TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY
Julkaisu 548 • Publication 548

Harri Lähdesmäki

Computational Methods for Systems Biology:
Analysis of High-Throughput Measurements and Modeling of Genetic Regulatory Networks

Tampere 2005

Tampereen teknillinen yliopisto. Julkaisu 548
Tampere University of Technology. Publication 548

Harri Lähdesmäki

Computational Methods for Systems Biology:
Analysis of High-Throughput Measurements and Modeling of Genetic Regulatory Networks

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB109, at Tampere University of Technology, on the 27th of October 2005, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2005

ISBN 952-15-1454-X (printed)
ISBN 952-15-1835-9 (PDF)
ISSN 1459-2045

Abstract

High-throughput measurement techniques have revolutionized the field of molecular biology by gearing biological research towards approaches that involve extensive collection of experimental data and integrated analysis of biological systems on a genome-wide scale. Integration of experimental and computational approaches to understand complex biological systems—computational systems biology—has the potential to play a profound role in making life science discoveries in the future. Analysis of massive amounts of measurement data and modeling of high-dimensional biological systems inevitably require advanced computational methods in order to draw valid biological conclusions.

This thesis introduces novel computational methods for the problems encountered in the field of systems biology. The content of the thesis is three-fold.

The first part introduces methods for high-throughput measurement preprocessing. Two general methods for correcting systematic distortions originating from sample heterogeneity and sample asynchrony are developed. The former distortion is typically present in experiments conducted on non-homogeneous cell populations and the latter is encountered in practically all biological time series experiments.

The second topic focuses on robust time series analysis. General methods for both robust spectrum estimation and robust periodicity detection are introduced. Robust computational methods are preferred because the exact statistical characteristics of high-throughput data are generally unknown and the measurements are also prone to contain other non-idealities, such as outliers and distortion from the original wave form.

The third part is devoted to integrated analysis of genetic regulatory networks, or biological networks as they are also called, on a global scale. The effect of certain Post function classes on general properties of genetic regulatory networks, such as robustness and ordered and chaotic behavior, is studied in the Boolean network framework. In order to facilitate the analysis of generic properties of biological networks, efficient spectral methods for testing membership in the studied Post function classes and the class of forcing functions (as well as its variants) are introduced. Fast optimized search algorithms are developed for the inference of regulatory functions from experimental data. Relationships between two commonly used stochastic networks models, probabilistic Boolean networks (PBN) and dynamic Bayesian networks (DBN), are also established. This connection provides a way of applying the standard tools of DBNs to PBNs and the other way around.

Acknowledgements

I am grateful to my advisor Prof. Olli Yli-Harja for guiding me through the doctoral studies. His sense of humor and continuous encouragement and support have inspired me along the way. I would also like to thank my friends and colleagues in our Computational Systems Biology group.

I am highly grateful to Prof. Ilya Shmulevich for his invaluable guidance and deep involvement during this process. Ilya's influence and contribution to this dissertation cannot be overemphasized. I am also indebted to Prof. Wei Zhang for the excited and insightful guidance he gave me on systems biology.

This work has been carried out at the Institute of Signal Processing in Tampere University of Technology and partly at the Cancer Genomics Laboratory in The University of Texas M. D. Anderson Cancer Center during my research visit there. Special thanks go to all the personnel in both institutes. The financial support of the Tampere Graduate School in Information Science and Engineering (TISE), Academy of Finland, Emil Aaltonen Foundation, Kauhajoki Cultural Foundation, Jenny and Antti Wihuri Foundation and Instrumentarium Foundation are gratefully acknowledged.

I would like express my gratitude to my parents, mother Liisa and father Kari, and to my sister Riitta for their constant support. And finally, my warm thanks goes to my wife Marianna.

Contents

List of Publications

This thesis is based on the following publications. In the text, these publications are referred to as Publication-I, Publication-II, etc.

I Lähdesmäki, H., Huttunen, H., Aho, T., Linne, M.-L., Niemi, J., Kesseli, J., Pearson, R. and Yli-Harja, O. (2003) Estimation and inversion of the effects of cell population asynchrony in gene expression time-series. *Signal Processing*, Vol. 83, No. 4, pp. 835–858.

II Lähdesmäki, H., Shmulevich, I. and Yli-Harja, O. (2003) On learning gene regulatory networks under the Boolean network model. *Machine Learning*, Vol. 52, No. 1–2, pp. 147–167.

III Shmulevich, I., Lähdesmäki, H., Dougherty, E.R., Astola, J. and Zhang, W. (2003) The role of certain Post classes in Boolean network models of genetic networks. *Proceedings of the National Academy of Sciences of the USA*, Vol. 100, No. 19, pp. 10734–10739.

IV Pearson, R.K., Lähdesmäki, H., Huttunen, H. and Yli-Harja, O. (2003) Detecting periodicity in nonideal datasets. In *SIAM International Conference on Data Mining 2003*, Cathedral Hill Hotel, San Francisco, CA, May 1-3.

V Shmulevich, I. Lähdesmäki, H. and Egiazarian, K. (2004) Spectral methods for testing membership in certain Post classes and the class of forcing functions. *IEEE Signal Processing Letters*, Vol. 11, No. 2, pp. 289–292.

VI Lähdesmäki, H., Shmulevich, I., Yli-Harja, O. and Astola, J. (to appear) Inference of genetic regulatory networks via Best-Fit extensions. To appear in W. Zhang and I. Shmulevich (Eds.) *Computational And*

Statistical Approaches To Genomics (2nd ed.), Boston: Kluwer Academic Publishers.

VII Lähdesmäki, H., Shmulevich, I., Dunmire, V., Yli-Harja, O. and Zhang, W. (2005) *In silico* microdissection of microarray data from heterogeneous cell populations. *BMC Bioinformatics*, 6:54.

VIII Lähdesmäki, H., Hautaniemi, S., Shmulevich, I. and Yli-Harja, O. (to appear) Relationships between probabilistic Boolean networks and dynamic Bayesian networks as models of gene regulatory networks. To appear in *Signal Processing*.

IX Ahdesmäki, M.,[†] Lähdesmäki, H.,[†] Pearson, R., Huttunen, H. and Yli-Harja, O. (2005) Robust detection of periodic time series measured from biological systems. *BMC Bioinformatics*, 6:117.

The author's contribution to Publications II, VI, VII and VIII is as follows. As the first author of these publications, H. Lähdesmäki designed and implemented the computational methods, derived the mathematical proofs, and wrote the manuscript for most part, with the exception that Publication VI was co-written with I. Shmulevich. W. Zhang and I. Shmulevich also contributed to Publication VII by providing essential ideas and assisting in drafting the manuscript.

Publication I was a result of collective efforts. As the first author, H. Lähdesmäki had a major role in writing the manuscript. The author was also mainly responsible for the development of those computational methods that are covered in this thesis. Other subtopics to which the author did not make the main contribution, such as the proposed blind deconvolution method developed by Dr. H. Huttunen, are not discussed in this thesis in detail.

In Publications III and V, the author assisted in developing the computational methods and co-performed the simulations. In Publication IV, the author performed the simulations and helped in refining the computational methods.

M. Ahdesmäki and H. Lähdesmäki were equal contributors to Publication IX. H. Lähdesmäki developed the statistical methods, assisted in performing the simulations and mainly drafted the manuscript. M. Ahdesmäki carried out an implementation of the methods, performed the most of the extensive simulations and co-drafted the manuscript.

The author has also published the following related publications. In the text, these publications are referred to as Publication-A, Publication-B and Publication-C.

A Lähdesmäki, H., Hao, X., Sun, B., Hu, L., Yli-Harja, O., Shmulevich, I. and Zhang, W. (2004) Distinguishing key biological pathways between primary breast cancers and their lymph node metastases by gene function-based clustering analysis. *International Journal of Oncology*, Vol. 24, No. 6, pp. 1589–1596.

B Hao, X., Sun, B., Hu, L., Lähdesmäki, H., Dunmire, V., Feng, Y., Zhang, S.-W., Wang, H., Wu, C., Wang, H., Fuller, G.N., Symmans, W.F., Shmulevich, I. and Zhang, W. (2004) Differential gene and protein expression in primary breast malignancies and their lymph node metastases as revealed by combined cDNA microarray and tissue microarray analysis. *Cancer*, Vol. 100, No. 6, pp. 1110–1122.

C Lähdesmäki, H., Yli-Harja, O., Zhang, W. and Shmulevich, I. (2005) Intrinsic dimensionality in gene expression analysis. In *IEEE International Workshop on Genomic Signal Processing and Statistics 2005*, Hyatt Regent Hotel, New Port, Rhode Island, May 22-24.

Chapter 1

Introduction

Technological developments have commonly preceded important discoveries in life sciences. For example, X-ray crystallography methods, among others, played an essential role in the discovery of the double-helical structure of DNA (Watson and Crick, 1953). Initiated with the first rapid DNA sequencing methods (Maxam and Gilbert, 1977; Sanger *et al.*, 1977), one of the latest milestones, completion of the Human Genome Project, was recently achieved (The Genome International Sequencing Consortium, 2001; Venter *et al.*, 2001) thanks to modern computing facilities and interdisciplinary efforts in developing more efficient DNA sequencing methods. These discoveries have had a profound effect in changing the face of the life sciences. Uncovering the structure of DNA provided researchers with the explanation of the heredity by means of passing the genetic information from one generation to another through DNA, hence filling in the missing piece of the well-known Darwinian view of the progression of life (Darwin, 1859). Understanding of the structure of DNA also equipped researchers with a basic understanding of its function. Furthermore, the whole genome sequences of different species already available enable a more refined understanding of the operation of complex biological machineries.

Although a cell's operational instructions are stored in its genome (see, e.g., Hood and Galas, 2003), the operation of the biological system in a living cell is only partly performed using DNA. The actual functional part is carried out to a great extent by proteins. The proteins, in turn, are products of DNA. More specifically, DNA is transcribed into messenger RNAs which are further translated into proteins with the help of ribosomes.

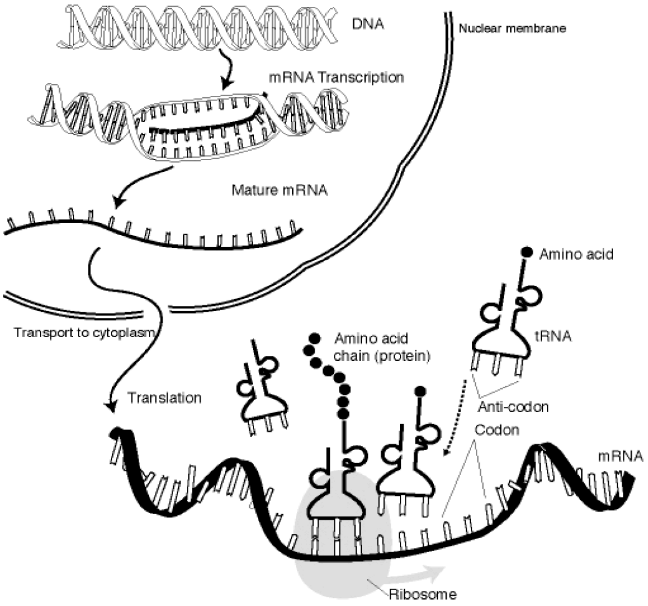


Figure 1.1: Illustration of cell's operation at genomic level. Image is taken from Access Excellence at The National Health Museum.

Proteins, such as transcription factors, or complexes they form with other molecules, can in turn bind back to DNA (see, e.g., Alberts *et al.*, 2002, and Figure 1.1 for illustration). Hence a loop in a biological system is obtained. The above description of a cell's operation at a genomic level is only illustrative since there are a number of other factors, both intra- and extracellular, affecting the overall biological processes. However, it should be evident that in order to have a more comprehensive view of a cell's operation, the whole-genome sequence information is not yet enough but some knowledge of the operational components themselves should also be available. This is the context where the latest technological innovations, such as microarrays and other developing measurement techniques, enter the scene.

Microarray technology, introduced about 10 years ago (Schena *et al.*, 1995), has established its role as a standard tool for probing cell populations. Being highly parallel, a single microarray chip can currently be used, e.g., to measure the transcription levels of all the genes in the human genome. Although the transcription levels do not directly correspond to the abundance of proteins, transcription levels are at the closest proximity to the protein levels that can be measured in high-throughput, genome-wide fashion at the moment. It is the genome-wide nature of the microarray technique that makes it particularly attractive. The possibility of collecting whole-genome measurements sets a turning-point in biological research. Contrary to the old-fashioned, reductionistic research approaches where a single or few components are studied at a time, these new methodologies are especially well-suited to study complex, integrated behavior of biological systems.

It is nowadays widely recognized that biological systems operate in highly parallel and integrated fashion (see, e.g., Davidson *et al.*, 2002). In other words, each component in a biological system rarely functions in isolation but usually co-operates with a larger group or module of interacting components. Biological systems also constantly process their complex machinery, e.g., by carrying out their basic functions, such as the fundamental cell cycle. Consequently, from a systems theoretical point of view, biological systems can be considered as highly parallel dynamical systems where the molecules form the components of the system and their reactions define the system dynamics. The next major landmarks in life sciences include uncovering a detailed understanding of the regulatory operation of a living cell. In other words, an important goal is to gain a system-level understanding of the manner in which genes and their products collectively form a biological system.

System-level description and modeling of biological systems inevitably requires formal modeling methods. Consequently, a significant role is played by the development and analysis of mathematical, statistical and computational methods to construct formal models of biological systems. In order to be able to address the questions and needs of the current systems biology research, the aforementioned high-throughput measurement techniques will play an essential role in future research. Although the high-throughput measurement techniques will most probably change over the years, the need for analyzing the massive amounts of data they produce will remain. Novel, high-throughput measurement techniques do not, however, come without their own puzzles. From a computational point of view, much needs to be done in developing proper and efficient ways of analyzing the complex measurement systems as well. That further emphasizes the necessity of the computational approaches.

Being an immense challenge, system-level understanding of biological systems cannot be developed overnight. Interdisciplinary efforts and achievements made world-wide will gradually lead to a better understanding and, hopefully, will finally provide a satisfactory solution. In the hope of providing useful information and advancing the field, this thesis introduces some results to the above-listed problems.

The results presented here are introduced in a linear fashion starting from the preprocessing of high-throughput measurements and ending up with their dynamical analysis. Each chapter is also expanded with necessary background and reviews of previously proposed methods. Chapter 2 focuses on preprocessing of high-throughput measurements. Two general methods for correcting systematic distortions stemming from heterogeneity and asynchrony of biological sample are introduced in Sections 2.2 and 2.3, respectively. Chapter 3 continues the analysis of gene expression time series already started in the previous chapter. A central theme of this chapter revolves around robust time series analysis. Methods for robust spectrum estimation and robust periodicity detection are introduced in Sections 3.2.2 and 3.3.2, respectively. It is also worth noting that although the computational methods are introduced in the context of microarray measurements in Chapters 2 and 3, the proposed methods are general and can be applied to other types of measurements as well. Chapter 4 is devoted to a more integrated and more comprehensive analysis of genetic regulatory networks, or biological networks, as they are commonly called. The first part of this chapter concentrates solely on generic principles of biological networks, such as robustness and ordered and chaotic behavior. The role of certain type of regulatory rules, namely Post functions, is studied in Section 4.1.2. In order to facilitate the study of generic properties of biological networks, efficient spectral membership testing methods for the studied Post function classes as well as the class of forcing functions are introduced in Section 4.2. Towards the end of this chapter, the emphasis is moved to more realistic approaches and more realistic network models. Two particular results are considered: an efficient inference of regulatory functions in Section 4.3, and relationships between different probabilistic network models in Section 4.4. Concluding remarks are given in Chapter 5 and the original publications are attached at the end of the thesis.

Chapter 2

Preprocessing of High-Throughput Measurements

The current high-throughput measurement techniques for probing biological samples are considerably complex. For example, in the case of microarrays the measurement process consists of several separate steps, such as extraction of the biological sample, isolation of the RNA, reverse transcription and labelling of the RNA, selection of specific probes (nucleotide sequences), printing or synthesis of the probes, hybridization of the fluorescent-labelled (and possibly amplified) biological sample, laser scanning, use of image processing methods, and storing the detected signals for further computer-based analysis (see, e.g., Schena *et al.*, 1995; Baldi and Hatfield, 2002, and Figure 2.1 for illustration). Many of the steps in the overall measurement process are likely to introduce noise or a systematic bias. Therefore, in order to be able to draw valid biological conclusions, microarray measurements require careful preprocessing and experiment design (see, e.g., Quackenbush, 2002; Speed, 2003) as well as quality control (see, e.g., Zhang *et al.*, 2004).

Microarray technology, either two-color cDNA arrays on glass slides or one-color oligonucleotide arrays on silicon chips, is the most commonly used high-throughput measurement technique. Therefore, this chapter focuses on the preprocessing of high-throughput measurements with a special emphasis on microarray data. However, the developed methods (to be in-

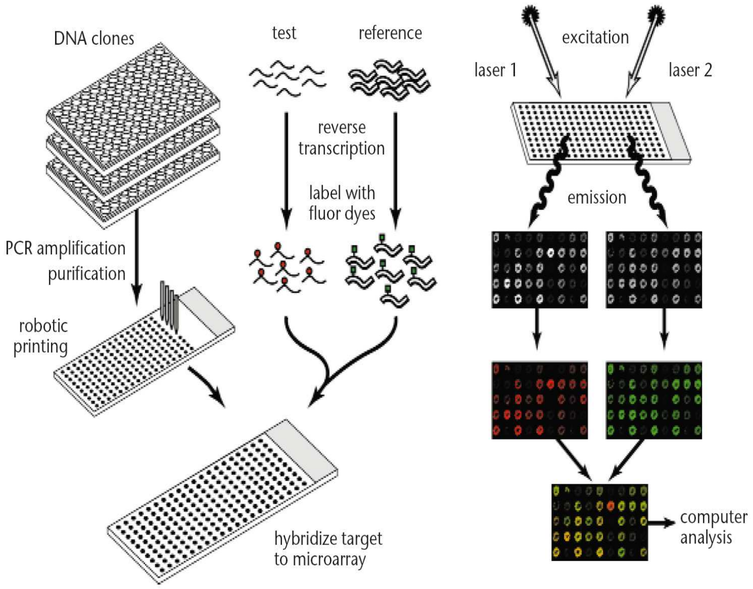


Figure 2.1: An illustration of the (cDNA) microarray experiment. Image is taken from (Duggan *et al.*, 1999).

troduced shortly in Sections 2.2 and 2.3) can be applied, with no or minor modifications, to other types of high-throughput data as well. Before introducing the developed methods we first give an overview of the standard preprocessing steps typically applied to all microarray data prior to further computational or statistical analysis.

2.1 Standard Preprocessing Steps for Microarray Data

The underlying assumption concerning the microarray data is that the measured intensities represent the relative transcription levels of all the genes present in the slide. There are, however, a number of disturbing effects that can make the measurements less quantitative and hinder comparison and analysis of the measured intensities. For example, unequal quantities of the labelled RNA hybridized on different slides are likely to result in different average expression values. Similarly, differences in labelling, emission and detection efficiencies of different fluorescent dyes over- and underemphasize the signals in different channels. Hence, they produce a systematic bias for the measured expression levels. The main purpose of data preprocessing, or normalization, is to facilitate more accurate comparison and analysis of transcription levels both within slide and between different slides by removing the disturbing biases from the measurements.

For the purposes of this and the following sections, it is not necessary to go into the details concerning the data extraction from the scanned microarray images. In the following we assume the recorded raw signal intensities to represent the relative, although non-normalized, transcription levels of different genes. However, it is worth mentioning that the microarray quality control is usually implemented right after the image analysis part and utilizes some image statistics, such as spot intensity, background intensity, pixel-wise variation in spot and background intensities, spot size, roundness of spot, alignment error, and bleeding (Hautaniemi *et al.*, 2003; Speed, 2003; Zhang *et al.*, 2004). Alternatively, if replicate spots or arrays are available, then statistical tests (Ideker *et al.*, 2000*a*) or measures such as coefficient of variation (Tseng *et al.*, 2001) can be used to filter out low quality expression values. Since low quality spots typically result in erroneous intensity values, quality control is important at least for two reasons. Obviously, erroneous (outlying) transcription values can lead to incorrect biological conclusions but, in addition, they can also interfere with the computational preprocessing methods. Further issues in quality control are discussed, e.g., in Zhang *et al.* (2004).

Although several potential noise and bias sources in the microarray technology can be pinpointed, currently no preprocessing method can handle all of them on an individual basis. Such a detailed analysis is prevented by insufficient knowledge of the underlying error mechanisms and their statistical characteristics, and by the limited amounts of data (typically too few replicates). Therefore, the current normalization methods handle the error sources in a quite general manner. Some details of the normalization methods are also platform dependent, i.e., whether cDNA or oligonucleotide chips are used. Yet another difference in preprocessing methods is dependent on whether or not replicates are available, and whether the replicates are within a single array or on different arrays. Replicates within a single array are commonly not considered as real (independent) replicates. Averaging of replicates within an array is appropriate though and results in more accurate expression values. Replicates on different slides are typically utilized, e.g., when the differentially expressed genes are sought. Although a variety of different types of replicated measurements can be considered (see, e.g., Speed, 2003), we do not discuss this issue further. A brief summary of the standard normalization methods follows.

2.1.1 Within Slide Normalization

The within slide normalization is particularly important for the measurements obtained using the two-color cDNA microarray technology. The most commonly used fluorescent dyes, Cy3 and Cy5, have different incorporation efficiencies during the labeling and are also detected by the scanner with different efficiencies. This obscuring systematic variation, so-called label bias, can be satisfactorily accounted for using a robust local regression in the scatterplots of the two channels, also called as loess normalization (Cleveland, 1979; Yang *et al.*, 2002). It is worth noting that the loess provides a nonlinear correction.

In order to correct the label bias, a sufficiently large set of non-differentially expressed genes should be identified to provide a necessary calibration for the loess curve construction. For that purpose, either house keeping genes, control spots, or all genes can be considered (see, e.g., Speed, 2003). In the case of house keeping genes, a predetermined set of genes assumed to be non-differentially expressed is used. The use of house keeping genes suffers at least from two problems. First, the expression levels of genes exhibit natural biological variation. Secondly, the construction of the normalization curve is prone to errors if the cardinality of the predetermined set of genes is small or if the expression values of the house keeping genes do not cover the whole dynamic range. The use of control spots may have the same problems, although a proper microarray design can alleviate that issue. Due to the above listed shortcomings, the most commonly used strategy is to use all the genes in the construction of the loess normalization curve. This approach is based on the assumption that most genes are non-differentially expressed or that the number of up- and down-regulated genes is roughly the same. This assumption is usually considered to be true in large-scale studies (Speed, 2003). Moreover, small deviations from the above conditions do not result in a failure since the robust local regression performed in the loess is error tolerant. The set of all genes can also be reduced by removing the most differentially expressed genes with the help of rank-invariant gene selection schemes (see, e.g., Speed, 2003, and references therein).

Spatial variation within a slide can also be remarkable, e.g., if the microarrays are generated with a robotic printing machine utilizing several print-tips. A standard solution to that problem is to perform the loess normalization for each pin separately. Alternatively, a composite method that combines both the print-tip dependent and independent methods can be applied (Yang *et al.*, 2002).

2.1.2 Between Slides Normalization

After correcting the label bias within each slide, the two-color cDNA data (log-ratios) are already mean-centered but the data are typically further adjusted between slides. The aim is to prevent any single array from having dominating expression values by performing between array scale normalization. Assuming the nonlinear loess normalization is already applied, a sufficient scale normalization can typically be obtained with multiplicative scaling. To that end, simple adjustments, such as the ones based on the sample variance, the median absolute deviation from the median or certain quantiles of individual arrays, have been used successfully (see, e.g., Huang and Pan, 2002; Smyth *et al.*, 2003). More refined adjustments that take the data from all the arrays into account have also been proposed, e.g., the sample variance for a particular array divided by the geometric mean of the sample variances for all the arrays (Yang *et al.*, 2002; Quackenbush, 2002). Similar between array scale corrections have also been developed for data coming from one-color oligonucleotide arrays. One of the first studies to derive scaling factors assumed a particular parametric (Gaussian) model (Hartemink *et al.*, 2001). Indeed, optimal scaling factors for the model Hartemink *et al.* considered were found to conform with certain weighted geometric means. Further discussion and comparison between different between slides normalization methods (for oligonucleotide arrays) is reported in (Hartemink, 2001).

The label bias does not play the same role in one-color oligonucleotide arrays as it plays in two-color cDNA arrays. However, nonlinear relations between one-color oligonucleotide arrays are common (Bolstad *et al.*, 2003). Since the standard scale corrections cannot cope with nonlinearities more advanced normalization methods are required. A recent comparison of normalization methods for oligonucleotide arrays is presented in (Bolstad *et al.*, 2003). So called cyclic loess method makes use of the standard loess normalization. Instead of applying the loess to data from two different channels, it is applied to expression values from two distinct arrays. If more than two arrays are present, then the loess is applied to all pairwise combinations of arrays in an iterative fashion. Quantile method, in turn, forces the distribution of the expression values for each array to be the same. Although this distributional adjustment sounds somewhat forceful and can potentially result in problems especially in the tails of the distribution, both the quantile method and the cyclic loess were found to perform favorably (Bolstad *et al.*, 2003).

Other approaches have also been proposed. The use of analysis of variance (ANOVA) shows a departure from the above listed methods. The ANOVA-based approach can potentially provide an individual treatment of some specific sources of variation, such as the effect of array, dye, sample, gene, and their combinations (Kerr *et al.*, 2000). The methods proposed in this framework so far, however, can only account for linear distortions.

2.1.3 Variance Stabilization, Missing Values and Model-Based Analysis

A common observation is that the homoscedasticity (i.e., equality of variance) does not always hold for microarray data but, instead, the noise variance is proportional to the underlying signal intensity (Chen *et al.*, 1997). Such heteroscedasticity, if not properly taken into account, may hinder further statistical analysis. Consequently, several variance stabilizing transforms have been proposed for both the one-color oligonucleotide and the two-color cDNA microarrays (see, e.g., Huber *et al.*, 2002; Rocke and Durbin, 2003; Durbin and Rocke, 2004). These data transforms are typically applied before other preprocessing steps, such as loess and between slides normalization.

Microarray data are also prone to contain missing values. Two different strategies can be considered. Missing values can be ignored during the preprocessing if the downstream analysis methods are flexible enough to handle the missing values. This usually results in a considerable increase in the computational burden and hence the missing values are typically imputed (Troyanskaya *et al.*, 2001; Bar-Joseph *et al.*, 2002; Zhou *et al.*, 2003*a*).

The final note on standard microarray data preprocessing concerns model-based analysis in which a specific model for the measurements is postulated. Model-based analysis is used especially in the case of identifying differentially expressed genes. A number of different models have been proposed for both the one-color oligonucleotide and the two-color cDNA array data, see, e.g., (Ideker *et al.*, 2000*a*; Rocke and Durbin, 2001; Dror *et al.*, 2003; Gottardo *et al.*, 2003; Cho and Lee, 2004). From a preprocessing point of view, an important aspect is that several factors related to data normalization, such as label effects and scale differences, can also be taken into account in the parametric models. A problem in the model-based microarray analysis is that no commonly agreed standard parametric model has been found so far. This issue is further complicated by the non-standard nature of the two-color cDNA microarray technology, i.e., different laboratories may have slightly different procedures in each step of the microarray experiment. The resulting microarray data is therefore likely to have more or less different statistical characteristics. Having that in mind, a noteworthy exception in the model-based analysis is a general data-driven approach taken in (Dror *et al.*, 2003).

The above discussion gives an overview of the most common non-biological sources of variation and the corresponding normalization methods. Since the microarray measurements are taken from biological samples they contain other general sources of variation as well. Two such noise sources, namely, sample heterogeneity and cell population asynchrony, have biological origin but they have an unwanted, confounding effect on the measurements. Those two noise sources together with their inversion methods are considered in detail in Sections 2.2 and 2.3.

2.2 Sample Heterogeneity

Although a number of different preprocessing methods have been proposed, very few computational approaches have been reported to resolve the variability in microarray measurements stemming from sample heterogeneity. For example, tissue samples used in cancer studies are usually contaminated with the surrounding or infiltrating cell types. This results in an obscuring mixing effect that hinders further statistical analysis, significantly so if different samples contain different proportions of these additional cell types. We studied this problem in Publication-VII and developed computational methods for reconstructing the expression values of the pure cell types from the expression values of the heterogeneous mixtures.

In traditional approaches (see, e.g., Fuller *et al.*, 1999), pathologists carefully evaluate the samples and only select those with more than a certain percentage of cells of interest (e.g., $> 90\%$). This prescreening step can result in the exclusion of many samples and thus decreases the sample size. Also note that the samples are still heterogeneous after the prescreening. Alternatively, laser capture microdissection (LCM) technology can be used to purify the target cells from mixed populations (Emmert-Buck *et al.*, 1996). This approach has seen limited success because it is challenging to maintain RNA stability during the microdissection process. LCM procedures are also time-consuming and yield insufficient quantities of RNA, thus requiring multiple amplification steps that may confound quantitative inferences from gene expression data. Thus, computational preprocessing methods are needed.

Computational methods for removing the mixing effect from heterogeneous samples have been previously proposed in (Lu *et al.*, 2003; Stuart *et al.*, 2004; Venet *et al.*, 2001). Lu *et al.* focused on estimating the fraction of cells in different phases of the cell cycle whereas Stuart *et al.* considered the problem of estimating the cell type specific expression patterns over all samples. In Publication-VII we focus on estimating both the sample and the cell type specific expression values. We also consider estimating the mixing percentages of different cell types in each heterogeneous mixture. Venet *et al.* introduced some preliminary methods for tackling the same problem as we consider here. Furthermore, we also provide non-parametric confidence intervals to facilitate downstream analysis and consider the problem of selecting the correct number of cell types using a general purpose model selection framework.

The developed methods were tested on carefully controlled microarray data consisting of five different heterogeneous mixtures of colon cancer and lymph node samples. For more details of the microarray data, preliminary preprocessing steps, and the computational methods, see Publication-VII.

2.2.1 Modeling and Inversion of Sample Heterogeneity

Since the two samples, colon cancer cells and normal lymphocytes, are mixed at the extracted RNA level in Publication-VII, it is natural to assume the mixing model to be linear. Let x_i^c and x_i^l denote the expression level of the ith gene in the colon cancer and in the lymph node samples, respectively. Let us first assume that only two different cell types are mixed. The sample heterogeneity is modeled by a simple linear model

$$y_i^k = \alpha_k x_i^c + (1 - \alpha_k) x_i^l, \tag{2.1}$$

where y_i^k denotes the expression value of the ith gene in the kth heterogeneous sample, and $0 \leq \alpha_k \leq 1$ denotes the fraction of the colon cancer cells in the kth mixture. Note that in Equation (2.1) it is assumed that the expression level in colon cancer (x_i^c) and lymph node (x_i^l) is "fixed" and does not change between heterogeneous measurements. The same model can be extended to more than two cell types (see Section 2.2.4 below).

The first objective is to invert the mixing effect shown in Equation (2.1). By making some distributional assumptions, one could use standard model-based estimation methods. However, in order to avoid making additional modeling assumptions, we prefer to use the general purpose least squares method. Let the number of genes be n and assume that one has measured the expression values for K different heterogeneous mixtures. Let us also assume for now that the mixing percentages are known or have been measured. For the ith gene the sample heterogeneity can be expressed as[1]

$$\begin{aligned}
\begin{pmatrix} Y_i^1 \\ \vdots \\ Y_i^K \end{pmatrix} &= \begin{pmatrix} \alpha_1 & 1-\alpha_1 \\ \vdots & \vdots \\ \alpha_K & 1-\alpha_K \end{pmatrix} \begin{pmatrix} x_i^c \\ x_i^l \end{pmatrix} + \begin{pmatrix} \epsilon_i^1 \\ \vdots \\ \epsilon_i^K \end{pmatrix} && (2.2) \\
&\Leftrightarrow && (2.3) \\
\mathbf{Y}_i &= \mathbf{A}\mathbf{x}_i + \epsilon_i, && (2.4)
\end{aligned}$$

where ϵ_i is a generic additive noise term. For the purposes of further anal-

[1] Throughout this thesis, vector- and matrix-valued quantities are in boldface. Uppercase letters, such as X and $\mathbf{X}$, are typically used to denote random variables and the lowercase letters, such as x and $\mathbf{x}$, denote the value of the corresponding random variables.

ysis, it is useful to rewrite the above model for all n genes as,

$$\begin{aligned}
\begin{pmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_n \end{pmatrix} &= \begin{pmatrix} \mathbf{A} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix} \\
&\Leftrightarrow \\
\mathbf{Y} &= \tilde{\mathbf{A}}\mathbf{x} + \epsilon \qquad (2.5)
\end{aligned}$$

where $\mathbf{0}$ denotes the K-by-2 zero matrix. Assuming the column rank of $\mathbf{A}$ is full, then so is $\tilde{\mathbf{A}}$, and the well-known least squares solution is given by (see, e.g., Johnson and Wichern, 1998)

$$\hat{\mathbf{x}} = (\tilde{\mathbf{A}}^T\tilde{\mathbf{A}})^{-1}\tilde{\mathbf{A}}^T\mathbf{y}, \qquad (2.6)$$

where $\mathbf{y}$ is the observed value of $\mathbf{Y}$.

As noted above, a common observation is that the homoscedasticity does not always hold for microarray data, but instead, the noise variance depends on the underlying signal intensity (Chen *et al.*, 1997; Huber *et al.*, 2002; Durbin and Rocke, 2004). Such heteroscedasticity may decrease the efficiency of the inversion method shown in Equation (2.6). Fortunately, using the properties of block matrix multiplication and inversion, it is easy to see that the structure of the matrix $\tilde{A}$ ensures that the least squares solution can also be obtained gene-wise as $\hat{\mathbf{x}}_i = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}_i$. Consequently, all we need to assume is that the noise variance is approximately constant for each gene separately.

2.2.2 Optimization of Mixing Fractions

Because the mixing percentages must be measured by some means, they are also likely to contain some error. So, in addition to estimating the expression values of the pure cell types, one would like to estimate the most likely value of the mixing percentages. As above, no assumptions on the noise distributions are being made and we use the least squares method.

This results in the following optimization problem

$$\begin{aligned} &\min_{\tilde{\mathbf{A}},\mathbf{x}} && \|\tilde{\mathbf{A}}\mathbf{x} - \mathbf{y}\| \\ &\text{subject to} && 0 \le \alpha_k \le 1 \quad \text{for all } 1 \le k \le K. \end{aligned} \tag{2.7}$$

It is worth noting that the Kn-by-$2n$ regression matrix $\tilde{\mathbf{A}}$ in Equation (2.7) contains only K free parameters.

Any general purpose iterative optimization method can be used to get a solution. Since iterative methods usually become inefficient/unstable as the number of parameters to be optimized increases we use a two-step approach in the optimization. In the first step, given a proper initial value for $\tilde{\mathbf{A}}$, the least squares solution for $\mathbf{x}$ is found using Equation (2.6).[2] In the second step, the mixing percentages are optimized in the least squares sense (subject to the constraints $0 \le \alpha_k \le 1$ for all $1 \le k \le K$) using the previously found value for $\mathbf{x}$.[3] These two steps are then repeated. Details of the optimization algorithm are shown in Figure 2.2 where $\hat{\mathbf{x}}^{(j)}$ (resp. $\hat{\mathbf{A}}^{(j)}$) denotes the value of $\mathbf{x}$ (resp. $\tilde{\mathbf{A}}$) after the jth iteration. Clearly, at each iteration of steps 2 and 3, the value of the objective function is decreased. Because the objective function is bounded below a minimum will be found.

It is important to note that Equation (2.7) no longer implements an independent inversion of the mixing effect for each gene. Consequently, the possible heteroscedasticity does not cancel out in the same way as it does in Equation (2.6). The possible effects of heteroscedasticity could be circumvented by estimating the mixing percentages for each gene separately but sample size of the current data set does not permit such an analysis.

2.2.3 Confidence Intervals

In order to facilitate the further statistical analysis, it is useful to assess the confidence intervals of the obtained expression estimates. Let us first

[2] Measured values of the mixing percentages were used as the initial values for $\tilde{\mathbf{A}}$.

[3] Given a value for $\mathbf{x}$, least squares solution for the mixing parameters can be obtained easily, e.g., from Equation (2.11). Denote $\mathbf{q} = \left(\hat{x}_1^c - \hat{x}_1^l, \ldots, \hat{x}_n^c - \hat{x}_n^l\right)$ and $\mathbf{p}^k = \left(y_1^k - \hat{x}_1^l, \ldots, y_n^k - \hat{x}_n^l\right)^T$, $k = 1, \ldots, K$, where $\hat{x}$ denotes the estimated expression value from the previous step (step 2 in Figure 2.2). Assuming optimal solution satisfies the constraint $0 \le \alpha_k \le 1$, then the closed-form solution for α_k is $\hat{\alpha}_k = \frac{1}{\mathbf{q}^T\mathbf{q}}\mathbf{q}^T\mathbf{p}^k$. If the constraint is violated, then optimal solution can be obtained, e.g., using a general purpose optimization algorithm.

```
1. Initialize $\hat{\mathbf{A}}^{(1)}$ and set $j = 1$.
2. Minimize $\|\hat{\mathbf{A}}^{(j)}\hat{\mathbf{x}}^{(j)} - \mathbf{y}\|$ for $\hat{\mathbf{x}}^{(j)}$:
       $\hat{\mathbf{x}}^{(j+1)} := (\hat{\mathbf{A}}^{(j)T}\hat{\mathbf{A}}^{(j)})^{-1}\hat{\mathbf{A}}^{(j)T}\mathbf{y}.$
3. Minimize $\|\hat{\mathbf{A}}^{(j)}\hat{\mathbf{x}}^{(j+1)} - \mathbf{y}\|$ for $\hat{\mathbf{A}}^{(j)}$ (subject to constraints
   $0 \leq \alpha_k \leq 1$ for all $1 \leq k \leq K$). Increase the iteration
   index $j := j + 1$.
4. Repeat steps 2 and 3.
```

Figure 2.2: Details of the two-step algorithm used for the optimization problem shown in Equation (2.7).

assume that the expression estimates are obtained by applying Equation (2.6). Should the noise ϵ_i^k be i.i.d. with a variance σ^2, then the variance of the estimated expression values would be $\mathbb{V}(\hat{\mathbf{x}}) = \sigma^2(\tilde{\mathbf{A}}^T\tilde{\mathbf{A}})^{-1}$. As explained above, the inversion (Gauss-Markov theorem) can also be applied gene-wise, which greatly alleviates the issue of heteroscedasticity. In such a scenario, the variance of the estimated expression values for the ith gene can be expressed as

$$\mathbb{V}(\hat{\mathbf{x}}_i) = \sigma_i^2(\mathbf{A}^T\mathbf{A})^{-1}, \tag{2.8}$$

where σ_i^2 is the noise variance for the ith gene. A straightforward way of obtaining an estimate of the variance is to compute the sample noise variance $\hat{\sigma}_i^2$ for each gene and then apply Equation (2.8) to get $\hat{\mathbb{V}}(\hat{\mathbf{x}}_i)$. Given our particular data set, that would result in somewhat sensitive variance estimates since there are only $K = 5$ error residuals associated with each gene. A better alternative is to pool genes which have approximately the same average expression value $1/K\sum_{k=1}^{K} y_i^k$ and then compute the sample noise variance from the error residuals of the pooled genes.

Although we do not assume a Gaussian noise distribution, we can resort to the Gaussian approximation when computing the confidence intervals. For example, using the Gaussian approximation, the $1 - 2\alpha$ confidence interval for the estimated expression value of the ith gene in the colon cancer cells is

$$\left[\hat{x}_i^c - \Phi^{-1}(1-\alpha)\sqrt{(\hat{\mathbb{V}}(\hat{\mathbf{x}}_i))_{11}}\ ,\ \hat{x}_i^c + \Phi^{-1}(1-\alpha)\sqrt{(\hat{\mathbb{V}}(\hat{\mathbf{x}}_i))_{11}}\right], \tag{2.9}$$

where $\Phi^{-1}(\cdot)$ is the inverse of the standard normal cumulative distribution function and $(\hat{\mathbb{V}}(\hat{\mathbf{x}}_i))_{11}$ denotes the $(1,1)$ element of the estimated variance matrix $\hat{\mathbb{V}}(\hat{\mathbf{x}}_i)$ (similarly for the lymph node sample). Alternatively, the confidence intervals can be obtained using the non-parametric bootstrap framework (Efron and Tibshirani, 1993). Here we consider the method in which one re-samples the error residuals with replacement (within the set of pooled genes) and computes the confidence intervals directly from the α and $1-\alpha$ percentiles of the bootstrap distribution of the expression estimates.

Let us then focus on confidence intervals of the expression estimates obtained using Equation (2.7). We propose to use the methodology described above in this case as well. However, as noted above, possible heteroscedasticity does not completely cancel out in Equation (2.7). Consequently, the expression estimates as well as the corresponding confidence intervals for individual genes are not completely independent regarding the possible heteroscedasticity. Therefore, the confidence intervals in this case are not completely in concordance with the model and the above discussion, but must be viewed as estimates that are constructed afterwards. The effect of this issue on the confidence intervals appears to be rather small though. This is seen, e.g., in Figure 2.3 that show the estimated 90% confidence intervals from a set of genes. The width of the confidence intervals varies for different genes and clearly correlates with the underlying expression values, e.g., about 10 units for a low-expressed gene TP53 and about 120 units for a high-expressed gene having an accession number NM_002765. Similar observations apply to other genes shown in Figures 2.3 and 2.4, too.

2.2.4 Selecting the Number of Cell Types

Although it is known that only two cell types are mixed in experiments in Publication-VII there may be other experimental settings where the number of cell types may be unknown. Then it is useful to assess the validity of the model as well. The linear mixing model can be extended to incorporate more than just two cell types using a straightforward extension:

$$y_i^k = \sum_j \alpha_k^j x_i^j, \tag{2.10}$$

where x_i^j denotes the expression value of the ith gene in the jth cell type, and $0 \leq \alpha_k^j \leq 1$ denotes the fraction of the jth cell type in the kth mixture. Naturally, the mixing percentages must also satisfy $\sum_j \alpha_k^j = 1$ for all k. Since the standard regression-based significance tests apply only to Gaussian noise we recommend using a general purpose cross-validation for model selection (see, e.g., Stone, 1974; Hastie *et al.*, 2001). Here we consider the leave-one-out cross-validation (LOOCV), i.e., each heterogeneous sample is left out from the training data at a time, the regression coefficients x_i^j are estimated based on the remaining four samples, and the model is then tested on the sample which was left out from the training data set. The relatively small sample size ($K = 5$) does not allow the estimation of the mixing fractions α_k^j within the cross-validation loop. Hence fixed (optimized, see Equation (2.7)) mixing fractions are used.

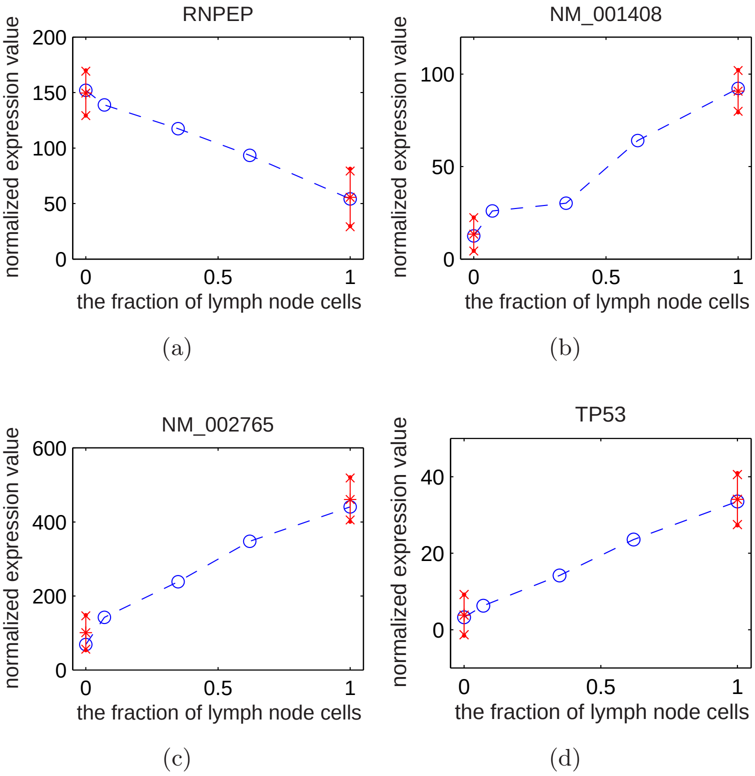
2.2.5 Examples and Discussion

We briefly illustrate the operation of the above described *in silico* microdissection methods on a carefully controlled heterogeneous microarray data set from Publication-VII. The results shown in Figure 2.3 are obtained by applying the above methods for inversion, optimization of mixing fractions, and confidence interval computation to some example genes. As the examples indicate, the expression values of the pure cell types can be estimated from the heterogeneous mixtures. A more comprehensive performance assessment of the methods is presented in Publication-VII, including also model selection using LOOCV.

Despite constant quality improvements, microarray data are quite noisy and impulses are not that uncommon. As was pointed out in Publication-VII, the effects of non-idealities, such as impulses, can be reduced by robust analysis. Although the general results remained largely unchanged after applying robust methods, improved results for some individual genes whose expression values contained an impulse were obtained. The well-known least squares method finds the optimum solution by minimizing

$$\arg\min_{x_i^c, x_i^l} \sum_{i=1}^{n} \sum_{k=1}^{K} \left(y_i^k - \alpha_k x_i^c - (1 - \alpha_k) x_i^l \right)^2 . \tag{2.11}$$

A number of robust estimation methods have been proposed (see, e.g., Hampel *et al.*, 1985; Rousseeuw and Leroy, 1987). In general, there might



(a) (b)

(c) (d)

Figure 2.3: Examples of the inversion of the sample heterogeneity and the corresponding 90% confidence intervals for some example genes. The x-axis (resp. y-axis) corresponds to the fraction of lymph node cells (resp. the normalized expression value). Shown are the measured expression values (blue circles), the estimated expression values of the pure cell types (red stars), confidence intervals based on Gaussian approximation (red points), and bootstrap-based confidence intervals (red x-marks).

be impulses in both regressors (position, α_ks) and outputs (error residuals, ϵ_i^ks). Due to physical constraints, however, there cannot be impulses in regressors in this application since the mixing fractions are known to be between zero and one. Outliers in error residuals can be taken into account by several standard robust regression methods. Let us use, e.g., the standard Huber's M-estimator whose influence function of the residuals is bounded (see, e.g., Hampel *et al.*, 1985). Briefly, instead of minimizing

Equation (2.11), a modified objective function is optimized

$$\arg\min_{x_i^c, x_i^l} \sum_{i=1}^{n} \sum_{k=1}^{K} \rho_c \left(\left(y_i^k - \alpha_k x_i^c - (1-\alpha_k) x_i^l \right) / \sigma_i \right), \tag{2.12}$$

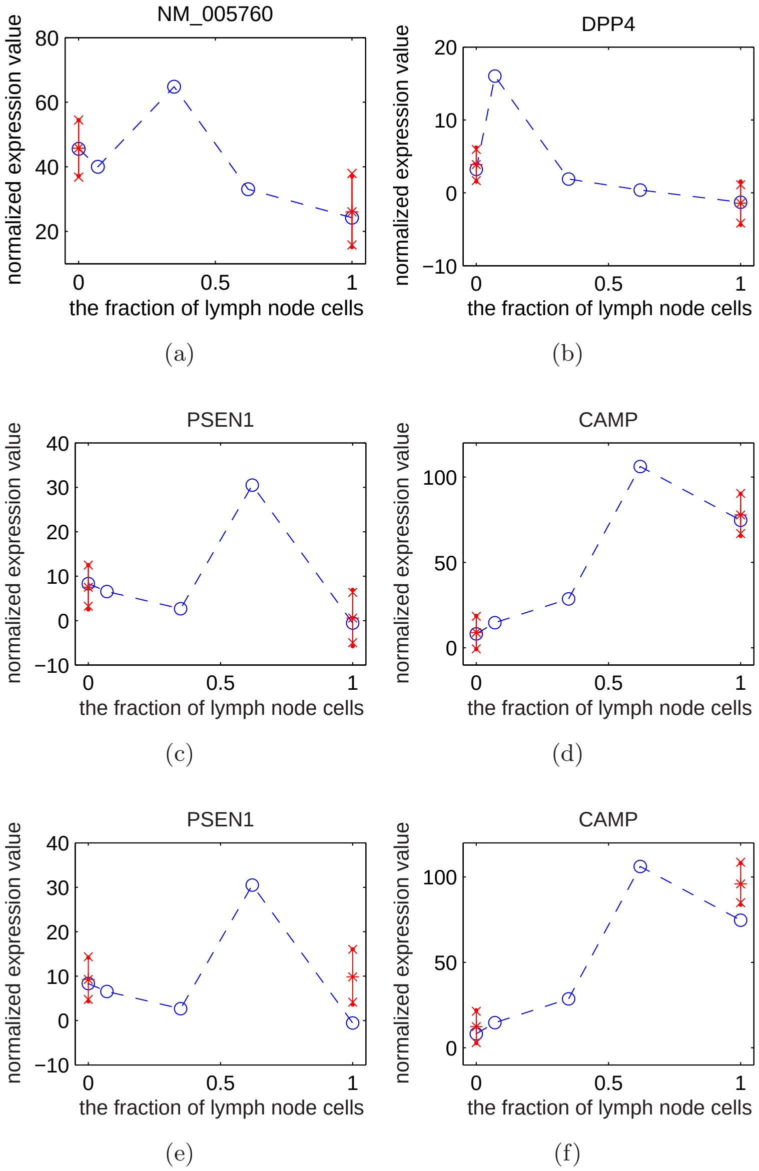
where σ_is are scaling factors and the quadratic function is replaced by the Huber estimator $\rho_c(\cdot)$

$$\rho_c(r) = \begin{cases} r^2/2, & \text{if } |x| \leq c \\ c(|r| - \frac{c}{2}), & \text{if } |x| > c \end{cases} . \tag{2.13}$$

Adjustable parameters are set to $c = 1$ and σ_is are chosen such that the resulting estimator is approximately 95% as efficient as the least squares estimator when applied to a normally distributed data with no outliers.[4] The robust objective function shown in Equation (2.12) is minimized using the iteratively reweighted least squares algorithm. Results for some genes contaminated with impulsive noise that serve as examples are shown in Figures 2.4 (a)–(d). As can be seen from the estimated expression values and the corresponding confidence intervals, robustness is clearly increased. For comparison purposes, Figures 2.4 (e)–(f) show the standard non-robust inversion results for the same genes as shown in Figures 2.4 (c)–(d).

As was discussed above, similar computational methods have been introduced in (Venet *et al.*, 2001; Lu *et al.*, 2003; Stuart *et al.*, 2004). In particular, the least squares inversion method we proposed resembles other methods introduced previously. Venet *et al.* also considered a similar method for optimizing the mixing percentages, but with slightly different constraints. Their analysis also focused on "deterministic" signals and they did not demonstrate performance of their methods on real heterogeneous measurements. Other aspects of our computational inversion methods, namely, the particular type of confidence interval computation, model selection and robust analysis are novel in this context.

[4] In particular, $\sigma_i = 1.345 \cdot \hat{s}\sqrt{1-h_i}$ is used for each i, where $\hat{s} = 1.4826 \cdot \text{mad}\{r_i\}$ is the scaled median absolute deviation of the residuals from their median and $h_i = (A(A^TA)^{-1}A^T)_{ii}$, i.e, the ith diagonal element of the "hat" matrix. For further details, see (Huber, 1981) and implementation details of `robustfit` function in (The MathWorks, Inc., 2005).



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2.4: Examples of the robust inversion (a)–(d) of the sample heterogeneity and the corresponding 90% confidence intervals for some example genes. Subgraphs (e) and (f) show the standard (non-robust) inversion results for the same genes as shown in subgraphs (c) and (d), respectively. See Figure 2.3 for explanation of the symbols.

2.3 Cell Population Asynchrony

Although most microarray experiments have been conducted for the purposes of static gene expression profiling, there is growing interest in monitoring the expression values over time as well. Time series experiments can provide temporal information about the development and dynamical operation of time-varying processes, such as the fundamental cell cycle. From a computational point of view, time series experiments provide a way of obtaining necessary dynamical data for studying regulatory effects in biological systems.

A single cell does not contain a sufficient amount of extractable mRNA to be measured using microarrays. Instead, the measurement procedure requires a sample that contains a very large number of cells. Since time series experiments are typically designed for studying a time-varying biological process, all the cells in the sample should operate exactly in the same phase of the process to be studied. Consequently, the cell population is usually forced to a synchrony prior to taking the measurements using an external synchronization method. For example, different synchronization methods have been used to synchronize the cell population relative to the cell cycle (Spellman *et al.*, 1998; Cho *et al.*, 1998; Whitfield *et al.*, 2002; Rustici *et al.*, 2004).

However, no matter what synchronization method has been used initially, the cell population gradually loses its synchrony. For example, in the case of cell cycle study, the cell population is distributed continuously into different cell cycle phases. It is useful to consider the loss of synchrony in terms of the distribution of the cell population over time. Perfect synchrony corresponds to the Dirac delta function whereas less synchronized cell populations correspond to wider distributions. Since the measurements are taken from the whole cell population, this results in time-varying (low-pass) filtering of the underlying gene expression time series. In Publication-I, we developed computational methods for inverting this smoothing effect from the gene expression time series. In addition, we also proposed methods for estimating the cell population distributions.

2.3.1 Modeling Cell Population Asynchrony

Although the measurements are taken from a finite cell population, it is convenient to describe the general model using continuous variables. Let

$x(t)$ denote the continuous expression value of a gene and p_t denote the continuous distribution of the cell population at time t. Assuming that each cell has, on average, an equal contribution to the resulting measurement $Y(t)$, the effect of the cell population asynchrony can be represented as

$$Y(t) = \int_{\tau=-\infty}^{\infty} p_t(\tau)x(t+\tau)d\tau + \epsilon(t), \tag{2.14}$$

where $\epsilon(t)$ is a continuous, generic noise term. Distributions are centered around the origin so that $p_t(-\tau)$ (resp. $p_t(\tau)$) denotes the fraction of cells having a negative (resp. positive) shift of size τ at time t. Note that the integral in Equation (2.14) corresponds to the standard continuous inner product.

So, if the cells were in perfect synchrony, p_t would correspond to the Dirac delta function and Equation (2.14) would reduce to $Y(t) = x(t) + \epsilon(t)$. In reality, however, the cell population gradually loses its synchrony, which results in wider distributions. That is, whenever p_t is not the Dirac delta function, measurements are "smoothed" by the distribution of the cell population p_t as shown in Equation (2.14).

Let us assume that gene expression time series data consists of m measurement time points t_i, $i = 1, \ldots, m$. In the following, we use the shorthand $Y(i)$ (resp. p_i) to denote $Y(t_i)$ (resp. p_{t_i}). Because only discrete measurements are available, we find it convenient to form a discrete approximation of the integral shown in Equation (2.14). Assume for now that we know the cell population distribution p_i at different time instants $i = 1, \ldots, m$ and let h_i denote their discrete approximations. Then, Equation (2.14) can be approximated as

$$Y(i) \approx \sum_j h_i(j)x(i+j) + \epsilon(i), \tag{2.15}$$

where the sum is computed over those j that satisfy $h_i(j) \neq 0$. A natural way of computing the coefficients h_i is as follows. The jth element of h_i is found by integrating p_i over an interval $I(j)$

$$h_i(j) = \int_{\tau \in I(j)} p_i(\tau)d\tau, \tag{2.16}$$

where a proper interval is

$$I(j) = \left[t_j - \frac{(t_j - t_{j-1})}{2},\ t_j + \frac{(t_{j+1} - t_j)}{2} \right]. \tag{2.17}$$

Basically, any discretization method could be used for the same purpose. The above procedure guarantees, however, that $\sum_j h_i(j) = 1$ for all i, assuming that each p_i is really a distribution.

Assuming that only expression measurements $Y(i)$ are available, then the smoothing cannot be inverted accurately since there are far too many adjustable parameters, even though there is redundancy in that the time-varying filter kernels p_i $(i = 1, \ldots, m)$ are the same for all the genes. Fortunately, the cell population distributions p_i can be estimated separately from additional measurements collected during the time series experiment.

If, for some reason, the cell population distributions cannot be estimated, then one can still consider so called spreading of the cell population distribution (see Publication-I for more details). The idea is based on the assumption that there is a convolution kernel h that maps the cell population distribution from any time t to the corresponding time $t + L$ at the next cycle, where L is the period length. That is, $p_{t+L} = h * p_t$, where $*$ stands for the convolution. Knowing a set of periodically behaving genes, the mean-squared optimal coefficients for the filter h can be obtained, e.g., by using methods of adaptive filtering (Haykin, 1996). A set of periodically behaving genes, in turn, can be obtained from prior biological knowledge, using the standard discrete time Fourier transform, or with the help of more advanced periodicity detection methods (see Chapter 3). Then, instead of inverting the whole smoothing model (Equation (2.14)), it is possible to invert the part of the smoothing effect that corresponds to the filter h.

2.3.2 Estimation of Cell Population Distribution

Accurate estimation of the cell population distribution is central to the inversion of the time-varying smoothing effect. That estimation step can utilize several different additional measurements gathered during the time course experiment. In Publication-I, we proposed several estimation methods for that purpose. Probably the most natural method is what we call the direct conversion of the distribution of a cell cycle regulated parameter.

Flow cytometry is an experimental technique for measuring certain physical and chemical characteristics of a cell population, such as the distribution of the amount of DNA. Since the amount of DNA grows from one unit up to two units during the cell cycle (see, e.g., Lodish *et al.*, 1999), the amount of DNA in a single cell defines its place in the cell cycle, assuming we know exactly the growth of the amount of DNA during the cell cycle.

Let us assume the cell cycle to start at 0 and end at L. Let $f_i(\tau)$ denote the measured distribution of the amount of DNA at time i, where $\tau \in [0, L]$ denotes the phase shift at the cell cycle. The amount of DNA in an ideal cell at phase τ of the cell cycle is denoted by $g(\tau)$. For simplicity, the function g is assumed to be strictly increasing, i.e., $(0 \leq \tau < \tau' \leq L) \Rightarrow (g(\tau) < g(\tau'))$. The assumption of a strictly monotone g is made mostly for mathematical convenience but that has some biological justification as well.[5] Under these assumptions the measured distributions can be converted into the cell population distributions simply by means of a combined mapping as

$$p_i(\tau) = f_i(g(\tau)) \tag{2.18}$$

for all $\tau \in [0, L]$. Depending on the amount and type of noise present in f_i, Equation (2.18) produces a noisy estimate of the underlying distribution. Hence some post-processing can be applied to the obtained distribution p_i.

The growth rate of a periodic parameter, such as the amount of the DNA, may be unknown. The best way of obtaining g is, of course, a measured distribution from a completely unsynchronized cell population (Niemistö *et al.*, 2004). Since the measured distributions are inherently discrete, Niemistö *et al.* also give a detailed discrete implementation of a method similar to that in Equation (2.18). The above ideas have been developed further. For example, advanced image processing algorithms have been proposed for the estimation of the bud size distributions from yeast microscope images (Niemistö *et al.*, 2003).

[5] Since DNA replication proceeds (bi)directionally (Lodish *et al.*, 1999) and most of the cells seem to obey the once-and-only-once principle of DNA replication (Boye *et al.*, 2000), it is natural to assume that the amount of DNA during the cell cycle, $g(\tau)$, can be modelled as a strictly increasing function. It is worth mentioning, however, that the DNA replication occurs during the synthesis (S) phase of the cell cycle that covers only a part of the whole cell cycle. In other words, $g(\tau)$ grows from one unit up to two units within a relatively short time interval. Consequently, under the assumption of strictly increasing $g(\tau)$, the amount of DNA remains approximately constant during the other cell cycle phases (G1, G2, and M) and that complicates the estimation of the cell population distribution.

In Publication-I, we also introduced other methods for estimating the distribution of the cell population, such as methods based on a rapidly changing parameter, fraction of cells having a bud of certain size (budding index), and blind channel estimation techniques. Other related estimation methods have been recently introduced in (Bar-Joseph *et al.*, 2004). Instead of using FACS or budding index data directly in a non-parametric fashion, Bar-Joseph *et al.* developed a method where parametric (Gaussian) distributions are fit to the data. Although constraining the cell population distributions to a certain parametric family can exclude the true underlying distributions, that can also provide a way of obtaining smooth and more accurate estimates if the parametric constraint is properly chosen.

2.3.3 Inversion of Cell Population Asynchrony

Assuming the cell population distributions, or their estimates, are known, then the final step of the analysis consists of the inversion of the time-varying smoothing effect. Two methods for that purpose were introduced in Publication-I, namely, a time-varying inverse filtering, and a regression-based method.

The inverse filtering becomes available by observing that Equation (2.15) corresponds to a discrete convolution with a time-reversed (and time-varying) filter kernel. Hence, a time-varying version of the inverse filter Wiener filter) can be applied (see Publication-I for more details). Even though the inverse filter can be shown to be optimal in the mean-square error sense (see, e.g., Dougherty, 1999), its application can be difficult because gene expression time series are short and the spectral density function of the true signal and the noise are unknown.

A better inversion method can be developed by expressing the smoothing as a type of regression problem. In particular, the possible periodicity in gene expression can be taken into account more easily. Let us first formalize the smoothing shown in Equation (2.15) in matrix form by expanding the inner products for all $1 \leq i \leq m$

$$\mathbf{Y} = \mathbf{H}\mathbf{x} + \epsilon, \tag{2.19}$$

where $\mathbf{Y} = (\ Y(1)\ \ \cdots\ \ Y(m)\ )^T$,

$$\mathbf{H} = \begin{pmatrix} \cdots & h_1(-1) & h_1(0) & h_1(1) & \cdots & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & \cdots & h_m(-1) & h_m(0) & h_m(1) & \cdots \end{pmatrix}, \tag{2.20}$$

$\mathbf{x} = (\ \cdots\ \ x(0)\ \ x(1)\ \ \cdots\ \ x(m)\ \ x(m+1)\ \ \cdots\ )^T$ and $\epsilon = (\ \epsilon(1)\ \ \cdots\ \ \epsilon(m)\ )^T$.

Let us first concentrate on the cell cycle regulated genes and assume that measurements from consecutive cell cycles are taken from the same phases of the cell cycle.[6] For the periodic genes we have that $x(i) = x(i+L)$ for all $i = 1, \ldots, m-L$. Combining that with Equations (2.19) and (2.20) one gets

$$\begin{pmatrix} Y(1) \\ \vdots \\ Y(m) \end{pmatrix} = \begin{pmatrix} h_1(0) & h_1(1) & \cdots & h_1(-1) \\ h_2(-1) & h_2(0) & \cdots & h_2(-2) \\ \vdots & \vdots & \ddots & \vdots \\ h_L(1) & h_L(2) & \cdots & h_L(0) \\ \vdots & \vdots & \ddots & \vdots \\ h_m(k) & h_m(k+1) & \cdots & h_m(k-1) \end{pmatrix} \begin{pmatrix} x(1) \\ \vdots \\ x(L) \end{pmatrix} + \epsilon \tag{2.21}$$

where k is selected properly. Equation (2.21) essentially sets up the standard regression problem. Assuming one has measurements from more than one cell cycle, $m > L$, then the rank of the kernel matrix $\mathbf{H}$ in Equation (2.21) is most probably full. Minimization of the least squares error criterion results again in the well-known estimate of $\mathbf{x}$

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}, \tag{2.22}$$

where $\mathbf{y}$ is the observed value of $\mathbf{Y}$.

If the measurements from the consecutive cell cycles cannot be controlled to be taken from the same positions, then the solution becomes a bit more complicated and less accurate. However, the problem can still be formalized

[6] Note that this requirement can be controlled in practise.

much in the same way as shown above, with the exception that the kernel matrix $\mathbf{H}$ belongs now to $[0,1]^{m\times M}$ instead of $[0,1]^{m\times L}$, where $M > L$ (see Publication-I for more details).

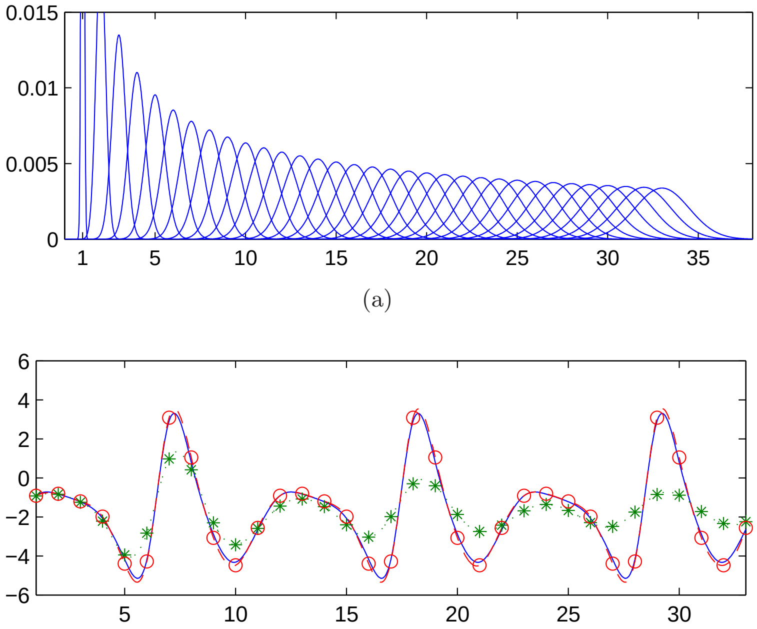
The above methods cannot be applied to genes that are not periodic. Equation (2.19) shows, however, that the filtered, but noise-free, expression profile $\mathbf{Hx}$ must lie in the space spanned by the columns of $\mathbf{H}$, i.e., $\mathrm{R}(\mathbf{H})$. The standard Gauss-Markov Theorem (see, e.g., Johnson and Wichern, 1998) can be applied to Equation (2.19) in order to get the best linear unbiased estimate of $\mathbf{Hx}$. This is achieved by projecting the measured vector $\mathbf{y}$ orthogonally into the space $\mathrm{R}(\mathbf{H})$. In this case, estimates of $\mathbf{x}$ constitute a space $S_{\mathbf{x}} = \{\mathbf{x}\,|\,\mathbf{x} = \hat{\mathbf{x}} + \mathbf{x}_0,\ \mathbf{x}_0 \in \mathcal{N}(\mathbf{H})\} \subset \mathbf{R}^m$, where $\hat{\mathbf{x}}$ is an optimal solution and $\mathcal{N}(\mathbf{H})$ is the null space of $\mathbf{H}$. Some extra constraints, e.g., on the smoothness, can be used to reduce the size of $S_{\mathbf{x}}$.

2.3.4 Examples and Discussion

The performance of the proposed method for correcting the cell population asynchrony is illustrated below. Let us concentrate on periodic time series in this simulation. Truncated Gaussian distributions with increasing variance are used for the cell population distributions, see Figure 2.5 (a). The measurements are generated according to Equation (2.14) both without and with the additive (white) noise. Furthermore, the measurements from different periods are assumed to be taken such that the inversion method shown in Equation (2.22) applies. The discrete kernels h_i are computed from the true distributions using Equations (2.16) and (2.17).

The optimal solution for the noise-free case is shown in Figure 2.5 (b). A cubic spline interpolation is used to obtain a continuous version of the discrete signals. The observed measurements clearly illustrate the smoothing effect that is caused by the cell population asynchrony. The regression-based method reconstructs the underlying signal almost exactly. Note that even though the measurements are noise-free the coarse-scale approximation of the true measurement process causes a loss of accuracy.

The same results are shown for the case of additive noise with signal-to-noise ratio (SNR) about 20 and 10 in Figures 2.6 (a) and 2.7 (a), respectively. The general behavior of the estimation method is assessed using Monte Carlo simulation. Figures 2.6 (b) and 2.7 (b) shows the Box-plot visualization of the estimates obtained from 100 independent runs, together



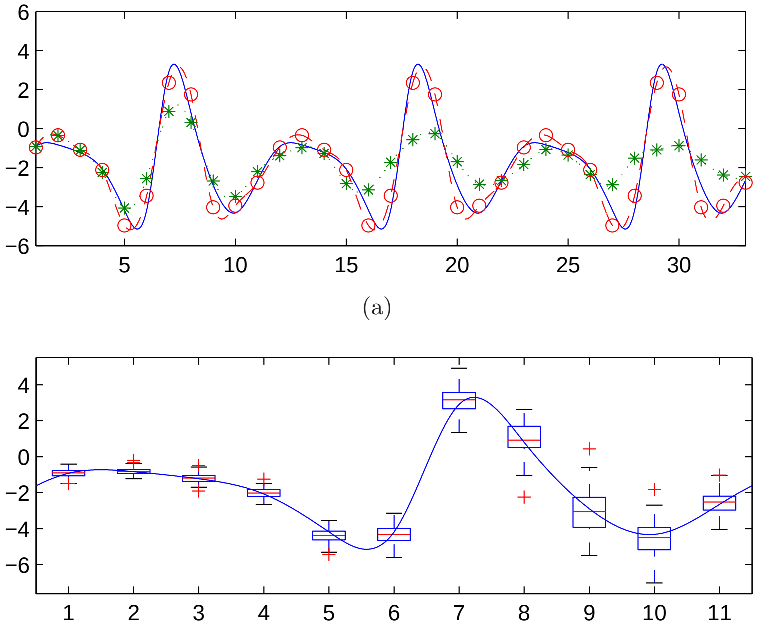
(a)

(b)

Figure 2.5: (a) The cell population distributions used in the simulation. Horizontal axis denotes time. Note that y-axis has been cut for visualization purposes. (b) Representative results for the inversion of the cell population asynchrony: the noise-free case. Symbols: continuous true gene expression time series (solid blue curve), the observed measurements (dotted green line with stars), and the corrected measurements (dashed red line with circles).

with the underlying continuous time series (solid line). The Box-plot is shown only for the first period of the signal since the estimates for the remaining cycles are the same. Overall, variability of the estimates increases relative to the variance of noise ϵ. For SNR about 10 (and smaller), the estimates are already likely to contain some artifacts, such the kind of overshoot seen in Figure 2.7 (a). It is worth noting that performance of the estimation method also depends on characteristics of the underlying signal (see the blue solid curve in Figures 2.5 (b), 2.6 (a) and 2.7 (a)) as well as on cell population distributions (see Figure 2.5 (a)).

The computational methods presented in Publication-I were, to our knowledge, the first ones that have been proposed for the inversion of the cell population asynchrony. Possible future research directions include, e.g., in-

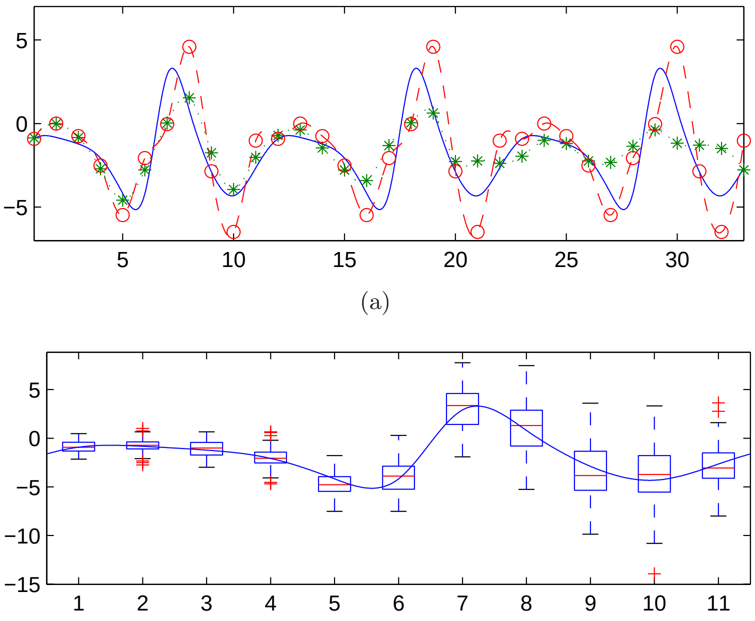


(a)

(b)

Figure 2.6: Representative results for the inversion of the cell population asynchrony. (a) additive white noise with signal-to-noise SNR $\approx$ 20, (b) the Box-plot visualization of the obtained estimates. See Figure 2.5 for encoding of the symbols.

corporating more specific noise models and developing methods that do not rely only on discrete-time modeling. For example, an approach based on a properly developed frequency domain analysis could be useful. Recently, computational inversion methods for the cell population asynchrony were further extended in (Bar-Joseph *et al.*, 2004). Indeed, Bar-Joseph *et al.* represent gene expression time series using spline interpolation and invert the smoothing effect without resorting to discrete approximations. Such an approach is more general in that fewer assumptions for measurement time instants are needed. In particular, measurements from consecutive cell cycles do not need to be taken from the same phases of the cell cycle. Bar-Joseph *et al.* also combine their analysis with a clustering method, hence providing a way of regularizing the actual inversion problem. Clustering-based regularization can be advantageous especially in the case of low signal-to-noise ratios, but the possible advantage also depends on the performance of a



(a)

(b)

Figure 2.7: Representative results for the inversion of the cell population asynchrony. (a) additive white noise with signal-to-noise SNR ≈ 10, (b) the Box-plot visualization of the obtained estimates. See Figure 2.5 for encoding of the symbols.

particular clustering method that is used. Since real biological time series are prone to contain different kinds of non-idealities, such as outliers, it is also worth noting that robust estimation methods can potentially be useful in this case as well.

Chapter 3

Robust Time Series Analysis

Although most of the first microarray experiments focused solely on static experiments, there is an increasing interest in measuring changes in expression values over time. Recently, analysis of periodic expression profiles has attracted much attention. Numerous studies are motivated by the fact that periodic phenomena are widespread in biology, including, among others, membrane potential oscillations, smooth muscle contraction, cardiac rhythms, calcium oscillations, glycolytic oscillations, cAMP oscillations, oscillations in neuronal signals, cell cycle, circadian rhythms, ovarian cycle, and others (see, e.g., Tyson, 2002). Consequently, there are numerous biological applications where periodicities must be detected from experimental biological data.

Detecting periodicity in gene expression is of particular importance because it can indicate cell-cycle regulation (Breeden, 2003), for example, as well as the effect of circadian rhythms (Correa *et al.*, 2003). The significance of the detection of cell-cycle regulated processes is further emphasized by the linkage between cell-cycle and cancer (see, e.g., Sherr, 1996; Whitfield *et al.*, 2002). To this end, microarrays have been used to study the circadian gene expression in *Neurospora crassa* (Correa *et al.*, 2003) as well as cell-cycle regulated genes in budding yeast (Spellman *et al.*, 1998), in fission yeast (Rustici *et al.*, 2004), and in human cells (Whitfield *et al.*, 2002).

A number of methods for detecting periodic transcripts have recently been proposed (Zhao *et al.*, 2001; Johansson *et al.*, 2003; Liu *et al.*, 2004; Lu *et al.*, 2004; Luan and Li, 2004; Wichert *et al.*, 2004). The proposed methods vary from applications of partial least squares regression to complex, combined procedures of estimating the cell population distributions and statistical testing. A major difference between the method proposed by Wichert *et al.* and other methods is that Wichert's method is capable of detecting unknown frequencies whereas other methods are designed for detecting fixed frequencies. From a computational point of view, the problem of finding unknown frequencies is even more demanding since no prior knowledge of the frequency to be detected is available. In many biological applications it is more important to search for periodicities having an unknown frequency. However, in some applications, such as large-scale cell cycle studies, the period length is usually known and thus provides additional information for testing.

In many applications, including those arising from bioinformatics, the exact noise characteristics are usually unknown and can be remarkably non-Gaussian. Furthermore, the observed gene expression time series can exhibit other non-idealities, such as outliers, missing values, short length and distortion from the original wave form. Therefore, the computational methods should preferably be robust against such anomalies in the data. The recently introduced methods for detecting periodic transcripts are not particularly robust, e.g., in the case of outlier contaminated data. In what follows, we introduce general, robust methods for spectrum estimation (Publication-IV) and periodicity detection of both fixed and unknown frequencies (Publication-IX). For clarity of terminology, notation and further analysis, we first give some background about spectral analysis of stochastic processes. For more details, see, e.g., (Priestley, 1981; Brockwell and Davis, 1991).

3.1 Spectral Theory of Stochastic Signals

In this and the following sections we are mainly interested in discrete time series. Since real world time series exhibit some inherent randomness, we first give the definition of a discrete stochastic process. A discrete stochastic process is a set of random variables $\{Y_t, t \in T\}$ defined in a probability space $(\Omega, \mathcal{F}, P)$, where T is a discrete set, Ω is a sample space, $\mathcal{F}$ is a sigma-algebra of subsets of Ω, and P is a probability measure on $\mathcal{F}$. Given a fixed element of Ω, $\{y_t, t \in T\}$ is a realization of the stochastic process $\{Y_t, t \in T\}$. Intuitively speaking, time series $\{y_t, t \in T\}$ is a realization of the family of random variables $\{Y_t, t \in T\}$. In the following we will assume the set of time indices to be $T = \mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$ and later in the case of real data $T = \{1, \ldots, N\}$.

Much of the following analysis is built on the following definitions. The autocovariance function of a stochastic process $\{Y_t, t \in T\}$ is defined as $\gamma(k,l) = \mathbb{E}[(Y_k - \mu_{Y_k})(Y_l - \mu_{Y_l})]$, where $\mathbb{E}[\cdot]$ denotes the expectation operator and $\mu_{Y_k} = \mathbb{E}[Y_k]$. The autocorrelation function of a stochastic process $\{Y_t, t \in T\}$ is defined as $r(k,l) = \mathbb{E}[Y_k Y_l]$. A useful class of processes, on which most of the classical theory of spectral analysis has focused, is the one whose statistical properties do not change over time. More precisely, a stochastic process $\{Y_t, t \in T\}$ is stationary[1] if *(i)* $\mathbb{E}[Y_t] = \mu$ for all $t \in T$, *(ii)* $\mathbb{E}[Y_t^2] < \infty$ for all $t \in T$, and *(iii)* $\gamma(k,l) = \gamma(k+t, l+t)$ for all $k, l, t \in T$. Note that for stationary processes the autocovariance function as well as the autocorrelation function depend on time only through the difference $k - l$. Therefore, it is convenient to redefine the autocovariance function as $\gamma(k) = \mathbb{E}[(Y_t - \mu)(Y_{t+k} - \mu)]$ and similarly for the autocorrelation function $r(k) = \mathbb{E}[Y_t Y_{t+k}]$.

In the following we will be assuming the autocovariance function $\gamma(k)$ to be absolutely summable, i.e., $\sum_k |\gamma(k)| < \infty$. Let us then state a fundamental and important result (see, e.g., Priestley, 1981; Brockwell and Davis, 1991). Let $\gamma(k)$ and $F(\omega)$ be the autocovariance function and the spectral distribution function, respectively, of a zero-mean stationary process $\{Y_t, t \in T\}$, then

$$\gamma(k) = \int_{(-\pi,\pi]} e^{i\omega k} dF(\omega), \qquad \text{for all } k \in T. \tag{3.1}$$

If the process $\{Y_t, t \in T\}$ is such that $F(\omega)$ is differentiable everywhere, then Equation (3.1) can be rewritten as

$$\gamma(k) = \int_{(-\pi,\pi]} e^{i\omega k} f(\omega) d\omega, \qquad \text{for all } k \in T, \tag{3.2}$$

where $f(\omega)$ is the spectral density function of the process $\{Y_t, t \in T\}$.

[1] This definition of stationarity is also known as weakly stationarity or second-order stationarity.

Moreover, Equation (3.2) can be inverted, i.e.,

$$f(\omega) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \gamma(k) e^{-i\omega k}, \qquad \text{for all } \omega \in [-\pi, \pi]. \tag{3.3}$$

Note that for zero-mean stationary processes the autocovariance function equals $\gamma(k) = \mathbb{E}[(Y_t - \mu)(Y_{t+k} - \mu)] = \mathbb{E}[Y_t Y_{t+k}] = r(k)$, i.e., the autocorrelation function. Hence, under the assumption of zero-mean stationarity, the spectral density can equally well be represented in terms of the autocorrelation function as

$$f(\omega) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} r(k) e^{-i\omega k}, \qquad \text{for all } \omega \in [-\pi, \pi]. \tag{3.4}$$

The above equation provides a useful formula for finding the spectral density function from $r(k)$. Consequently, many spectral estimators of the form of Equation (3.4) have been proposed in the literature.

In the following we are interested in both spectrum estimation and periodicity detection. Loosely speaking, those two goals correspond to the estimation of $f(\omega)$ from observed data and finding statistically significant peaks in the estimated spectrum, respectively. As noted above, characteristics of the data measured from biological systems are typically unknown. Hence our aim is to achieve the two goals using robust methods. We start with the spectrum estimation and then move towards the periodicity detection.

3.2 Spectrum Estimation

A number of different methods have been proposed for spectrum estimation (see, e.g., Priestley, 1981; Brockwell and Davis, 1991; Stoica and Moses, 1997). One categorization divides different approaches into parametric and nonparametric methods, depending on whether a parametric model for the signal is assumed. Although we define a generic model for periodic signals later in Section 3.3, we are primarily interested in nonparametric spectrum estimation. The choice of nonparametric approach has at least two motivations in the context of bioinformatics. First, as already stated above, the characteristics of the data are typically unknown, thus precluding postulation of specific noise models. Second, the same argument applies to the actual periodic signal models as well, since they do not necessarily follow any predetermined wave form. In the following, we use a shorthand $\{Y_t\}$ (resp. $\{y_t\}$) to denote $\{Y_t, t \in T\}$ (resp. $\{y_t, t \in T\}$) if the discrete index set is known.

3.2.1 Nonparametric Spectrum Estimation

A fundamental, nonparametric tool for spectrum estimation is the periodogram defined as

$$I(\omega) = \frac{1}{N} \left| \sum_{t=1}^{N} y_t e^{-i\omega t} \right|^2, \quad \omega \in [0, \pi]. \tag{3.5}$$

Although the periodogram is a basic spectrum estimation tool widely applied in different applications, under rather general assumptions the periodogram is not a consistent estimator of the spectral density. On the other hand, consistent estimators can be constructed from the periodogram, e.g., by applying linear smoothing filters to the periodogram. More importantly, however, the exact distributional characteristics of the periodogram are known and useful. Although these exact characterizations mostly apply to Gaussian sequences, an assumption widely invoked in spectral estimation, they form the basis of traditional statistical inference methods for the spectrum (see Section 3.3).

Let us turn back to the spectrum estimation problem. It is well-known that the periodogram $I(\omega)$ is equivalent to the correlogram spectral estimator (see, e.g., Brockwell and Davis, 1991)

$$S(\omega) = \sum_{k=-N+1}^{N-1} \hat{r}(k) e^{-i\omega k}, \tag{3.6}$$

where $\hat{r}(k)$ is the biased estimator of the autocorrelation function

$$\hat{r}(k) = \frac{1}{N} \sum_{t=1}^{N-m} y_t y_{t+k}. \tag{3.7}$$

Note that the required values for $\hat{r}(k)$ for $k < 0$ are obtained by invoking the inherent symmetry of the autocorrelation function, i.e., $r(-k) = r(k)$. The use of the correlogram (and thereby the periodogram) is directly motivated by the theoretical result shown in Equation (3.4). As our goal is to obtain robust time series analysis methods, a natural choice is to try to obtain robustness by replacing $\hat{r}(k)$ with a robust alternative.

3.2.2 Robust Spectrum Estimation

Relatively few methods have been introduced for robust spectrum estimation. Most of the proposed methods are based on complex procedures of robust precleaning of the data and weighting of the residuals using autoregressive models (Kleiner *et al.*, 1979; Martin and Thomson, 1982); see also (Spangl and Dutter, 2005). Robust versions of the Fourier transform have also been proposed (Tatum and Hurvich, 1993). Our robust method is built on the standard principles introduced above, but modified to obtain robustness. Our approach is particularly motivated by and designed for robust periodicity detection (to be discussed shortly in Section 3.3.2).

Before reviewing the robust method, it is important to note that, especially in the case of gene expression time series, the data is often contaminated with missing values. Let I_k be the set of time indices t for which both y_t and y_{t+k} are available and $K_k = |I_k|$. As long as $K_k > 0$, a missing data-adapted version of the unbiased estimate of the autocorrelation can be obtained as

$$\tilde{r}(k) = \frac{1}{K_k} \sum_{t \in I_k} y_t y_{t+k}. \tag{3.8}$$

Only versions adapted to missing data are considered in the following since they are equal to the standard estimators in case of complete data sets.

Next we review the proposed robust, rank-based autocorrelation estimator for the problem of spectrum estimation (Publication-IV; Publication-IX). This estimator is a moving-window extension of the Spearman rank correlation coefficient, quantifying the association between the sequences $\{Y_t\}$ and $\{Y_{t+k}\}$. The resulting quantity is actually an alternative estimator of the standard correlation coefficient $\rho(k, l)$ between these sequences (see, e.g., Priestley, 1981)

$$\rho(k, l) = \frac{\mathbb{E}[(Y_k - \mu_{Y_k})(Y_l - \mu_{Y_l})]}{\sqrt{\mathbb{E}[(Y_k - \mu_{Y_k})^2]}\sqrt{\mathbb{E}[(Y_l - \mu_{Y_l})^2]}}. \tag{3.9}$$

Recall that the sample correlation coefficient between two N length sequences $\{x_i\}$ and $\{y_i\}$ is defined as

$$\rho_{xy} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \overline{y})^2}}. \tag{3.10}$$

where $\overline{x}$ denotes the sample mean of $\{x_i\}$.

Under the assumption of stationarity, the correlation coefficient depends on time only through the difference $k-l$ and can hence be redefined as $\rho(k)$. Further, it immediately follows from Equation (3.9) that the correlation coefficient $\rho(k)$ is related to the autocorrelation function $r(k)$ by $r(k) = \mu_Y^2 + \sigma_Y^2\rho(k)$, where $\sigma_Y^2 = \mathbb{E}[(Y_t - \mu_Y)^2]$ is the variance of the sequence. Since it is important to remove the mean of the sequence prior to spectrum estimation to avoid low frequency artifacts and since σ_Y^2 is simply a scale factor, the problem of detecting periodic components in a data sequence may equally well be based on $\rho(k)$ as $r(k)$. Consequently, we consider spectral estimators of the form

$$\tilde{S}(\omega) = \sum_{k=-L}^{L} \tilde{\rho}(k)e^{-i\omega k}, \tag{3.11}$$

where $\tilde{\rho}(k)$ estimates the correlation coefficient between $\{Y_t\}$ and $\{Y_{t+k}\}$ and L is the maximum lag for which the correlation coefficient is computed. More specifically, we consider the correlation coefficient between the data ranks $R_y(t)$ and $R'_y(t)$, defined by

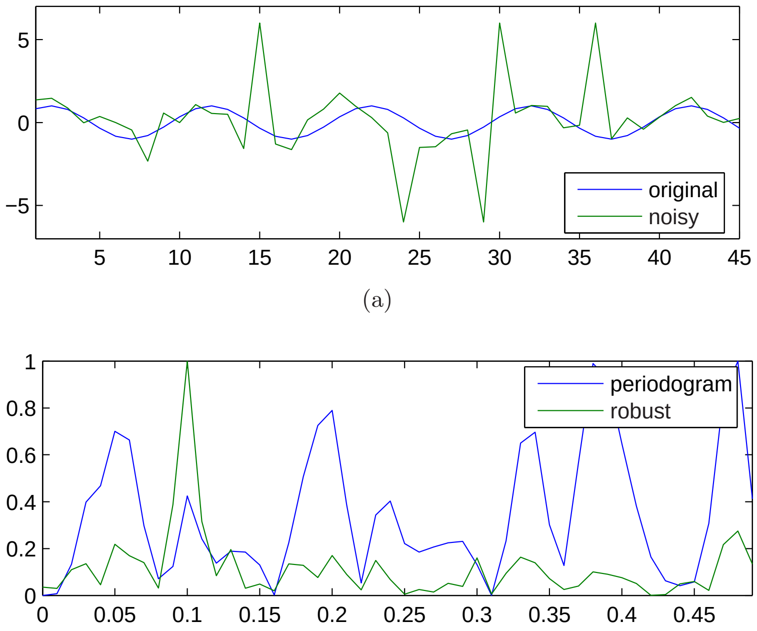
$$\tilde{\rho}(k) \quad = \quad \frac{1}{C}\cdot\frac{12}{K_k^2 - 1}\sum_{t\in I_k}\left(R_y(t) - \frac{K_k+1}{2}\right)\left(R'_y(t) - \frac{K_k+1}{2}\right) \tag{3.12}$$

where C is a normalisation factor, $R_y(t)$ denotes the rank of y_t in the set $S = \{y_t : t \in I_k\}$ and $R'_y(t)$ denotes the rank of y_{t+k} in the set $S' = \{y_{t+k} : t \in I_k\}$. By selecting either $C = K_k$ or $C = N$, Equation (3.12) yields the unbiased or the biased estimate of the correlation coefficient between the rank sequences, respectively. See Publication-IX for some properties of the proposed robust estimator.

Recall that the periodogram $I(\omega)$ is equivalent to the correlogram $S(\omega)$ when the correlogram is implemented with the biased estimator of the standard autocorrelation function (Equation (3.7)). For that reason, we use the biased estimator for the robust correlation coefficient, i.e., $C = N$, in the following. Moreover, the use of the biased estimate in spectrum estimation (Equations (3.6) and (3.7)) can be interpreted as triangular weighting of the autocorrelation function estimate. Windowing is usually applied to reduce the variance of the autocorrelation function estimate and to reduce the scalloping loss effect (Priestley, 1981; Stoica and Moses, 1997). Windowing is discussed in a bit more detail in Section 3.3.2. Contrary to the standard periodogram, the robust spectral estimator is not guaranteed to be non-negative. Hence the absolute value of $\tilde{S}(\omega)$ is used in the following.

Figures 3.1 and 3.2 show some example time series and the corresponding scaled spectral estimates obtained by the standard periodogram and the robust method. The noise-free reference signal consists of a single sinusoid at normalized frequency 0.1 (see Equation (3.13)) and the two noisy time series in Figures 3.1 and 3.2 are both contaminated with additive Gaussian noise, plus about 10% impulses and cubic distortion, respectively. In these examples the robust method clearly outperforms the standard periodogram. A more extensive Monte Carlo simulation of the performance of the robust spectrum estimator is conducted in Publication-IV. Although the raw periodogram is known to be unconsistent and not the best possible spectrum estimator, we use it for comparison purposes. Motivation for doing so is three-fold. First, both spectrum estimators, the periodogram and the proposed robust estimator, are extensively discussed and used as basic building blocks of periodicity detection methods in the next section. Secondly, the consistency property may be of little use in small sample settings typically encountered in high-throughput systems biology experiments. The last, but not the least important point is that the proposed robust spectrum estimator can be viewed as a similar elementary method as the periodogram whose performance can be improved. In particular, the simple tricks that provide consistency for the periodogram, such as linear filtering of the periodogram or windowing of the correlogram, can alike be applied to the robust spectrum estimator.

Overall, the proposed method is found to provide remarkably robust spectrum estimates. A main ingredient of the obtained robustness is the use of a rank-based approach. Moreover, because the proposed method is build on a moving-window extension of the Spearman rank correlation coefficient, it provides a straightforward robust analog for the traditional autocorrelation estimator. That further emphasizes the use of the robust



(a)

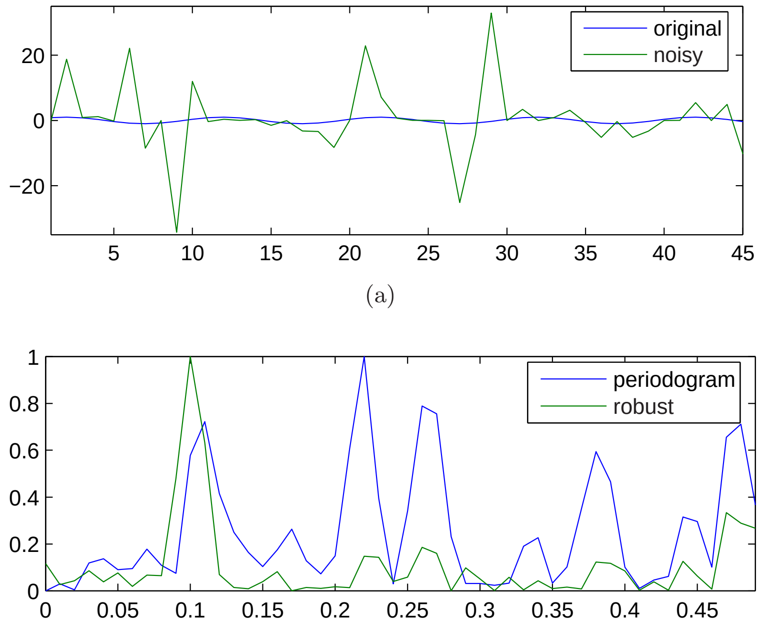
(b)

Figure 3.1: (a) A noise-free sinusoid (blue) and its noisy version contaminated with additive Gaussian noise and about 10% outliers (green). (b) Scaled spectrum estimates of the above noisy sinusoid obtained by the standard periodogram (blue) and the robust method (green).

estimator as a basic building block of spectral estimators via the well-known connection between the spectral density and the autocorrelation function (see Equation (3.4)). Because missing data points are commonly present in high-throughput biological measurements, the method is also adapted to handle such cases. The proposed method is also efficient from computational point of view because it does not require the use of computationally intensive estimation or optimization procedures. Interesting future research directions, especially from periodicity detection point of view, are discussed in Section 3.3.2.

3.3 Detection of Periodic Time Series

Let us now turn to our primary problem of detecting periodic time series. To our knowledge, no particularly robust periodicity detection method has



(a)

(b)

Figure 3.2: (a) A noise-free sinusoid (blue) and its noisy version contaminated with additive Gaussian noise and the cubic distortion (green). (b) Scaled spectrum estimates of the above noisy sinusoid obtained by the standard periodogram (blue) and the robust method (green).

been proposed in the literature so far. Since periodicity detection can be viewed as a decision problem based on the spectral content of the signal, it has a natural connection to spectrum estimation. In Publication-IX, we proposed a robust periodicity detection method which is build on the robust spectrum estimator described above. Also note that periodicity detection problems can be divided into two categories depending on whether unknown or known (fixed) frequencies are to be detected. We proposed a solution to both of the problems in Publication-IX.

3.3.1 Standard Periodicity Detection Methods

For the purposes of introduction and comparison, we first give a brief review of the well-known, exact tests for periodicity detection under the Gaussian noise assumption, namely, Fisher's test for detection of unknown frequencies and standard regression (i.e., ANOVA-based) methods for the detection of fixed frequencies. Also note that Fisher's test has recently been applied to the detection of periodic gene expression time series in Wichert *et al.* (2004).

In the following we consider a generic model for the periodic time series

$$Y_t = \beta \cos(\omega t + \phi) + \epsilon_t, \tag{3.13}$$

where $\beta \geq 0$, $\omega \in (0, \pi)$, $t = 1, \ldots, N$, $\phi \in (-\pi, \pi]$, and ϵ_t is an i.i.d. noise sequence.[2] To test for periodicity, define the null hypothesis as $H_0 : \beta = 0$, i.e., time series consists of the noise sequence alone. The corresponding alternative hypothesis is $H_1 : \beta > 0$.

Despite the inconsistency of the periodogram as a spectrum estimator, the periodogram is a useful tool for developing statistical inference methods for the spectrum since its statistical properties are known. Consequently, many of the traditional statistical tests for the detection of periodic time series can be expressed in terms of the periodogram. Moreover, improvements over the standard periodogram, developed originally for spectrum estimation purposes, can also be incorporated into periodicity detection (see Section 3.3.2 for further discussion).

Consider the periodogram $I(\omega)$ shown in Equation (3.5) evaluated at the harmonic frequencies

$$\omega_l = \frac{2\pi l}{N}, \qquad l = 0, 1, \ldots, a, \tag{3.14}$$

where $a = [(N-1)/2]$ and $[x]$ denotes the integer part of x. Fisher's test utilizes the so-called g-statistic

$$g = \frac{\max_{1 \leq l \leq a} I(\omega_l)}{\sum_{l=1}^{a} I(\omega_l)}. \tag{3.15}$$

Since the g-statistic divides the maximum periodogram ordinate by the sum of all periodogram ordinates large values of g indicate a strong periodic component and can lead to the rejection of the null hypothesis. More precisely, the exact null distribution of the g-statistic, under the Gaussian

[2] The above model assumes that the periodic time series consists of only a single frequency. For possible extensions to multiple frequencies, see discussion in Publication-IX and references therein.

noise assumption, is shown to be (Fisher, 1929)

$$\begin{aligned} P(g > x) = {} & a(1-x)^{a-1} - \frac{a(a-1)}{2}(1-2x)^{a-1} \\ & + \ldots + (-1)^b \frac{a!}{b!(a-b)!}(1-bx)^{a-1}, \end{aligned} \tag{3.16}$$

where b is the largest integer less than $1/x$ and x is the observed value of the g-statistic (see also, e.g, Brockwell and Davis, 1991; Wichert *et al.*, 2004). To summarize, Equations (3.5),[3] (3.14), (3.15) and (3.16) form Fisher's test.

If the harmonic frequency ω_l, $l = 1, \ldots, a$, to be detected is known *a priori*, then even a simpler approach suffices (see, e.g., Brockwell and Davis, 1991).[4] Note that Equation (3.13) can be rewritten as

$$Y_t = \beta' \cos(\omega_l t) + \beta'' \sin(\omega_l t) + \epsilon_t, \tag{3.17}$$

for all $t = 1, \ldots, N$. The null and the alternative hypotheses can now be defined as $H_0 : \beta' = \beta'' = 0$ and $H_1 : \beta' \neq 0$ or $\beta'' \neq 0$, respectively. Equation (3.17) can be further formulated using the standard linear regression model $\mathbf{Y} = \mathbf{X}\tilde{\beta} + \epsilon$, where $\mathbf{Y}$ (resp. ϵ) is the column vector of Y_ts (resp. ϵ_ts), $\tilde{\beta} = (\ \beta' \ \ \beta'' \ )^T$, and the tth row of $\mathbf{X}$ is $(\ \cos(\omega_l t) \ \ \sin(\omega_l t)\ )$. Assuming again Gaussian noise, then the standard method of analysis of variance (ANOVA) can be used to construct an F-distributed test statistic with 2 and $N-2$ degrees of freedom, i.e.,

$$\frac{\hat{\beta}(\mathbf{X}^T\mathbf{X})^{-1}\hat{\beta}}{2\hat{\sigma}^2} \sim F(2, n-2), \tag{3.18}$$

where $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ and $\hat{\sigma}^2 = ||\mathbf{Y} - \mathbf{X}\hat{\beta}||^2/(n-2)$. Although the above methods provide exact tests, because they are based on a Gaussian assumption and a type of least squares estimation, they are not robust and can fail if the original noise assumptions do not hold.

[3] Due to the equivalence between the periodogram and the correlogram, Equation (3.5) can be replaced with Equations (3.6) and (3.7).

[4] A modification for a nonharmonic frequency can also be constructed.

3.3.2 Robust Periodicity Detection Methods

The robust periodicity detection methods introduced in Publication-IX are motivated by Fisher's test and the robust spectrum estimator introduced in Publication-IV (see Section 3.2.2). Let us first focus on the detection of unknown frequencies. In the same way as in Fisher's test, periodicity detection is based on the g-statistic

$$g = \frac{\max_{1 \le l \le a} |\tilde{S}(\omega_l)|}{\sum_{l=1}^{a} |\tilde{S}(\omega_l)|}. \tag{3.19}$$

Instead of using the fixed harmonic frequencies shown in Equation (3.14), we can also adjust the number of frequencies at which the spectrum estimator is evaluated, i.e.,

$$\omega_l = \frac{2\pi l}{K},\ l = 0, 1, \ldots, [(K-1)/2], \tag{3.20}$$

The frequencies ω_l in Equation (3.19) are changed accordingly. In particular, the parameter a is set to $[(K-1)/2]$. Although the performance is not very sensitive to the selection of K, the value $K = 2N$ was found to provide generally good results.

The null distribution shown in Equation (3.16) is no longer valid for Equation (3.19). Since the exact null distribution is hard to obtain for the robust method, two commonly used methods for the significance value computation were considered in Publication-IX, namely Monte Carlo simulations and permutations.

Before continuing, it is important to note a highly useful property of the robust periodicity detection method: it is distribution free. A statistic T is said to be distribution-free over a collection of distributions $\mathcal{D}$ if the distribution of T is the same for every joint distribution in $\mathcal{D}$. Under the null hypothesis $H_0 : \beta = 0$ the signal model Y_t in Equation (3.13) can be considered as a set of i.i.d. random variables. Because robust periodicity detection depends on $\{Y_t, t \in T\}$ only through its rank sequence, it follows that the g-statistic, when evaluated using $\tilde{S}(\omega)$, is distribution-free over the class of all joint distributions of N i.i.d. continuous univariate random variables (see, e.g., Randles and Wolfe, 1979). In other words, for each N, regardless of the type of the noise, the g-statistic has exactly the same null distribution as long as the noise term is continuous and i.i.d.

In the Monte Carlo approach, the null distribution is simulated by generating a large number of time series from the null distribution and computing the g-statistic. The distribution can be estimated, e.g., using kernel density estimation methods (see, e.g., Silverman, 1986). The significance values can then be obtained by integrating relative to the estimated distribution.

Alternatively, permutation tests can be used for significance value computation (see, e.g., Good, 2000). The idea of the standard permutation procedure is illustrated below:

1. Evaluate the test statistic on the original data to get g.

2. Randomly permute the original time series P times and evaluate the test statistic on every permutation π_i, $i = 1, \ldots, P$, to get $g^{(i)}$.

3. Estimate the significance by computing the fraction of times the permutation-based g values are larger than the one obtained from the original time series, i.e.,

$$p = \frac{\sum_{i=1}^{P} I(g^{(i)} \geq g)}{P}. \tag{3.21}$$

A sufficient condition for validity of the permutation tests is exchangeability. A sequence of random variables $\{Y_t\}, t = 1, 2, \ldots, N$ is said to be exchangeable, if the joint distribution of $Y_{\pi_1}, Y_{\pi_2}, \ldots, Y_{\pi_N}$ is the same as that of the original sequence $Y_1, Y_2, \ldots, Y_N$ for all permutations π. Under the null hypothesis H_0, the elements of the stochastic process in Equation (3.13) are i.i.d. and hence exchangeable. From a less formal point of view, since the application of a random permutation destroys any periodic structure that is present in the original sequence, permutation tests can be used to assess how highly structured the given time series is in the light of the chosen test statistic versus other permutations of the sequence.

A final point should be made regarding the detection of known periodic components. If ω' is the known frequency to be tested, then a modified g-statistic, g', can be used

$$g' = \frac{|\tilde{S}(\omega')|}{\sum_{l=1}^{a} |\tilde{S}(\omega_l)|}. \tag{3.22}$$

The distribution free property holds for the modified g-statistic as well and the significance value computation can be carried out the same way as explained above.

In practice, periodicity detection is typically applied to a very large number of time series (e.g. genes). Therefore, the issue of multiple testing must be appreciated in the overall testing procedure. In Publication-IX, a general purpose method from (Benjamini and Hochberg, 1995) for controlling the false discovery rate was considered. Since the issue of multiple testing is inevitable in most genomics studies, correction methods for multiple testing are under active research (see, e.g., Dudoit *et al.*, 2003).

Some generalizations and improvements over traditional periodicity detection methods have been proposed. A recent review of different methods can be found in (Artis *et al.*, 2004). Two particular methods are generally found to have a good performance. The first modification by Priestley and Bhansali (see, e.g., Priestley, 1981; Artis *et al.*, 2004) uses a certain type of windowing of the correlogram in order to reduce the variance, i.e.,

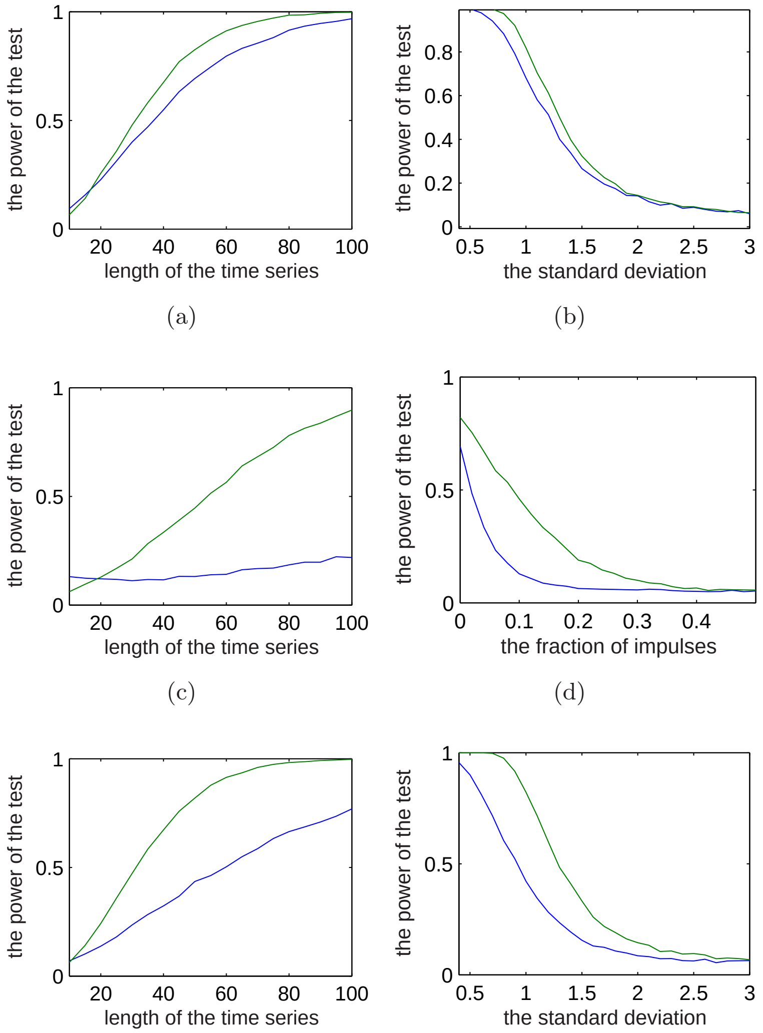
$$S'(\omega) = \sum_{k=-N+1}^{N-1} w(k)\hat{r}(k)e^{-i\omega k}, \tag{3.23}$$

where $w(k)$ is a proper window function, such as Bartlett, Daniel, Parzen, etc. The second modification by Chiu (Chiu, 1989) modifies the g-statistic by replacing the average spectrum in the denominator with a proper trimmed mean of the ranked periodogram ordinates $I_1 \leq I_2 \leq \ldots \leq I_a$, i.e.,

$$g_\gamma = \frac{I_a}{\sum_{l=1}^{[\gamma a]} I_l}, \tag{3.24}$$

where $\gamma \in (0, 1]$ is a trimming parameter. The use of the above modifications in the proposed robust framework is as straightforward as in the case of standard periodogram/correlogram. Those issues are discussed in Publication-IX and will be further investigated in our future studies.

In order to illustrate performance, Figure 3.3 shows the power of the proposed robust test for the detection of unknown frequencies. The power of Fisher's test is shown for comparison purposes. The power of the test is estimated using the signal model shown in Equation (3.13) with three different noise scenarios: Figures 3.3 (a) and (b) Gaussian noise, Figures 3.3 (c) and (d) Gaussian noise and about 10% impulses, and Figures 3.3 (e) and (f) Gaussian noise and cubic distortion. The graphs are computed using 10000 Monte Carlo runs with the significance level $\alpha = 0.05$. In the left



(a) (b)

(c) (d)

(e) (f)

Figure 3.3: The power of the robust periodicity detection method (green) for different test scenarios. The power of the Fisher's (blue) test is shown for comparison. See text for more details.

(resp. right) column of Figure 3.3, the time series length (resp. the noise parameter) is varied while keeping the noise parameters (resp. the time series length ($N = 50$)) fixed. Results shown in Figure 3.3 demonstrate remarkable robustness of the proposed method under a variety of noise conditions.

More extensive performance evaluations are shown in Publication-IX. Periodicity detection results on real gene expression time series as well as the type of bench marking suggested in de Lichtenberg *et al.* (2005) are also provided in Publication-IX. Both extensive simulations and applications to real data demonstrate the good performance and, in particular, remarkably good robustness properties of the proposed methods.

The robust periodicity detection method introduced in this section was build on the robust spectrum estimator described in Section 3.2.2. Separate methods for detecting unknown and fixed frequencies were developed. Both simulation and permutation-based approaches were considered for computing significance values. The distribution-free property of the robust method can be highly useful in practice since the distribution of the test statistic remains the same under the general assumption of continuous i.i.d. noise. Extensive simulations on both synthetic and real time series data show the good properties of the proposed method. In particular, the proposed testing procedure is insensitive to a heavy contamination of outliers, missing-values, short time series, nonlinear distortions, and is completely insensitive to any monotone nonlinear distortions.

Chapter 4

Modeling and Analysis of Genetic Regulatory Networks

During recent years, it has become evident that biological systems are executed in a highly parallel and integrated fashion. It has also been noticed that computational modeling approaches can provide powerful methodologies for gaining deeper insight into the operation of biological systems. In particular, with the help of recent developments in high-throughput measurement techniques, computational methods can have enormous potential in the context of model inference from real measurement data.

Construction of large scale biological models of transcriptional regulation originates back to the work of Stuart Kauffman who envisioned the use of random Boolean networks as coarse-scale models of genetic regulatory networks (Kauffman, 1969). Since then research around regulatory network modeling has attracted ever increasing interest. A central question concerns the level of approximation in network modeling: discrete vs. continuous, deterministic vs. stochastic, and genome-scale analysis vs. subnetworks (for a recent review, see, e.g., de Jong, 2002). Often used coarse-scale model classes are Boolean networks (NK)[1], probabilistic Boolean networks (PBN) and Bayesian networks (BN), whereas more refined models include, among others, differential equations and their stochastic extensions. Coarse-scale

[1] This abbreviation of Boolean networks has a somewhat historical flavor but it is used to make a clear distinction between Boolean and Bayesian networks.

and detailed modeling frameworks are geared towards different modeling goals: the former emphasizes fundamental, generic principles between interacting components, whereas the latter can be used for a refined representation of biochemical reactions.

It is known that real expression levels are (at least approximately) continuous and stochastic (McAdams and Arkin, 1997, 1999) and follow some type of differential equations. Despite the known stochasticity, mainly deterministic differential equations have been studied in this context so far. Moreover, the use of those detailed models is still limited, especially from a model inference point of view, which is generally considered to be the most important problem in computational systems biology. Although computational improvements have also been reported for the use of differential equations as models of gene regulatory networks (Chen *et al.*, 1999; Sakamoto and Iba, 2001; de Hoon *et al.*, 2003; Tabus *et al.*, 2004), coarse-scale approaches have received the most emphasis in the literature.

Coarse-scale, and especially Boolean, network analysis is further motivated by the logical nature of gene regulation (see, e.g., Yuh *et al.*, 1998; Davidson *et al.*, 2002). The dynamical behavior of such networks can be used to model many biological phenomena, e.g., cellular state dynamics with switch-like behavior, stability, and hysteresis (Huang, 1999). Although coarse-scale models cannot represent molecular details, they can capture the fundamental, generic properties of regulation in large biological networks, without modeling the actual quantitative details.

In this chapter, we focus solely on coarse-scale network models. The content of the chapter is two-fold. The first part (Sections 4.1 and 4.2) deals with fundamental properties of discrete large-scale network models, whereas the second part (Sections 4.3 and 4.4) focuses on inference of regulatory rules from data and other computational and application issues, such as relationships between two commonly used probabilistic network models.

4.1 Generic Properties of Genetic Networks

Regardless of the chosen network model class, comprehensive modeling at whole genome-scale is still prohibitive. Hence, it is instructive to study the general properties of biological network models and possibly compare them with those of real cell populations. Of particular interest are the sources of order and robustness observed in genetic regulatory networks. In other words, the ensemble approach, to use a term from Kauffman (2004), focuses on large network ensembles, tries to identify their generic properties, and matches them to the corresponding features of real cells.

4.1.1 Analysis of Boolean Networks

The study of the properties of random Boolean networks has proven to provide insight into the general properties of biological network models (Kauffman, 1969, 1993). Despite the inherent simplicity of the NK model, NKs are capable of representing a wide variety of complex behavior and have much in common with other dynamical systems. One of the most well-studied properties of NK models is perhaps the phase transition between ordered and chaotic behavior (for a recent review, see, e.g., Aldana-Gonzalez *et al.*, 2002). This is also interesting from the evolutionary point of view since it has generally been argued that the boundary between ordered and chaotic regimes, called the critical phase, can provide the necessary robustness and stability for real genetic regulatory networks (Kauffman, 1993). Boolean networks also provide a simplified modeling framework within which the properties and plausibility of different network parameters, such as regulatory mechanisms or network topologies, can be studied.

Let us briefly review the NK and random NK models and then discuss their generic properties. The NK model $G(V, F)$ is defined by a set of n binary-valued nodes $V = \{x_1, x_2 \ldots, x_n\}$ and a list of Boolean functions $F = (f_1, f_2, \ldots, f_n)$. The value of each node x_i at time $t + 1$ is determined by a Boolean function f_i which depends on the value of some controlling elements $x_{j_1(i)}, x_{j_2(i)}, \ldots, x_{j_{k_i}(i)}$ at time t, i.e., $x_i(t+1) = f_i(x_{j_1(i)}(t), x_{j_2(i)}(t), \ldots, x_{j_{k_i}(i)}(t))$. The value of the nodes are updated synchronously in accordance with the updating functions in F.[2] In the random NK model the list of functions as well as their controlling elements (specific input variables) are selected randomly. The state vector of a NK (at time t) is thus an n-dimensional binary vector $\mathbf{x}(t) = (x_1(t), x_2(t), \ldots, x_n(t))^T$. The functions in a NK can also be defined using a vector-valued mapping $\mathbf{f} = (f_1, f_2, \ldots, f_n)$ from $\mathbb{B}^n$ to $\mathbb{B}^n$, where $\mathbb{B} = \{0, 1\}$.

Since NK models are inherently deterministic and have finite state space they cannot be chaotic in a strict sense. However, various parameters

[2] Alternative updating schemes have also been considered. Here we focus only on the original synchronous model.

specifying local or global properties of the networks can be adjusted such that the network is operating in one of two different regimes. In the ordered regime, the system behaves in a simple way, with most of its nodes being constant. In the chaotic regime, the system behaves in the opposite way, with a perturbation of one node propagating to many other nodes. Thus, networks in the chaotic regime are very sensitive to initial conditions and perturbations. The boundary between the ordered and chaotic regimes is the critical phase (or the complex regime).

There are two different ways of analyzing NKs: the so called quenched model and the annealed approximation. In the quenched model, the same realization of the (random) functions is kept through all time whereas in the annealed model a new realization of all the functions is selected after each time step. Many of the theoretical results have been obtained for the annealed model on which we also focus here.

The phase transition between ordered and chaotic regimes of NKs is usually analyzed using so called Derrida plots (Derrida and Pomeau, 1986; Derrida and Stauffer, 1986). To use the notation from (Kesseli *et al.*, 2005), let us define $P_{\rho,n} = \{\mathbf{y} \in \mathbb{B}^n : |\mathbf{y}| = \rho n\}$, where $\rho = 0, 1/n, \ldots, n/n$ and $|\cdot|$ denotes the Hamming weight. Then the Derrida plot of a NK is defined as

$$d_{\mathbf{f}}(\rho) = \frac{1}{2^n \binom{n}{\rho n}} \sum_{\mathbf{x} \in \mathbb{B}^n} \sum_{\mathbf{y} \in P_{\rho,n}} \frac{1}{n} \left| \mathbf{f}(\mathbf{x}) \oplus \mathbf{f}(\mathbf{x} \oplus \mathbf{y}) \right|, \tag{4.1}$$

where $\oplus$ denotes addition modulo two and $\mathbf{f}$ is the vector-valued network function. Note that Equation (4.1) corresponds to the quenched model. The Derrida plot of the annealed model can be obtained by taking the expectation of $d_{\mathbf{f}}(\rho)$ relative to the random functions. The actual Derrida plot is obtained by plotting $\mathbb{E}[d_{\mathbf{f}}(\rho)]$ versus ρ (see Figure 4.2 for examples). Instead of using the definition directly, Derrida plots are usually approximated by averaging over different random networks, random initial points ($\mathbf{x}$) and random perturbations ($\mathbf{y}$) in Monte Carlo fashion. More advanced spectral-based methods have recently been introduced (Kesseli *et al.*, 2005).

In the annealed approximation framework, the propagation of the perturbation over time is obtained by iterating the mapping defined by the Derrida plot. Two cases are of interest. If the Derrida plot is above the main diagonal, then (at least some) perturbations tend to diverge and the random NKs are called chaotic. Alternatively, if the Derrida plot is completely below the main diagonal, then perturbations tend to vanish and the random NKs are considered to be ordered. If, in addition to the Derrida plot not being above the main diagonal, the derivative of the Derrida plot at the origin is equal to one, then the random NKs are in the critical phase.

Other related ways of analyzing the network dynamics include percolation analysis where nodes of the network are place on a grid. For each node in the network, the neighboring nodes in the grid are used as the input variables. For more details, see Publication-III and, e.g., (Derrida and Stauffer, 1986; Aldana-Gonzalez *et al.*, 2002). Sensitivity of the functions in random NKs have also been studied and a connection to the Derrida plot has been established (Shmulevich and Kauffman, 2004; Kesseli *et al.*, 2005).

Apparently, there are two parameters in the NK model that determine its operation mode: the random functions and the topology of the controlling elements (wiring of the input variables). In its original formulation (Kauffman, 1969), the random NK model was defined to have a constant number k of controlling elements for each node (selected uniformly randomly) and each function was an instantiation of a random p-biased function, i.e., for each input variable configuration $(x_{j_1(i)}, x_{j_2(i)}, \ldots, x_{j_k(i)}) \in \mathbb{B}^k$ the probability of the function being equal to one is p. In terms of the Derrida plot analysis, the critical connectivity and bias for the phase transition, when $n \to \infty$, were shown to be related as $1/k = 2p(1-p)$ (Derrida and Pomeau, 1986). Natural relaxations of the NK model allow the updating functions to have a more meaningful structure, or the number of controlling elements to vary.

It is well-known that forcing functions (also called as canalizing functions) provide one of the few mechanisms for preventing chaotic behavior in NKs (Stauffer, 1987). (For a precise definition of the forcing functions, see Section 4.1.2). There is also an abundance of evidence that functions from this class are commonly utilized in higher vertebrate gene regulatory mechanisms. Recently, Harris *et al.* studied more than 150 known transcriptional regulatory functions with varying numbers of regulating components and found that these controlling rules are strongly biased toward forcing functions (Harris *et al.*, 2002).

Recent results also show that if the number of controlling elements (input variables) follows a power law distribution, $P(k) = \frac{1}{\zeta(\gamma)} k^{-\gamma}$ $(k \geq 1, \gamma > 1$, $\zeta(\gamma)$ is the Riemann Zeta function), the dynamics of the networks exhibit phase transition for $\gamma \in (2.0, 2.5)$, the specific value being determined by the bias p, with ordered and chaotic regimes being obtained for larger and smaller values of γ, respectively (Aldana and Cluzel, 2003; Aldana, 2003). In particular, when $\gamma \geq 2.5$ the system is always in the ordered regime, regardless of the bias p. Real genetic networks have also been suggested to exhibit scale-free topology (Fox and Hill, 2001; Oosawa and Savageau, 2002).

4.1.2 The Role of Certain Post Classes in Boolean Networks

In Publication-III we studied the general properties of certain Post function classes and, especially, the role that they play in the emergence of order in NKs. All classes of Boolean functions that are closed under composition were characterized by Post (1921, 1941) and have thereafter been called Post function classes. Several well-known function classes, such as monotone, linear, self-dual, and the class of all functions, belong to this family of function classes. In Publication-III, we were only interested in certain special classes of Post functions, namely, so called A^μ, a^μ, A^∞, and a^∞ function classes. For convenience, let us first review the definition of the above function classes as well as the definition of the class of forcing functions.

A k-variable Boolean function is said to be forcing if there exist an index $i \in \{1, \ldots, k\}$ and $u, v \in \mathbb{B}$ such that for all $(x_1, \ldots, x_k) \in \mathbb{B}^k$, if $x_i = u$ then $f(x_1, \ldots, x_k) = v$. The input variable x_i is called the forcing variable, u the forcing value, and v the forced value.

The set of true and false vectors of a Boolean function f are $T(f) = \{x : f(x) = 1\}$ and $F(f) = \{x : f(x) = 0\}$, respectively. A Boolean function f belongs to class A^μ, $\mu \geq 2$, if any μ true vectors share a common component equal to 1 (note that some of these μ vectors may be repeated). Analogously, a Boolean function f belongs to class a^μ, $\mu \geq 2$, if any μ false vectors share a common component equal to 0. Moreover, a Boolean function f belongs to class A^∞ if all true vectors share a common component equal to 1. The class a^∞ is again obtained by duality.

The motivation of Publication-III was to characterize and analyze a class of functions with some desirable properties. The main results of that study are summarized below. The considered Boolean function class (A^μ) has the following characteristics:

1. It is much larger than the class of forcing functions;

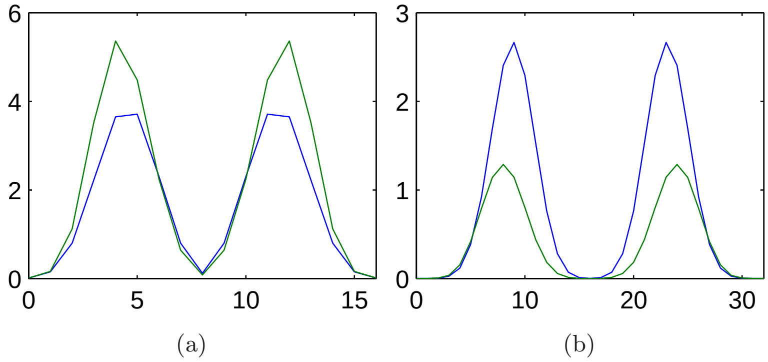
2. Functions from this class are characterized by internal homogeneity;
3. An abundance of functions from this class prevent chaotic behavior in NKs;
4. Functions from this class ensure robustness against noise and uncertainty; and
5. Functions from this class are closed under composition.

Let us briefly elaborate on these five findings. More discussion can be found in Publication-III.

A proper class of regulatory functions should be large enough to be plausible from an evolutionary point of view. The average connectivity (the number of controlling elements in regulatory functions) in real genetic regulatory networks is considered to be considerable higher than two (Arnone and Davidson, 1997). However, the fraction of functions that are forcing, one of the few known mechanisms preventing chaotic behavior in NKs, gets very small as the connectivity increases. Assuming forcing functions indeed were selected randomly by evolution, then it must have been quite a difficult task to find so few regulatory rules from such a large set of all regulatory rules. The cardinalities of the proposed functions classes (A^μ and a^μ with $\mu = 2$) are much larger than the cardinality of the class of forcing functions. Moreover, the difference in cardinalities gets more significant for higher connectivity (see Publication-III for further details and discussion). It is also worth noting that the exact number of forcing functions has been derived recently (Just *et al.*, 2004).

Forcing functions are similar to random p-biased functions in that they exhibit a preference towards biased functions (many ones or zeros in the truth table). The A^μ and a^μ Post function classes were found to have the same kind of preference. Figure 4.1 shows the histograms of the number of functions versus the number of ones in their truth table for 4 and 5-variable forcing and $A^2 \cup a^2$ functions.

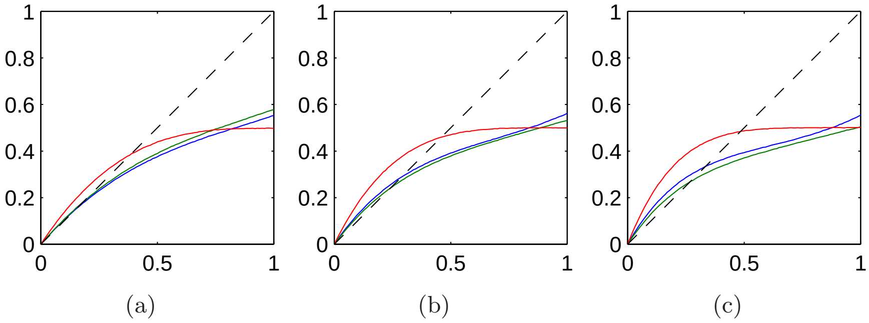
One of the most important findings is that networks constructed from functions belonging to the considered Post classes provide a mechanism to prevent chaotic behavior. In Publication-III, we analyzed the dynamics using both Derrida plots (see Equation (4.1)) and percolation on square lattices. Both approaches showed that the considered Post classes exhibit a tendency for ordered network dynamics. Figure 4.2 shows some example



(a)

(b)

Figure 4.1: The histograms of the number of functions versus the number of ones in their truth table for (a) 4 and (b) 5-variable forcing (green) and $A^\mu \cup a^\mu$ ($\mu = 2$) (blue) function classes. y-axis shows (a) hundreds (b) hundreds of thousands of functions.



(a)

(b)

(c)

Figure 4.2: Derrida plots for BNs constructed from (a) 3, (b) 4, and (c) 5-variable random p-biased functions (red), forcing functions (green) and A^2 and a^2 Post (blue) functions.

Derrida plots for 100-node networks constructed from 3, 4 and 5-variable random p-biased functions ($p = 1/2$), forcing functions, and A^2 and a^2 Post functions. The class of forcing functions as well as the class of A^2 and a^2 functions clearly provide much more ordered dynamics than the class of all Boolean functions.

The robustness properties of the A^μ and a^μ Post function classes have been known for quite some time (see, e.g., Muchnik and Gindikin, 1962). In particular, Boolean circuits synthesized using the considered Post function classes are strongly fault tolerant, i.e., the correct output of a device is guaranteed even when some of the components are erroneous. The robustness properties rely on the concept of closure, which leads to the final note.

By construction, all the Post function classes are closed. This has interesting implications for NKs. For example, any node at any number of time steps in the future is guaranteed to be controlled by a function from the same closed function class. Consider random p-biased functions. They have a similar closure property only in the annealed approximation framework since it is easy to see that any node at any number of time steps in the future is controlled by a p-biased function. However, only Post (closed) function classes can have the closure property in the quenched framework. Another interesting implication is related to time in the discrete-time model and time in the real, continuous-time systems. Obviously, discrete-time models are idealizations of continuous-time systems, and time in the model may not correspond to time of the real system. This discrepancy, however, is greatly alleviated by the Post function classes. If the genes are regulated by rules belonging to a closed function classes, then, no matter how many discrete time steps correspond to the time interval of the actual physical regulation, the overall rule regulating the gene still belongs to the same closed function class.

To this end, we would like to mention that other function classes, such as nested canalizing (or nested forcing) functions (Kauffman *et al.*, 2003) and chain functions (Gat-Viks and Shamir, 2003), have also been proposed in the context of gene regulatory network modeling recently.

4.2 Testing Membership in Certain Boolean Function Classes

In order to facilitate the analysis of Boolean networks, it is useful to have efficient tools for testing the membership of a given function in different function classes. In Publication-V we developed methods for testing membership in the classes of forcing functions, the A^{μ} and a^{μ} functions, and the A^{∞} and a^{∞} functions. The novel method for testing membership in the class of forcing functions can easily be extended to cover some subclasses, such as the classes of nested canalizing functions and chain functions, too. A mathematically elegant solution to testing membership in all Post classes has been introduced in (Levchenkov, 2000). Our goal in Publication-V was to develop efficient spectral methods for membership testing. Spectral methods for membership testing are preferred over their direct analogs since they are usually more efficient from a time complexity point of view (Agaian *et al.*, 1995). Spectral methods for testing several other (Post) function classes, such as monotone, self-dual and linear, are described in (Agaian *et al.*, 1995). In the following we give an overview of the proposed methods. Proofs of the theorems can be found in Publication-V.

4.2.1 Testing Membership in the Class of Forcing Functions

Let $\mathbf{R}_n(0,1)$ be the Rademacher $(0,1)$ matrix of order n, whose rows correspond to all n-element binary vectors in lexicographical order. As an example,

$$\mathbf{R}_3(0,1) = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}^T, \tag{4.2}$$

where T denotes matrix transpose. Let the truth table of an n-variable Boolean function f be denoted as the 2^n-element binary column vector $\mathbf{f}$. Define the forcing transform as

$$\mathbf{c}^{(1,1)} = \mathbf{R}_n^T(0,1) \cdot \mathbf{f}. \tag{4.3}$$

Note that $\cdot$ stands for the regular matrix-vector (or scalar-vector) multiplication. For convenience, define three other related quantities as

$$\begin{aligned} \mathbf{c}^{(0,1)} &= |\mathbf{f}| \cdot \mathbf{1} - \mathbf{c}^{(1,1)} & (4.4)\\ \mathbf{c}^{(0,0)} &= \left(2^{n-1} - |\mathbf{f}|\right) \cdot \mathbf{1} + \mathbf{c}^{(1,1)} & (4.5)\\ \mathbf{c}^{(1,0)} &= 2^{n-1} \cdot \mathbf{1} - \mathbf{c}^{(1,1)}, & (4.6) \end{aligned}$$

where $\mathbf{1}$ is a column vector containing all 1s.

Membership testing in the class of forcing functions can now be stated as follows. The function f is a forcing function if and only if there exist $u, v \in \mathbb{B}$ and $i \in \{1, \ldots, n\}$, such that $\mathbf{c}_i^{(u,v)} = 2^{n-1}$. In that case, x_i is the forcing variable, u is the forcing value and v is the forced value. Furthermore, information about all forcing variables, forcing values, and forced values is contained in the vectors $\mathbf{c}^{(u,v)}$.

4.2.2 Testing Membership in the Class of A^μ and a^μ Functions

Let us focus on the A^μ property because a^μ is similar by duality. Let S be an n-element set of natural numbers $\{1, 2, \ldots, n\}$. For each family Ω of subsets of S, let f be the corresponding indicator function, i.e., $f(x_1, \ldots, x_n) = 1$ if and only if Ω contains a set $T \subseteq S$ such that

$$x_i = \begin{cases} 1, \text{ if the } i\text{th element of } S \text{ is in } T \\ 0, \text{ otherwise.} \end{cases} \tag{4.7}$$

Thus, there is one-to-one correspondence between Ω and f. It is easy to see that the function f is in A^μ iff any μ members of Ω have a non-empty intersection. Such sets are called μ-inseparable. Let the rows of matrix $\mathbf{B}_{n,\mu}$ contain all indicator functions of μ-element sets that are not μ-inseparable, with the exception of sets containing the empty set. In other words, the first column of $\mathbf{B}_{n,\mu}$ contains zeros. For example, for $n = 3$ and $\mu = 2$,

$$\mathbf{B}_{3,2} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}. \tag{4.8}$$

Define the A^μ transform as

$$\mathbf{c}_{n,\mu} = \mathbf{B}_{n,\mu} \cdot \mathbf{f}. \tag{4.9}$$

The problem of membership testing in the class A^μ can now be stated as follows. Let $\mu \geq 2$ be given. Let f be an n-variable Boolean function that is equal to 0 on the all-zero vector and has at least 2 true vectors, i.e., $|T(f)| \geq 2$. Let $k = \min(\mu, |T(f)|, n)$. Then, f is in A^μ if and only if $\mathbf{c}_{n,k}$

does not contain an element equal to k. If f has a single true vector, then f is in A^μ if and only if f is equal to 0 on the all-zero vector. The constant zero function $f(x) \equiv 0$ is in A^μ by definition.

4.2.3 Testing Membership in the Class of A^∞ and a^∞ Functions

There are at least two ways of testing the A^∞ and a^∞ properties. Remember that a Boolean function f belongs to class A^∞ if all true vectors share a common component equal to 1. Thus, if we express f in its (canonical) disjunctive normal form, then there must exist a variable that is contained in all the conjunctions and f can be expressed as

$$f(x_1, \dots, x_i, \dots, x_n) = x_i \cdot g(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n), \tag{4.10}$$

where g is another Boolean function not depending on variable x_i. Setting the variable x_i to 0 forces the function to take on value 0. Therefore, if a function is A^∞, then it is also a forcing function with $u = v = 0$. The same reasoning applies to the a^∞ with $u = v = 1$. This allows us to use the methods developed for testing the forcing property.

Alternatively, the following theorem from (Yablonsky *et al.*, 1966) allows an alternative approach: if an n-variable Boolean function f is A^μ, then for $\mu \geq n$ it is also A^∞. Thus, for the A^∞ property one only needs to test the membership in the class of A^n functions.

4.2.4 Testing Membership in Some Related Function Classes

Recall that the definition of the nested forcing functions (Kauffman *et al.*, 2003) is the following. Given an n-variable nested forcing function f, there exists a forcing variable x_i with a forcing value u_i. Fixing the first forcing variable x_i to its non-forcing value $\overline{u}_i$ defines another $(n-1)$-variable Boolean function. This reduced function is also forcing with another forcing variable and forcing value, and so on and so forth. It immediately follows that the above methods for testing the general forcing property can be used to test the nested forcing property with minor modifications. Moreover, since the chain functions (Gat-Viks and Shamir, 2003) are known to be a special case of the nested forcing functions (Kauffman *et al.*, 2003) the same argument applies to the class of chain functions as well.

The proposed methods can also be extended to so-called generalized forcing functions, where the forcing property is defined in terms of several forcing variables which all need to take a specific value in order to force the output value of the function.

4.3 Inference of Regulatory Functions from Data

As already pointed out in the Introduction Chapter, in contrast to reductionistic approaches in biology, it is apparent that the behavior of genes needs to be studied in a global rather than in an individual manner. Consequently, most of the recent work on biological network modeling has focused of learning multivariate regulatory models from experimental data. Such approaches inevitably require advanced computational methods to process massive amounts of data and to make useful predictions about system behavior in the presence of known conditions.

There have been a number of attempts to model gene regulatory networks, including Boolean networks (Kauffman, 1969), Bayesian networks (Murphy and Mian, 1999; Friedman *et al.*, 2000), linear models (van Someren *et al.*, 2000), nonlinear models (Kim *et al.*, 2000), neural networks (Weaver *et al.*, 1999), differential equation models (Chen *et al.*, 1999), and models including stochastic components on the molecular level (McAdams and Arkin, 1997). Due to reasons outlined in the beginning of this chapter, the use of detailed models is quite limited and, therefore, coarse-scale model classes, such as Boolean networks (NK), probabilistic Boolean networks (PBN), and static and dynamic Bayesian networks (BN/DBN) have received a considerable amount of attention. The use of coarse-scale models is also motivated by the encouraging studies of the capacity of discrete DBNs for revealing gene regulatory networks (Smith *et al.*, 2002; van Berlo *et al.*, 2003; Husmeier, 2003). For example, Husmeier generated short time series from a realistic differential equation model and analyzed the capabilities of DBN inference methods for revealing the underlying network structure. Discussion on discrete DBNs and related model classes is continued in Section 4.4.

The above mentioned coarse-scale models are both discrete-time and discrete-state.[3] A special case is the Boolean framework where the state of a

[3] Bayesian networks can be both continuous-time and continuous-valued, but discrete versions are considered here.

gene is represented by a Boolean variable (ON or OFF) and the interactions between the genes are represented by Boolean functions, which determine the state of a gene, either deterministically or stochastically, on the basis of the state of some other genes. Let us focus on deterministic predictive models for a while. A number of studies have focused on identifying the structure of Boolean networks from binarized expression data (Liang *et al.*, 1998; Akutsu *et al.*, 1999; Karp *et al.*, 1999; Akutsu *et al.*, 2000; Ideker *et al.*, 2000*b*; Akutsu *et al.*, 2003). Although we do not recommend using standard Boolean networks as models of gene regulatory networks *per se*, it is useful to look at the problem of inferring binary predictors.

Most of the previous approaches have focused on the so-called Consistency Problem, i.e., the problem of determining whether there exists a network that is consistent with the examples. Such an approach may not be applicable in a realistic setting in which noisy observations or other errors are contained, as is the case with microarray data. A learning paradigm that can incorporate such inconsistencies is called the Best-Fit Extension Problem (Boros *et al.*, 1998). Shmulevich *et al.* showed that if the Best-Fit Extension Problem is solvable in polynomial time for one Boolean function from a class $\mathcal{C}$, then the Best-Fit Extension problem is polynomial time also for the whole network in which all functions belong to class $\mathcal{C}$ (Shmulevich *et al.*, 2002). In Publication-II we developed fast optimized search algorithms under the Best-Fit Extension paradigm for the inference of binary predictors for the NK models (see Section 4.3.2). Applications of the inference methods to real gene expression data are also illustrated in Publication-II.

4.3.1 Best-Fit Extension Problem

Let us briefly review the Best-Fit Extension problem for Boolean functions (Boros *et al.*, 1998). A partially defined Boolean function pdBf is defined by a pair of sets (T, F) such that $T, F \subseteq \mathbb{B}^n$, where T is the set of true vectors and F is the set of false vectors. A function f is called an extension of pdBf(T, F) if $T \subseteq T(f)$ and $F \subseteq F(f)$. Suppose that we are also given positive weights $w(\mathbf{x})$ for all vectors $\mathbf{x} \in T \cup F$ and define $w(S) = \sum_{\mathbf{x} \in S} w(\mathbf{x})$ for a subset $S \subseteq T \cup F$. Then, the error size of function f is defined as

$$\varepsilon(f) = w(T \cap F(f)) + w(F \cap T(f)). \tag{4.11}$$

The goal is then to find sets T^* and F^* such that $T^* \cap F^* = \emptyset$ and $T^* \cup F^* = T \cup F$ for which the pdBf(T^*, F^*) has an extension in some class of functions $\mathcal{C}$ (chosen *a priori*) and so that $w(T^* \cap F) + w(F^* \cap T)$ is minimum. Consequently, any extension $f \in \mathcal{C}$ of pdBf(T^*, F^*) has minimum error size. Note that the Consistency Problem is a special case of the Best-Fit Extension Problem when $\varepsilon(f) = 0$.

The generalization of the above definition to networks is obtained by simply repeating it to all the nodes. Consider a single node. In the Best-Fit Extension framework, T and F consists of observed values of the predictor variables (e.g., vectors $\mathbf{x} = (x_{j_1(i)}, x_{j_2(i)}, \ldots, x_{j_{k_i}(i)})^T$ in the NK-framework). Each observation vector $\mathbf{x}$ is in T (resp. F) if the corresponding value of the target gene (node to be predicted) is ON (resp. OFF). Note that the weight vector w is an additional parameter to the prediction problem. Let us assume that the different observations are weighted equally, although that does not need to be the case in general. If the weights are unspecified, then a reasonable definition for the weight of an observation $\mathbf{x}$ is the absolute difference in the number of times $\mathbf{x}$ is true or false. Alternatively, since we consider the class of all Boolean functions in the following, we can equivalently define a separate weight for the same vector in T and F. For example, the weight of $\mathbf{x} \in T$, $w_T(\mathbf{x})$, (resp. $\mathbf{x} \in F$, $w_F(\mathbf{x})$) is the sum of all weights of observation $\mathbf{x}$ belonging to T (resp. F). Naturally, weights can also incorporate, e.g., information about the quality of different measurements (see Publication-II for further discussion).

4.3.2 Optimized Search Algorithms

Application of the Best-Fit Extension paradigm for networks relies on a type of brute-force search where the same inference method is applied to each of the nodes and to all possible predictor variable combinations. Let us assume that the network consists of n nodes and that one is interested in inferring k-variable ($k \leq n$) predictor functions. Consider a single node and a single predictor variable combination. Let $\mathbf{c}^{(0)}, \mathbf{c}^{(1)} \in \mathbb{R}^{2^k}$, and $\mathbf{c}^{(0)}$ and $\mathbf{c}^{(1)}$ are indexed from 1 to 2^k and initially zero vectors. Let s be a bijective mapping $s : \{0,1\}^k \rightarrow \{1, \ldots, 2^k\}$ that encodes all binary vectors of length k to positive integers. Then, during one pass over the given examples in T

and F, $\mathbf{c}^{(0)}$ and $\mathbf{c}^{(1)}$ can be updated to

$$\begin{aligned} \mathbf{c}_i^{(0)} &= w(x),\ \text{if } x \in F \wedge s(x) = i \\ \mathbf{c}_i^{(1)} &= w(x),\ \text{if } x \in T \wedge s(x) = i. \end{aligned} \tag{4.12}$$

Note that Equation (4.12) can also be redefined for two weights vector w_F and w_T. Elements of $\mathbf{c}_i^{(0)}$ and $\mathbf{c}_i^{(1)}$ that are not set in Equation (4.12) remain zero-valued due to initialization. Let $\overline{\mathbf{f}}$ denote the complement of $\mathbf{f}$. Then, the error size of function f can be written as

$$\varepsilon(f) = \sum_{i=1}^{2^k} \mathbf{c}_i^{(\overline{\mathbf{f}}_i)}. \tag{4.13}$$

It is easy to see that error size is minimized when $\mathbf{c}_i^{(\overline{\mathbf{f}}_i)}$ is minimum or, conversely, $\mathbf{c}_i^{(\mathbf{f}_i)}$ is maximum for all i. Thus, the truth table of the optimal Boolean function is $\mathbf{f}_i^{\text{opt}} = \arg\max_j \mathbf{c}_i^{(j)}$.

The generalization for the entire network is as straightforward as explained above. That is, the above method must be applied to all $\binom{n}{k}$ variable combinations and all n nodes. When the time spent on memory initialization is ignored, the optimal solution of the Best-Fit Extension Problem for the entire network can be found in time

$$O\left(n^k \cdot m \cdot n \cdot \text{poly}\,(k)\right). \tag{4.14}$$

The time complexity notation deserves a special remark due to the extra constraint $k \leq n$ (see Publication-II for more details).

Due to the noise and limited amounts of data, a single Best-Fit function may not stand out sufficiently uniquely, i.e., there may be other functions with a comparable error size. Selecting only a single predictor function may lead to incorrect results. That can be circumvented by finding all functions having the error size below some threshold ε_{max}.

Consider again only a single node and a single variable combination. Let us assume we know vectors $\mathbf{c}^{(0)}$ and $\mathbf{c}^{(1)}$, the error-size of the Best-Fit function $\varepsilon(f^{\text{opt}}) = \varepsilon_{\text{opt}}$, and the optimal binary function f^{opt} itself through its truth table $\mathbf{f}^{\text{opt}}$. Define $\mathbf{c}$ as $\mathbf{c}_i = |\mathbf{c}_i^{(0)} - \mathbf{c}_i^{(1)}|$, $i = 1, \ldots, 2^k$, and let $\mathbf{f}'$ denote the truth table of a non-optimal function f'. The truth table of f' can now be written as $\mathbf{f}' = \mathbf{f}^{\mathrm{opt}} \oplus \mathbf{d}$, where $\mathbf{d} \in \{0,1\}^{2^k}$ defines the distortion from the optimal function. From Equation (4.13) is follows that

$$\varepsilon(f') = \sum_{i=1}^{2^k} \mathbf{c}_i^{(\overline{\mathbf{f}'_i})} = \sum_{i=1}^{2^k} \mathbf{c}_i^{(\overline{\mathbf{f}_i^{\mathrm{opt}}})} + \sum_{i\,:\,\mathbf{d}_i=1} \mathbf{c}_i = \varepsilon_{\mathrm{opt}} + \mathbf{c}^T\mathbf{d}. \tag{4.15}$$

The above equation allows to rewrite the set of functions to be found $\{f : \varepsilon(f) \leq \varepsilon_{\max}\}$ in terms of truth tables as

$$\{\mathbf{f}^{\mathrm{opt}} \oplus \mathbf{d} : \mathbf{c}^T\mathbf{d} \leq \varepsilon_{\max} - \varepsilon_{\mathrm{opt}}\}. \tag{4.16}$$

In Publication-II we introduced a simple recursive algorithm for finding the set of functions $\{f : \varepsilon(f) \leq \varepsilon_{\max}\}$. Conceptually, the algorithm builds a tree where the root corresponds to the optimal function $\mathbf{f}^{\mathrm{opt}}$ and the nodes below the root correspond to acceptable (permuted) vectors $\mathbf{d}$. See Publication-II for more details.

Yet another important matter that deserves to be mentioned is a connection between the Best-Fit Extension and standard pattern recognition methods (Publication-VI). In a pattern recognition framework, the input-output patterns are usually modelled as random variables $\mathbf{X}$ and Y and are assumed to have a joint distribution π. It is common to search for a predictor (classifier) which has the smallest probability of misprediction, i.e., $f = \arg\min_{f\in\mathcal{C}} P(f(\mathbf{X}) \neq Y)$. When the joint distribution π is unknown a classification rule is applied to the sample data to construct a predictor. In the discrete setting, the most often used rule is the so-called histogram rule. The plug-in estimate of π is $\hat{\pi}(\mathbf{x}, y) = n(\mathbf{x}, y)/m$, where $n(\mathbf{x}, y)$ is the number of times $\mathbf{x}$ and y are observed jointly in the data and m denotes the number of samples. The histogram rule finds a predictor which minimizes the resubstitution error on the given data set, i.e., $\hat{f} = \arg\min_{f\in\mathcal{C}} \hat{P}(f(\mathbf{X}) \neq Y)$, where the probability is computed relative to the plug-in estimate $\hat{\pi}$.

The connection between the Best-Fit Extension and the histogram rule is the following. Given the plug-in estimate $\hat{\pi}$, define the corresponding Best-Fit weights for the observed inputs $\mathbf{x}$ as $w(\mathbf{x}) = |\hat{\pi}(\mathbf{x}, 0) - \hat{\pi}(\mathbf{x}, 1)|$, and set $\mathbf{x} \in T$ if $\hat{\pi}(\mathbf{x}, 1) \geq \hat{\pi}(\mathbf{x}, 0)$, otherwise $\mathbf{x} \in F$. It is easy to see that the error size of a function f satisfies $\varepsilon(f) = w(T \cap F(f)) + w(F \cap T(f)) = \hat{P}(f(\mathbf{X}) \neq Y) - \hat{P}(\hat{f}(\mathbf{X}) \neq Y)$, where $\hat{f}$ is the optimal (unconstrained) Boolean predictor. Thus, minimizing the error size will also minimize the resubstitution error. Using different weights for T and F, w_T and w_F, allows a direct computation of the resubstitution error estimate as well. This connection gives a sound basis for the use of different error estimation and model selection strategies, such as cross-validation (Stone, 1974), bootstrap (see, e.g., Efron and Tibshirani, 1993), bolstered error estimation (Braga-Neto and Dougherty, 2004), and several different distribution-free error bounds (see, e.g., Devroye *et al.*, 1996). This also shows a close connection to information theoretic methods, such as the minimum description length principle in gene regulatory network inference (Tabus and Astola, 2001) and the normalized maximum likelihood based binary regression (Tabus *et al.*, 2002).

The above methods provide efficient and optimized search algorithms under the Best-Fit Extension paradigm for the inference of logical regulatory rules from experimental data. The Best-Fit Extension Problem has been extensively studied for several Boolean function classes in (Boros *et al.*, 1998) and for Boolean networks in (Shmulevich *et al.*, 2002). For the class of all Boolean functions, the proposed methods provide more efficient search algorithms for the inference of both Boolean functions and Boolean networks. The algorithm for finding all functions having a limited error size provides an efficient way of finding a set of top ranked predictors. That can be particularly useful in the cases of small samples and noisy environments where the best predictor do not stand out sufficiently uniquely. Moreover, the connection between the Best-Fit Extension Problem and the discrete classification rule enables applying standard model validation tools under the Best-Fit Extension paradigm. Possible future research directions, such as adding measurement quality information into the Best-Fit Extension Problem via the weights w, are discussed in Publication-II and Publication-VI.

4.4 Probabilistic Models for Regulatory Networks

Regulatory network modeling is typically confounded by a considerable amount of uncertainty. This uncertainty arises from several sources, the major ones being the stochastic nature of biological regulation and the noise present in the measurements. A common approach to tackling the issue of uncertainty is the use of probabilistic models. Two widely used stochastic modeling frameworks are considered here, namely, probabilistic Boolean networks (PBN) (Shmulevich *et al.*, 2002*a*), and static and dynamic Bayesian networks (BN/DBN) (Pearl, 1988; Cowell *et al.*, 1999; Murphy, 2002).

In Publication-VIII we showed that PBNs and a certain subclass of DBNs can represent the same joint probability distributions. In other words, the two models are statistically equivalent. This relationship is useful because it opens up the possibility of applying the advanced tools of these network models to both of them. In other words, this extends the collection of analysis tools of both model classes. Let us briefly review the two model classes and then summarize the main results and implications of Publication-VIII.

4.4.1 Probabilistic Boolean Networks

PBN is a model class that has been recently introduced in the context of gene regulatory network modeling (Shmulevich *et al.*, 2002*a*). PBN is a stochastic extension of the standard Boolean network that incorporates rule-based dependencies between variables but is also stochastic in nature. PBNs and closely related probabilistic gene regulatory network models have been further studied and developed in numerous papers (Kim *et al.*, 2002; Shmulevich *et al.*, 2002*b*,*c*; Datta *et al.*, 2003; Dougherty and Shmulevich, 2003; Hashimoto *et al.*, 2003; Zhou *et al.*, 2003*b*; Datta *et al.*, 2004; Zhou *et al.*, 2004); see also (Dougherty *et al.*, 2005).

A PBN $G(V, F)$ is defined by a set of binary-valued nodes $V = \{X_1, \dots, X_n\}$ and a list of function sets $F = (F_1, \dots, F_n)$, where each function set F_i consists of $l(i)$ Boolean functions, i.e., $F_i = \{f_1^{(i)}, \dots, f_{l(i)}^{(i)}\}$. At each time step, the value of each node X_i is updated by a Boolean function taken from the corresponding set F_i. In the case of independent PBNs, the predictor functions are selected independently for each node X_i according to the corresponding selection probabilities $P(F^{(i)} = f_j^{(i)})$, $1 \leq j \leq l(i)$, where $F^{(i)}$ denotes a random variable taking values in $F_i = \{f_1^{(i)}, \dots, f_{l(i)}^{(i)}\}$. A realization of the PBN is defined by a vector of Boolean functions $\mathbf{f} = (f_{i_1}^{(1)}, f_{i_2}^{(2)}, \dots, f_{i_n}^{(n)})$. In the case of dependent PBNs, the predictor functions are selected for each time step from a joint distribution $P(\mathbf{F} = \mathbf{f})$, where $\mathbf{F}$ denotes a multivariate random variable taking values in $F_1 \times \cdots \times F_n$.

4.4.2 Dynamic Bayesian Networks

DBN is a general model class that is capable of representing complex temporal stochastic processes (see, e.g., Murphy, 2002). DBNs are also known to be able to capture several other modeling frameworks, such as hidden Markov models (and its variants) and Kalman filter models, as its special cases. DBNs and their non-temporal versions have been successfully used in a variety of problems, such as in speech recognition, target tracking and identification, genetics, and medical diagnostic systems (see, e.g., Cowell *et al.*, 1999, and the references therein). BNs and DBNs have also been intensively studied in the context of modeling genetic regulation (Friedman *et al.*, 1998; Murphy and Mian, 1999; Friedman *et al.*, 2000; Hartemink *et al.*, 2001; Pe'er *et al.*, 2001; Hartemink *et al.*, 2002; Smith *et al.*, 2002; Yoo *et al.*, 2002; Yu *et al.*, 2002; Husmeier, 2003; Imoto *et al.*, 2003; Perrin *et al.*, 2003; Friedman, 2004; Imoto *et al.*, 2004; Pournara and Wernisch, 2004; Rangel *et al.*, 2004; Yu *et al.*, 2004; Beal *et al.*, 2005; Bernard and Hartemink, 2005).

For BNs and DBNs we use the notation from (Friedman *et al.*, 1998). Let $\mathbf{X} = \{X_1, \ldots, X_n\}$ denote the discrete random variables in the network. A BN for $\mathbf{X}$ is a pair $B = (G, \Theta)$ that encodes a joint probability distribution over $\mathbf{X}$. The first component, G, is a directed acyclic graph whose vertices correspond to the variables in $\mathbf{X}$. The network structure induces conditional independencies between the variables in $\mathbf{X}$. The second component, Θ, defines a set of local conditional probability distributions for G. Let $\mathbf{Pa}(X_i)$ denote the parents of the variable X_i in the graph G. Then, a BN B defines a unique joint probability distribution over $\mathbf{X}$ given by the well-known formula

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathbf{pa}(X_i)). \tag{4.17}$$

A DBN that represents the first-order Markov processes of variables in $\mathbf{X}$ is a pair (B_0, B_1), where $B_0 = (G_0, \Theta_0)$ is an initial BN defining the joint distribution of the variables in $\mathbf{X}(0)$, and $B_1 = (G_1, \Theta_1)$ is a transition BN specifying the transition probabilities $P(\mathbf{X}(t)|\mathbf{X}(t-1))$ for all $t > 0$. The following constraints are assumed: $\mathbf{Pa}(X_i(0)) \subseteq \{X_1(0), \ldots, X_n(0)\}$ for all i, and $\mathbf{Pa}(X_i(t)) \subseteq \{X_1(t-1), \ldots, X_n(t-1)\}$ for all i and $t > 0$.

4.4.3 Relationships between PBNs and DBNs

To show the relationships between the two model classes, we introduced a way of conceptually expressing a PBN as a DBN and *vice versa.* For a given independent PBN, it is relatively easy to show that the probability of a finite time series can be expressed as

$$P(\mathbf{x}(0), \mathbf{x}(1), \ldots, \mathbf{x}(T)) = P(\mathbf{x}(0)) \prod_{t=1}^{T} \prod_{i=1}^{n} A(\mathbf{x}(t-1), (\mathbf{x}(t))_i), \qquad (4.18)$$

where $A(\mathbf{x}(t-1), (\mathbf{x}(t))_i)$ denotes the probability that the ith element of $\mathbf{X}(t)$ will be $(\mathbf{x}(t))_i$ after one step of the network, given that the current state is $\mathbf{x}(t-1)$. Similarly, using the definition of DBNs it immediately follows that the probability of the same finite time series in a given DBN is (Friedman *et al.*, 1998)

$$P(\mathbf{x}(0), \mathbf{x}(1), \ldots, \mathbf{x}(T)) = \prod_{i=1}^{n} P(x_i(0)|\mathbf{pa}(X_i(0))) \prod_{t=1}^{T} \prod_{j=1}^{n} P(x_j(t)|\mathbf{pa}(X_j(t))). \qquad (4.19)$$

Equations (4.18) and (4.19) already resemble each other. The final step of the analysis consists of showing that the Boolean functions and the corresponding selection probabilities (resp. initial and transition BNs) can be defined such that any given DBN (resp. PBN) can be expressed as a PBN (resp. DBN). The technical details are given in Publication-VIII. This can be summarized as the following theorem. Independent PBNs $G(V, F)$ and binary-valued DBNs (B_0, B_1) whose initial and transition BNs B_0 and B_1 are assumed to have only within and between consecutive slice connections, respectively, can represent the same joint distribution over their common variables. Thus, the two models are statistically equivalent.

Interestingly, a similar statistical equivalence can also be established between dependent PBNs and discrete-valued DBNs. Without going into the details, the essential result can be stated as follows. Dependent PBNs $G(V, F)$ and discrete-valued DBNs (B_0, B_1) whose initial and transition BNs B_0 and B_1 are assumed to have only within and between consecutive slice connections, respectively, can represent the same joint distribution over their corresponding variables.

In Publication-VIII, we also showed the above types of relationships between more general DBNs and some extensions of PBNs, such as PBNs including so called random node perturbations (Shmulevich *et al.*, 2002*b*), and PBNs including additional random network changes (Zhou *et al.*, 2004). Because there are many PBNs that can represent the statistical behavior of a DBN, we also discussed the issue of constructing optimal PBNs. Note that although the relationships are presented in the binary setting, extensions to finer models (more discretisation levels) are also possible.

4.4.4 The Use of Relationships

Having shown the fundamental connection between PBNs and DBNs, the tools originally developed for PBNs become available in the context of DBNs, e.g., by using the detailed conversion of a DBN to a PBN. The same argument also applies the other way around. The main new tools now available for DBNs and PBNs are briefly reviewed below. Further discussion can be found from Publication-VIII.

From the DBN point of view, the tools for controlling the stationary behavior of PBNs, by means of interventions (Shmulevich *et al.*, 2002*b*), structural modifications of the network (Shmulevich *et al.*, 2002*c*), and optimal external control (Datta *et al.*, 2003, 2004), become available for DBNs. To our knowledge, no such methods have been introduced in the context of DBNs so far. The same applies to efficient learning schemes, strength of connection based subnetwork inference methods (Hashimoto *et al.*, 2004), as well as mappings between different networks (Dougherty and Shmulevich, 2003), in particular, projections onto subnetworks, which at the same time preserve consistency with the original probabilistic structure.

From the PBN point of view, both exact and approximate inference tools developed for BNs (see, e.g., Pearl, 1988; Cowell *et al.*, 1999) give a natural way of handling the missing values in PBNs which are often present in gene expression measurements. Well-developed learning methods of BNs can also be applied to PBNs (see, e.g., Heckerman, 1996; Friedman *et al.*, 1998; Pearl, 2003). Active learning methods can also be potentially very useful (Tong and Koller, 2000; Murphy, 2001; Tong and Koller, 2001; Pournara and Wernisch, 2004). However, it is probably even more important to be able to combine several different information sources. In Bayesian framework, a natural way of incorporating additional information into the model inference is via the prior distributions. For example, the use of so called location data and other sources of information for the construction of priors have been considered in (Hartemink *et al.*, 2002; Imoto *et al.*, 2004; Bernard and Hartemink, 2005). It is also tempting to speculate that biological knowledge of plausible regulatory rules (see the beginning of this chapter) could be incorporated into the prior distributions of the parameters Θ.

The main result introduced in Publication-VIII is the connection between the two model classes. The main benefit of such a connection is that tools developed in different modeling frameworks can be applied to both model classes.

Chapter 5

Conclusions

This chapter summarizes the computational methods introduced in the previous chapters. In the following concluding remarks we discuss some advantages as well as limitations of the proposed methods and point out some possible extensions for future work.

Sample Heterogeneity

Publication-VII serves as a proof-of-principle study by showing that sample and cell type specific expression values can be recovered from expression values of heterogeneous mixtures. The proposed methods have a potential to be highly useful, especially in experiments where the surrounding or infiltrating additional cell types cannot be successfully separated from the cells of interest, either manually or using LCM methods. Cancer studies serve as a typical example.

An inevitable limitation is that the proposed methods require several measurements from the same heterogeneous sample, with different mixing proportions of the underlying cell types. However, this inherent limitation is problem related, not a limitation of the proposed methods, since the expression values of all the underlying pure cell types simply cannot be estimated from a single expression profile. Moreover, if the mixing percentages of the underlying cell types are not known, then the combined estimation of both the expression values and the mixing fractions of the pure cell types further increases the sample size requirement. The same argument naturally applies to the case of in which the number of cell types is unknown (model selection).

In more challenging heterogeneous experiments involving complex tissues rather than cell lines, it would be worth studying more than just the linear mixing model shown in Equation (2.1). Other possible extensions include incorporating more assumptions, such as specific noise models, into the computation. It is also worth emphasizing robust estimation methods since real high-throughput data are prone to contain outliers or other nonidealities. The importance of robust computational procedures in discussed throughout the whole thesis, especially in Chapter 3.

Sample Asynchrony

A similar smoothing effect as in the case of heterogeneous mixtures is also present in many biological time series experiments. Computational methods for correcting the smoothing effect caused by sample asynchrony were described in Section 2.3. Examples shown in Section 2.3 and in Publication-I demonstrate the potential of the inversion methods. Description of the computational methods can also be thought of as guidelines for the design of time series experiments, such that the proposed preprocessing methods can be applied most easily and most efficiently.

Discrete approximation of a continuous process (see Equations (2.14) and (2.15)) results in an approximation error. Consequently, possible extensions include developing inversion methods that operate entirely in the continuous domain (see Bar-Joseph *et al.*, 2004, for an extension to that direction). Advanced, automated methods for estimating the underlying cell population distributions also deserve more research efforts.

Robust Time Series Analysis

The proposed robust spectrum estimation and robust periodicity detection methods were introduced in Chapter 3. The examples in Chapter 3 and more extensive performance evaluations in Publication-IV and Publication-IX clearly show the excellent robustness properties of the proposed methods. In addition, periodicity detection is also based on a test statistic that is distribution free. This is a highly useful property, e.g., for the simulation (Monte Carlo) based significance value computation.

Some possible straightforward extensions, such as windowing of the autocorrelation function and Chiu's modification of the g-statistic, were already discussed in Section 3.3.2. In general, the fields of robust spectrum estimation and robust periodicity detection have attracted little attention and hence there is room for several new ideas and methods.

Analysis of Boolean Networks as Models of Regulatory Networks

The general properties of certain Post function classes were studied in Section 4.1.2. The findings were interesting since, e.g., the studied Post function classes are one of the few known methodologies for preventing chaotic behavior in NK models. However, discrete network models are quite theoretical and their relations to real biological systems are not straightforward. One of the next major research steps to be undertaken is the ensemble approach described in (Kauffman, 2004). In other words, the goal is to compare the general properties of large network ensembles with the ones of real biological systems. That requires, e.g., more theoretical results for the discrete network models, and careful design of experiments so that the relevant general properties of real biological systems, such as the propagation of perturbations, can be revealed.

Efficient spectral methods for testing membership in the studied Post classes and in the class of forcing functions (and its variants) were also introduced. These methods are valuable for analysis of the properties of the NK models.

Inference of Predictive Models

Inference of predictive models was studied under the Best-Fit Extension paradigm in Section 4.3. Potentially useful extensions include development of efficient Best-Fit Extension methods for the studied Post function classes and the class of forcing functions. Note that the Best-Fit Extension Problem has been extensively studied for several other function classes in (Boros *et al.*, 1998).

Relationships between PBNs and DBNs

The last topic focused on two widely used stochastic modeling approaches, PBNs and DBNs. We believe that the established connections between the two modeling frameworks will increase researchers' awareness of the new analysis tools, both in the context of PBN and DBN, that now become available. PBNs and DBNs themselves provide several interesting future research problems. Of particular interest is the problem of model inference from experimental data. That includes, e.g., development of efficient methods for incorporating several different data sources into the inference process.

Bibliography

Agaian,S., Astola,J. and Egiazarian,K. (1995) *Binary Polynomial Transforms and Nonlinear Digital Filters.* Marcel Dekker Inc., New York.

Akutsu,T., Kuhara,S., Maruyama,O. and Miyano,S. (2003) Identification of genetic networks by strategic gene disruptions and gene overexpressions under a Boolean model. *Theoretical Computer Science,* **298** (1), 235–251.

Akutsu,T., Miyano,S. and Kuhara,S. (1999) Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. In *Proceedings of Pacific Symposium on Biocomputing (PSB 99)* vol. 4, pp. 17–28 World Scientific, Singapore.

Akutsu,T., Miyano,S. and Kuhara,S. (2000) Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics,* **16** (8), 727–734.

Alberts,B., Johnson,A., Lewis,J., Raff,M., Roberts,K. and Walter,P. (2002) *Molecular Biology of The Cell.* 4th edition, Gerland Publishing Inc.

Aldana,M. (2003) Boolean dynamics of networks with scale-free topology. *Physica D,* **185** (1), 45–66.

Aldana,M. and Cluzel,P. (2003) A natural class of robust networks. *Proceedings of the National Academy of Sciences of the USA,* **100** (15), 8710–8714.

Aldana-Gonzalez,M., Coppersmith,S. and Kadanoff,L.P. (2002) Boolean dynamics with random couplings. In *Perspectives and Problems in Nonlinear Science*, (Kaplan,E., Marsden,J. and Sreenivasan,K., eds),. Springer pp. 23–89.

Arnone,M.I. and Davidson,E.H. (1997) The hardwiring of development: organization and function of genomic regulatory systems. *Development,* **124** (10), 1851–1864.

Artis,M., Hoffmann,M., Nachane,D. and Toro,J. (2004). The detection of hidden periodicities: a comparison of alternative methods. Working paper ECO 2004/10 European University Institute. (Available on-line at `http://www.iue.it/PUB/ECO2004-10.pdf`).

Baldi,P. and Hatfield,G.W. (2002) *DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modeling.* Cambridge University Press.

Bar-Joseph,Z., Farkash,S., Gifford,D.K., Simon,I. and Rosenfeld,R. (2004) Deconvolving cell cycle expression data with complementary information. *Bioinformatics,* **20** (Suppl. 1), I23–I30.

Bar-Joseph,Z., Gerber,G., Gifford,D.K., Jaakkola,T.S. and Simon,I. (2002) A new approach to analyzing gene expression time series data. In *Proceedings of The Sixth Annual International Conference on Research in Computational Molecular Biology (RECOMB)* pp. 39–48 ACM Press, New York.

Beal,M.J., Falciani,F., Ghahramani,Z., Rangel,C. and Wild,D.L. (2005) A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics,* **21** (3), 349–356.

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

Bernard,A. and Hartemink,A. (2005) Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data. In *Proceedings of Pacific Symposium on Biocomputing (PSB 05)* vol. 10, pp. 459–470 World Scientific, Singapore.

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

Boros,E., Ibaraki,T. and Makino,K. (1998) Error-free and Best-Fit Extensions of partially defined Boolean functions. *Information and Computation,* **140** (2), 254–283.

Boye,E., Løbner-Olesen,A. and Skarstad,K. (2000) Limiting DNA replication to once and only once. *EMBO Reports,* **1** (6), 479–483.

Braga-Neto,U.M. and Dougherty,E.R. (2004) Bolstered error estimation. *Pattern Recognition,* **37** (6), 1267–1281.

Breeden,L.L. (2003) Periodic transcription: a cycle within a cycle. *Current Biology,* **13** (1), R31–R38.

Brockwell,P.J. and Davis,R.A. (1991) *Time Series: Theory and Methods.* 2nd edition, Springer-Verlag, New York.

Chen,T., He,H.L. and Church,G.M. (1999) Modeling gene expression with differential equations. In *Proceedings of Pacific Symposium on Biocomputing (PSB 99)* vol. 4, pp. 29–40 World Scientific, Singapore.

Chen,Y., Dougherty,E.R. and Bittner,M.L. (1997) Ratio-based decisions and the quantitative analysis of cDNA microarray images. *Journal of Biomedical Optics,* **2** (4), 364–374.

Chiu,S.T. (1989) Detecting periodic components in a white Gaussian time series. *Journal of the Royal Statistical Society: Series B,* **51** (2), 249–259.

Cho,H. and Lee,J.K. (2004) Bayesian hierarchical error model for analysis of gene expression data. *Bioinformatics,* **20** (13), 2016–2025.

Cho,R.J., Campbell,M.J., Winzeler,E.A., Steinmetz,L., Conway,A., Wodicka,L., Wolfsberg,T.G., Gabrielian,A.E., Landsman,D., Lockhart,D.J. and Davis,R.W. (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell,* **2** (1), 65–73.

Cleveland,W.S. (1979) Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association,* **74** (368), 829–836.

Correa,A., Lewis,Z.A., Greene,A.V., March,I.J., Gomer,R.H. and Bell-Pedersen,D. (2003) Multiple oscillators regulate circadian gene expression in *Neurospora. Proceedings of the National Academy of Sciences of the USA,* **100** (23), 13597–13602.

Cowell,R.G., Dawid,A.P., Lauritzen,S.L. and Spiegelhalter,D.J. (1999) *Probabilistic Networks and Expert Systems.* Statistics for Engineering and Information Science, Springer, New York.

Darwin,C. (1859) *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life.* 1st edition, John Murray, London.

Datta,A., Choudhary,A., Bittner,M.L. and Dougherty,E.R. (2003) External control in Markovian genetic regulatory networks. *Machine Learning,* **52** (1–2), 169–181.

Datta,A., Choudhary,A., Bittner,M.L. and Dougherty,E.R. (2004) External control in Markovian genetic regulatory networks: the imperfect information case. *Bioinformatics,* **20** (6), 924–930.

Davidson,E.H., Rast,J.P., Oliveri,P., Ransick,A., Calestani,C., Yuh,C.H., Minokawa,T., Amore,G., Hinman,V., Arenas-Mena,C., Otim,O., Brown,C.T., Livi,C.B., Lee,P.Y., Revilla,R., Rust,A.G., Pan,Z.j., Schilstra,M.J., Clarke,P.J.C., Arnone,M.I., Rowen,L., Cameron,R.A., McClay,D.R., Hood,L. and Bolouri,H. (2002) A genomic regulatory network for development. *Science,* **295** (5560), 1669–1678.

de Hoon,M.J.L., Imoto,S., Kobayashi,K., Ogasawara,N. and Miyano,S. (2003) Inferring gene regulatory networks from time-ordered gene expression data of *Bacillus subtilis* using differential equations. In *Proceedings of Pacific Symposium on Biocomputing (PSB 03)* vol. 8, pp. 17–28 World Scientific, Singapore.

de Jong,H. (2002) Modeling and simulation of genetic regulatory systems: a literature review. *Journal of Computational Biology,* **9** (1), 67–103.

de Lichtenberg,U., Jensen,L.J., Fausbøll,A., Jensen,T.S., Bork,P. and Brunak,S. (2005) Comparison of computational methods for the identification of cell cycle regulated genes. *Bioinformatics,* **21** (7), 1164–1171.

Derrida,B. and Pomeau,Y. (1986) Random networks of automata: a simple annealed approximation. *Europhysics Letters,* **1** (2), 45–49.

Derrida,B. and Stauffer,D. (1986) Phase transitions in two-dimensional Kauffman cellular automata. *Europhysics Letters,* **2** (10), 739–745.

Devroye,L., Györfi,L. and Lugosi,G. (1996) *A Probabilistic Theory of Pattern Recognition.* Springer, New York.

Dougherty,E.R. (1999) *Random Processes for Image and Signal Processing.* SPIE Press/IEEE Press, Bellingham.

Dougherty,E.R. and Shmulevich,I. (2003) Mappings between probabilistic Boolean networks. *Signal Processing,* **83** (4), 745–761.

Dougherty,E.R., Shmulevich,I., Chen,J. and Wang,Z.J., eds (2005) *Genomic Signal Processing and Statistics.* EURASIP Book Series on SP&C, Volume 2, Hindawi.

Dror,R.O., Murnick,J.G., Rinaldi,N.J., Marinescu,V.D., Rifkin,R.M. and Young,R.A. (2003) Bayesian estimation of transcript levels using a general model of array measurement noise. *Journal of Computational Biology,* **10** (3–4), 433–452.

Dudoit,S., Shaffer,J.P. and Boldrick,J.C. (2003) Multiple hypothesis testing in microarray experiments. *Statistical Science,* **18** (1), 71–103.

Duggan,D.J., Bittner,M., Chen,Y., Meltzer,P. and Trent,J.M. (1999) Expression profiling using cDNA microarrays. *Nature Genetics,* **21** (Suppl. 1), 10–14.

Durbin,B.P. and Rocke,D.M. (2004) Variance-stabilizing transformations for two-color microarrays. *Bioinformatics,* **20** (5), 660–667.

Efron,B. and Tibshirani,R.J. (1993) *An Introduction to the Bootstrap.* 1st edition,, Chapman & Hall, New York.

Emmert-Buck,M.R., Bonner,R.F., Smith,P.D., Chuaqui,R.F., Zhuang,Z., Goldstein,S.R., Weiss,R.A. and Liotta,L.A. (1996) Laser capture microdissection. *Science,* **274**, 998–1001.

Fisher,R.A. (1929) Tests of significance in harmonic analysis. *Proceedings of the Royal Society of London Series A,* **125**, 54–59.

Fox,J.J. and Hill,C.C. (2001) From topology to dynamics in biochemical networks. *Chaos,* **11** (4), 809–815.

Friedman,N. (2004) Inferring cellular networks using probabilistic graphical models. *Science,* **303**, 799–805.

Friedman,N., Linial,M., Nachman,I. and Pe'er,D. (2000) Using Bayesian networks to analyze expression data. *Journal of Computational Biology,* **7** (3–4), 601–620.

Friedman,N., Murphy,K. and Russell,S. (1998) Learning the structure of dynamic probabilistic networks. In *Proceedings of Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI)* pp. 139–147 Morgan Kaufmann.

Fuller,G.N., Rhee,C.H., Hess,K.R., Caskey,L.S., Wang,R., Bruner,J.M., Yung,W.K.A. and Zhang,W. (1999) Reactivation of insulin-like growth factor binding protein 2 expression in glioblastoma multiforme: a revelation by parallel gene expression profiling. *Cancer Research,* **59**, 4228–4232.

Gat-Viks,I. and Shamir,R. (2003) Chain functions and scoring functions in genetic networks. *Bioinformatics,* **19** (Suppl. 1), i108–i117.

Good,P. (2000) *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypothesis.* 1st edition, Springer, New York.

Gottardo,R., Raftery,A.E., Yeung,K.Y. and Bumgarner,R.E. (2003). Robust estimation of cDNA microarray intensities with replicates. Technical report 438 Department of Statistics, University of Washington.

Hampel,F.R., Ronchetti,E.M., Rousseeuw,P.J. and Stahel,W.A. (1985) *Robust Statistics: The Approach Based on Influence Function.* 1st edition, John Wiley.

Harris,S.E., Sawhill,B.K., Wuensche,A. and Kauffman,S.A. (2002) A model of transcriptional regulatory networks based on biases in the observed regulation rules. *Complexity,* **7** (4), 23–40.

Hartemink,A. (2001). *Principled Computational Methods for the Validation and Discovery of Genetic Regulatory Networks.* Ph.D. Thesis Massachusetts Institute of Technology.

Hartemink,A., Gifford,D., Jaakkola,T. and Young,R. (2001) Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. In *Proceedings of Pacific Symposium on Biocomputing (PSB 01)* vol. 6, pp. 422–433 World Scientific, Singapore.

Hartemink,A., Gifford,D., Jaakkola,T. and Young,R. (2002) Combining location and expression data for principled discovery of genetic regulatory network models. In *Proceedings of Pacific Symposium on Biocomputing (PSB 02)* vol. 7, pp. 437–449 World Scientific, Singapore.

Hartemink,A.J., Gifford,D.K., Jaakkola,T.S. and Young,R.A. (2001) Maximum likelihood estimation of optimal scaling factors for expression array normalization. In *Microarrays: Optical Technologies and Informatics, 20–26 January, San Jose, CA, USA*, (Bittner,M., Chen,Y., Dorsel,A. and Dougherty,E., eds), Proceedings of SPIE, Vol. 4266. SPIE pp. 132–140.

Hashimoto,R.F., Dougherty,E.R., Brun,M., Zhou,Z.Z., Bittner,M.L. and Trent,J.M. (2003) Efficient selection of feature sets possessing high coefficients of determination based on incremental determinations. *Signal Processing,* **83** (4), 695–712.

Hashimoto,R.F., Kim,S., Shmulevich,I., Zhang,W., Bittner,M.L. and Dougherty,E.R. (2004) Growing genetic regulatory networks from seed genes. *Bioinformatics,* **20** (8), 1241–1247.

Hastie,T., Tibshirani,R. and Friedman,J. (2001) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* 1st edition, Springer-Verlag.

Hautaniemi,S., Edgren,H., Vesanen,P., Wolf,M., Järvinen,A.K., Yli-Harja,O., Astola,J., Kallioniemi,O. and Monni,O. (2003) A novel strategy for microarray quality control using Bayesian networks. *Bioinformatics,* **19** (16), 2031–2038.

Haykin,S. (1996) *Adaptive Filter Theory.* 3rd edition, Prentice-Hall, Englewood Cliffs, NJ.

Heckerman,D. (1996). A tutorial on learning with Bayesian networks. Technical report msr-tr-95-06 Microsoft Corporation, Redmond, USA.

Hood,L. and Galas,D. (2003) The digital code of DNA. *Nature,* **421** (6921), 444–448.

Huang,S. (1999) Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery. *Journal of Molecular Medicine,* **77**, 469–480.

Huang,X. and Pan,W. (2002) Comparing three methods for variance estimation with duplicated high density oligonucleotide arrays. *Functional & Integrative Genomics,* **2** (3), 126–133.

Huber,P.J. (1981) *Robust Statistics.* Wiley, New York.

Huber,W., von Heydebreck,A., Sültmann,H., Poustka,A. and Vingron,M. (2002) Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics,* **18** (Suppl. 1), S96–S104.

Husmeier,D. (2003) Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. *Bioinformatics,* **19** (17), 2271–2282.

Ideker,T., Thorsson,V., Siegel,A.F. and Hood,L.E. (2000*a*) Testing for differentially-expressed genes by maximum-likelihood analysis of microarray data. *Journal of Computational Biology,* **7** (6), 805–817.

Ideker,T.E., Thorsson,V. and Karp,R.M. (2000*b*) Discovery of regulatory interactions through perturbation: inference and experimental design. In *Proceedings of Pacific Symposium on Biocomputing (PSB 00)* vol. 5, p. 302–313 World Scientific, Singapore.

Imoto,S., Higuchi,T., Goto,T., Tashiro,K., Kuhara,S. and Miyano,S. (2004) Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. *Journal of Bioinformatics and Computational Biology,* **2** (1), 77–98.

Imoto,S., Kim,S., Goto,T., Miyano,S., Aburatani,S., Tashiro,K. and Kuhara,S. (2003) Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. *Journal of Bioinformatics and Computational Biology,* **1** (2), 231–252.

Johansson,D., Lindgren,P. and Berglund,A. (2003) A multivariate approach applied to microarray data for identification of genes with cell cycle-coupled transcription. *Bioinformatics,* **19** (4), 467–473.

Johnson,R.A. and Wichern,D.W. (1998) *Applied Multivariate Statistical Analysis.* 4th edition,, Prentice-Hall, Englewood Cliffs, New Jersey.

Just,W., Shmulevich,I. and Konvalina,J. (2004) The number and probability of canalizing functions. *Physica D,* **197** (3–4), 211–221.

Karp,R.M., Stoughton,R. and Yeung,K.Y. (1999) Algorithms for choosing differential gene expression experiments. In *Proceedings of Third Annual International Conference on Computational Molecular Biology (RECOMB'99)* pp. 208–217 ACM Press.

Kauffman,S., Peterson,C., Samuelsson,B. and Troein,C. (2003) Random Boolean network models and the yeast transcriptional network. *Proceedings of the National Academy of Sciences of the USA,* **100** (25), 14796–14799.

Kauffman,S.A. (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology,* **22** (3), 437–467.

Kauffman,S.A. (1993) *The Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press, New York.

Kauffman,S.A. (2004) The ensemble approach to understand genetic regulatory networks. *Physica A,* **340** (4), 733–740.

Kerr,M.K., Martin,M. and Churchill,G.A. (2000) Analysis of variance for gene expression microarray data. *Journal of Computational Biology,* **7** (6), 819–837.

Kesseli,J., Rämö,P. and Yli-Harja,O. (2005) On spectral techniques in analysis of Boolean networks. *Physica D,* **206** (1–2), 49–61.

Kim,S., Dougherty,E.R., Bittner,M.L., Chen,Y., Sivakumar,K., Meltzer,P. and Trent,J.M. (2000) General nonlinear framework for the analysis of gene interaction via multivariate expression arrays. *Journal of Biomedical Optics,* **5** (4), 411–424.

Kim,S., Li,H., Dougherty,E.R., Cao,N., Chen,Y., Bittner,M. and Suh,E.B. (2002) Can Markov chain models mimic biological regulation? *Journal of Biological Systems,* **10** (4), 431–445.

Kleiner,R., Martin,R.D. and Thomson,D.J. (1979) Robust estimation of power spectra. *Journal of the Royal Statistical Society: Series B,* **41** (3), 313–351.

Levchenkov,V.S. (2000) Boolean equations and Post classes. *Doklady Akademii Nauk,* **373** (3), 316–319. (in Russian).

Liang,S., Fuhrman,S. and Somogyi,R. (1998) REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In *Proceedings of Pacific Symposium on Biocomputing (PSB 98)* vol. 3, pp. 18–29 World Scientific, Singapore.

Liu,D., Umbach,D.M., Peddada,S.D., Li,L., Crockett,P.W. and Weinberg,C.R. (2004) A random-periods model for expression of cell-cycle genes. *Proceedings of the National Academy of Sciences of the USA,* **101** (19), 7240–7245.

Lodish,H., Berk,A., Zipursky,L.S., Matsudaira,P., Baltimore,D. and Darnell,J.E.L. (1999) *Molecular Cell Biology.* 4th edition, W. H. Freeman & Co, New York.

Lu,P., Nakorchevskiy,A. and Marcotte,E.M. (2003) Expression deconvolution: a reinterpretation of DNA microarray data reveals dynamic changes in cell populations. *Proceedings of the National Academy of Sciences of the USA,* **100** (18), 10370–10375.

Lu,X., Zhang,W., Qin,Z.S., Kwast,K.E. and Liu,J.S. (2004) Statistical resynchronization and Bayesian detection of periodically expressed genes. *Nucleic Acids Research,* **32** (2), 447–455.

Luan,Y. and Li,H. (2004) Model-based methods for identifying periodically expressed genes based on time course microarray gene expression data. *Bioinformatics,* **20** (3), 332–339.

Martin,R.D. and Thomson,D.J. (1982) Robust-resistant sprectrum estimation. *Proceedings of the IEEE,* **70**, 1097–1115.

Maxam,A. and Gilbert,W. (1977) A new method for sequencing DNA. *Proceedings of the National Academy of Sciences of the USA,* **74** (2), 560–564.

McAdams,H.H. and Arkin,A. (1997) Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences of the USA,* **4** (94), 814–819.

McAdams,H.H. and Arkin,A. (1999) It's a noisy business! Genetic regulation at the nanomolar scale. *Trends in Genetics,* **15** (2), 65–69.

Muchnik,A.A. and Gindikin,S.G. (1962) On the completeness of systems of unreliable elements which realize functions of the algebra of logic. *Doklady Akademii Nauk SSSR,* **144** (5), 1007–1010. (In Russian).

Murphy,K. (2001). Active learning of causal Bayes net structure. Technical report University of California, Berkeley.

Murphy,K. and Mian,S. (1999). Modelling gene expression data using dynamic Bayesian networks. Technical report University of California, Berkeley.

Murphy,K.P. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning.* Ph.D. Thesis University of California, Berkeley.

Niemistö,A., Aho,T., Thesleff,H., Tiainen,M., Marjanen,K., Linne,M.L. and Yli-Harja,O. (2003) Estimation of population effects in synchronized budding yeast experiments. In *Image Processing: Algorithms and Systems II, Proceedings of SPIE* vol. 5014, pp. 448–459 SPIE.

Niemistö,A., Nykter,M., Aho,T., Jalovaara,H., Marjanen,K., Ahdesmäki,M., Ruusuvuori,P., Tiainen,M., Linne,M.L. and Yli-Harja,O. (2004) Distribution estimation of synchronized budding yeast population. In *Proceedings of the Winter International Symposium on Information and Communication Technologies 2004 (WISICT'04), Cancun, Mexico, January 5-8* pp. 243–248.

Oosawa,C. and Savageau,M.A. (2002) Effects of alternative connectivity on behavior of randomly constructed Boolean networks. *Physica D,* **170** (2), 143–161.

Pearl,J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Representation and Reasoning, Morgan Kaufmann.

Pearl,J. (2003) *Learning Bayesian Networks.* Prentice Hall.

Pe'er,D., Regev,A., Elidan,G. and Friedman,N. (2001) Inferring subnetworks from perturbed expression profiles. *Bioinformatics,* **17** (Suppl. 1), 215S–224S.

Perrin,B.E., Ralaivola,L., Mazurie,A., Bottani,S., Mallet,J. and d'Alché-Buc,F. (2003) Gene networks inference using dynamic Bayesian networks. *Bioinformatics,* **19** (Suppl. 2), ii138–ii148.

Post,E.L. (1921) Introduction to a general theory of elementary propositions. *American Journal of Mathematics,* **43**, 163–185.

Post,E.L. (1941) *The Two-Valued Iterative Systems of Mathematical Logic.* Annals of Mathematics Studies, Number 5, Princeton University Press, Princeton.

Pournara,I. and Wernisch,L. (2004) Reconstruction of gene networks using Bayesian learning and manipulation experiments. *Bioinformatics,* **20** (17), 2934–2942.

Priestley,M.B. (1981) *Spectral Analysis and Time Series, Volume 1.* Academic Press, London.

Quackenbush,J. (2002) Microarray data normalization and transformation. *Nature Genetics,* **32** (Suppl.), 496–501.

Randles,R.H. and Wolfe,D.A. (1979) *Introduction to the Theory of Nonparametric Statistics.* Wiley.

Rangel,C., Angus,J., Ghahramani,Z., Lioumi,M., Sotheran,E., Gaiba,A., Wild,D.L. and Falciani,F. (2004) Modeling T-cell activation using gene expression profiling and state-space models. *Bioinformatics,* **20** (8), 1361–1372.

Rocke,D.M. and Durbin,B. (2001) A model for measurement error for gene expression arrays. *Journal of Computational Biology,* **8** (6), 557–569.

Rocke,D.M. and Durbin,B. (2003) Approximate variance-stabilizing transformations for gene-expression microarray data. *Bioinformatics,* **19** (8), 966–972.

Rousseeuw,P.J. and Leroy,A.M. (1987) *Robust Regression and Outlier Detection.* 1st edition, John Wiley.

Rustici,G., Mata,J., Kivinen,K., Lió,P., Penkett,C.J., Burns,G., Hayles,J., Brazma,A., Nurse,P. and Bähler,J. (2004) Periodic gene expression program of the fission yeast cell cycle. *Nature Genetics,* **36** (8), 809–817.

Sakamoto,E. and Iba,H. (2001) Inferring a system of differential equations for a gene regulatory network by using genetic programming. In *Proceedings of Congress on Evolutionary Computation '01* pp. 720–726 IEEE Press.

Sanger,F., Nicklen,S. and Coulson,A. (1977) DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the USA,* **74** (12), 5463–5467.

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

Sherr,C.J. (1996) Cancer cell cycles. *Science,* **274** (5293), 1672–1677.

Shmulevich,I., Dougherty,E.R., Kim,S. and Zhang,W. (2002*a*) Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics,* **18** (2), 261–274.

Shmulevich,I., Dougherty,E.R. and Zhang,W. (2002*b*) Gene perturbation and intervention in probabilistic Boolean networks. *Bioinformatics,* **18** (10), 1319–1331.

Shmulevich,I., Dougherty,E.R. and Zhang,W. (2002*c*) Control of stationary behavior in probabilistic Boolean networks by means of structural intervention. *Journal of Biological Systems,* **10** (4), 431–445.

Shmulevich,I. and Kauffman,S.A. (2004) Activities and sensitivities in Boolean network models. *Physical Review Letters,* **93** (4), 048701(1–4).

Shmulevich,I., Saarinen,A., Yli-Harja,O. and Astola,J. (2002) Inference of genetic regulatory networks under the Best-Fit Extension paradigm. In *Computational And Statistical Approaches To Genomics*, (Zhang,W. and Shmulevich,I., eds),. Kluwer Academic Publishers, Boston.

Silverman,B.W. (1986) *Density Estimation for Statistics and Data Analysis.* Chapman and Hall.

Smith,V.A., Jarvis,E.D. and Hartemink,A.J. (2002) Evaluating functional network inference using simulations of complex biological systems. *Bioinformatics,* **18** (Suppl. 1), S216–S224.

Smyth,G.K., Yang,Y.H. and Speed,T. (2003) Statistical issues in microarray data analysis. In *Functional Genomics: Methods and Protocols*, (Brownstein,M.J. and Khodursky,A.B., eds), Methods in Molecular Biology Volume 224. Humana Press, Totowa, NJ pp. 111–136.

Spangl,B. and Dutter,R. (2005) On robust estimation of power spectra. *Austrian Journal of Statistics,* **34** (2), 199–210.

Speed,T. (2003) *Statistical Analysis of Gene Expression Microarray Data.* Chapman & Hall/CRC.

Spellman,P.T., Sherlock,G., Zhang,M.Q., Iyer,V.R., Anders,K., Eisen,M.B., Brown,P.O., Botstein,D. and Futcher,B. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell,* **9** (12), 3273–3297.

Stauffer,D. (1987) On forcing functions in Kauffman random Boolean networks. *Journal of Statistical Physics,* **46** (3–4), 789–794.

Stoica,P. and Moses,R.L. (1997) *Introduction to Spectral Analysis.* Prentice Hall, New Jersey.

Stone,M. (1974) Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B,* **36** (2), 111–147.

Stuart,R.O., Wachsman,W., Berry,C.C., Wang-Rodriguez,J., Wasserman,L., Klacansky,I., Masys,D., Arden,K., Goodison,S., McClelland,M., Wang,Y., Sawyers,A., Kalcheva,I., Tarin,D. and Mercola,D. (2004) *In silico* dissection of cell-type-associated patterns of gene expression in prostate cancer. *Proceedings of the National Academy of Sciences of the USA,* **101** (2), 615–620.

Tabus,I. and Astola,J. (2001) On the use of MDL principle in gene expression prediction. *Journal of Applied Signal Processing,* **4**, 297–303.

Tabus,I., Giurcaneanu,C.D. and Astola,J. (2004) Genetic networks inferred from time series of gene expression data. In *Proceedings of First International Symposium on Control, Communications and Signal Processing, ISCCSP 2004, Hammamet, Tunisia, 21-24 March 2004* pp. 755–758.

Tabus,I., Rissanen,J. and Astola,J. (2002) Normalized maximum likelihood models for Boolean regression with application to prediction and classification in genomics. In *Computational And Statistical Approaches To Genomics*, (Zhang,W. and Shmulevich,I., eds),. Kluwer Academic Publishers, Boston.

Tatum,L.G. and Hurvich,C.M. (1993) High breakdown methods of time series analysis. *Journal of the Royal Statistical Society: Series B,* **55** (4), 881–896.

The Genome International Sequencing Consortium (2001) Initial sequencing and analysis of the human genome. *Nature,* **409** (6822), 860–921.

The MathWorks, Inc. (2005) *Statistics Toolbox. User's Guide.* 5th edition,, The MathWorks, Inc. (Available on-line at `http://www.mathworks.com`).

Tong,S. and Koller,D. (2000) Active learning for parameter estimation in Bayesian networks. In *Proceedings of Neural Information Processing Systems* pp. 647–653.

Tong,S. and Koller,D. (2001) Active learning for structure in Bayesian networks. In *Proceedings of International Joint Conference on Artificial Intelligence* pp. 863–869 Morgan Kaufmann.

Troyanskaya,O., Cantor,M., Sherlock,G., Brown,P., Hastie,T., Tibshirani,R., Botstein,D. and Altman,R.B. (2001) Missing value estimation methods for DNA microarrays. *Bioinformatics,* **17** (6), 520–525.

Tseng,G.C., Oh,M.K., Rohlin,L., Liao,J.C. and Wong,W.H. (2001) Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects. *Nucleic Acids Research,* **29** (12), 2549–2557.

Tyson,J.J. (2002) Biochemical oscillations. In *Computational Cell Biology: An Introductory Text on Computer Modeling in Molecular and Cell Biology*, (Fall,C., Marland,E., Wagner,J. and Tyson,J., eds),. Springer-Verlag, New York pp. 230–260.

van Berlo,R.J.P., van Someren,E.P. and Reinders,M.J. (2003) Studying the conditions for learning dynamic Bayesian networks to discover genetic regulatory networks. *Simulation,* **79** (12), 689–702.

van Someren,E.P., Wessels,L.F.A. and Reinders,M.J.T. (2000) Linear modeling of genetic networks from experimental data. In *Proceedings of Eight International conference on Intelligent Systems for Molecular Biology (ISMB 2000), San Diego, August 19-23* pp. 355–366.

Venet,D., Pecasse,F., Maenhaut,C. and Bersini,H. (2001) Separation of samples into their constituents using gene expression data. *Bioinformatics,* **17** (Suppl. 1), S279–S287.

Venter,J.C., Adams,M.D., Myers,E.W., Li,P.W., Mural,R.J., Sutton,G.G., Smith,H.O., Yandell,M., Evans,C.A., Holt,R.A. and *et al.* (2001) The sequence of the human genome. *Science,* **291** (5507), 1304–1351.

Watson,J.D. and Crick,F.H. (1953) Molecular structure of nucleic acids. *Nature,* **171** (4356), 737–738.

Weaver,D.C., Workman,C.T. and Stormo,G.D. (1999) Modeling regulatory networks with weight matrices. In *Proceedings of Pacific Symposium on Biocomputing (PSB 99)* vol. 4, pp. 112–123 World Scientific, Singapore.

Whitfield,M.L., Sherlock,G., Saldanha,A.J., Murray,J.I., Ball,C.A., Alexander,K.E., Matese,J.C., Perou,C.M., Hurt,M.M., Brown,P.O. and Botstein,D. (2002) Identification of genes periodically expressed in the human cell cycle and their expression in tumors. *Molecular Biology of the Cell,* **13** (6), 1977–2000.

Wichert,S., Fokianos,K. and Strimmer,K. (2004) Identifying periodically expressed transcripts in microarray time series data. *Bioinformatics,* **20** (1), 5–20.

Yablonsky,S.V., Gavrilov,G.P. and Kudryavtsev,V.B. (1966) *Functions of the Algebra of Logic and Post Classes.* Nauka, Moscow, Russia. (In Russian).

Yang,Y.H., Dudoit,S., Luu,P., Lin,D.M., Peng,V., Ngai,J. and Speed,T.P. (2002) Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research,* **30** (4), e15.

Yoo,C., Thorsson,V. and Cooper,G.F. (2002) Discovery of causal relationships in a gene-regulation pathway from a mixture of experimental and observational DNA microarray data. In *Proceedings of Pacific Symposium on Biocomputing (PSB 02)* vol. 7, pp. 498–509 World Scientific, Singapore.

Yu,J., Smith,V.A., Wang,P.P., Hartemink,A.J. and Jarvis,E.D. (2002) Using Bayesian network inference algorithms to recover molecular genetic regulatory networks. In *Proceedings of International Conference on Systems Biology (ICSB02)* vol. 3,.

Yu,J., Smith,V.A., Wang,P.P., Hartemink,A.J. and Jarvis,E.D. (2004) Advances to Bayesian network inference for generating causal networks from observational biological data. *Bioinformatics,* **20** (18), 3594–3603.

Yuh,C.H., Bolouri,H. and Davidson,E.H. (1998) Genomic Cis-regulatory logic: experimental and computational analysis of a sea urchin gene. *Science,* **279** (5358), 1896–1902.

Zhang,W., Shmulevich,I. and Astola,J. (2004) *Microarray Quality Control.* John Wiley & Sons, New Jersey.

Zhao,L.P., Prentice,R. and Breeden,L. (2001) Statistical modeling of large microarray data sets to identify stimulus-response profiles. *Proceedings of the National Academy of Sciences of the USA,* **98** (10), 5631–5636.

Zhou,X., Wang,X. and Dougherty,E.R. (2003*a*) Missing-value estimation using linear and non-linear regression with Bayesian gene selection. *Bioinformatics,* **19** (17), 2302–2307.

Zhou,X., Wang,X. and Dougherty,E.R. (2003*b*) Construction of genomic networks using mutual-information clustering and reversible-jump Markov-chain-Monte-Carlo predictor design. *Signal Processing,* **83** (4), 745–761.

Zhou,X., Wang,X., Pal,R., Ivanov,I., Bittner,M. and Dougherty,E.R. (2004) A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks. *Bioinformatics,* **20** (17), 2918–2927.

Publications

Publication-I

Lähdesmäki, H., Huttunen, H., Aho, T., Linne, M.-L., Niemi, J., Kesseli, J., Pearson, R. and Yli-Harja, O. (2003) Estimation and inversion of the effects of cell population asynchrony in gene expression time-series. *Signal Processing*, Vol. 83, No. 4, pp. 835–858.

Publication-II

Lähdesmäki, H., Shmulevich, I. and Yli-Harja, O. (2003) On learning gene regulatory networks under the Boolean network model. *Machine Learning*, Vol. 52, No. 1–2, pp. 147–167.

Publication-III

Shmulevich, I., Lähdesmäki, H., Dougherty, E.R., Astola, J. and Zhang, W. (2003) The role of certain Post classes in Boolean network models of genetic networks. *Proceedings of the National Academy of Sciences of the USA*, Vol. 100, No. 19, pp. 10734–10739.

Publication-IV

Pearson, R.K., Lähdesmäki, H., Huttunen, H. and Yli-Harja, O. (2003) Detecting periodicity in nonideal datasets. In *SIAM International Conference on Data Mining 2003*, Cathedral Hill Hotel, San Francisco, CA, May 1-3.

Publication-V

Shmulevich, I. Lähdesmäki, H. and Egiazarian, K. (2004) Spectral methods for testing membership in certain Post classes and the class of forcing functions. *IEEE Signal Processing Letters*, Vol. 11, No. 2, pp. 289–292.

Publication-VI

Lähdesmäki, H., Shmulevich, I., Yli-Harja, O. and Astola, J. (to appear) Inference of genetic regulatory networks via Best-Fit extensions. To appear in W. Zhang and I. Shmulevich (Eds.) *Computational And Statistical Approaches To Genomics (2nd ed.)*, Boston: Kluwer Academic Publishers.

Publication-VII

Lähdesmäki, H., Shmulevich, I., Dunmire, V., Yli-Harja, O. and Zhang, W. (2005) *In silico* microdissection of microarray data from heterogeneous cell populations. *BMC Bioinformatics*, 6:54.

Publication-VIII

Lähdesmäki, H., Hautaniemi, S., Shmulevich, I. and Yli-Harja, O. (to appear) Relationships between probabilistic Boolean networks and dynamic Bayesian networks as models of gene regulatory networks. To appear in *Signal Processing*.

Publication-IX

Ahdesmäki, M.,[†] Lähdesmäki, H.,[†] Pearson, R., Huttunen, H. and Yli-Harja, O. (2005) Robust detection of periodic time series measured from biological systems. *BMC Bioinformatics*, 6:117.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O. Box 527
FIN-33101 Tampere, Finland