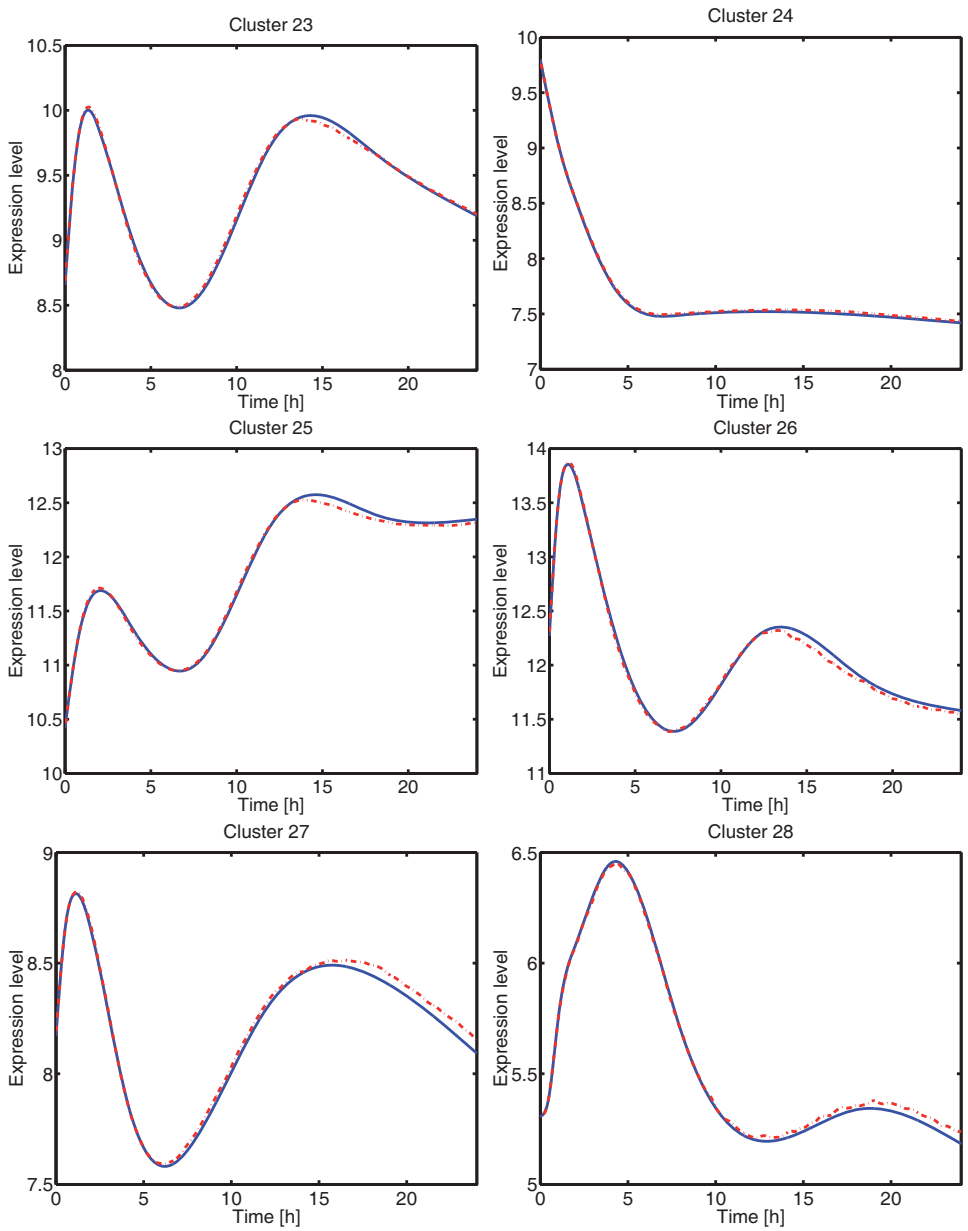


Figure E.5: Model trajectories of cluster representatives 23-28

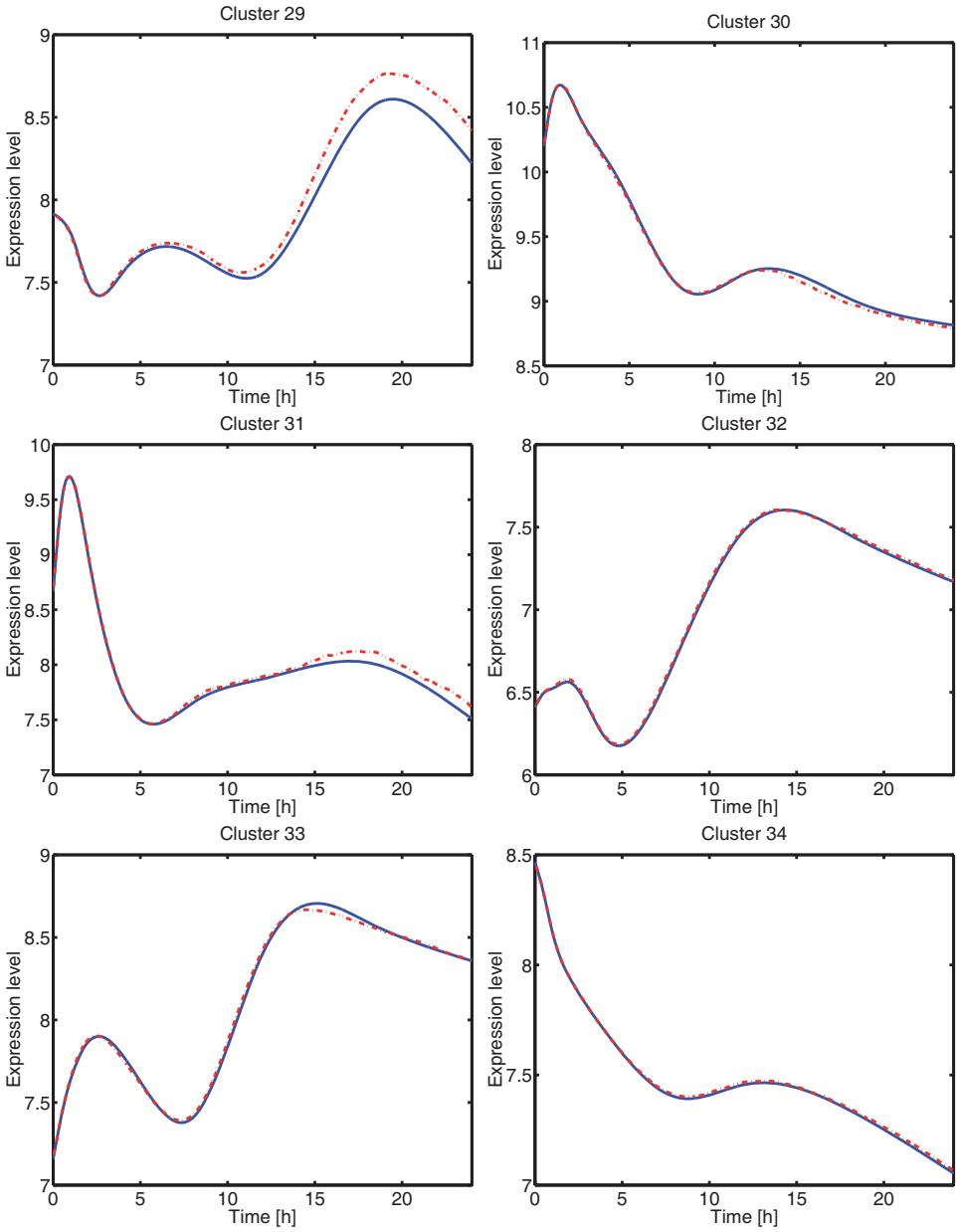


Figure E.6: Model trajectories of cluster representatives 29-34

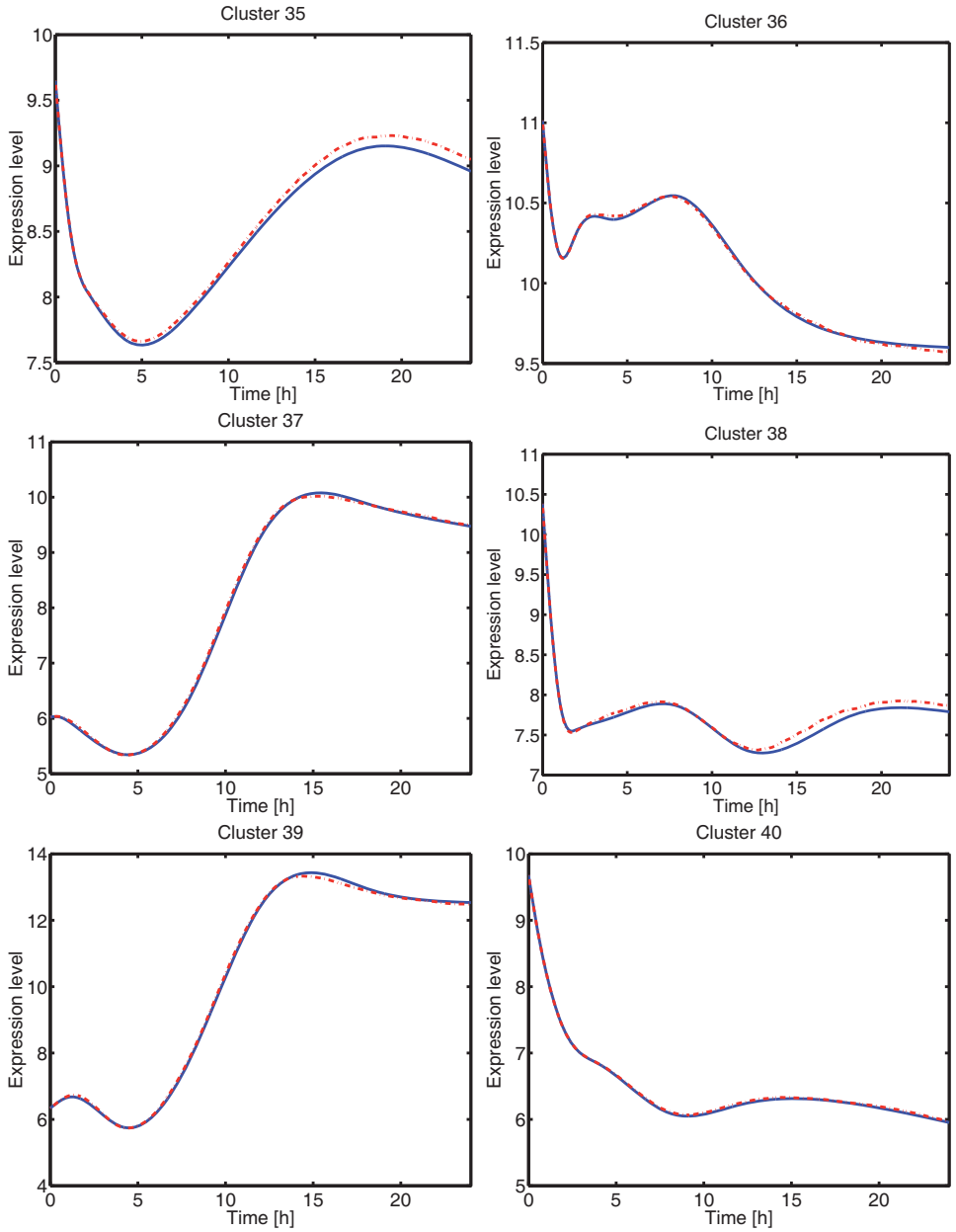


Figure E.7: Model trajectories of cluster representatives 35-40

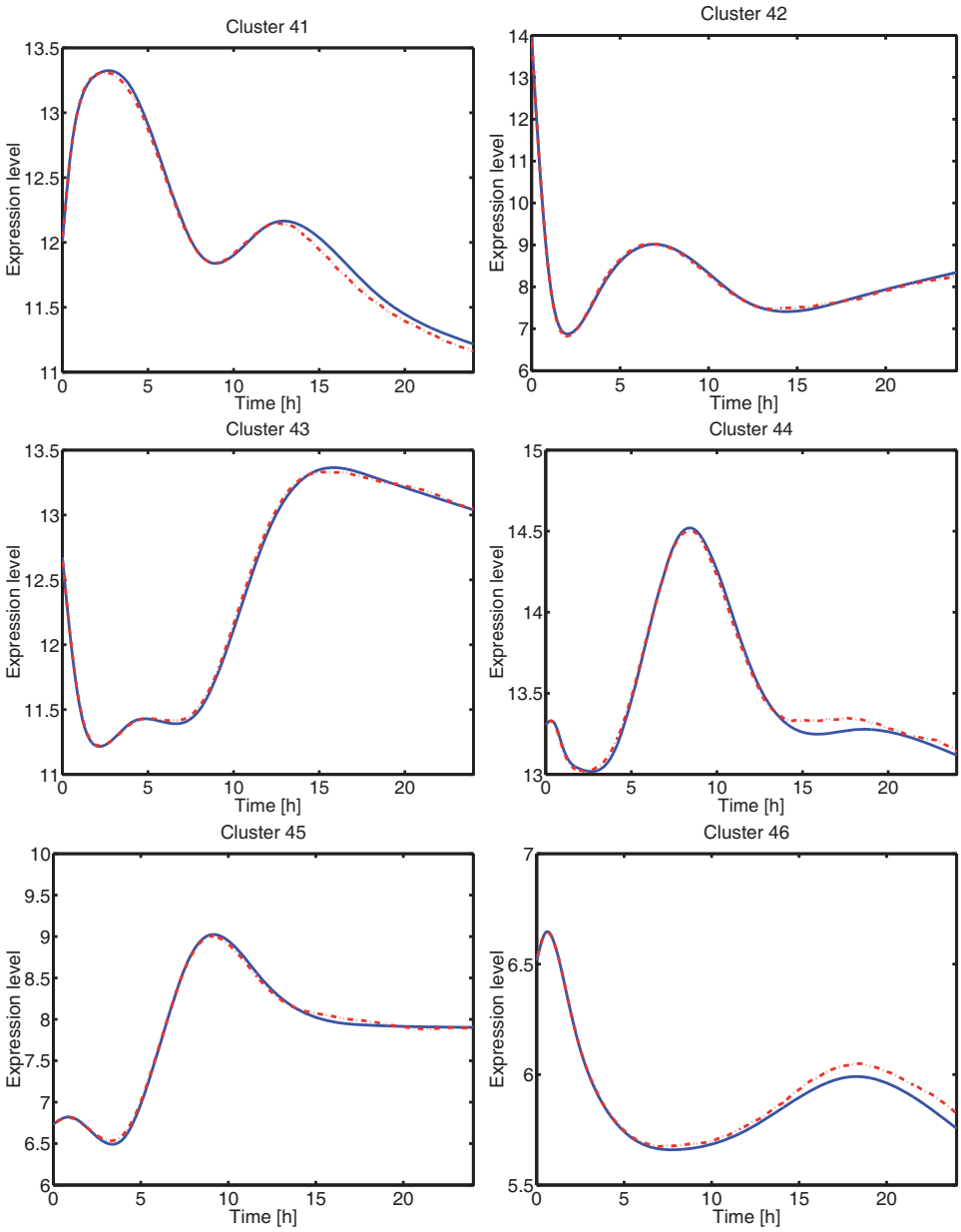


Figure E.8: Model trajectories of cluster representatives 41-46

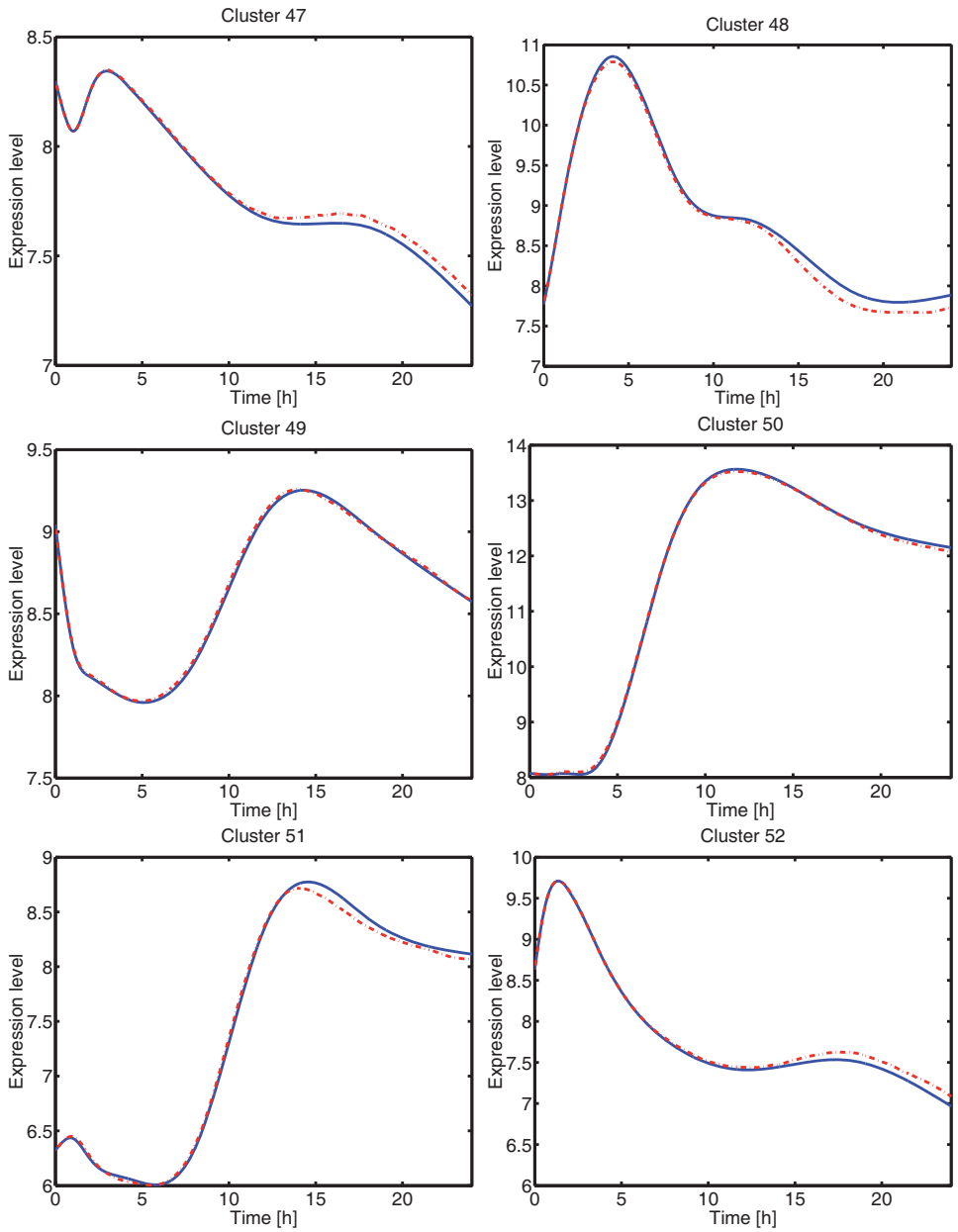


Figure E.9: Model trajectories of cluster representatives 47-52

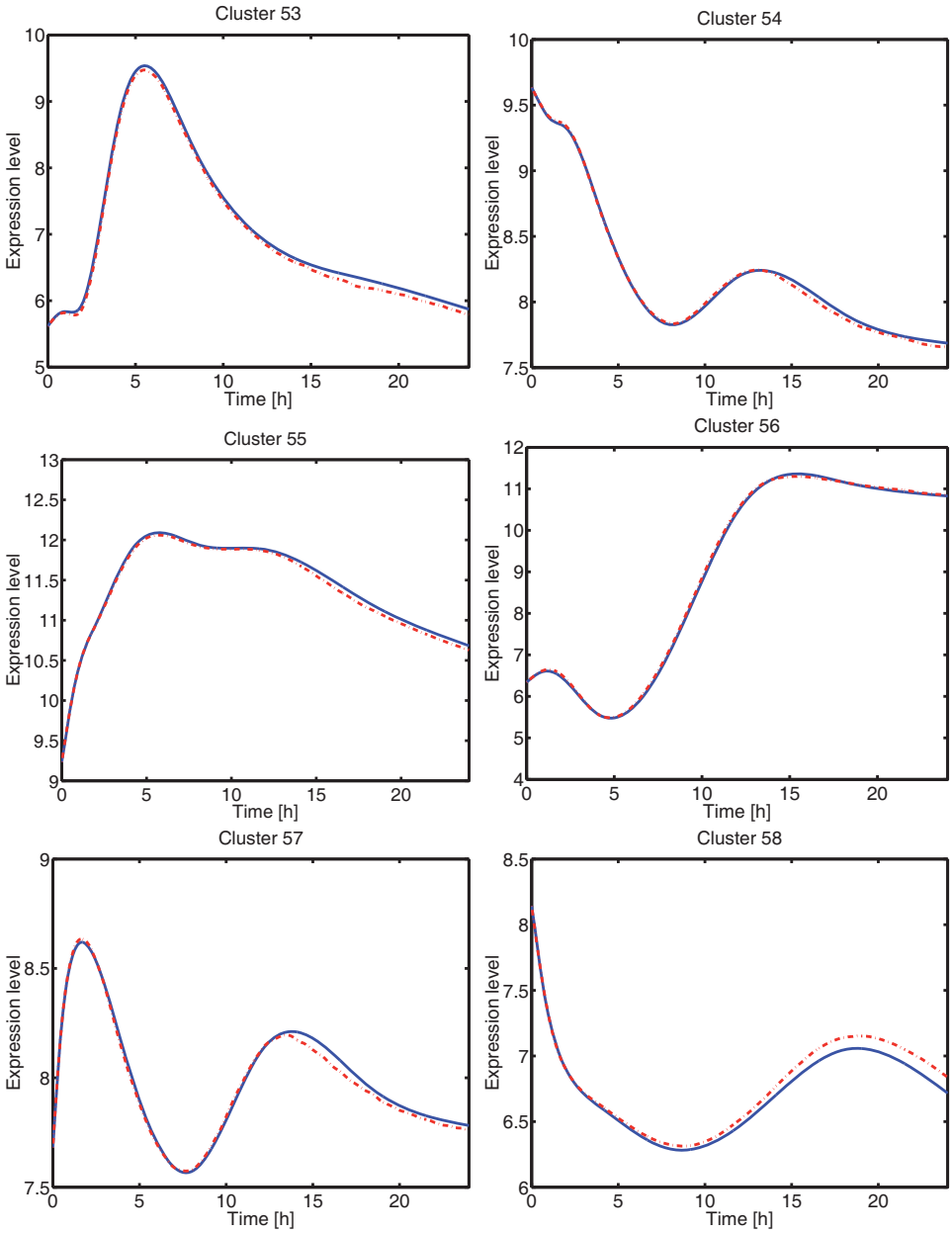


Figure E.10: Model trajectories of cluster representatives 53-58

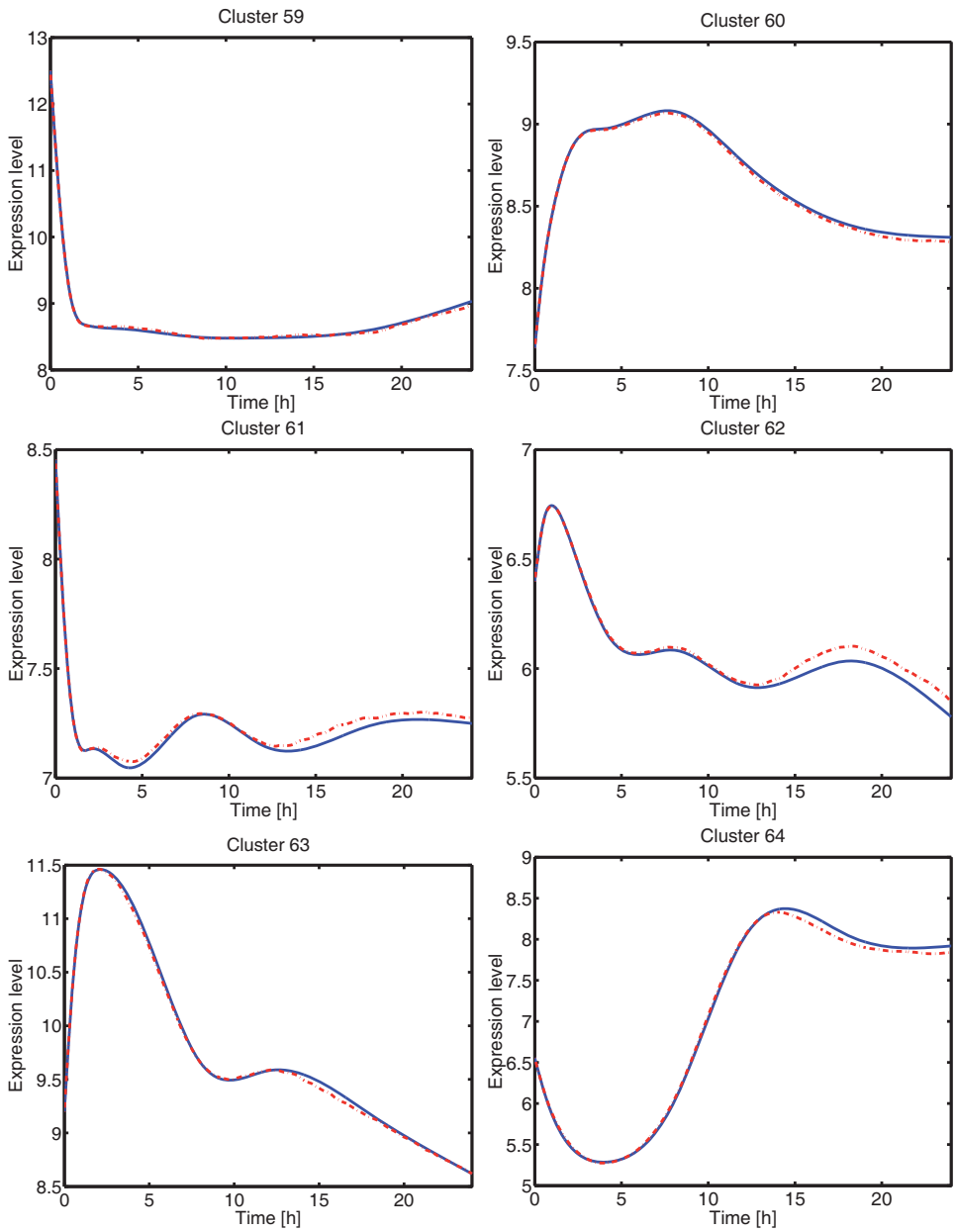


Figure E.11: Model trajectories of cluster representatives 59-64

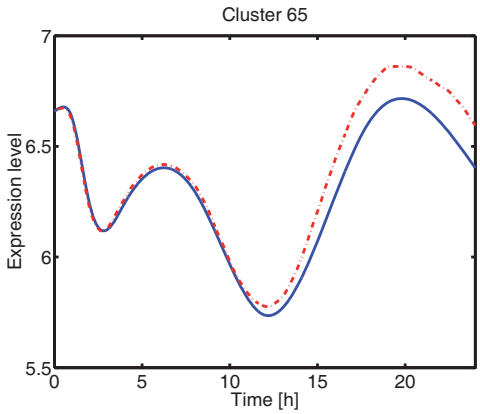


Figure E.12: Model trajectories of cluster 65

List of Figures

List of Tables

Notations

Sets and Spaces

$\mathbb{N}$	the set natural numbers $\{1,2,3,\dots\}$
$\mathbb{N}_0$	the set natural numbers including 0
$2\mathbb{N}$	the set of even natural numbers
$2\mathbb{N}-1$	the set of odd natural numbers
$\mathbb{R}$	the set of real numbers
$\mathbb{R}^+$	the set of positive real numbers
$\mathbb{R}_0^+$	the set of non-negative real numbers
$\mathbb{R}^-$	the set of negative real numbers
$\mathbb{R}^n$	the n-dimensional real vector space
$\mathbb{R}^{m\times n}$	the set of all real m by n matrices
$\mathscr{C}^0$	the set of continuous functions
$\mathscr{C}^n$	the set of n-times continuously differentiable functions
$\{a_i\}_{i\in I}$	the set $\{a_i \mid i \in I\}$ where $I \subset \mathbb{N}$ is the index set
$X \times Y$	the Cartesian product of the sets X and Y
X^n	n-th Cartesian power of the set X, $n \in \mathbb{N}$
$\mathbb{P}_n$	the set of all permutations of $\{1,\dots,n\}$, with $n \in \mathbb{N}$
Π_δ	the set of real-valued polynomials with degree equal to or less than $\delta \in \mathbb{N}_0$
$\mathscr{S}^m(t_1,\dots,t_n)$	the set of splines of order $m \in \mathbb{N}$ with knot set $\{t_1,\dots,t_n\} \subset \mathbb{R}$, $n \in \mathbb{N}$
$\mathscr{N}^m(t_1,\dots,t_n)$	the set of natural splines of order $m \in 2\mathbb{N}$ with knot set $\{t_1,\dots,t_n\} \subset \mathbb{R}$, $n \in \mathbb{N}$
$L_p[a,b]$	the Lebesgue space on $[a,b] \subset \mathbb{R}$ containing the measurable and p-th power absolute integrable functions on $[a,b] \subset \mathbb{R}$

$W_2^m[a,b]$	the Sobolev space of order $m \in \mathbb{N}$ on $[a,b] \subset \mathbb{R}$
$\mathscr{O}(h(x_1,\ldots,x_n))$	Bachmann-Landau notation, set of all functions $f : \mathbb{R}^n \to \mathbb{R}$ with $\exists c, m \in \mathbb{R}^+ \forall x_1,\ldots,x_n > m \, \lvert f(x_1,\ldots,x_n)\rvert \leq c \, \lvert h(x_1,\ldots,x_n)\rvert$

Functions and Operators

$\lvert x\rvert$	the absolute value of $x \in \mathbb{R}$
$\lVert x\rVert_p$	the p-norm of vector $x \in \mathbb{R}^n$
$\lVert f\rVert_p$	the p-norm of function $f \in L^p$
$\lvert X\rvert$	the number of elements of the finite set X
$\overline{X}$	the arithmetic mean of the finite set X
$\operatorname{med}(X)$	the median of the finite set X
$\min X$	the minimum of the finite set X
$\max X$	the maximum of the finite set X
$\min\limits_{x\in X} f(x)$	the minimum of the finite set $\{f(x) \mid x \in X\}$
$\max\limits_{x\in X} f(x)$	the maximum of the finite set $\{f(x) \mid x \in X\}$
$\arg\min\limits_{x\in X} f(x)$	the value $x \in X$ where $f(x)$ becomes minimal, with $\lvert X\rvert < \infty$. (if not unique, an additional rule is given.)
$\lfloor x\rfloor$	the largest integer not greater than x
$\lceil x\rceil$	the lowest integer not smaller than x
$\log_2$	the binary logarithm
$f\lvert_{[a,b]}$	the function f restricted to the interval $[a,b]$
$f^{(k)}$	the k-th derivative of the function f (for convenience let $f^{(0)} \equiv f$)
δ_{ij}	Kronecker delta function, being 1 if $i = j$ and 0 otherwise
$\lim\limits_{x \nearrow x_0} f(x)$	the one-sided limit of the function f from below

Miscellaneous

x_i	the i-th component of the vector $i \in \mathbb{R}^n$, $n \in \mathbb{N}$
A'	the transpose of $A \in \mathbb{R}^{m \times n}$
A_{ij}	the entry in the i-th row and j-th column of matrix A
$A_{i\cdot}$	the i-th row of matrix A
$A_{\cdot j}$	the j-th column of matrix A
$\wedge$	logical AND
$\binom{n}{k}$	binomial coefficient; n choose k

Modeling Notations

$M_{g,t}$	set of normalized measurements of gene g under treatment t
$m_{g,t}^{(i)}$	i-th normalized measurement of gene g under treatment t
$n_{g,t}$	number of measurements of gene g under treatment t
T	set of treatments
τ	number of treatments
Γ	number of genes
$y_i^{(g)}$	median of the measurements of gene g at time t_i
$w_i^{(g)}$	accuracy measure of the median of the measurements of gene g at time t_i
S_g	interpolating cubic smoothing spline of gene g
$\Xi = \{C_1, \ldots, C_\xi\}$	clustering of the underlying gene set
ξ	number of clusters
$\hat{g}_i$	representative of cluster C_i
$S_{\hat{g}_i}$	interpolating cubic smoothing spline of cluster C_i

Bibliography

[Agi07a] Agilent Technologies. *Agilent Feature Extraction Software (v9.5) Reference Guide*, 4th edition, February 2007.

[Agi07b] Agilent Technologies. *Agilent Feature Extraction Software (v9.5) User Guide*, 4th edition, February 2007.

[AHSV99] Esa Alhoniemi, Jaakko Hollmén, Olli Simula, and Juha Vesanto. Process Monitoring and Modeling Using the Self-Organizing Map. *Integr. Comput.-Aided Eng.*, 6(1):3–14, 1999.

[And99] Eric C. Anderson. Monte Carlo Methods and Importance Sampling. http://ib.berkeley.edu, 1999.

[Apa00] Samuel A J R Aparicio. How to Count...Human Genes. *Nature Genetics*, 25:129–130, 2000.

[Ari] Aristotle. *On the Generation of Animals*. eBooks@Adelaide, translated by Arthur Platt, 2007.

[ASV97] Esa Alhoniemi, Olli Simula, and Juha Vesanto. Analysis of Complex Systems using the Self-Organizing Map. In *Progress in Connections Based Information Systems. Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems*, pages 1313–1317. Springer, 1997.

[BBC02] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation Clustering. In *Machine Learning*, pages 238–247, 2002.

[BFT04] Eli Ben-Naim, Hans Frauenfelder, and Zoltan Toroczaki, editors. *Complex Networks*. Lecture Notes in Physics. Springer-Verlag, 2004.

[BGL+00] Michael P. S. Brown, William Noble Grundy, David Lin, Nello Cristianini, Charles Walsh Sugnet, Terrence S. Furey, Manuel Ares, and David Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America*, 97(1):262–267, 2000.

[BGOT04] Karla V. Ballman, Diane E. Grill, Ann L. Oberg, and Terry M. Therneau. Faster cyclic loess: normalizing RNA arrays via linear models. *Bioinformatics*, 20(16):2778–2786, 2004.

[BGWK06] Lubica Benuskova, Simei Gomes Wysoski, and Nikola K. Kasabov. Computational Neurogenetic Modeling: A Methodology to Study Gene Interactions Underlying Neural Oscillations. In *International Joint Conference on Neural Networks*, pages 4638–4644, July 2006.

[BHJ+04] John Berger, Sampsa Hautaniemi, Anna-Kaarina Jarvinen, Henrik Edgren, Sanjit Mitra, and Jaakko Astola. Optimized LOWESS normalization parameter selection for DNA microarray data. *BMC Bioinformatics*, 5(1):194, 2004.

[BIAS03] B.M. Bolstad, R.A Irizarry, M. Astrand, and T.P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

[Bil06] Stephen C. Billups. Analyzing Time Course Microarray Data With Temporal Uncertainty. http://www.iscb.org, December 2006.

[Bla03] Eric Blalock, editor. *A Beginner's Guide To Microarrays*. Kluwer Academic Publishers, 2003.

[BLB00] Jürgen Bortz, Gustav A. Lienert, and Klaus Boehnke. *Verteilungsfreie Methoden in der Biostatistik*. Springer-Verlag, 2nd edition, 2000.

[Boo78] Carl de Boor. *A Practical Guide to Splines*. Number 27 in Applied Mathematical Sciences. Springer-Verlag, 1978.

[BPD+04] Monica Benito, Joel Parker, Quan Du, Junyuan Wu, Dong Xiang, Charles M. Perou, and J. S. Marron. Adjustment of systematic microarray data biases. *Bioinformatics*, 20(1):105–114, 2004.

[Bro93] Terence A. Brown. *Moderne Genetik, Eine Einführung*. Spektrum Akademischer Verlag, 1993. Original edition: Genetics: A Molecular Approach, Chapman & Hall, 1992.

[CGM07] Olivier Cappé, Simon J. Godsill, and Eric Moulines. An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

[Chu02] Gary A. Churchill. Fundamentals of experimental design for cDNA microarrays. *Nature Genetics Supplement*, 32:490–495, December 2002.

[CK02] Brett C. Couch and Linda M. Kohn. A multilocus gene genealogy concordant with host preference indicates segregation of a new species, Magnaporthe oryzae, from M. grisea. *Mycologia*, 94(4):683–693, 2002.

[Cla01] Jean-Michel Claverie. Gene Number. What if There are Only 30,000 Human Genes? *Science*, 291(5507):1255–1257, 2001.

[CMC99] Hung-Han Chen, Michael T.B Manry, and Hema Chandrasekaran. A Neural Network Training Algorithm Utilizing Multiple Sets of Linear Equations. *Neurocomputing*, 25(1):55–72, April 1999.

[Con04] International Human Genome Sequencing Consortium. Finishing the euchromatic sequence of the human genome. *Nature*, 431:931–945, October 2004.

[Cri58] Francis Harry Compton Crick. On Protein Synthesis. *Symposia of the Society for Experimental Biology*, 12:139–163, 1958.

[CSC+09] N. Castilla, S. Savary, C.M. Vera Cruz, , and H. Leung. Rice Fact Sheets - Rice Blast. Produced by the International Rice Research Institute (IRRI), June 2009.

[Dai03] Yang Dai. Mathematical Modeling Lecture 19: Modeling of Data: LOWESS. http://array.bioengr.uic.edu, 2003.

[D'h00] Patrik D'haeseleer. *Reconstructing Gene Networks from Large Scale Gene Expression Data*. PhD thesis, University of New Mexico, Albuquerque, New Mexico, December 2000.

[DHHR02] B.P. Durbin, J.S. Hardin, D.M. Hawkins, and D.M. Rocke. A variance-stabilizing transformation for gene-expression microarray data. *Bioinformatics*, 18(Suppl.1):S105–110, 2002.

[DHJ+04] Adrian Dobra, Chris Hans, Beatrix Jones, Joseph R. Nevins, and Mike West. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, 90(1):196–212, 2004. Special Issue on Multivariate Methods in Genomic Data Analysis.

[DKM+04] Lori E. Dodd, Edward L. Korn, Lisa M. McShane, G.V.R. Chandramouli, and Eric Y. Chuang. Correcting log ratios for signal saturation in cDNA microarrays. *Bioinformatics*, 20(16):2685–2693, 2004.

[DLS00] Patrik D'haeseleer, Shoudan Liang, and Roland Somogyi. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8):707–726, 2000.

[DMBB07] S. Déjean, P. G. P. Martin, A. Baccini, and P. Besse. Clustering Time-Series Gene Expression Data Using Smoothing Spline Derivatives. *EURASIP Journal on Bioinformatics and Systems Biology*, 2007.

[DPCL96] Jean-Philippe Draye, Davor Pavisic, Guy Cheron, and Gaëtan Libert. Dynamic Recurrent Neural Networks: a Dynamical Analysis. *IEEE Trans. on Systems Man and Cybernetics, Part B*, 26:692–706, 1996.

[DR04] B. Durbin and D. M. Rocke. Variance-stabilizing transformations for two-color microarrays. *Bioinformatics*, 20(5):660–667, 2004.

[DS02] Kevin Dobbin and Richard Simon. Comparison of microarray designs for class comparison and class discovery. *Bioinformatics*, 18(11):1438–1445, 2002.

[DS05] Kevin Dobbin and Richard Simon. Sample size determination in microarray experiments for class comparison and prognostic classification. *Biostat*, 6(1):27–38, 2005.

[DTEF05] Ralph A. Dean, Nicholas J. Talbot, Daniel J. Ebbole, and Mark L. Farman. The genome sequence of the rice blast fungus Magnaporthe grisea. *Nature*, 434(7036):980–986, April 2005.

[DTG+96] W. Dioh, D. Tharreau, R. Gomez, E. Roumen, M. Orbach, J.L. Notteghem, and Lebrun M.H. Mapping avirulence genes in the rice blast fungus Magnaporthe grisea. In *Rice genetics III. Proceedings of the third international rice genetics symposium*, pages 916–920, 1996.

[DW95] E. Drougge and J. Wroldsen. A Robust Algorithm For Pruning Neural Networks, 1995.

[DYCS00] Sandrine Dudoit, Yee Hwa Yang, Matthew J. Callow, and Terence P. Speed. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. Technical Report 578, Stanford University School of Medicine, Department of Biochemistry, August 2000.

[Ebb07] Daniel J. Ebbole. Magnaporthe as a model for understanding host-pathogen interactions. *Annual Review of Phytopathology*, 45(1):437–456, 2007.

[Edw03] David Edwards. Non-linear normalization and background correction in one-channel cDNA microarray studies. *Bioinformatics*, 19(7):825–833, 2003.

[Efr79] Bradley Efron. Bootstrap Methods: Another Look at the Jackknife. *Annals of Statistics*, 7(1):1–26, 1979.

[EG00] Brent Ewing and Phil Green. Analysis of Expressed Sequence Tags Indicates 35,000 Human Genes. *Nature Genetics*, 25:232–234, 2000.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231. AAAI Press, 1996.

[ESBB98] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95(25):14863–14868, December 1998.

[ET93] Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

[ETST01] Bradley Efron, Robert Tibshirani, John D. Storey, and Virginia Tusher. Empirical Bayes Analysis of a Microarray Experiment. *Journal of the American Statistical Association*, 96(456):1151–1160, 2001.

[Eub88] Randall L. Eubank. *Spline Smoothing and Nonparametric Regression*. Marcel Dekker Inc., 1988.

[EZT04] Olivia Eriksson, Yishao Zhou, and Jesper Tegner. Modeling complex cellular networks - robust switching in the cell cycle ensures a piecewise linear reduction of the regulatory network. In *43rd IEEE Conference on Decision and Control*, December 2004.

[FG96] Jianging Fan and Irene Gijbels. *Local Polynomial Modelling and its Applications*. Chapman & Hall, 1996.

[Fic05] Gabriella Ficz. *Protein dynamics in the nucleus: Implications for gene expression*. PhD thesis, Georg August University Göttingen, June 2005.

[Fie93] Emile Fiesler. Minimal and High Order Neural Network Topologies. In *Proc. SPIE Vol. 2204, 5th Workshop on Neural Networks: Academic/Industrial/NASA/Defense. An International Conference on Computational Intelligence: Neural Networks, Fuzzy Systems, Evolutionary Programming and Virtual Reality*, 1993.

[Fie96] Emile Fiesler. Neural Network Topologies, 1996.

[GC02] Debashis Ghosh and Arul M. Chinnaiyan. Mixture modelling of gene expression data from microarray experiments. *Bioinformatics*, 18(2):275–286, 2002.

[GHL05] Mika Gustafsson, Michael Hörnquist, and Anna Lombardi. Constructing and Analyzing a Large-Scale Gene-to-Gene Regulatory Network-Lasso-Constrained Inference and Biological Validation. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 2(3):254–261, 2005.

[GPSB84] M. Ghosh, W. C. Parr, K. Singh, and G. J. Babu. A Note on Bootstrapping the Sample Median. *Annals of Statistics*, 12(3):1130–1135, 1984.

[GS05] Xin Gao and Peter Song. Nonparametric tests for differential gene expression and interaction effects in multi-factorial microarray experiments. *BMC Bioinformatics*, 6(1):186, 2005.

[Hau08] Jan Hauth. *Grey-Box Modelling for Nonlinear Systems*. PhD thesis, University of Technology Kaiserslautern (Germany), December 2008.

[HdH85] M. F. Hutchinson and F. R. de Hoog. Smoothing Noisy Data with Spline Functions. *Numerische Mathematik*, 47:99–106, 1985.

[Hen98] Wolfgang Hennig. *Genetik*. Springer-Verlag, 2nd edition, 1998.

[HHS+02] Wolfgang Huber, Anja von Heydebreck, Holger Sultmann, Annemarie Poustka, and Martin Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18(Suppl.1):S96–104, 2002.

[HKL+05] Iiris Hovatta, Katja Kimppa, Antti Lehmussola, Tomi Pasanen, Janna Saarela, Ilana Saarikko, Juha Saharinen, Pekka Tikkainen, Teemu Toivanen, Martti Tolvanen, Mauno Vihinen, and Garry Wong. *DNA Microarray Data Analysis*. CSC Scientific Computing, Helsinki, 2005.

[HMC+01] Neal S. Holter, Amos Maritan, Marek Cieplak, Nina V. Fedoroff, and Jayanth R. Banavar. Dynamic modeling of gene expression data. *Proceedings of the National Academy of Sciences of the United States of America*, 98(4):1693–1698, 2001.

[HS05] Sariel Har-Peled and Bardia Sadri. How fast is the k-means Method? *Algorithmica*, 41(3):185–202, January 2005.

[JC04] Sung Hoon Jung and Kwang-Hyun Cho. Identification of Gene Interaction Networks Based on Evolutionary Computation. In *Artificial Intelligence and Simulation*, pages 428–439, 2004.

[JGHP03] Hidde de Jong, Johannes Geiselmann, Celine Hernandez, and Michel Page. Genetic Network Analyzer: qualitative simulation of genetic regulatory networks. *Bioinformatics*, 19(3):336–344, 2003.

[Kat01] Hajime Kato. Rice Blast Disease. *Pesticide Outlook*, February 2001.

[KBGW04] Nikola Kasabov, Lubica Benuskova, and Simei Gomes Wysoski. Computational neurogenetic modelling: gene networks within neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 1203–1208, July 2004.

[Ker03] M. Kathleen Kerr. Linear Models for Microarray Data Analysis: Hidden Similarities and Differences. *Journal of Computational Biology*, 10(6):891–901, 2003.

[KMOK00] Koji M. Kyoda, Mineo Morohashi, Shuichi Onami, and Hiroaki Kitano. A gene network inference method from continuous-value gene expression data of wild-type and mutants. *Genome Inform Ser Workshop Genome Inform*, 11:196–204, 2000.

[Kre08] Anette Krengel. Clustering of High-Volume Gene Expression Data from Time Course Microarray Experiments. Diploma Thesis, Technische Universität Kaiserslautern, Fachbereich Mathematik, December 2008.

[Kuy00] Devlin Kuyek. Blast, biotech and big business - Implications of corporate strategies on rice research in asia. http://www.grain.org, August 2000.

[KVLB09] Dries Knapen, Lucia Vergauwen, Kris Laukens, and Ronny Blust. Best practices for hybridization design in two-colour microarray analysis. *Trends in Biotechnology*, 27(7):406–414, July 2009.

[Lap96a] Harri Lappalainen. A computationally efficient algorithm for finding sparse codes. Master Thesis, Helsinki University of Technology, May 1996.

[Lap96b] Harri Lappalainen. Soft multiple winners for sparse feature extraction. In *Neural Networks, 1996., IEEE International Conference on*, volume 1, pages 207–210, June 1996.

[Lee04] Thomas C. M. Lee. Improved smoothing spline regression by combining estimates of different smoothness. *Statistics & Probability Letters*, 67(2):133–140, April 2004.

[LLB+01] International Human Genome Sequencing Consortium: Eric S. Lander, Lauren M. Linton, Bruce Birren, Chad Nusbaum, Michael C. Zody, and Jennifer Baldwin et al. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, February 2001.

[LM97] Tomas Lundin and Perry Moerland. Quantization and Pruning of Multilayer Perceptrons: Towards Compact Neural Networks. Technical Report 2, Institut Dalle Molle d'Intelligence Artificielle Perceptive, March 1997.

[Mac67] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, page 281–297. University of California Press, 1967.

[MBE+09] Shaowu Meng, Douglas Brown, Daniel Ebbole, Trudy Torto-Alalibo, Yeon Oh, Jixin Deng, Thomas Mitchell, and Ralph Dean. Gene Ontology annotation of the rice blast fungus, Magnaporthe oryzae. *BMC Microbiology*, 9(Suppl 1):S8, 2009.

[MBP02] G. J. McLachlan, R. W. Bean, and D. Peel. A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, 18(3):413–422, 2002.

[MCB00] Peter Morgan, Bruce Curry, and Malcom Beynon. Pruning neural networks by minimization of the estimated variance. *European Journal of Economic and Social Systems*, 14(1):1–16, 2000.

[MCZL06] Ping Ma, Cristian I. Castillo-Davis, Wenxuan Zhong, and Jun S. Liu. A data-driven clustering method for time course gene expression data. *Nucleic Acids Research*, 34(4):1261–1269, 2006.

[Men66] Gregor Johann Mendel. Versuche über Pflanzenhybriden. *Verhandlungen des naturforschenden Vereines in Brünn*, 4:3–47, 1866.

[MS02] Mario Medvedovic and Siva Sivaganesan. Bayesian infinite mixture model based clustering of gene expression profiles. *Bioinformatics*, 18(9):1194–1206, 2002.

[MYB04] M. Medvedovic, K.Y. Yeung, and R.E. Bumgarner. Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics*, 20(8):1222–1232, 2004.

[Nat00] The Nature of the Number. *Nature Genetics*, 25:129–130, 2000. (editorial).

[NGW01] Javier Nunez Garcia and Olaf Wolkenhauer. Dynamic Modelling of Microarray Time Course Data. Technical report, University of Manchester Institute of Science and Technology, February 2001.

[NIS09] NIST/SEMATECH. e-Handbook of Statistical Methods. http://www.itl.nist.gov, September 2009.

[NL04] Markus Neuhäser and Fred C. Lam. Nonparametric approaches to detecting differentially expressed genes in replicated microarray experiments. In *APBC '04: Proceedings of the second conference on Asia-Pacific bioinformatics*, pages 139–143, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[ORSS06] Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman, and Chaitanya Swamy. The Effectiveness of Lloyd-Type Methods for the k-Means Problem. *47th Annual IEEE Symposium on Foundations of Computer Science*, pages 165–176, 2006.

[Pea95] Barak A. Pearlmutter. Gradient Calculations for Dynamic Recurrent Neural Networks: A Survey. *IEEE Transactions on Neural Networks*, 6:1212–1228, 1995.

[Pen03a] Elizabeth Pennisi. A Low Number Wins the GeneSweep Pool. *Science*, 300(5625):1484, 2003.

[Pen03b] Elizabeth Pennisi. Gene Counters Struggle to Get the Right Answer. *Science*, 301(5636):1040–1041, 2003.

[PL09] PaDIL Pests and Diseases Image Library. Diagnostic Methods for Rice Blast Magnaporthe grisea. http://www.padil.gov.au, April 2009.

[Pol99] D.S.G. Pollock, editor. *Handbook of Time Series Analysis, Signal Processing, and Dynamics (Signal Processing and its Applications)*. Academic Press, 1999.

[PS05] Lior Pachter and Bernd Sturmfels, editors. *Algebraic Statistics for Computational Biology*. Cambridge University Press, August 2005.

[RD03] David M. Rocke and Blythe Durbin. Approximate variance-stabilizing transformations for gene-expression microarray data. *Bioinformatics*, 19(8):966–972, 2003.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning internal representations by error propagation*, chapter 8, pages 318–362. Parallel Distributed Processing. MIT Press, Cambridge, 1986.

[Rie98] Britta Riege. Strukturmaße für dynamische Systeme. Technical Report 1, Gerhard-Mercator-Universität - GH Duisburg, 1998.

[RMV05] Francesca Ruffino, Marco Muselli, and Giorgio Valentini. Biological Specifications for a Synthetic Gene Expression Data Generation Model. In *Fuzzy Logic and Applications*, pages 277–283, 2005.

[RN88] Joseph Lee Rodgers and W. Alan Nicewander. Thirteen Ways to Look at the Correlation Coefficient. *The American Statistician*, 42(1):59–66, February 1988.

[RSO+07] Matthew E. Ritchie, Jeremy Silver, Alicia Oshlack, Melissa Holmes, Dileepa Diyagama, Andrew Holloway, and Gordon K. Smyth. A comparison of background correction methods for two-colour microarrays. *Bioinformatics*, 23(20):2700–2707, 2007.

[Sav76] Michael A. Savageau. *Biochemical Systems Analysis*. Addison-Wesley Publishing Company, 1976.

[SAW+00] Zhou Shui-geng, Zhou Ao-ying, Jin Wen, Fan Ye, and Qian Wei-ning. FDBSCAN: A Fast DBSCAN Algorithm. *Journal of Software*, 11(6):735–744, June 2000.

[SB05] Benno Stein and Michael Busch. Density-based Cluster Algorithms in Low-dimensional and High-dimensional Applications. *Fachberichte Informatik*, pages 45–56, 2005.

[Sch09] Bernhard Schmitt. Krylov-Iterationsverfahren, Vorlesung, Wintersemester 2008/09.

[Sch46] I. J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. *Quarterly of Applied Mathematics*, 4:45–99, 112–141, 1946.

[Sch64] I. J. Schoenberg. Spline Functions And The Problem Of Graduation. *Proceedings of the National Academy of Sciences of the United States of America*, 52(4):947–950, 1964.

[Sch07] Larry L. Schumaker. *Spline Functions: Basic Theory*. Cambridge Mathematical Library. Cambridge University Press, 3 edition, 2007.

[SH09] Lothar Sachs and Jürgen Hedderich. *Angewandte Statistik*. Springer-Verlag, 2009.

[SHS01] Kenji Suzuki, Isao Horiba, and Noboru Sugie. A Simple Neural Network Pruning Algorithm with Application to Filter Synthesis. *Neural Processing Letters*, 13(1):43–53, 2001.

[SJ01] S. Sreenivasaprasad and R. Johnson, editors. *Major Fungal Diseases of Rice, Recent Advances*. Kluwer Academic Publishers, 2001.

[SL00] Rudy Setiono and Wee Kheng Leow. Pruned Neural Networks for Regression. In *In Proc. of the 6th Pacific Rim Conference on Artificial Intelligence, PRICAI 2000, Lecture Notes in AI 1886*, pages 500–509, 2000.

[SMS05] Gordon K. Smyth, Joelle Michaud, and Hamish S. Scott. Use of within-array replicate spots for assessing differential expression in microarray experiments. *Bioinformatics*, 21(9):2067–2075, 2005.

[Smy04] Gordon K. Smyth. Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 2004.

[SPT+04] Fatima Sanchez-Cabo, Andreas Prokesch, Gerhard G. Thallinger, Zlatko Trajanoski, Philip D. Butcher, Jason Hinds, Leah E. A. Holmes, Susan G. Campbell, Mark P. Ashe, Simon Hubbard, and Kwang-hyun Cho. Assessing the Efficiency of Dye-Swap Normalization to Remove Systematic Bias from Two-Color Microarray Data. http://www.sbi.uni-rostock.de, 2004.

[Spä73] Helmut Späth. *Spline-Algorithmen zur Konstruktion glatter Kurven und Flächen*. Verfahren der Datenverarbeitung. R. Oldenbourg Verlag, 1973.

[SRDM03] Richard Simon, Michael D. Radmacher, Kevin Dobbin, and Lisa M. McShane. Pitfalls in the Use of DNA Microarray Data for Diagnostic and Prognostic Classification. *Jornal of the National Cancer Institute*, 95(1):14–18, 2003.

[SS03] Gordon K. Smyth and Terry Speed. Normalization of cDNA microarray data. *Methods*, 31(4):265–273, 2003. Candidate Genes from DNA Array Screens: application to neuroscience.

[SSDB95] Mark Schena, Dari Shalon, Ronald W. Davis, and Patrick O. Brown. Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray. *Science*, 270:467–470, October 1995.

[SSS03] Alexander Schliep, Alexander Schonhuth, and Christine Steinhoff. Using hidden Markov models to analyze gene expression time course data. *Bioinformatics*, 19(Suppl.1):i255–i263, 2003.

[Ste04] Licoln D. Stein. Human Genome: End of the beginning. *Nature*, 431:915–916, October 2004.

[Str97] Nikko Ström. Sparse Connection And Pruning In Large Dynamic Artificial Neural Networks, 1997.

[SWG+04] Jeremy Schmutz, Jeremy Wheeler, Jane Grimwood, Mark Dickson, Joan Yang, and Chenier Caoile et al. Quality assessment of the human genome sequence. *Nature*, 429:365–368, May 2004.

[SWGG07] Jürgen Schmidhuber, Daan Wierstra, Matteo Gagliolo, and Faustino Gomez. Training Recurrent Networks by Evolino. *Neural Computation*, 19(3):757–779, 2007.

[SX03] Minghe Sun and Momiao Xiong. A mathematical programming approach for gene selection and tissue classification. *Bioinformatics*, 19(10):1243–1251, 2003.

[Tal03] Nicholas J. Talbot. On the Trail of a Cereal Killer: Exploring the Biology of Magnaporthe grisea. *Annual Review of Microbiology*, 57(1):177–202, 2003.

[TAW04] Eckhard Thines, Heidrun Anke, and Roland W. S. Weber. Review Article: Fungal secondary metabolites as inhibitors of infection-related morphogenesis in phytopathogenic fungi. *Mycological Research*, 108(1):14–25, 2004.

[TCS+01] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[TDG99] Frank L. Tobin, Valeriu Damian-Iordache, and Larry D. Greller. Towards the Reconstruction of Gene Regulatory Networks. http://www.nsti.org, 1999.

[TF95] Georg Thimm and Emile Fiesler. Evaluating Pruning Methods. In *National Chiao-Tung University*, page 2, 1995.

[TF96a] Georg Thimm and Emile Fiesler. A Neural Networks Construction Method based on Boolean Logic. In *IEEE International Conference on Tools with Artificial Intelligence*, 1996.

[TF96b] Georg Thimm and Emile Fiesler. Neural Network Pruning and Pruning Parameters. In *1st Online Workshop on Soft Computing*, 1996.

[TF97] Georg Thimm and Emile Fiesler. Pruning of Neural Networks. Technical Report 3, Institut Dalle Molle d'Intelligence Artificielle Perceptive, February 1997.

[THC+99] Saeed Tavazoie, Jason D. Hughes, Michael J. Campbell, Raymond J. Cho, and George M. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, 1999.

[TSM+99] Pablo Tamayo, Donna Slonim, Jill Mesirov, Qing Zhu, Sutisak Kitareewan, Ethan Dmitrovsky, Eric S. Lander, and Todd R. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences of the United States of America*, 96(6):2907–2912, 1999.

[TYHC03] Jesper Tegnér, M. K. Stephen Yeung, Jeff Hasty, and James J. Collins. Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *Proceedings of the National Academy of Sciences of the United States of America*, 100(10):5944–5949, 2003.

[VAM+01] J. Craig Venter, Mark D. Adams, Eugene W. Myers, Peter W. Li, Richard J. Mural, and Granger G. Sutton et al. The Sequence of the Human Genome. *Science*, 291(5507):1304–1351, February 2001.

[VVK06] Jarkko Venna, Jarkko Venna, and Samuel Kaski. Visualizing gene interaction graphs with local multidimensional scaling. In *European Symposium on Artificial Neural Networks*, pages 557–562, Bruges (Belgium), April 2006.

[WBHW03] D.L. Wilson, M.J. Buckley, C.A. Helliwell, and I.W. Wilson. New normalization methods for cDNA microarray data. *Bioinformatics*, 19(11):1325–1332, 2003.

[Wer90] P. J. Werbos. Backpropagation through time: what it does and how to do it. In *Proceedings of the IEEE*, volume 78, pages 1550–1560, 1990.

[Wer92a] Jochen Werner. *Numerische Mathematik 1; Lineare und nichtlineare Gleichungssysteme, Interpolation, numerische Integration*. Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, 1992.

[Wer92b] Jochen Werner. *Numerische Mathematik 2; Eigenwertaufgaben, lineare Optimierungsaufgaben, unrestringierte Optimierungsaufgaben*. Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, 1992.

[WH00] Matthias Wahde and John Hertz. Coarse-grained reverse engineering of genetic regulatory networks. *Biosystems*, 55(1-3):129–136, February 2000.

[WH01] Mattias Wahde and John Hertz. Modeling Genetic Regulatory Dynamics in Neural Development. *Journal of Computational Biology*, 8(4):429–442, 2001.

[Wil92] Ronald J. Williams. Training Recurrent Networks Using the Extended Kalman Filter. In *In Proceedings International Joint Conference on Neural Networks*, pages 241–246, 1992.

[WJJ+02] Christopher Workman, Lars Juhl Jensen, Hanne Jarmer, Randy Berka, Laurent Gautier, Henkik Bjørn Nielsen, Hans-Henrik Saxild, Claus Nielsen, Søren Brunak, and Steen Knudsen. A new non-linear normalization method for reducing variability in DNA microarray experiments. *Genome Biology*, 3(9):research0048.1–research0048.16, August 2002.

[WT09] Richard A. Wilson and Nicholas J. Talbot. Under pressure: investigating the biology of plant infection by Magnaporthe oryzae. *Nature Reviews Microbiology*, 7(3):185–195, March 2009.

[WV09] Guo-Liang Wang and Barbara Valent, editors. *Advances in Genetics, Genomics and Control of Rice Blast Disease*. Springer, 2009.

[YDLS01] Yee Hwa Yang, Sandrine Dudoit, Percy Luu, and Terence P. Speed. Normalization for cDNA microarray data. Technical report, SPIE BiOS, San Jose, California, 2001.

[YFM+01] Ka Yee Yeung, Chris Fraley, Alejandro Murua, Adran E. Raftery, and Walter L. Ruzzo. Model-Based Clustering and Data Transformations for Gene Expression Data. Technical Report 396, University of Washington, Department of Statistics, April 2001.

[YHR00] Ka Yee Yeung, David R. Haynor, and Walter L. Ruzzo. Validating Clustering for Gene Expression Data. *Bioinformatics*, 17:309–318, 2000.

[YK06] Ming Yuan and Christina Kendziorski. A Unified Approach for Simultaneous Gene Clustering and Differential Expression Identification. *Biometrics*, 62(4):1089–1098, 2006.

[YMB03] Ka Yeung, Mario Medvedovic, and Roger Bumgarner. Clustering gene-expression data with repeated measurements. *Genome Biology*, 4(5):R34, 2003.

[YTC02] M. K. Stephen Yeung, Jesper Tegnér, and James J. Collins. Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences of the United States of America*, 99(9):6163–6168, April 2002.

[Zer02] Eva Zerz. Introduction to Systems and Control Theory. Technical Report 248, University of Kaiserslautern, Department of Mathematics, Berichte der AG Technomathematik, July 2002.

[ZHP01] Eva Zerz, Uwe Helmke, and Dieter Prätzel-Wolters. Mathematical Theory of Neural Networks. Technical Report 238, University of Kaiserslautern, Department of Mathematics, Berichte der AG Technomathematik, July 2001.

Furthermore, the following sources were used:

- MathWorks Matlab online documentation.
  http://www.mathworks.com
- The R Project for Statistical Computing. R online documentation.
  http://www.r-project.org/
- Websites of the Wikimedia Foundation.
  http://wikimediafoundation.org
  Pictures based on the GNU Free Documentation License are marked in the List of Figures.
- A. S. Hornby. Oxford Advanced Learner's Dictionary of Current English.
  Cornelsen & Oxford, 5th edition, 1995
- LEO Dictionary German - English. online service of the LEO GmbH.
  http://dict.leo.org/
- Helmut Kopka. LaTeX Einführung Band 1. Addison-Wesley, 2000
- Peter Mösgen. Makeindex - Sachregister erstellen mit LaTeX. Schriftenreihe des Universitätsrechenzentrums Nr. 14, Katholische Universität Eichstätt, 1998

The reference date for all webpages cited or used in this thesis is the 4th of January 2010, unless otherwise noted.

Index

T

U

V

Wissenschaftlicher Werdegang

- 1999 Abitur am Kant-Gymnasium in Boppard
- 2000–2005 Studium im Diplomstudiengang Technomathematik mit Anwendungsfach Elektrotechnik an der Technischen Universität Kaiserslautern
- 2005 Diplom
- 2005–2008 Stipendiat der Graduate School des Fachbereichs Mathematik an der Technischen Universität Kaiserslautern
- seit 2008 Wissenschaftlicher Mitarbeiter am Fraunhofer Institut für Techno- und Wirtschaftsmathematik Kaiserslautern

Scientific Career

- 1999 Abitur at the Kant-Gymnasium in Boppard (Germany)
- 2000–2005 Studies in Mathematics with minor subject Electrical Engineering at the University of Technology in Kaiserslautern (Germany)
- 2005 German Diplom
- 2005–2008 Scholarship of the Graduate School at the Department of Mathematics at the University of Technology in Kaiserslautern
- since 2008 Scientific assistant at the Fraunhofer Institute for Industrial Mathematics Kaiserslautern