Helsinki University of Technology
Dissertations in Computer and Information Science
Espoo 2005 Report D11

EXPLORATORY CLUSTER ANALYSIS OF GENOMIC HIGH-THROUGHPUT DATA SETS AND THEIR DEPENDENCIES

Janne Nikkilä

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Computer Science and Engineering for public examination and debate in Auditorium T2 at Helsinki University of Technology (Espoo, Finland) on the 1st of December, 2005, at 12 o'clock noon.

Helsinki University of Technology
Department of Computer Science and Engineering
Laboratory of Computer and Information Science

Distribution:

Helsinki University of Technology
Laboratory of Computer and Information Science
P.O.Box 5400
FIN-02015 TKK
Finland

Tel. +358-9-451 3272
Fax +358-9-451 3277
http://www.cis.tkk.fi/

Available in pdf format at http://lib.hut.fi/Diss/2005/isbn9512279096/

ISBN 951-22-7908-8 (printed version)
ISBN 951-22-7909-6 (electronic version)
ISSN 1459-7020

Otamedia Oy
Espoo 2005

Nikkilä, J. (2005): **Exploratory Cluster Analysis of Genomic High-Throughput Data Sets and Their Dependencies**. Doctoral thesis, Helsinki University of Technology, Dissertations in Computer and Information Science, Report D11, Espoo, Finland.

Keywords: bioinformatics, clustering, data integration, dependency modeling, exploratory data analysis, gene expression, human, learning metrics, mouse, self-organizing map, systems biology, transcription, yeast.

ABSTRACT

This thesis studies exploratory cluster analysis of genomic high-throughput data sets and their interdependencies. In modern biology, new high-throughput measurements generate numerical data simultaneously from thousands of molecules in the cell. This enables a new perspective to biology, which is called systems biology. The discipline developing methods for the analysis of the systems biology data is called bioinformatics. The work in this thesis contributes mainly to bioinformatics, but the approaches presented are general purpose machine learning methods and can be applied in many problem areas.

A main problem in analyzing genomic high-throughput data is that the potentially useful new findings are hidden in a huge data mass. They need to be extracted and visualized to the analyst as overviews.

This thesis introduces new exploratory cluster analysis methods for extracting and visualizing findings of high-throughput data. Three kinds of methods are presented to solve progressively better-focused problems. First, visualizations and clusterings using the self-organizing map are applied to genomic data sets. Second, the recently developed methods for improving the visualization and clustering of a data set with auxiliary data are applied. Third, new methods for exploring the dependency between data sets are developed and applied. The new methods are based on maximizing the Bayes factor between the model of independence and the model of dependence for finite data.

The methods outperform their alternatives in numerical comparisons. In applications they proved capable of confirming known biological findings, which validates the methods, and also generated new hypotheses. The applications included exploration of yeast gene expression data, yeast gene expression data in a new metric learned with auxiliary data, the regulation of yeast gene expression by transcription factors, and the dependencies between human and mouse gene expression.

Contents

PREFACE

This work has been carried out in Neural Networks Research Centre of Laboratory of Computer and Information Science, Helsinki University of Technology. The main sources of funding have been the Graduate School in Computational Methods of Information Technology (ComMIT) and project funding from the Academy of Finland and the National Technology Agency of Finland. The project numbers are listed in the acknowledgements of the individual publications. Additional funding from University of Helsinki, European Neural Networks Society, and from the graduate school of Computational Biology, Bioinformatics, and Biometry (COMBI) is also gratefully acknowledged. Of special encouragement have been the personal grants received from Jenny and Antti Wihuri foundation, and the Finnish Foundation for the Promotion of Technology.

The work done for this thesis has taught me essentially everything I know about science now. For this, I wish to thank most sincerely Professor Samuel Kaski. His scientific contribution and supervision in every stage of the actual research has been invaluable for my learning, but even more influential for me has been his indefatigable ambition to conduct research that is capable to create new and to question the old.

I express my gratitude for Professor Erkki Oja, Academician Teuvo Kohonen, and Professor Olli Simula, for I feel privileged to have been able to enjoy the excellent facilities of the Centre, with a well-run and easy-going management.

My sincere compliments belong to the reviewers of the thesis: Professor Olli Yli-Harja and Docent Matej Oresic. Their expertise reasserted the quality of the thesis. I am additionally obliged to Prof. Kaski, Dr. Janne Salo, and Docent Christophe Roos for reading and commenting the manuscript of my thesis, and helping me to improve it significantly.

I am indebted to all the other co-authors. Firstly, I am honored to have had a possibility to work with Academician Teuvo Kohonen. On the methodological side, I wish to thank Dr. Janne Sinkkonen, Merja Oja, Eerika Savia, Jarkko Venna, and Leo Lahti for guiding me through occasional quagmires of theoretical and algorithmic problems. On the biological side, Prof. Eero Castrén, together with Prof. Garry Wong, Dr. Petri Törönen, and Juha Knuuttila, had an unenviable task to acquaint me, a novice computer science postgraduate, into biology. I am grateful for Docent Christophe Roos for many discussions that have helped me to understand more complicated biology, and its requirements for models. I also thank all the co-authors for extraordinary flexibility when deadlines have been approaching a way too fast.

While the work has reshaped me and my thoughts in a most satisfactory way, it has, at times, been tiresome. I wish to thank the whole research group, Arto, Eerika, Jaakko, Janne S., Jarkko S., Jarkko V., Leo, Merja, and Sami, for support and encouragement in such times. Thanks belong also to the whole laboratory for creating an easy atmosphere to work in.

I am grateful for my parents and brothers, since all of them have teached and supported me, each in their own way. And of course, all of my friends are to be thanked for still being there, despite of me being lost in my work. I also wish to thank everyone in our company for allowing me freedom to finalize my thesis.

Most of all, I wish to thank Johanna.

LIST OF PUBLICATIONS

The thesis consists of an introduction and the following publications:

1. Samuel Kaski, Janne Nikkilä, and Teuvo Kohonen. Methods for Exploratory Cluster Analysis. In: *Szczepaniak, Segovia, Kacprzyk, Zadeh (Eds.): Intelligent Exploration of the Web*, pp. 136–151, Springer, Berlin, 2003.

2. Janne Nikkilä, Petri Törönen, Samuel Kaski, Jarkko Venna, Eero Castrén and Garry Wong. Analysis and Visualization of Gene Expression Data using Self-Organizing Maps. *Neural Networks, Special issue on New Developments on Self-Organizing Maps*, vol. 15, issue 8-9, pages 953–966, 2002.

3. Samuel Kaski, Janne Sinkkonen, and Janne Nikkilä. Clustering Gene Expression Data by Mutual Information with Gene Function. In *Dorffner, Bischof, Hornik (Eds.): Artificial Neural Networks - ICANN 2001 (Proceedings of the ICANN 2001, International Conference on Neural Networks 2001)*, pages 81–86, Springer-Verlag, Berlin, Germany, 2001.

4. Merja Oja, Janne Nikkilä, Petri Törönen, Garry Wong, Eero Castrén, and Samuel Kaski. Exploratory Clustering of Gene Expression Profiles of Mutated Yeast Strains. In *Zhang and Shmulevich (Eds.): Computational And Statistical Approaches To Genomics*, pages 65–78, Kluwer Academic Publishers, 2002.

5. Janne Sinkkonen, Samuel Kaski, and Janne Nikkilä. Discriminative Clustering: Optimal Contingency Tables by Learning Metrics. In *Elomaa, Mannila, Toivonen (Eds.): Machine Learning: ECML 2002 (Proceedings of the ECML'02, 13th European Conference on Machine Learning)*, Lecture Notes in Artificial Intelligence 2430, Springer, Berlin, pages 418–430, 2002.

6. Samuel Kaski, Janne Nikkilä, Merja Oja, Jarkko Venna, Petri Törönen, and Eero Castrén. Trustworthiness and Metrics in Visualizing Similarity of Gene Expression. *BMC Bioinformatics*, 4:48, 2003.

7. Samuel Kaski, Janne Nikkilä, Eerika Savia, and Christophe Roos. Discriminative Clustering of Yeast Stress Response. In *Seiffert, Jain, Schweizer (Eds.): Bioinformatics using Computational Intelligence Paradigms*, pages 75–92. Springer, Berlin, 2005.

8. Samuel Kaski, Janne Nikkilä, Janne Sinkkonen, Leo Lahti, Juha Knuuttila, and Christophe Roos. Associative Clustering for Exploring Dependencies between Functional Genomics Data Sets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics: Special Issue on Machine Learning for Bioinformatics - Part 2*, vol. 2, no. 3, pages 203–216, July-September 2005.

9. Janne Nikkilä, Christophe Roos, Eerika Savia, and Samuel Kaski. Explorative Modeling of Yeast Stress Response and its Regulation with gCCA and Associative Clustering. *International Journal of Neural Systems, Special Issue on Bioinformatics*, vol. 15, no. 4, pages 237–246, 2005.

SUMMARY OF PUBLICATIONS AND THE AUTHOR'S CONTRIBUTION

Here a general overview of the author's contributions is given, but the contributions of the other authors are not discussed thoroughly. The author's main contribution is two-fold: first, application of the methods to genomic data, and second, development of new methods. In all cases the actual writing of the papers has been a collaborative effort.

Publication 1 introduces new methods for interpreting the mapping formed by the self-organizing map (SOM). First, it suggests a new method to visualize the cluster structures of the data, based on SOM, and second, a new method for interpreting the areas of the data space, based on the SOM and a local factor analysis. The original idea of interpreting SOM in a more detailed way was offered by Academician Teuvo Kohonen and Professor Samuel Kaski. The detailed methods were jointly developed by the author and Samuel Kaski and all the simulations were done by the author.

In Publication 2 gene expression data is visualized and clustered with SOM and the methods introduced in Publication 1. Additionally, the visualization performance of SOM and hierarchical clustering is compared. The initial idea for the application was jointly developed in a collaboration project with the group of Professor Eero Castrén. The author performed the SOM-related simulations in the paper, and was largely responsible for coordinating the analysis and the visualization of the SOM-related results.

In Publication 3 a discriminative clustering algorithm developed earlier is applied to combine gene expression data with gene functional classes. The author took part in developing the idea for the case study, and designing it. The author also performed the simulations.

In Publication 4 the gene expression profiles of yeast mutation strains are clustered and visualized with SOM. The idea of the research was jointly developed, and the author supervised the simulations. The author also coordinated the analysis of the results jointly with Samuel Kaski.

Publication 5 introduces a new version of discriminative clustering, based on *maximum a posteriori* estimation. The framework was also connected to classical methods of measuring the dependency of categorical data by deriving a connection between the contingency tables and MAP-DC. The author contributed mainly to the connection to contingency tables.

In Publication 6 two key aspects of various data analysis methods are studied: i) the visualization performance, and ii) the similarity metric. The methods are applied to several gene expression data sets. The research idea was developed jointly, and the author participated in designing the case studies and partly coordinated the simulations.

In Publication 7 the improved discriminative clustering developed in Publication 5 is applied to study the regulation of yeast stress by two known regulator proteins. The author developed the initial idea, and participated in planning the experiments and in further development of the idea together with Samuel Kaski and Eerika Savia.

An extension of discriminative clustering called associative clustering is introduced in Publication 8. It is applied to three problems in the field of bioinformatics. The author participated in developing theoretical ideas for associative clustering, developed the ideas for the case studies jointly with Samuel Kaski, largely planned the experimental settings in case studies, performed the simulations concerning the yeast, and supervised the other simulations.

The first attempt to extend associative clustering to multiple data sets with generalized canonical correlations is introduced in Publication 9. The author developed the overall idea jointly with Samuel Kaski. The way of applying generalized canonical correlations was jointly developed with Eerika Savia and Samuel Kaski. The author made all the simulations and contributed the numerical and visual analysis of the results.

LIST OF ABBREVIATIONS AND SYMBOLS

AC	Associative clustering
BF	Bayes factor
CCA	Canonical correlation analysis
DC	Discriminative clustering
DNA	Deoxyribonucleic acid
IB	Information bottleneck
gCCA	Generalized canonical correlation analysis
LDA	Linear Discriminant Analysis
LM	Learning metrics
MAP	Maximum a posteriori
MAP-DC	Maximum a posteriori discriminative clustering
MDA	Mixture discriminant analysis
ML	Maximum likelihood
PCA	Principal component analysis
RNA	Ribonucleic acid
SOM	Self-organizing map
SOM-LM	Self-organizing map in learning metrics
TF	Transcription factor

$d(\mathbf{x}, \mathbf{y})$	distance between data samples $\mathbf{x}$ and $\mathbf{y}$
h_{ij}	neighborhood function of SOM unit i at unit j
$H(X)$	entropy of a random variable X
$I(X, Y)$	mutual information of random variables X and Y
$\mathbf{m}_j$	model vector of jth SOM unit, or prototype of jth cluster.
ψ_j	prototype distribution of auxiliary data, in cluster j
$\mathbf{x}, \mathbf{y}$	vectors in input spaces

Chapter 1

Introduction

This thesis is about applying exploratory cluster analysis methods to genomic data sets. Here genomic data means any data associated to the functionality and the structure of genes: DNA sequence, proteins, metabolites, transcription factor binding sites, etc., and, in particular, transcripts of genes. Exploratory cluster analysis, in turn, refers to computational methods that aim at giving an overview of data by grouping them. This chapter is an overview of the motivations and the needs in biology and data analysis, and a summary of the contributions of the thesis.

1.1 General motivation and background

Biology is the study of living things, and obtaining reliable, quantitative observations from living organisms has traditionally been a major challenge. As a consequence, the development of biology has been strongly connected to the advances made in measurement technologies. In particular, only the development of the measurement methods and tools has enabled the discovery process of the cellular components of organisms. From a certain point of view, this gradual revealing of the particles of living organisms has also influenced the analysis and the hypotheses in biology. It could be argued that, so far, the emphasis in biology has been largely on the individual components of the organisms, and not on the functional, highly interacting system they constitute.

The era of discovering components smaller than the human eye could observe started with an important discovery made using a new analysis tool called microscope. In 1665 Robert Hooke found small vesicles in plants and coined them "cells" (Hooke, 1665). This lead to the realization that there were cells in almost every living organism. Eventually the cell was accepted as one of the fundamental units of life.

The introduction of the microscope also made possible the series of discoveries of ever smaller components of the cell. The discovery of the molecular structure of the DNA in 1953 (Watson and Crick, 1953) was a breakthrough that opened up a new perspective on life. Then the genes in the DNA could be established as a kind of fundamental units of life, explaining the inheritance of traits and diseases. However, while the most individual system components had been observed and described, the lack of description of the higher system structures still hampered the biological research.

During the nineties, a new page was turned in the book of the measurement methods in biology: laboratory techniques capable of measuring the activity of thousands of genes at the same time started developing (Schena et al., 1995; Lockhart et al., 1996). For the first time, with these *high-throughput methods* scientists were able to get an overview of which genes are active in cells at a certain moment and condition. Boosted by large consortium projects, particularly by the Human Genome Project (for a history review, see Roberts et al. (2001)), the techniques continued to be developed and be applied to new problems and they spread all over the world. The field of biology started to become populated with various high-throughput data sets containing information about states of the cell from many different perspectives: gene expression, metabolite concentrations, protein–DNA interactions, sequences of genes and proteins, etc. The centuries-long tradition of analyzing single compounds in the cell started to shift towards the analysis of the cell-wide collection of components and their interactions.

Along the wide-spread use of the high-throughput technologies, a new problem emerged: the measurement techniques produced massive, noisy data sets describing the state of cells, but due to their huge size they were incomprehensible for a human as such. It could be argued that the scientists' ability to generate the data had exceeded their ability to analyze that data. At this point the practitioners of data analysis and data mining using *machine learning* techniques started to move in on the field. This started a new boom of *bioinformatics*, or *computational biology*, a discipline that emerged earlier in the analysis and managing of genomic sequence data. The difference between bioinformatics and computational biology is subtle, but both bioinformatics and computational biology can be seen as the disciplines in which new computational methods are developed for analyzing genomic data and solving biological research problems. The terms are used interchangeably in this thesis.

The goal of machine learning is to build methods that learn from the data in an automated way. The methods usually benefit from having a large amount of data, and are thus well suited also for the analysis of data from biological high-throughput measurement methods. The use of advanced data analysis methods and high-throughput data has actually opened up a new stage in biology, often called *systems biology* (Ideker et al., 2001), which tries to understand the cell as one complex system. But as the stage changed, the hypotheses possible to conceive changed also. In order to understand whether the genomic high-throughput data and machine learning could offer answers or at least a tool in biology, new questions have to be formulated. Therefore, since scientists only have the mass of data, it is logical to start looking for those questions from that.

Exploratory data analysis (EDA) is a branch of statistics that aims to make new discoveries from data. In its modern form, the techniques of computational modeling and machine learning are often used. When applied to genomic data, it is hoped that it can offer ideas for what could be inferred from the data, how to use the existing data in a more specific way, or how to carry out the next set of measurements. Exploratory data analysis is thus quite different from the traditional hypothesis-based analysis in statistics, in which the aim often is to answer one specific question formulated as a null hypothesis. While statistics is much more than hypothesis testing, one characterizing viewpoint is that EDA tries to give an overview of the data in order to facilitate the generation of new hypotheses concerning it.

In this thesis, the self-organizing map (SOM) (Kohonen, 1982, 2001) is used as an exploratory tool to cluster and visualize gene expression data sets. New methods for interpretation of SOM are introduced, and the viability of the SOM approach in exploration of gene expression data is demonstrated in several case studies.

In this thesis a recently introduced principle, *learning metrics (LM)* (Kaski et al., 2001; Sinkkonen and Kaski, 2002; Kaski and Sinkkonen, 2004), is used to guide clustering and visualization of gene expression data. LM tackles a crucial problem in data analysis: similarity metric. For example, the use of Euclidean metric is commonly accepted in clustering, but it implicitly assumes the equal importance of the variables of data. This is not necessarily true at all. In contrast, the methods using learning metrics derive the optimal metric from an auxiliary data, as for example from gene functional classes. The analysis then focuses on the variation in gene expression that is relevant for the functional classes, and, in a sense, combines two information sources.

During the new era of systems biology, the use of multiple and various information sources is becoming increasingly important. An example is modeling of gene expression regulation using both expression data and protein–DNA binding data. The classical methods in exploratory data analysis have rarely focused on the problems of data integration, hence no established approaches for data fusion yet exist. While there are many possible ways to integrate data sources, the problem calls for a rigorous definition to allow proper interpretation of the results. At the same time, the methods should be generally applicable, since new high-throughput technologies are developed constantly. In this thesis a framework for general data integration is proposed: maximization of statistical dependency between the mathematical representations of information sources. The underlying key idea in the framework is that the effects in data that are common to sources are relevant, while source-specific effects are considered noise.

The primary aim in this thesis is to develop new generic methods that are appropriate for analyzing high-throughput genomic data. A main motivation for the research has been the fact that, at the moment, the biological hypotheses concerning high-throughput genomic data are still largely unformulated. Consequently, while the thesis does not try to answer specific biological questions, it certainly aims to contribute to the formulation of the new perspectives in biology.

1.2 Contributions and organization of the thesis

In this thesis, exploratory cluster analysis of (multiple) genomic data sets is studied as a new approach to make discoveries from the genomic high-throughput data, especially from gene expression data. The specific contributions include

- application and development of new self-organizing map-based exploratory data analysis approaches for gene expression data
- application of exploratory cluster analysis methods to genomic data sets, based on the principle of learning metrics
- development of new exploratory methods capable of integrating genomic data sets by maximizing their mutual dependencies, and in particular their application to genomic data sets.

In Chapter 2 background information about the functionality of the cell and the data measured from it is presented. In Chapter 3 some of the most important paradigms of machine learning are reviewed and concepts of exploratory data analysis and clustering are introduced. In Chapter 4 the approach of exploratory cluster analysis, applied to genomic data sets, is motivated and discussed together with several case studies. In Chapter 5 the exploratory cluster analysis methods for dependency analysis, together with their applications, are motivated and introduced. The related works from the literature are reviewed in connection with the appropriate methods throughout the chapters.

Chapter 2

Biological background

To understand what is required from the computational methods in bioinformatics, it is essential to know the basics of cell biology and the measurement techniques used to gather the data from the cell. Particularly, it is of great use to understand the various noise sources and recognize which data can be regarded as approximately commensurable and when. For example, certain measurements can be grouped together to enhance the statistical power of the analysis in the presence of noise, and to an extent inferences can be extended from the organism under study to other organisms. Additionally, it is valuable to know whether some auxiliary data is available for the validation or enhancement of the analysis. This chapter aims to give a short overview of these matters from the perspective of a computational analyst. The part considering elementary biology is largely based on the book by Campbell et al. (2001).

2.1 The cell

Practically every living organism is composed of cells. The main exception are viruses, but it is in fact questionable whether they are actually alive. Cells have an invariable basic structure across the organisms: they are small compartments of an aqueous solution of chemicals encapsulated by a membrane (Alberts et al., 1994). Thus many methods working on and the inferences made on the cell level apply to a wide range of organisms. This is routinely utilized in biology, for example by using *model organisms* like yeast or mouse to study human.

However, cells are not exactly similar in all organisms. A main classification of the organisms is made by the presence of nuclear envelope in their cells: *eukaryotes* have the membrane around their nucleus, but *prokaryotes* do not. Examples of eukaryotic organisms are yeasts, plants, and animals, whereas all bacteria belong to prokaryotes. Figure 2.1 is a schematic illustration of an eukaryotic cell and its various parts. In this thesis the main focus is on eukaryotic cells, in particular, on the baker's yeast *Saccharomyces cerevisiae* which is the most understood eukaryotic organism.

There are also different cell types within individual multicellular eukaryotic organisms. This is mainly due to the specialization of the cells in different *tissues*. Although genome in the cells is the same, the sizes and appearances of cells in different tissues can vary dramatically. Even slight differences in cells between

Figure 2.1: A schematic figure of an eukaryotic cell and its various parts. The figure is an excerpt from Campbell et al. (2001). Figure omitted due copyright reasons.

tissues may add undesirable variance to the analysis, for example, when taking samples from a certain tissue surrounded by some other tissue. This is because the chemical content of the cells in different tissues varies, which is both the cause and the effect of the genes being activated differentially in different tissues.

2.2 Genes and proteins

Genes are the basic units of any organism. Slightly simplifying, it can be said that they determine the type of the organism and its individual characteristics. In a eukaryotic cell the nucleus (see Figure 2.1) contains the deoxyribonucleic acid (DNA) molecule in which the genes are encoded. Another molecule associated heavily with genes is the ribonucleic acid (RNA). Both the DNA and RNA molecules consist of a sequence of *nucleotides* of four kinds: adenosine (A) (uracil, U, in the RNA), thymine (T), guanine (G), and cytosine (C). Nucleotides can bind to each other in certain pairs: A (U) + T and G + C, which is called hybridization. The hybridization takes place in many reactions in the cell. In particular, it is the mechanism that makes DNA a double-stranded molecule, binding the two complementary nucleotide strands to each other.

The concept of gene can be defined in various ways, but most often its definition is based on i) the gene being the hereditary unit of an organism, and ii) the gene coding a protein. For example, before any molecular knowledge existed, in 1865 Mendel described genes as "particulate factors" that pass unchanged from a parent to the offspring. A more recent and functional definition states that *the gene is a sequence of the DNA in nucleus that contains the information the cell needs to manufacture a protein.* The latter definition is adopted in this thesis.

Proteins are involved in practically all of the cell's subprocesses. This is an important fact, since cells are not static objects, but are in a continuous process of living. This process is normally partly self-regulated, partly dependent on various external chemical and physical signals. Whether the signal in the cell is of external or internal origin does not matter: the actions induced by the signal and also the actual handling of the signal are mainly taken care of by proteins.

The protein synthesis is of crucial importance for the cell, since the protein concentrations in the cell in part determine its internal state. Figure 2.2 **A** represents the multistage process of making the proteins schematically. The code for each protein is stored in the DNA and the production process can be regulated at many stages to control the concentrations of the various proteins in the cell (Figure 2.2 **B**).

Perhaps the most dominant regulatory mechanism for controlling the protein concentrations in the cell is *transcriptional regulation.* It means regulation by controlling which genes are transcribed and how much. Transcriptional regulation is performed by a set of proteins called *transcription factors (TFs)*, which bind to a certain DNA sequence nearby a gene, called *the promoter region of the gene.* Depending on the configuration of the TFs in the gene's promoter region, the gene is either transcribed to pre-mRNA, a.k.a. *primary transcript*, or not. It is also possible that the rate of the transcription is controlled in the sense that the TF

Figure 2.2: **A**: The protein synthesis in both prokaryotic cells and in eukaryotic cells. The double-stranded DNA is first opened, and a molecule called polymerase attaches at the DNA in the start of the gene sequence to be transcribed. The polymerase advances along the DNA strand and composes a *pre-messenger RNA (pre-mRNA)* molecule which is the complement of the DNA strand it is based on. The DNA sequence, and pre-mRNA, of a gene consists of two kinds of sequences: *exons* and *introns*. In the next processing stage, *splicing* of the pre-mRNA, the introns are spliced out and only the exons end up in the mature mRNA, which is then transported out of the nucleus to cytoplasm in the eukaryotic cell. This is not the case in the prokaryotic cells, which have no nucleus. In the cytoplasm *ribosomes* read the mRNA strand and compose an amino acid sequence based on the subsequential triplets of nucleic acids of mRNA: each nucleic acid triplet, called *a codon*, corresponds to one amino acid. The newborn amino acid next folds in a certain way to get its final three-dimensional structure, resulting in a potentially functional protein. **B**: The possible control stages of gene expression in eukaryotic cell. The figures are excerpts from Campbell et al. (2001). Figure omitted for copyright reasons.

configuration can make the transcription more or less probable, not just active or in-active.

The *alternative splicing* of the primary transcript is another possible regulation process for the protein production (RNA processing in Figure 2.2 **B**). Alternative splicing enables the generation of different RNAs from the same pre-mRNA sequence. The mechanism takes place at the stage in which the non-coding regions inside the gene, *introns*, are sliced out, and only the coding parts, *exons*, are joined together and exported from the nucleus. In alternative splicing, the splicing sites are different from the "normal" version of the mRNA, resulting in a different mature mRNA. This is one of the reasons why no one-to-one mapping exists between the genes in the DNA and the actual proteins.

A relatively recently discovered regulatory process is *RNA interference (RNAi)*. It takes place right after transcription and works by inactivating the mature mRNA by hybridizing to it a complementary RNA strand (Dykxhoorn et al., 2003). It is still unclear how common RNAi regulation is in a normal cell but it has a great potential in gene therapy and an important role in certain diseases.

Post-translational modifications refer to all processes that take place after the mRNA is translated and folded into a functional protein. An example of this is provided by various interactions between proteins that results in formation of larger units, called protein complexes, which have functions different from their individual protein components.

The final functional proteins are the key players in the cell. They are used for building the various components of the cell, for delivering signals, for controlling gene expression, and they also take part in metabolic processes.

The interactions between proteins and other molecules in the cell form complex networks that are extremely diverse, in size and in topography. Unveiling these interaction networks of the various processes in the cell is an essential part of understanding the functionality of the cell. But the task is difficult both due to the complexity of the cell's system and due to lack of information about the cell. Figure 2.3 shows an example of the known interactions between proteins and other molecules in one process of the cell, which demonstrates the potential complexity of such networks.

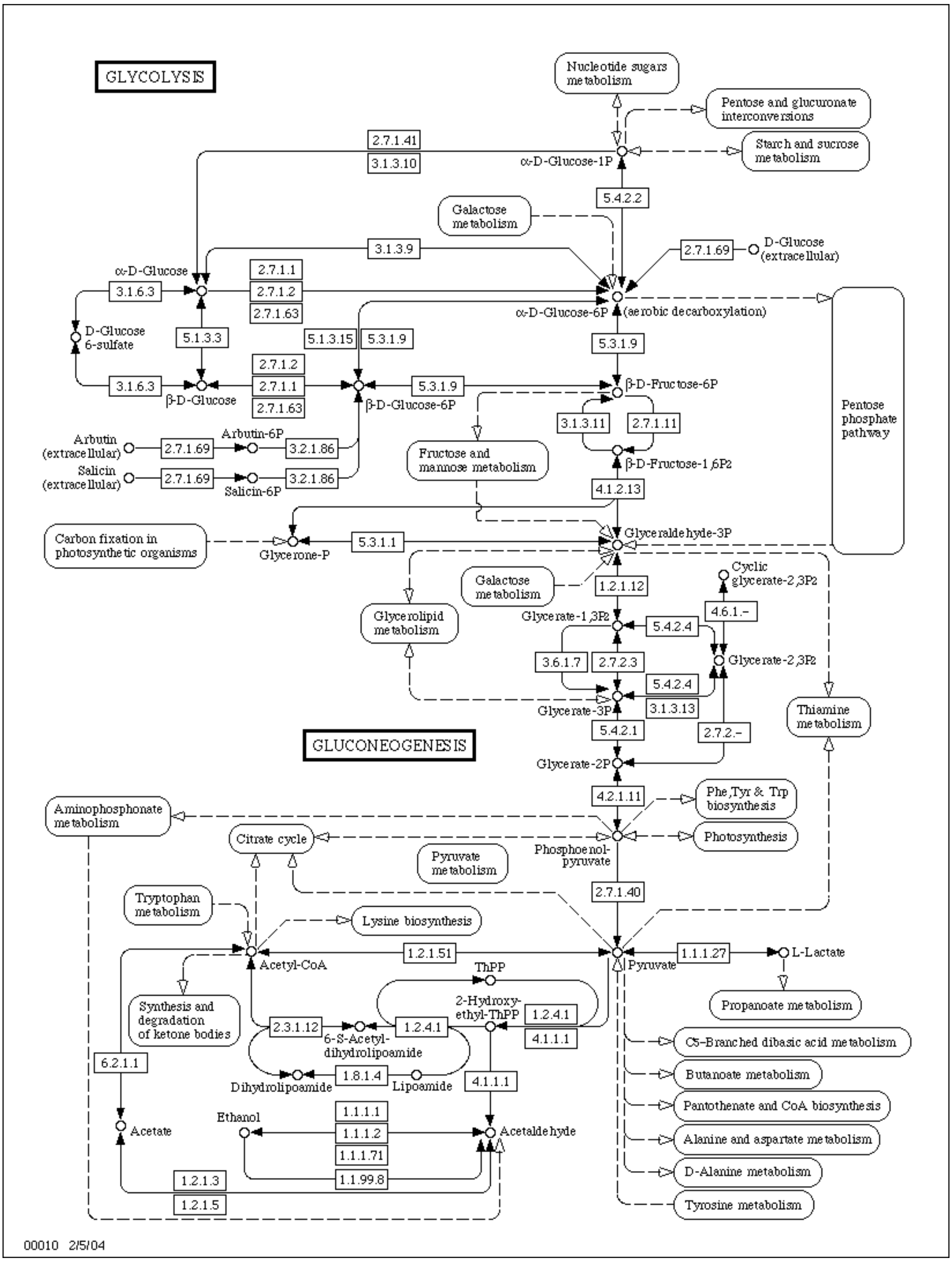


Figure 2.3: The glycolysis/gluconeogenesis process in the cell demonstrating the potential complexity of the interactions on a molecular level. The squares represent proteins or RNA, circles represent other molecules, and squares with round corners are other processes. The figure is taken from Kyoto Encyclopedia of Genes and Genomes (KEGG).

2.3 Data concerning the cell

Understanding a dynamic, complex system requires, first of all, time series measurements of most of the variables in the system. Unfortunately, for the cell this requirement can not be fulfilled. In fact, making measurements from the cell is currently one of the largest challenges in biology. Of special interest are the protein concentrations in cells in certain conditions at a certain time, but so far there are no practical or reliable methods to gather this information. Recent advances in laboratory techniques, however, have enabled the measurement of the gene activity on a genomic scale.

2.3.1 Measuring gene expression

Gene expression means the activity of a gene, measured by the amount of the mRNA produced from it and present in the cell at some specific time or environmental condition. Gene expression of thousands of genes can be measured simultaneously from cells with *DNA microarrays* (Schena et al., 1995; Lockhart et al., 1996). Currently, there are several microarray techniques; all of them are based on the same mechanism: hybridization of the mRNA in a sample(s) to the complementary nucleotide sequences attached on the arrays.

The basic idea in all microarrays is that a set of *probes* that correspond each known gene and expressed piece of mRNA (called *ESTs, expressed sequence tags*) in a certain organism has been immobilized on a small piece of suitable media (often glass). When measuring the gene expression in a cell population, a sample of cells of interest is collected and the mRNA is isolated. After possible preprocessing, the mRNA is *labeled* with certain marker molecules and hybridized on the microarray. The relative amount of mRNA hybridized on each probe reflects the mRNA concentration in the sample cells. The amount of mRNA on each probe is nowadays usually determined by measuring the intensity of light, characteristic to label molecules, with a laser scanner.

The different types of microarrays can be categorized in two main classes from the perspective of analyzing them: comparative and non-comparative arrays. In comparative arrays there are always two samples that are hybridized on the array, one typically being some sort of a reference sample (Schena et al., 1995). The comparative arrays are often called *cDNA arrays* or *spotted arrays*, referring either to the material of the probe or to the probe attachment technique: *spotting* (or *printing*). The reason for using comparative hybridization in spotted arrays is usually the so-called spot variation, which means that the probes for the different genes are not equally sensitive and thus can not be trivially compared on the absolute intensity level. Without several replications of the array measurement, the only available data from the comparative arrays are the ratios of the intensities for each probe.

In non-comparative hybridization arrays the probes are not spotted, but build directly on the array. They are more comparable and enable, at least in principle, measurement of the absolute levels of mRNA in the hybridized sample. More importantly, the reproducibility is better in this kind of microarrays and they do not require as many replications of the array measurement. However, they are more expensive respectively. The most common of the non-comparative microarrays currently is Affymetrix GeneChip (Lockhart et al., 1996).

The microarray experiments are complex processes with many parameters and choices. Hence, standardization of microarray experiments is extremely important. Attempts aiming for this include, for example, Minimum Information About a Microarray Experiment (MIAME) (Brazma et al., 2001) that specifies the subjects that should be reported with every microarray experiment. The common standards for the experiments ease the publication, the reproduction, and the storage of microarray data in public repositories, see for example Barrett et al. (2005); Brazma et al. (2003).

2.3.2 Other high-throughput information about the cell

Microarrays measuring gene expression offer one of the first ways to make high-throughput measurements about the cell. However, the power of the gene expression measurements will become fully utilized only after development of other high-throughput measurement techniques, when all the information can be integrated. The following sections browse the selection of the most widely used other high-throughput technologies briefly.

Hybridization-based methods

Other array-based hybridization approaches include at least the scanning of *single-nucleotide polymorphisms (SNPs)* from a genome (SNP-chips), comparative genome hybridizations (CGH-arrays), and protein–DNA interaction measurements. These methods are based on the hybridizing genomic DNA on microarray, not mRNA.

In SNP analysis the probes on the array consist only of the specific parts of the DNA that are known to include single nucleotide mutations. The resulting data are of binary nature (though noisy): either a SNP is present in the sample or it is not (see Lipschutz et al. (1999)). Detection of SNPs is relevant, for example, in research of hereditary diseases.

Measurement of the gene copy number and DNA multiplications is called *comparative genome hybridization (CGH) analysis* (Snijders et al., 2001). It is based on hybridizing the sample genome with a reference genome on an array containing probes for each gene present in the genomes. The data obtained are of continuous nature in the same sense as in the gene expression measurements, and reflect the multiplications and deletions in the DNA of the sample genome. The gene copy number and other chromosomal changes play a very important role in cancers, where they are assumed to be a reason for abnormal gene expression, and a mechanism in part responsible for a normal cell turning into a cancer cell.

Analysis of genome-wide protein–DNA interactions with microarrays is a recent and promising technique. One way to do that is to use so-called *chromatin immunoprecipitation (ChIP)* combined with cDNA arrays (Ren et al., 2000; Lee et al., 2002). The method is based on i) immobilizing the proteins (for example transcription factors) binding the chromatin of DNA, ii) digesting the DNA and immunoprecipitating the protein–DNA complexes with an antibody for the protein from the rest of the solution (enriching the solution with the protein of interest), and iii) hybridizing the solution on a microarray against the un-enriched solution. The ratios in each spot then reflect the amount of the tagged regulator attached to the corresponding gene. Note that the resulting data are real-valued, since there can be multiple binding sites and a lot of measurement noise.

Other methods

Several other methods exist to collect data from the cell, and the number of techniques is growing all the time.

Perhaps the most traditional and fundamental genomic-scale data from the cell is the DNA sequence itself. Although the actual sequencing (deciphering the nucleotide order of the genome) is not necessarily trivial, the information about the nucleotide sequence of each gene is presently freely available for many organisms in the databases accessible through internet. These sequences can be used to study, among other things, the alternative splicing as well as the binding sites of transcription factors. In particular, note that sequencing of DNA is the prerequisite for the transcriptome analysis, that is, gene expression microarrays.

Multiple techniques for *mass spectrometry* can be used to analyze the contents of any sample at the molecular level. For example, they can be used to study the metabolic compounds in blood, or to obtain protein "fingerprints" of the samples.

There are also attempts to develop microarrays that are not based on the principle of hybridization. Among these, the protein chips are of great interest since they could potentially be used to analyze the protein content of cells in a high-throughput fashion. However, the technology is challenging and the quality of the data is not yet acceptable (Michaud et al., 2003).

Finally, so-called *tissue chips* are designed for the analysis of hundreds of tissue samples at a time. The data from these chips are largely qualitative, but greatly speeds up the analysis in comparison to the traditional methods (Kononen et al., 1998).

2.3.3 Public databases

The existing information about the cell is largely stored in various databases accessible through the internet, both commercial and non-commercial. In addition to the databases containing measurements, there are also numerous databases containing higher-level information about the cell, like various classifications for genes. These databases are in a fundamental position in creating a holistic perspective on the cell and its functions. Some examples of the cellular-level information contained in the databases include gene ontology, molecular reaction pathways, molecular interactions, and phylogenetic information. The most important of these are presented briefly in the following.

Gene ontology

The main goal of the Gene Ontology (GO) project (Consortium, 2000) is to provide a consistent vocabulary for genes, which is applicable to all eukaryotic organisms. In practice, GO can be used as a classification for genes and it includes three separate ontologies (classifications): biological process, molecular function, and cellular component. Each annotated gene is mapped to one node (class) in each ontology.

A common practice to utilize GO is to check whether its classes are overrepresented in some gene groups, obtained, for example, from clustering analysis. If they are, it serves as a validation for the gene group, since it is then more likely that it represents some biologically interpretable entity, or part of it, in the cell.

Metabolic and regulatory pathways

Pathways represent knowledge concerning causalities in the cell. These causalities are of utmost importance, since, in principle, they enable inference and prediction of the effects of individual experiments on the cell. However, the available knowledge is still rather limited, but it should nonetheless be used whenever possible to utilize all the existing information in the analyses.

Molecular interactions

Interactions between various molecules can be used in models as prior information or as validation data, hence they are highly important. The Biomolecular Interaction Network Database (BIND) is a storage of molecular interaction information. Three kinds of interactions are documented in BIND: molecules that associate with each other to form interactions, molecular complexes that are formed from one or more interaction(s), and pathways that are defined by a specific sequence of two or more interactions.

Proteins

UniProt (Universal Protein Resource) is the most comprehensive database of information on proteins. It is a central repository of protein sequence.

Genomes

The annotated sequences of organisms are stored, for example, in Ensembl database. Ensembl offers the sequence data and software for analyzing it. In principle, all the information concerning DNA sequences can be found there.

Availability

Links to these databases and to many more can be found for example in Galperin (2005) and its supplement in the WWW,
`http://nar.oupjournals.org/cgi/content/full/33/suppl_1/D5/DC1`.

Chapter 3

Methodological background

A vast array of computational methods exist in the literature that have been, or could have been, applied to genomic data sets. It is impractical to try to present individual methods comprehensively here, this chapter rather focuses on the general paradigms relevant to many of them, in particular from the perspective of explorative cluster analysis. The chapter begins with a short introduction to the branch of computational modeling called statistical machine learning and its paradigms, providing some background for the actual methods presented in Chapters 4 and 5. The concepts of explorative data analysis and clustering are explained in the last section of the chapter.

3.1 Representation of data

In computational modeling the observations concerning some phenomena are usually treated in numerical format. This means that, for a set of objects, we have a set, or sets, of variables or features for which we have observed numerical values. For example, for a set of genes (objects) we could have measured their expression (values) in a set of treatments (variables). The concept "data" hence refers to the numerical values, and is often denoted with a symbol D. If the data consist of observations for the same variables for each object, it is convenient to represent the data and the variables as a matrix, often denoted with $\mathbf{X}$, where each row corresponds to an object and each column a variable. Then the observed values for the ith object can be represented as a vector $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{in})$, where n is the number of the variables, or the *dimensionality* of the data.

The observed data can also sometimes be described as a set of observations $\{\mathbf{x}_i\}_{i=1}^{N}$, $\mathbf{x}_i \in \mathbb{R}^n$, which is often abbreviated as $\{\mathbf{x}\}$ in this thesis.

3.2 Statistical machine learning

Computational models are representations composed of mathematical and statistical concepts that can be manipulated with algorithms in computers. In general, computational modeling aims at characterization, summarization, prediction, and simulation of phenomena. If there is uncertainty in the observations, the computational models often use the concepts of *statistical modeling.*

Statistical machine learning is a relatively new branch of computational modeling that encompasses a variety of methods from different fields including statistics, mathematics, neural networks, and information theory. Its goal is to develop computational, statistically justified models that learn efficiently from the data in an automated way. Learning here means that the models have some structure specified *a priori* with parameters that are optimized in the sense that the model best describes the data, or some property of it. Statistical machine learning is usually differentiated from the traditional statistics and information theory usually by its more complex models, and in part by its emphasis on efficient algorithms. The motivation and algorithms of the neural networks, on the other hand, are often based on the biological neural networks and heuristics derived from them, but rigorous neural network methods can be seen as a part of the statistical machine learning genre. In the end, good results, practicality, and generality of the machine learning methods often allow compromises on the theoretical side. The methods applied and developed in this thesis fall mostly in the category of statistical machine learning by their techniques, but their usage and goals are exploratory (see Section 3.3).

3.2.1 Overview of concepts in statistical machine learning

Machine learning approaches dealing with real-world data are often based on building a model at least partly on statistical and information-theoretic concepts, optimizing it with mathematical tools in a computer-aided way, and then possibly visualizing the results. This section overviews the most important paradigms used in the design of machine learning models.

Statistical modeling

Statistics is a branch of applied mathematics concerned with the collection and interpretation of quantitative data and the use of probability theory to estimate population parameters. Population here refers to the idea that there exists some "true" distribution or population of the data from which the observed sample (data set) is drawn.

Practically oriented statistics is centered around the observed set of data, D, and some model $M(\theta)$ with parameters θ for that data. The relation between the model and the data in statistics is usually defined with a *likelihood*,

$$p(D|M,\theta), \tag{3.1}$$

which measures the probability of the observed data given the model and the specific parameter values. The concept of likelihood is more or less common to all probability-based statistics, but opinions differ on whether one should use it as a primary source of inference about the data and the model.

There are numerous possible categorizations for the methods in statistical modeling. Three of the most salient characterizations are reviewed here: Bayesian modeling, generative/discriminative methods, and supervised/unsupervised methods.

Bayesian modeling is sometimes regarded as a fundamentally different philosophical school in statistics. It differs from the traditional approaches especially in the interpretation of the probabilities of events: instead of treating them as ratios of the number of successes against the number of events based on a large number of replications, in Bayesian modeling they are viewed more generally as subjective degrees of belief. The second big difference is the treatment of all parameters as random variables in the Bayesian statistics, which results in representing them with probability distributions. Lastly, the Bayesian statistics include the concepts of *a priori* information and *a posteriori* information, which mean that the inference made is incremental: the prior distribution is based on the previous data or belief, and the posterior distribution is computed when the new data is observed. The Bayesian way to do inference is based on the ideas of Reverend Thomas Bayes from the 18th century, and on the later derived *Bayes' formula* that gives the posterior probability of the model given the data

$$p(M|D) = \frac{p(D|M)p(M)}{P(D)} = \frac{\int_\theta p(D|M,\theta)p(\theta|M)d\theta p(M)}{\sum_M p(D|M)p(M)}. \tag{3.2}$$

The key idea is to take into account the uncertainty in the parameter values θ by defining a probability distribution, *a prior*, $p(\theta|M)$ for it, and integrating it out in the *marginal likelihood* term $p(D|M)$. The use of Eq. 3.2 results in the inference about how well the model M describes the data, and it can be used in model selection.

Whether the chosen model M is the correct one is a fundamental question in statistical modeling. In principle, and from the Bayesian point of view, one is interested in knowing which model M_m will produce the highest posterior probability for the observed data $P(M_m|D) = \frac{P(D|M_m)P(M_m)}{\sum_m P(D|M_m)P(M_m)}$. It is often not possible to compute the posterior probabilities of all the models, but they are compared in a pairwise manner with the ratio:

$$\frac{P(M_1|D)}{P(M_2|D)} = \frac{P(D|M_1)}{P(D|M_2)}\frac{P(M_1)}{P(M_2)}, \tag{3.3}$$

where, in the right side, the first ratio is called the *Bayes factor*. The second ratio, the models' prior probabilities, is often assumed to equal one, and only the Bayes factor is used to select the model (see for example Kass and Raftery (1995)). Instead of fully marginalized likelihoods $P(D|M)$, it is possible to use the likelihoods $p(D|\hat{\theta}, M)$, where $\hat{\theta}$ is a maximum likelihood estimate of θ. This leads the Bayes factor to the so-called *likelihood ratio*, which has a long history in various tests used in traditional statistics. Bayes factors are used in Publications 5, 8, and 9 as optimization criteria. For a more comprehensive introduction to the subject see for example (Gelman et al., 2003) and (Kass and Raftery, 1995).

However, the question about the correct model family is also often neglected and the M is assumed to be fixed. Then the primary interest are usually the parameters θ. The Bayesian method is then to compute the *posterior*, that is, the probability distribution over the parameter configurations given the data and the model,

$$p(\theta|D,M) = \frac{p(D|\theta,M)p(\theta|M)}{\int_\theta p(D|\theta,M)p(\theta|M)d\theta}, \tag{3.4}$$

where the normalizing denominator is the marginalized likelihood $P(D|M) = \int_\theta p(D|\theta,M)p(\theta|M)d\theta$ from Eq. 3.2.

If the posterior of the parameters has to be summarized, or the integral in Eq. 3.4 is too difficult to compute, the posterior is sometimes maximized. This is called *maximum a posteriori (MAP)* estimation of the parameters. If the prior of the parameters $p(\theta|M)$ is assumed uniform over all possible θ, the approach reduces to the one used in traditional statistics and is called *maximum likelihood (ML)* estimation of the parameters. MAP estimates are used in Publications 5 and 7, and ML estimates in Publication 3.

Generative and discriminative modeling division is historically related to classification tasks ("discrimination"), but it is nowadays associated with a wider array of modeling tasks. The decisive question is whether to model the whole phenomenon related to the task and solve the task based on that model (generative modeling), or to build a model solving the task directly (discriminative modeling). Naturally, the phrasing of the question is applicable only if the task can be seen from both perspectives, as, for example, classification and regression. The discussion below concerns such tasks.

The idea of generative modeling is to build a model that is assumed to have generated the observed data. Generative models rely heavily on the properties of the full probabilistic modeling in their attempt to describe the entire phenomenon with probability distributions. Given that the model is correct and there is only a little data available, they can be superior to discriminative models (Ng and Jordan, 2002). However, they can also be computationally very heavy. Examples of generative models are *mixture models* (see for example Gelman et al. (2003)) and *graphical models* (see for example Jordan (1999)).

In contrast, discriminative models are designed to solve a specific task, and they often neglect the aspects of probability distributions that are not straightforwardly relevant to the task. An example of a discriminative model is *support vector machine (SVM)* designed for classification (Vapnik, 1998; Shawe-Taylor and Cristianini, 2004). It carries out the classification solely based on the class boundary in a feature space. Given enough data, the discriminative models often outperform the generative ones in their task (Ng and Jordan, 2002).

The terminology is sometimes rather vague. For instance, the modeling of conditional distributions is often held equal to discriminative modeling, but it can be argued that it represents only a subset of discriminative tasks. From a wider perspective, all models directly solving the task of interest and neglecting some irrelevant aspects of the phenomenon are discriminative. For a recent textbook about the subject, see for example Jebara (2003).

The supervised and unsupervised learning dichotomy is related to discriminative learning. In particular, discriminative learning is practically always supervised learning, but not vice versa. Supervised learning usually means that there exist some "correct answers" in the task the model is trying to solve. Usually this is some relevant auxiliary information for the primary data, such as class labels. A supervised model then focuses solely on modeling this auxiliary information, disregarding all the aspects in the primary data that are irrelevant for the auxiliary data. Unsupervised learning, on the other hand, means that there exists no correct answer for the task the model is trying to solve, an example of this being density estimation.

The division of computational models into supervised and unsupervised methods can sometimes be misleading. The main reasons for that are too vague definitions of the terms, and the fact that the two categories do not cover all possible methods. The models searching for dependencies or utilizing some auxiliary data, which are studied and applied in this thesis, are examples of the methods not really fitting into either of the categories.

Information theory

A field of research closely related to statistical modeling is *information theory*, which focuses on data compression and the transmission rate of the data (Cover and Thomas, 1991). Information theory uses concepts of statistics in its definitions of various notions, and, on the other hand, many probabilistic models apply information-theoretic concepts. The elementary concepts of information theory will be presented here.

The *uncertainty* of a discrete random variable X is defined in information theory as *entropy H(X)*

$$H(X) = -\sum_{x} p(x) \log_2 p(x). \tag{3.5}$$

Loosely said, entropy measures the information content in the distribution $p(\mathbf{x})$.

Relative entropy or *Kullback-Leibler divergence D* measures the difference between two probability mass functions $p(x)$ and $q(x)$:

$$D(p||q) = \sum_{x} p(x) \log_2 \frac{p(x)}{q(x)}. \tag{3.6}$$

A very important concept in this thesis is the *mutual information I(X,Y)*, which is a measure of dependence between two discrete random variables X and Y

$$I(X,Y) = \sum_{x} \sum_{y} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}. \tag{3.7}$$

In other words, mutual information is the measure of information that X contains about Y and vice versa (in the sense of their probability distributions). Note that mutual information can be interpreted as relative entropy between the factorized joint distribution $p(x)p(y)$ and the full joint distribution $p(x,y)$. Mutual information, or its finite-data variant, is used in Publications 3, 5, 7, 8, and 9.

Originally, mutual information was defined only for two random variables, but it can be extended to multiple variables. The most straightforward extension is the so-called *multi-information* (see for example Studený and Vejnarovà (1999)),

$$MI(X_1, \ldots, X_n) = \sum_{x_1} \cdots \sum_{x_n} p(x_1, \ldots, x_n) \log_2 \frac{p(x_1, \ldots, x_n)}{p(x_1) \ldots p(x_n)}, \tag{3.8}$$

which measures the dependency of any two or more variables. The most important practical difference to mutual information is thus that a large multi-information does not necessarily imply the dependency between all the variables. Publication 9 uses multi-information.

The information-theoretic concepts were presented for discrete data here, but their natural generalizations for continuous data exist as well (Cover and Thomas, 1991).

Neural networks

Neural networks refer to computational models that are inspired by biological neural nets. They usually consist of small conceptual components called *neurons* and an interaction network connecting them. Estimation of the parameters in these models is usually referred to as *learning* or *training* of the neural network.

Neural networks are sometimes referred to as *non-parametric* models. This means that the observed phenomenon is modeled with a flexible model with parameters without a trivial interpretation. In contrast, in parametric models parameters have some straight-forward interpretation in the context of the application. Note that although non-parametric models might perform numerically better in analysis tasks, their interpretation can sometimes be tedious.

Neural networks approaches can be regarded as the pioneering models to more flexible statistical machine learning models, and they have produced many efficient and elegant methods for data analysis. For textbooks see for example Haykin (1999); Kohonen (2001); Bishop (1995). An example of neural networks is the self-organizing map (SOM) (Kohonen, 1982, 2001) that is studied and applied in Publications 1, 2, 4, and 6.

Generalizability of the computational models

The key idea in all modeling is to capture the aspects of the phenomenon that remain the same for the future, or unseen, data from the same source. Usually all other aspects in the data are regarded as noise that is random and does not reflect the true behavior of the system. If the model succeeds in this it is said that the model *generalizes* well or it *predicts* well. A big practical problem is how to build such models, and a number of different approaches exist to achieve this goal.

There are two underlying reasons for a model not generalizing well: i) the model may be incorrect in the sense it cannot reflect the behavior of the system (induces bias), or ii) the model might be too flexible (complex) for the amount of available data (induces variance). These two reasons and their interactions are often characterized in supervised learning by the *bias-variance trade-off* (Geman et al., 1992), which has also been generalized to any maximum-likelihood-based cost functions (Heskes, 1998).

When an over-complex model is fitted to too scarce data, the model is actually fitted partially to the noise, and consequently the results with that model for new data are bad. This phenomenon is called *over-fitting.*

In theory, in the Bayesian framework the averaging over parameter distributions together with a good choice of their priors usually protects from over-fitting. However, in practice the fitting of complex Bayesian models can be very time-consuming and has its own problems (Gelman et al., 2003; Jordan, 1999). Additionally, the choice of the priors can sometimes be challenging.

Another way to avoid over-fitting, especially when searching for a point estimate of the parameters such as ML or MAP, is to use some *regularization* during the fitting of the model. In practice, this kind of methods prevent the model from being fitted too well to the training data, usually by optimizing some other objective in addition to the actual cost function of the model. Such regularizations are used in Publications 5, 7, 8, and 9.

In addition to regularization, over-fitting of a model can be controlled in point estimation by averaging the parameter values over the multiple fittings of the model to different training sets. In its optimal form, with multiple truly independent training and test sets this kind of averaging requires a lot of data and is impractical nearly always. The idea itself, however, is still applicable when the training and the test sets are composed with *re-sampling methods*, such as *cross-validation* and *bootstrap*.

Cross-validation (Stone, 1974) is a method in which the data set is divided randomly into N subsets and the model is fitted with a data set consisting of $N-1$ subsets and evaluated using the subset that was left out. This one computation is called *fold*, and it is then repeated so that all the subsets are left out once. If the parameters are identifiable from one fold to another and their distribution is unimodal, they can be averaged over folds to get a robust estimate of them. While this is not always possible, it is usually at least possible to average the value of the cost function or of the prediction. These robust estimates for unseen data are often used for model comparisons. Note that the folds are not completely independent and estimates are thus always slightly biased. Cross-validation is used in Publications 3, 5, 7, 8, and 9, in model comparison.

In bootstrap (Efron, 1979; Efron and Tibshirani, 1993) the idea is to generate many *bootstrap sets* from the original data set by sampling it uniformly with replacement. Bootstrap sets are of the same size as the original one, and based on them it is often possible to estimate the biases and variances of the parameters of the model. Bootstrap is used in Publications 8 and 9 to reduce the uncertainty of the clustering.

3.3 Exploratory data analysis (EDA)

Exploratory data analysis (EDA) is a certain perspective to data analysis, not a collection of methods. It aims at giving an overview of data, which is of crucial importance when the data is completely new, or when it is not clear what question the data should answer. For a new data set, it is advisable to perform at least a basic EDA to discover possible unexpected behavior in the data, such as measurement errors.

EDA has its roots in the seventies, when Tukey published his seminal book *Exploratory Data Analysis* (Tukey, 1977). The book opened the era of "looking-at-data" with relatively simple boxplots and graphs that are still in use. From those days researchers have moved towards more advanced visualizations and other methods, but the aim has remained the same: to explore unknown data in order to gain insights into the phenomenon and to generate more detailed hypotheses.

The most dominant method types in EDA are perhaps *clustering* and *visualization* of the data. The reason for their predominance is that both of them can naturally give summaries of the data, and especially combinations of the two work well in practice. In the literature, both types have been applied extensively to genomic data sets, in particular to gene expression data (see Chapter 4). This thesis focuses on exploratory cluster analysis.

3.3.1 Clustering

The intuitive aim of clustering is to compose groups of data in such a way that the data items are more similar within each group than between the groups. Despite the intuitiveness of the goal, the clustering task is inherently difficult in general.

For example, it has been argued that it is impossible to satisfy a scale-invariance, a richness, and a consistency simultaneously with any clustering function (Kleinberg, 2002). Moreover, a rather common tendency to apply clustering methods to tasks that should be solved by other means leads to arbitrary and suboptimal results. Nonetheless, clustering can be a highly useful and even necessary tool for summarization of vast data sets, when used in a considerate way.

The specific aim of the clustering analysis can vary depending on the task at hand. At least the following related objectives can be distinguished:

- summarization of a data set (with discovered clusters and their prototype features)
- class discovery (discovery of some assumed class structure in the data)
- local generalization over data (to improve the parameter estimation in models).

Although the objectives can, in principle, be reached with an identical computational machinery, the clustering result obtained is not necessarily interpretable in the perspectives of all the objectives. Especially the last goal is not always sensible in all applications, whereas the first two generally are. Note that, in the last objective, the primary aim is not to discover the assumed latent class structure, but only to use it in parameter estimation. Actually, the preferable alternative for the last objective would, of course, be to measure more replicates of individual data items, but this is usually infeasible.

The clustering methods can be coarsely divided into *partitional* and *hierarchical* methods (Jain and Dubes, 1988). Partitional methods partition the data into non-overlapping clusters, and hierarchical methods create a hierarchy of clusters. Partitional clusterings are often model-based, meaning that the clustering introduces a model, such as a set of parameter vectors in the same space as the observed data, and an algorithm that optimizes the parameters of the model according to some cost function. Often partitions equal so-called *Voronoi regions* that are areas of the data space where, for all the points in that region, the closest parameter vector is the same (Voronoi (1908); see for example Kohonen (2001)). Hierarchical methods, on the other hand, operate more often on the similarity matrix, and the data points are clustered more directly based on their mutual similarities. The most common clustering methods are reviewed in Chapter 4.

3.3.2 EDA in the process of statistical data analysis

Data analysis in any field is usually a multi-stage and iterative process. In the analysis of genomic high-throughput data, the first stage is, or at least should be, the formulation of the objective. This means that there should be at least one well-defined objective which the data and the experimental design for the measurements aim to fulfill.

The *experimental design* of a study focuses on the plan which defines what should be measured, with how many replicates, and in what conditions, in order to render the obtained data maximally useful for the analysis goal.

After the measurements have been made, it is advisable to use exploratory analysis methods such as visualizations to check the properties of the data. This exploratory stage gives guidance for the subsequent analysis stages, and may also indicate if some part of the measurements needs to be redone. This stage is sometimes called *quality control* of the data.

The initial exploratory analysis confirms the suitability of the data for analysis, and the following phase is to build the computational model that aims to answer the primary question. Naturally, there exists a multitude of different kinds of questions and models, each combination producing a different analysis. Slightly simplifying, it can be said that this stage is a *hypothesis testing* step in which a model of some assumption (question) is formulated and its correctness is tested based on the measured data.

After these stages the first analysis is usually concluded. This is the point at which the exploratory analysis can be used again in the most fruitful way, this time in search for new possible hypotheses (questions) concerning the data. This stage is also often referred to as *data mining*. As the number of public data sets increases constantly, the possibility of new discoveries and the need for new exploratory methods gets more and more significant all the time.

As a summary, exploratory data analysis is an integral part of the data analysis process and should never be overlooked.

Chapter 4

Exploratory cluster analysis of genomic data sets

This chapter presents two of the main contributions of the thesis: i) the new self-organizing map-based approaches for exploratory cluster analysis of genomic data, and ii) the applications of the methods that integrate continuous primary data and discrete auxiliary information to analysis of genomic data. In the first section, the exploratory cluster analysis of genomic data is motivated. The following sections present the specific approaches including the summaries of the methods and biological settings, and the main results of the analyses, as well as the related work presented in the literature.

4.1 Motivation of EDA for genomic data

Firstly, *exploratory data analysis (EDA)* provides means to understand the key properties of unknown public data sets. Along the recent increase in the amount of available biological data, it has become a necessity to perform EDA for any analysis including some publicly available data with unknown characteristics.

Secondly, an important area of application for EDA is the initial quality control of the genomic high-throughput measurements. Since the measurement techniques are able to produce thousands of numerical values about the cell, it is not possible to check the quality of the data simply by “eyeballing” the numerical values. Already simple visualizations like box-plots and histograms help a lot, but more complicated methods, such as clusterings and projections, may reveal unexpected or undesired properties in the measured data. EDA offers information for performing the appropriate pre-selection and preprocessing of the data.

The third reason for the importance of EDA in bioinformatics is the potential complexity of both data and the hypotheses that one is able to conceive. These complexities are a natural result of the complexity of the system that generates the data. In this case the system is the fusion of living cells and the laboratory techniques used to measure them. Together they generate data that are inherently hierarchical, have dependencies between the levels of hierarchy, and are from highly interacting sources. The most common example of this is perhaps the gene expression data that have both biological and technical noise, a natural complex hierarchy in the form of the cell processes, and interactions between the genes.

EDA, being unsupervised by nature, offers a possibility to search and discover the phenomena caused by that complexity and reflected, sometimes even unexpectedly, in the data.

The fourth argument for the necessity of EDA in bioinformatics is the need to take advantage of multiple information sources. This need is generated by numerous unpredictable interactions between data sets that should be utilized. However, building of highly detailed models without any overview of the dependencies between the data sets may lead to inferior results and wasted efforts. The future of computational biology lies in systems biology, and it will only increase the need for new exploratory methods capable of summarizing the relations between multiple data sets.

4.1.1 Clustering of genomic data

Since clustering is a basic method in exploratory data analysis, its applicability deserves some extra attention. Naturally, the publications of the thesis and the results therein, as well as applications presented in the literature, provide evidence in favor of the success of the clustering and visualization of genomic data. However, there also exists *a priori* justifications for the applicability of clustering methods to genomic data specifically.

Clustering of gene expression profiles operates in all three different clustering perspectives: i) summarization, ii) class discovery, and iii) local data averaging (see Section 3.3). First, the classical motivation to summarize any vast data set with a small set of clusters is relevant: typical genomic data sets are large. This is because genomes may contain tens of thousands of genes which are studied in numerous experimental conditions.

Second, it is known that the genes are expressed in groups. This is due to many functions in the cell requiring a large variety of proteins (i.e., activated genes). Hence, clustering of gene expression profiles can be seen as a well-justified tool for discovering functional groups of genes. Additionally, if clustering is carried out for the variables (experimental conditions), another latent class structure can also be found, for example, in the form of disease subtypes. This kind of natural grouping is also present in a number of other types of genomic data, such as in protein sequence data, and in protein–DNA interaction data.

Third, model parameters can be estimated more robustly by taking advantage of the group-wise activation of genes. Currently, both the cost and the noise of the biological experiments are high, which results in carrying out an inadequate number of replicates per gene in one data set. This hinders reliable estimation of parameters, but can be overcome in part by local averaging of data.

In computational biology the data sets are generally very large and clustering is practically always needed at some point, also in visualization. Hence, the focus in this thesis will be on clustering methods, some of which lend themselves naturally for visualization.

4.2 Exploratory cluster analysis of a single genomic data set

Exploration of a single genomic data set is a common problem setting in bioinformatics. One of the main goals is usually to get an overview of the data, preferably with some visualization. Since the data sets often consist of thousands of genes they should also be summarized in the overview, for example, by clustering. Combinations of clustering and visualization are thus of great interest. However, performing the tasks separately and sequentially usually results in a suboptimal solution.

The self-organizing map (Kohonen, 1982, 2001) performs both the clustering and the projection of the data simultaneously. It enables the visualization of the density structure of the data naturally, it is not sensitive to the shapes or the number of clusters, and it provides a groundwork where additional data, if available, can be visualized as well.

This section presents a self-organizing map -based approach for analyzing single genomic data sets. It is based on Publications 1, 2, and 4. The most commonly used alternatives are reviewed and contrasted to the self-organizing map approach.

4.2.1 Self-organizing map (SOM)

Self-organizing map (Kohonen, 1982, 2001) is an algorithm that maps a high-dimensional data onto a lower-dimensional lattice in a fashion that aims at preserving the topology. In particular this means that high-dimensional data can be visualized on a 2-dimensional display in such way that items that are close-by on the display are also close-by in the original space. The virtue of SOM is that it can be used as a clustering and a visualization tool simultaneously, and therefore it facilitates a quick way to search for interesting density structures in the data. It lacks, however, a proper cost function as well as a probabilistic interpretation, and assessing the uncertainty in the SOM visualization in a rigorous way is difficult. Nonetheless, SOM serves as an excellent tool for exploratory data analysis (Kaski, 1997; Vesanto, 2002).

SOM is composed of a lattice of the ***prototypes*** $\mathbf{m}_i \in \mathbb{R}^n$ in the same space as the data $\{\mathbf{x}_j\}$, $\mathbf{x}_j \in \mathbb{R}^n$. The lattice defines the neighborhood relationships between the prototypes and is usually rectangular or hexagonal. The nodes of the lattice are commonly referred to as units or neurons of the map. During the learning of SOM the prototypes are fitted to the data.

The fitting of the SOM to data takes place, in short, by moving the prototypes and their neighbors towards their closest data points in data space until convergence. Intuitively, it can be thought of as a flexible fishing-net that is stretched to cover the observed data points. The initial locations of the prototypes, $\mathbf{m}_i(t = 0)$, may be either random or, for example, on the plane spanned by the two first principal components of the data. The original iterative learning rule for SOM for the ith prototype is

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t)(\mathbf{x}(t) - \mathbf{m}_i(t)), \tag{4.1}$$

where t is the iteration index, $\mathbf{x}(t)$ is the data point sampled from the observed data at iteration step t, and $h_{ci}(t)$ is the ***neighborhood function*** defined over the lattice and centered on the prototype c closest to $\mathbf{x}(t)$, defined by

$$c = \arg\min_i \{||\mathbf{x}(t) - \mathbf{m}_i(t)||\}. \tag{4.2}$$

The neighborhood function $h_{ci}(t)$ is the essence of SOM: it induces the self-organization of the map by allowing not only the closest prototype, but also its neighboring prototypes on the lattice to be moved towards the observed point $\mathbf{x}(t)$. The exact form of $h_{ci}(t)$ may vary, but it is often chosen to be Gaussian. The width and the value of the neighborhood function decrease during the learning, in order to ensure the convergence and a global ordering of the map.

Other crucial parameters of SOM are the size and the dimensionality of the lattice, topology of the lattice, and the training time. The most important of these are the first two. For visualization purposes it is common to choose the dimensionality to be two, but higher dimensionality is possible if the emphasis is on the preservation of the topology. The size of the lattice, that is, the number of prototypes, is sometimes thought to reflect trivially the number of clusters in the data. This, however, leads to a suboptimal usage of SOM, since for visualization purposes it is advisable to use as many prototypes as possible to increase the resolution of the map. This naturally raises the question about over-fitting the map to the data, but it can be controlled by increasing the final width of the neighborhood function while increasing the number of the lattice nodes. The joint effect of these parameters, the map size and the neighborhood width, is sometimes called the *stiffness* of the map, and intuitively it means the flexibility of the lattice in the data space.

There are many variants of SOM, of which one of the most relevant for bioinformatics is the so-called dot-product SOM (Kohonen, 2001). In dot-product SOM the best matching prototype is defined by

$$c = \arg\min_i \{\mathbf{x}^T(t)\mathbf{m}_i(t)\} \tag{4.3}$$

and the update rule is

$$\mathbf{m}_i(t+1) = \frac{\mathbf{m}_i(t) + h_{ci}(t)\mathbf{x}(t)}{||\mathbf{m}_i(t) + h_{ci}(t)\mathbf{x}(t)||}, \tag{4.4}$$

which normalizes the prototype vectors automatically to unit length.

The importance of the dot-product SOM in gene expression analysis is due to the occasional need to normalize the expression profiles (vectors) to the unit length in order to emphasize the similarities in their *shape*. The underlying assumption is that it is important whether the expressions of two genes correlate over various conditions or time points, rather than whether the expressions in the individual conditions are of the same absolute magnitude. In such normalizations the data will become projected to the surface of a hypersphere, and it is natural to require that the SOM lattice should also lie on the same sphere. Allowing SOM to cut through the empty sphere would make its interpretation difficult, and it even might cause SOM to get stuck in local optima in the middle of the hypersphere far away from the data. The dot-product SOM has been used in Publications 1, 2, 4, and 6.

Another variant is the batch SOM (Kohonen, 2001). It introduces an improvement to the update rule that is based on first mapping all the data onto their closest prototype vectors, and then computing the new locations of the prototype vectors as weighted averages of the data. The batch SOM has less convergence problems and it is faster to optimize than the original SOM.

An enhancement to the visualization property of SOM was introduced in the ViSOM algorithm (Yin, 2002) that aimed at preserving the density structure of the data on the lattice by forcing the distances between prototypes to be approximately constant in the data space. Another interesting variant in bioinformatics is the

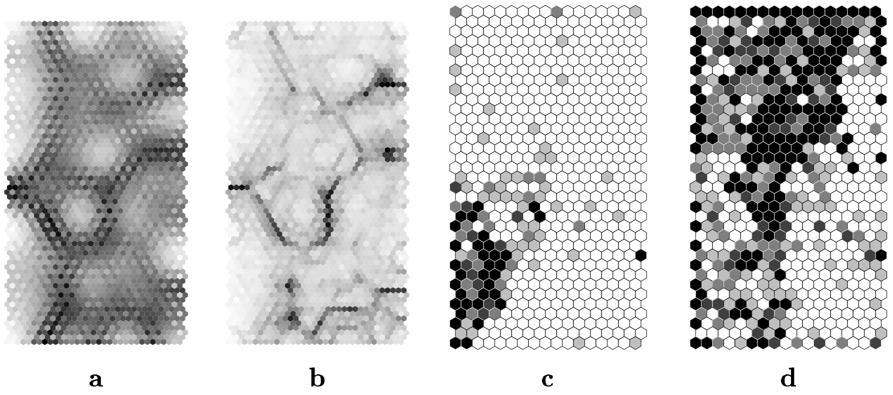


Figure 4.1: **a** U-matrix of the SOM trained with patent abstract data for visualizing the general density structure of the data. Light shade denotes high density of the data. **b** New visualization method reveals smaller details of the density structure. **c** Distribution of the class "optical computing devices" plotted on the SOM. The class characterizes the left lower corner cluster. **d** Distribution of the patent class "electrical digital data processing" revealing which clusters include patents associated with digital data processing. Note: In the distribution figures the dark color reflects a high amount of data. The figure is taken from Publication 1.

SOM of symbol strings (Kohonen and Somervuo, 1998), that has been used for example in the analysis of human endogenous retroviruses (Oja et al., 2003b). For an extensive list of SOM variants (and their applications), see (Oja et al., 2003a).

Summarizing, SOM is well suited to exploratory data analysis (Kaski, 1997) and it has been applied to a multitude of cases including, for example, process industry (Vesanto, 2002; Alhoniemi, 2002), text mining (Lagus, 200), image retrieval (Koskela, 2003), gene expression analysis (Tamayo et al., 1999; Törönen et al., 1999; Golub et al., 1999), and to an analysis of a collection of SOM-related papers (Oja et al., 2003a).

Interpreting the mapping of SOM

In visualization tasks the lattice of the self-organizing map is used as a groundwork on which various aspects of the data can be plotted. The main interest is usually the density structure of the data which gives an overview of the similarities between the data items.

One of the most widely used methods to visualize the density structure of the data on the SOM lattice is U-matrix (Ultsch and Siemon, 1990). It is based on computing the distances of the prototype vectors in the original data space and then visualizing these on the SOM lattice, for example with gray shades. The fundamental reason why this works is that the point density of the prototype vectors in SOM approximates the point density of the data in some sense, although obtaining the exact results of the nature of the approximation in a general case has turned out to be problematic (Kohonen, 1999). Figure 4.1 **a** is an example of U-matrix visualization of the SOM.

The U-matrix display reveals the dominant cluster structure of the data. However, it is not a perfect visualization. In particular, small clusters and areas of homogeneous density may become neglected in the conventional U-matrix visualization. With the new method presented in Publication 1, it is possible to visualize density structures that U-matrix is likely to miss. In short, the new method visualizes the changes in the gradient of the density, from one map unit to another. A large change in the gradient indicates the presence of a cluster border between the units. In a way, the new method can be seen as emphasizing the areas where SOM and U-matrix do not manage to visualize the data density well enough (for more details see Publication 1). Figures 4.1 **a** and **b** demonstrate the differences between the two visualizations in a case study analyzing patent abstract data.

After the areas of interesting changes in the data density have been detected, it is usually of interest to know how these can be interpreted in terms of the original variables. A natural way is to compute local factors by *principal component analysis (PCA)* (see Timm (2002)) from the prototypes as is done in Publication 1. The rationale of the local PCA approach is the close connection of SOM to so-called principal curves (Hastie and Stuetzle, 1989; Cherkassky and Mulier, 1998) that are an extension of the standard PCA. In Publication 2 a variant of the method, computing simple differences between the average cluster prototype and the surrounding area prototype, is used for the same task of interpreting the clusters.

Another very effective way to utilize the SOM groundwork is to plot some auxiliary data, if such are available, on the lattice. In this way various areas of the lattice can be characterized in terms of auxiliary data rather easily, given that the auxiliary data is somehow localized in the data space. Figures 4.1 **c** and **d** show examples of this in analyzing the patent abstract data, and Figures 4.2 **b** and **c** in analyzing gene expression data.

All visualizations of multidimensional data make some compromises. It is not possible in general to visualize high-dimensional data in two or three dimensions without losing information, and the compromise defines what kind of information is lost. The type of compromise is then of interest when choosing the optimal method for visualization. Quantitative evaluation of visualizations is, however, rather difficult.

One possible measure is the *trustworthiness* of the visualization (Venna and Kaski, 2001) (used in Publication 2 and in Publication 6) that measures how well the neighborhood of each data point is preserved in the visualization. The trustworthiness is based on comparing the rank order of the closest data points, for each data point, in the visualization to the rank order of the same points in the original space.

SOM-based analysis of similarities between expression profiles of yeast genes

The first attempts to utilize SOM in gene expression data analysis were published in 1999 (Tamayo et al., 1999; Golub et al., 1999; Törönen et al., 1999). Tamayo et al. (1999) and Törönen et al. (1999) focused on the ability of the SOM to organize similar clusters close to each other, whereas in Golub et al. (1999) the SOM was used rather suboptimally in the sense that the lattice consisted only of few nodes and was used more as a pure clustering method in an attempt to find subtypes of a cancer. However, the visualization abilities of the SOM have largely been neglected. An exception in this tendency has recently been presented in Hautaniemi et al. (2003), where visualization is emphasized in the SOM analysis of gene expression data from cancer patients.

Exploratory data analysis of gene expression data with the SOM and the new

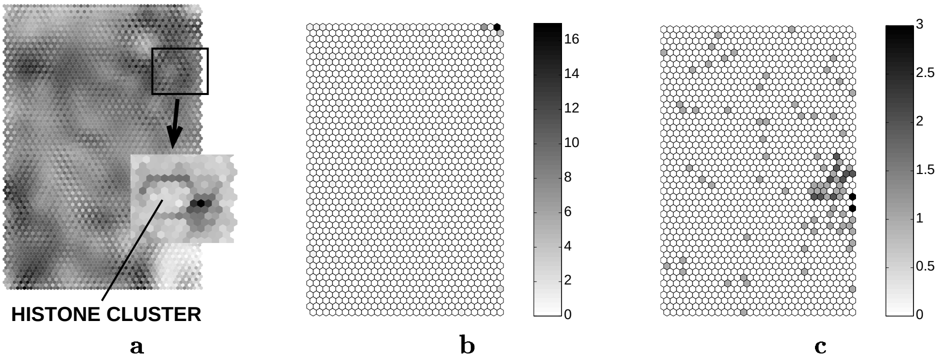


Figure 4.2: **a**: U-matrix visualization of the gene expression data from Publication 2 revealing the density structure of the data. The cluster of histone genes was found with the aid of the new visualization method developed in Publication 1. **b** and **c**: The number of genes belonging to a particular functional class in the SOM unit, revealing the localization of the classes in this expression data. The scale on the right explains the correspondence of the gray shade and the number of the data items in each node. The figure is taken from Publication 2.

visualization methods is performed in Publication 2. The aim is to explore the similarities between the expression profiles of the yeast genes and to demonstrate the advantages of visualization with the SOM-based methods.

The analyzed data is one of the first public gene expression data sets (Eisen et al., 1998). It consists of eight time series experiments measured in different conditions. They have simply been concatenated resulting in a total of 79 dimensions for 2460 genes (for which the functional class could be determined). At this point the genome-wide analysis of gene expressions was not yet a routinely performed task, and the main points were to study the applicability of the SOM-based exploratory data analysis to gene expression data, and to demonstrate the extensive use of data visualization with SOM.

Figure 4.2 **a** shows how visualizations can be used to explore the cluster structures of the data, and Figure 4.2 **b** and **c** demonstrates how auxiliary information, gene functional classification here, can be used to characterize the density structures.

A main result in Publication 2 was that the SOM was found to be more trustworthy than hierarchical clustering, suggesting its suitability for visualizing gene expression data.

SOM-based analysis of the similarities between yeast genome-wide responses to mutations

Of great interest are often also the similarities between the variables, that is, the measurement conditions or perhaps patients, of gene expression data. In fact, most of the time they are the actual interest in the analysis but usually there are so few of them (from tens to hundreds at most) compared to the amount of genes (thousands) that the problem becomes statistically ill-posed for most computational methods. Nonetheless, similarities between the variables can reveal the similarities between patients, individual organisms, and experimental situations, and they are thus worth exploring.

An exceptionally large collection of gene expression data (at that time) was made public in year 2000 consisting of 300 hundred experiments for yeast *Saccharomyces cerevisiae* (Hughes et al., 2000). Each experiment was a knockout mutation or a chemical treatment for one of the yeast genes. Knockout experiments are widely used in biology to study effects of the removal of one specific gene from the genome, and in principle they can reveal which other genes this specific gene affects. In Publication 4 SOM-based exploratory analysis was applied to this data to discover the similarities between the knockout experiments.

Figure 4.3 summarizes the key results of Publication 4, revealing how the groupings presented previously in the literature can be found with SOM-based analyses, complemented with suggestions for new groups and with enhancements to the previous ones.

4.2.2 Other approaches

The main types of clustering methods applicable for the exploratory analysis of a single genomic data set are reviewed in this section. The section covers the majority of the current types of clustering and their possible applications on gene expression data. In addition, the methods are contrasted to the SOM-based approaches applied in this thesis.

Hierarchical clustering

Perhaps the first clustering method applied to gene expression data (Eisen et al., 1998) was similarity-matrix-based *hierarchical clustering* (see Jain and Dubes (1988)). It has established its position as a standard method to analyze gene expression data. Hierarchical clustering can be performed either in an *agglomerative* or *a divisive* way. In the agglomerative method the data points are in the initial phase each in their own cluster, and the clusters are then progressively combined to new clusters, starting from the two clusters closest to each other. This finally results in one cluster consisting of all the data points, and the informative part of the clustering is actually the process that is often visualized as a *dendrogram*, a tree, where clusters are represented as lines that are then connected. The height of the connection visualizes the similarity of the clusters. If a set of clusters is wanted, the tree can be cut at some specific height, or some other criteria can be used to extract a set of clusters from the tree. In the divisive variant the process is started from the big cluster consisting of all the data points, which is then progressively divided into smaller clusters, ending up in singletons.

The advantages of hierarchical clustering include a short computation time and a non-parametric clustering model. Additionally, the simplicity of hierarchical clustering makes it rather easy to interpret the results. There also exist some theoretical results guaranteeing, for certain variants of hierarchical clustering, the performance (see Dasgupta (2002) and the references therein). On the other hand, hierarchical clustering is completely at the mercy of the chosen distance measure, it does not necessarily produce clustering where data points are in reasonably sized clusters, it does not handle uncertainty in a justified way, it is not trivially generalizable for unseen data, and its visualization properties for large data sets are not adequate, since the order of the leaves in the dendrogram is arbitrary.

Hierarchical clustering is the reference method in Publications 2 and 6. Additionally, it is used to summarize the results in Publications 8 and 9.

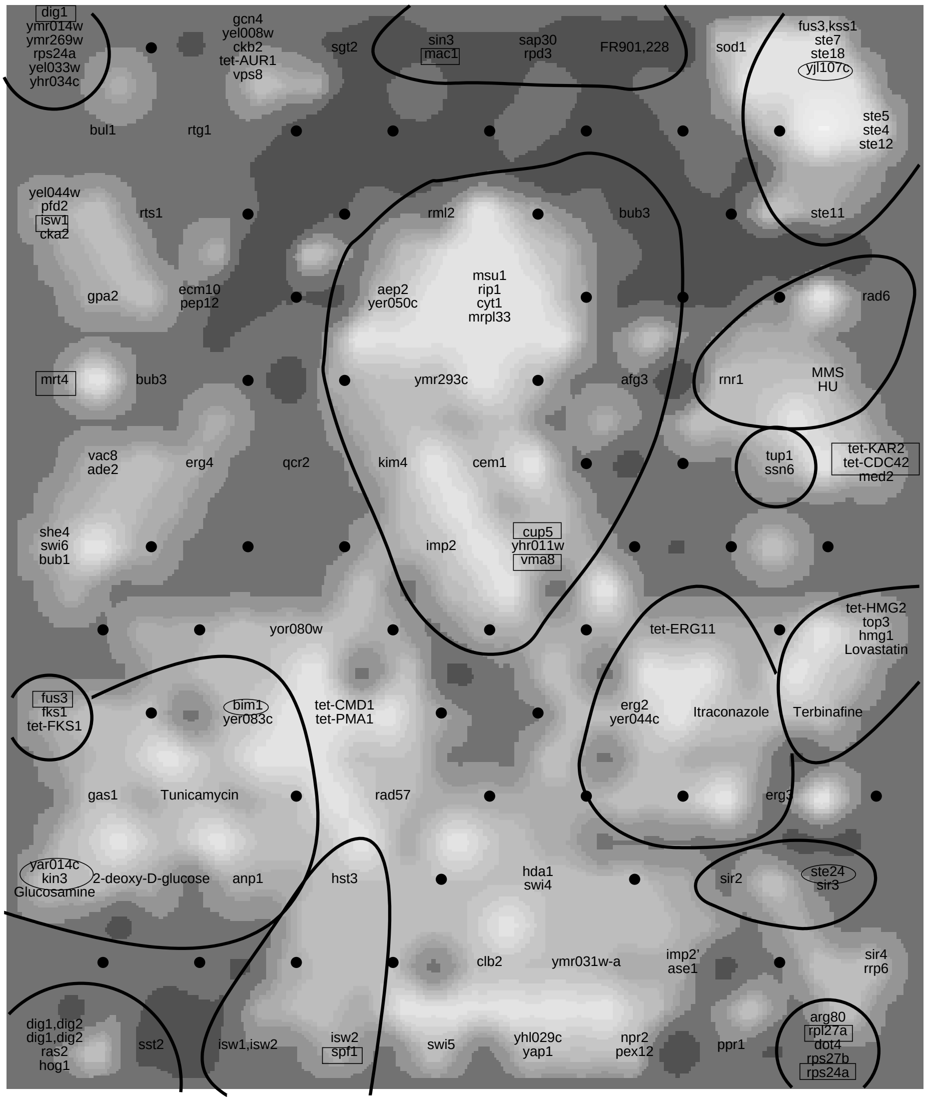


Figure 4.3: A smoothed U-matrix of the SOM that reveals the clusters reported earlier in the literature, and additionally suggests new groupings in Publication 4. The data are the expression profiles from yeast knockout treatments. The labels on the map are the names of the genes that have been knocked out in the yeast strain, and the dots are empty SOM map units. White shade denotes high density of the data (clusters) and dark low density (sparse, non-cluster area in the data space). The areas encircled with a hand-drawn line are the clusters from the literature, and the boxes and the ellipses denote differences between the SOM and the literature clusters: the boxed treatments were grouped to a different cluster in the literature, and encircled treatments are additions to the clusters from the literature, proposed by the SOM.

K-means

K-means clustering (MacQueen (1967); see Jain and Dubes (1988)) is a basic prototype clustering method for multivariate continuous data, where clusters are defined with K parameter vectors $\{\mathbf{m}_k\}$ in the same space as the observed data $\{\mathbf{x}_i\}$. Each data point is mapped to the cluster which produces the shortest distance between the prototype and the data point. K-means is optimized by minimizing the cost function

$$\sum_k \sum_i d(\mathbf{m}_k, \mathbf{x}_i)^2,$$

where $d(\mathbf{m}_k, \mathbf{x}_i)$ is usually the Euclidean distance. K-means has been applied to gene expression analysis in several cases, see for example (Tavazoie et al., 1999; Vilo et al., 2000; Beer and Tavazoie, 2004). It can be very useful when the optimal number of clusters is not crucial, but it is known approximately. Unseen data can trivially be placed in the clusters. As for downsides, it is not straight-forward to visualize the clusterings, K-means always favors Gaussian shaped clusters, and K-means does not treat the uncertainty in clustering properly. Note that if the neighborhood of the SOM is set to zero the SOM reduces to K-means.

K-means-based reference methods are used in Publications 7, 8, and 9.

Mixture-model-based clustering

Mixture models are primarily density estimation methods, that is, they aim to model the probability density distribution from which the observed data could have been sampled. They can, however, be used for clustering since the idea of the mixture model is that the observed data has been generated by C generators, which can be regarded as clusters. Mixture models impose the following model for the data $\mathbf{x}$:

$$p(\mathbf{x}|M, \boldsymbol{\theta}) = \sum_{i=1}^{C} p(\mathbf{x}|\alpha_i, \boldsymbol{\theta}, M) p(\alpha_i|M),$$

where α_i denotes the ith generator, $\boldsymbol{\theta}$ its parameters, and $p(\alpha_i|M)$ describes the probability of the generator given model M. A common model is the mixture of Gaussians where each generator is assumed to generate normally distributed data.

Mixture models have been applied increasingly to gene expression data (Yeung et al., 2001; Medvedovic and Sivaganesan, 2002), because they treat the uncertainty in data in a probabilistic manner. In particular, the uncertainty in cluster assignments, in prototype locations, and in the number of clusters can be handled in a principled way. The last becomes possible, for example, with a Bayesian version of *infinite mixture models* (Rasmussen, 2000), where the amount of clusters is just one parameter among others, and can be estimated based on the data.

The main problems with mixture models are related to unindentifiability of the generators, the visualization of the results, and the computational complexity when the optimization of the model is based on sampling. Unidentifiability becomes a problem when computing the posterior probability. Then any permutation of the generators results in the same posterior probability (if the other parameters of the generators are identical which usually is the case). As a consequence, forming the actual clusters is not a trivial issue. One of the simplest and the most common solutions is to form a new similarity matrix for the data that represents how often the pairs of data objects are generated by the same generator (Medvedovic and

Sivaganesan, 2002). This similarity matrix can then be summarized, for example, by hierarchical clustering (Kerr and Churchill, 2001).

Visualization is problematic for the same reason as in K-means: the clusters and their relations are not trivial to visualize.

A mixture model is used as a reference method in Publication 3, and it is introduced in more detail in Section 4.3.

Graph-theoretic clustering

Graph-theoretic clustering means here the algorithms that treat the observed data as a (weighted) graph where the edge weigths are computed from the distance matrix. The clustering problem then becomes a graph partitioning problem.

One of the first graph-based clusterings applied to gene expression data was Clustering Affinity Search Technique (CAST) (Ben-Dor et al., 1999). It assumes that the observed data is generated by a corrupted clique-graph model, where the disjoint cliques are regarded as the underlying true clusters. Given the similarity matrix of the data and the threshold of significant similarity it searches for the closest clique-graph to the thresholded similarity graph.

Other examples of graph-theoretic clusterings include CLuster Identification via Connectivity Kernels (CLICK) (Sharan and Shamir, 2000), which finds clusters by first detecting tightly connected sets of items, kernels, and then expands them, and a method by Xu et al. (2002), which defines clusters as subtrees in minimum spanning trees.

The biggest advantages of graph-clusterings are that the cluster shape is very non-parametrically defined, and that there usually exists an algorithm whose properties are analytically tractable (or this at least holds for an ideal version of the algorithm). However, the methods have problems: they are not trivially suitable for mapping future data to the existing clustering, visualization is not straightforward, and it is also sometimes tedious to take into account uncertainty in the data.

Spectral clustering

Spectral clustering (see for example Weiss (1999); Ng et al. (2002); Bie et al. (2005); Kluger et al. (2003)) is a specific class of clusterings that has recently gained popularity. In short, it is based on computing the eigenvectors of the affinity matrix of the data items, and inferring the cluster memberships from them. Spectral clustering has been applied to gene expression data to discover subtypes of lymphoma (Ding, 2002) and in combination with biclustering (Kluger et al., 2003) to analyze cancer gene expression data.

The main advantages of spectral clusterings are the short computational time facilitated by linear operations and the ability to find clusters of very diverse forms. The downsides include the possible difficulties in interpretation due to arbitrary forms of the clusters, inability to incorporate future data into the clustering result, and the lack of a natural visualization framework.

Information bottleneck

The *Information bottleneck (IB)* principle (Tishby et al., 1999) is about clustering a discrete variable (for example words in documents) in such a way that the resulting clusters are maximally informative with respect to some discrete auxiliary variable (for example topics of the documents).

The name "information bottleneck" is due to the principle it is based on: given some discrete variable X and an auxiliary discrete variable Y associated with it, find the clustering $\tilde{X}$ such that the following cost is minimized:

$$\mathcal{L}(p(\tilde{x}|x)) = I(\tilde{X}; X) - \beta I(\tilde{X}; Y).$$

Here β is the Lagrange multiplier controlling the resolution of the clustering. In other words, the information bottleneck tries to maximize the mutual information between the clusters and the auxiliary variable, and minimize the mutual information between the clusters and the original variable, the clusters $\tilde{X}$ being the bottleneck.

Although IB, in principle, assumes discrete co-occurrence data, it has also been used for clustering a single gene expression data set through so-called Markovian relaxation and information bottleneck (Tishby and Slonim, 2001). This method has been applied to gene expression data measured from a colon cancer (Slonim, 2002) and is one possible way to apply information bottleneck methods on continuous data. The method is theoretically interesting, but not intuitively interpretable and requires the determination of the correct step t of the emerged cluster structures, which may be difficult in practice.

Biclustering and subspace clustering

A very popular group of methods in the bioinformatics community are the biclustering methods (also called two-way clustering and co-clustering; cf. Getz et al. (2000); Lazzeroni and Owen (2002); Tanay et al. (2002); Sheng et al. (2003); Kluger et al. (2003)). Their aim is to find clusters of data by grouping both the columns and the rows of the data matrix. This is an appealing aim in particular in clustering gene expression data, since often a set of genes (rows) is assumed to behave similarly only in a subset of conditions (columns). For reviews of biclustering applied on biological data, see Tanay et al. (2005); Madeira and Oliveira (2004).

A special, biologically motivated type of biclustering is introduced by Segal et al. (2003a). The gene expression matrix is decomposed into "cellular processes" that consist of a set of genes and a set conditions. The authors claim that the method is able to find clusters of genes that reflect the known cellular processes better than, for example, the method presented in Lazzeroni and Owen (2002).

Another biologically oriented approach has been presented in (Ihmels et al., 2002), where a heuristic algorithm iteratively clusters genes and conditions to find a cluster of genes that are active in some subset of conditions.

If the data are discrete and they can be interpreted as co-occurrence data, certain information bottleneck methods (Friedman et al., 2001), and a closely related information theoretic co-clustering introduced by Dhillon et al. (2003), can be seen as a special kind of bi-clustering. However, the framework of IB methods is more general (for more details, see Section 4.3 and 5).

A group of clustering methods strongly related to biclustering are *subspace clustering methods*, or *projected clustering methods*, see for example (Agrawal et al., 1998; Parsons et al., 2004). Their aim is generally the same as in biclustering: find subsets of items from the data matrix, but in these cases the algorithms are designed for very large and high-dimensional databases. The motivation for them comes from the fact that it is in general very difficult to find clusters in a high-dimensional data space, but the task comes much easier if the clusters are assumed to lie in some subspace. From the methodological perspective, they generally differ from biclustering methods in their grid-based way (histogram) to estimate the density of the data, which makes the algorithms fast and suitable for large data sets but suboptimal in the sense of probabilistic modeling. For applications of subspace clustering to genomic data, see for example Yip et al. (2004).

Since the biclustering methods form a very diverse set, it is difficult to say anything general about their methodology. They may provide a good alternative when the data matrix has many columns that are assumed to have some latent group structure, or the clusters can be assumed to exist in some subspace of the original data space. In particular, they may ease the interpretation of the clusters. Their emphasis is on clustering, and consequently they do not trivially offer any visualization possibilities.

Other clustering methods

Gene shaving (Hastie et al., 2000) is a clustering technique specifically designed for gene expression data. It finds gene clusters (a set of rows) in which genes have similar expression, but additionally the gene expression varies maximally over columns (treatments). The name "shaving" comes from the technique that discards the genes from the cluster that has the smallest contribution to the variance over columns. The interpretation of the clusters is likely to be easy since the method specifically produces sets of genes behaving differentially in different conditions. In addition, the method can be supervised by treating the rows or columns partially or fully labeled. However, the visualization of the clusters and cluster relationships is not trivial, and the algorithm is computationally intensive due to its iterative nature.

4.2.3 Discussion

This section introduced the SOM-based approach for exploratory cluster analysis of a single gene expression data set, demonstrating new interpretation methods and visualization properties of SOM. The clustering approaches proposed in the literature were reviewed, concentrating on the representative main types of clusterings and on the applications to the gene expression data.

Most of the previously introduced clustering approaches for gene expression analysis do not focus on visualization. On the other hand, they conveniently summarize the data as lists of similar genes. This is naturally desirable and understandable, but often results in an over-optimistic view of the data at hand: that there would exist some clear-cut clusters in the data. Most often this is not the case, but the lists merely represent a partitioning of the expression space, more or less an arbitrary one. Visualization, as performed simultaneously to clustering with SOM, conveys the intuition of the type of data density structure to the analyst who then has a possibility to asses the cluster quality intuitively. However, if the analyst already has enough knowledge about the density structure of the data and the objective is purely to generate lists of similar genes, then methods focusing solely on clustering are more suitable than SOM.

Another point, also related to the visualization, deserves extra attention: the shape of the clusters does not need to be determined in the SOM reduce *a priori*, since the clusters are determined based on the visualization. This is in contrast to many other clustering methods, and is not usually recognized in the literature. The usual claim is that the SOM is equivalent to K-means with respect to the number of clusters, but this is due to misinterpreting the map nodes as clusters. The number of map nodes only sets the resolution at which the data space is quantized, and the actual clusters should be inferred from the map lattice based on the density structure of the data.

While none of the clusterings in the literature are trivially applicable for visualization, it is naturally possible to visualize the clusters as a post-processing step. For example, it is very easy to use the SOM for visualization of the clusters obtained by other means, and then potentially revise them based on the visualization. An example of this is Figure 4.3.

While the SOM-based approaches presented here are clearly applicable to gene expression data, true biological discoveries are not likely to be made based on the analysis of a single genomic information source alone. Already in this section some promising interpretations were made based on some auxiliary information, like functional classes of the genes. The next section and the subsequent chapter introduce advances in the data analysis of multiple genomic information sources.

4.3 Integrating discrete auxiliary information into cluster analysis

A crucial problem in unsupervised data analysis is that the similarity metric is usually arbitrary. For example, in cluster analysis conventionally a data sample is assigned into that cluster for which the Euclidean distance between the sample and the cluster is shorter than between the sample and any other cluster. However, the Euclidean distance implicitly assumes that the dimensions, or features, of the data space are equally important which is not necessarily true. In particular, some dimensions consisting of mainly noise may dominate, locally or even globally, the variation in the data and obscure the interesting cluster structures. The problem is then how to choose the correct metrics for the analysis.

Often there exists some relevant auxiliary information about the data items. This auxiliary information can be regarded as prior information about the data, that should be utilized in the analysis. For example, if it is of interest to find gene clusters or study density structures of the expression data that are maximally related to the functional classes of the genes, the functional classification should be incorporated into the analysis. In particular, if such relevant data is available, the correct metric should be inferred from this auxiliary information, and the primary data (expression) under study. Developing the methods for this problem setting provides an interesting machine learning problem that is highly relevant to genomic data analysis.

Any type of method that affects the feature space based on some auxiliary class information can be seen as a method changing the metric. For example, a classical method to find linear combinations of the original components that best separate the known classes is *linear discriminant analysis (LDA)*, see for example Timm (2002). Application of LDA would then produce a projection of the data to the dimensions that would be very informative in the sense of a linear classifier that assumes that the class distributions are Gaussian. However, that metric would not necessarily be optimal for any other method. The metric estimation should be compatible with the actual analysis method, or a very general one.

The recently introduced *learning metrics* concept (Kaski et al., 2001; Kaski and Sinkkonen, 2004; Sinkkonen and Kaski, 2002) estimates a metric for multivariate, continuous primary data that optimally reflects the changes in the distribution of auxiliary data. The learning metrics is a general principle that can be used as a preprocessing step prior to the analysis method of choice, or it can be integrated to the method. This section reviews two algorithms using the learning metrics: Self-Organizing Map in learning metrics and discriminative clustering, and their applications to genomic data.

The section is organized as follows: First, the approaches related to learning metrics, presented previously in the literature are reviewed. Second, the learning metrics concept and the SOM in learning metrics applied to genomic data are introduced. Third, related clustering methods presented in the literature are reviewed. Fourth, clustering methods using auxiliary data are reviewed. Finally, the discriminative clustering, which is motivated by the learning metrics, is introduced with its application.

4.3.1 Existing metric estimation methods that use auxiliary data

Assume that some expert knowledge about the data $\{\mathbf{x}_i\}$, $\mathbf{x}_i \in \mathbb{R}^n$, exists in the form of class labels $\{c_i\}$. If one then weighted or pruned dimensions as a preprocessing stage in a way that neglects the class labels, the preprocessing would be suboptimal for the tasks that use the class labels. A better option is to preprocess variables in such a way that the remaining features are maximally relevant for auxiliary data. This leads either to the methods of *feature selection* or *feature extraction.* Feature (variable) selection usually refers to methods that select a subset of original features, but do not modify them, whereas feature extraction is used to refer to the methods that transform the original features, for example by taking a linear combination of them. These methods are not traditionally considered to be methods changing the metric, but they can be regarded as such.

Feature selection

Feature selection methods can be categorized into three groups: *filter, wrapper, and embedded* methods (Guyon and Elisseeff, 2003). The feature selection methods are most often used in connection with classification tasks, but are in principle applicable also to any other method relying on some similarity metric. The methods of filter category are independent of the actual analysis method; they are used as a pure preprocessing step. Wrapper methods treat the actual analysis method as a black box, and iteratively improve the analysis result. In embedded methods the feature selection is integrated into the analysis method. While feature selection facilitates good interpretability of the results, assuming the original features are interpretable, the methods using it can be suboptimal in computational performance and accuracy. The feature extraction methods will be the main focus in this section, but note that feature selection has been applied to genomic data problems for example in classification (Xing et al., 2001) and was observed to improve the classification results.

Feature extraction and metric estimation

Feature extraction is sometimes also called feature construction or metric estimation, but the objective is the same: to improve the analysis by changing the metric with some auxiliary data. Here we review methods that can be used to produce a new metric that is then available for utilization in the analysis method of choice.

Perhaps the simplest example is the traditional Linear Discriminant Analysis (LDA, see Timm (2002)). LDA finds the components in the data space that maximally separate the given classes of data. It is traditionally used for classification tasks, but also for visualizations for which the maximal separability of the subgroups of the data is important. LDA assumes that the distribution of the data in each class is a Gaussian with a covariance matrix common to all classes.

More recent linear component methods have been proposed, for example, by Peltonen and Kaski (2005), based on likelihood for a generative model, and by Torkkola (2003), based on mutual information between an auxiliary variable and the components, in Shental et al. (2002); Bar-Hillel et al. (2003) based on emphasizing the components where the variation is largely due to between-class variations, and in Xing et al. (2003) based on minimizing the within-class distances of data by quadratic optimization while simultaneously avoiding collapsing the data to one point.

Globally non-linear metric estimation approaches have been presented in Chang and Cheung (2004), based on locally linear metric transformation for every data point pair defined similar by auxiliary data, and then applying the transformation also to the neighboring points with an estimated weight. In Zhang et al. (2003) the metric is learned with non-linear regression.

For a more detailed review of the methods related to metric estimation from the technical perspective, see for example Sinkkonen (2002); Peltonen (2004).

4.3.2 Learning metrics

The *learning metrics (LM)* principle (Kaski et al., 2001; Sinkkonen and Kaski, 2002; Kaski and Sinkkonen, 2004) is a framework in which the metric of the primary data space, used in data analysis, is learned using auxiliary data. The primary data space is a multidimensional real space, and the auxiliary data is in the form of the class labels. The key idea is to make distances in the primary space proportional to the changes in the class distribution.

More formally, given a paired data set $\{\mathbf{x}, c\}$, $\mathbf{x} \in \mathbb{R}^n$ and c multinomially distributed, the learning metric is defined locally by

$$d^2(\mathbf{x}, \mathbf{x} + d\mathbf{x}) = D_{KL}(p(c|\mathbf{x}), p(c|\mathbf{x} + d\mathbf{x})) \equiv \frac{1}{2} d\mathbf{x}^T \mathbf{J}(\mathbf{x}) d\mathbf{x}, \tag{4.5}$$

where $\mathbf{J}(\mathbf{x})$ is the Fisher information matrix that describes the change of $p(c|\mathbf{x})$ with respect to the coordinates of the primary data space.

In practice, the problem is then to estimate the conditional probability distributions $p(c|\mathbf{x})$, which can be done, for example, by Bayes' rule from some standard estimator of the joint distribution $p(c, \mathbf{x})$.

Learning metrics bears a resemblance to supervised and discriminative learning, in particular to classification. In general, the aim of supervised modeling is to model the conditional distribution $p(c|\mathbf{x})$. Conventional classification is a special case where the primary interest are the changes of the most probable class in the primary space. The main differences between the purely supervised methods and LM are summarized as follows: in LM the aim is not to model $p(c|\mathbf{x})$ itself, but only to transform the metric in $\mathbf{x}$-space; the data analysis still takes place in the primary space after that. Additionally, LM does not concentrate on the changes in the dominant class, in contrast to conventional classification, but takes into account the changes in the whole *class distribution*.

The learning metrics can be regularized with the Euclidean metric in order to avoid the possible singularities in pure learning metrics, see Publication 6. The first papers used rather crude approximations in computing distances, but it is also possible to compute non-local distances more accurately with learning metrics by using various approximations for path integrals representing the distances in the primary data space (see Peltonen et al. (2004)).

After the metric has been learned, it is possible to use various data analysis methods in the new metric. Examples of these are the self-organizing map (Kaski et al., 2001) and Sammon's mapping (Peltonen et al., 2002), of which the SOM in learning metrics will be reviewed in more detail here. In addition to the original publications, extensive introductions to the learning metrics principle can be found in Kaski and Sinkkonen (2004); Sinkkonen (2002); Peltonen (2004).

Self-organizing map in learning metrics (SOM-LM)

The interpretation of the SOM is often partly based on visualizing distribution of some class information on the SOM lattice (see for example Section 4.2). Unexpected localizations of the classes on the lattice can then be held as hints that the classes depend on the primary data features. However, if the class information available is of primary relevance in the analysis, this is suboptimal in the sense that the SOM itself has no information of the classes, and most likely will not represent the class distributions optimally. Particularly, it may concentrate on variation that is irrelevant to the auxiliary information.

SOM in learning metrics (SOM-LM) (Kaski et al., 2001) is a recently developed method that fits a SOM to the continuous primary data to represent the changes in some auxiliary information. Examples of applications are companies described as a feature vector based on their financial statements and as the auxiliary information the class label telling whether they went bankrupt, or gene expression profiles with the functional class labels.

The main difference between SOM and SOM-LM using local distance approximations is the best matching unit search in learning metrics. With the natural gradient, the update rule remains the same (Kaski et al., 2001).

SOM-LM of yeast gene expressions and functional classes

The functional classes of genes describe approximately the roles of the genes in the cell. Despite the occasional crudeness of the functional classification, the classes are one of the most important ways to interpret and summarize groups of genes, as well as various visualizations. They also enable interpretation of gene expression experiments by letting the researcher see which functions are coordinated in concert in some specific experiment, for example activated or perhaps shut-down altogether.

In Publication 6 SOM in learning metrics is used to visualize the gene expression changes related to changes in functional class distribution. This enables more efficient visualization of the functional classes than visualizations in arbitrary metrics. It may also reveal novel combinations of the existing functional classes, in which the genes behave similarly in experiments despite that they belong to different functional classes.

In Publication 6 SOM-LM was applied to yeast gene expression data measured in 300 mutations and chemical treatments (Hughes et al., 2000), and to a human gene expression data measured in different tissues (Su et al., 2002). The auxiliary information used in learning metrics was the MIPS functional classification (Mewes et al., 2002) for the yeast, and a discretized expression of the homologous genes in mouse for the human. Figure 4.4 shows SOM-LM of the yeast knock-out data, visualizing the density structures of the data, and some sample clusters. In Publication 6 SOM-LM was found to represent the auxiliary data better then a dot-product SOM (see Section 4.2.1) for both the yeast and the human data sets.

4.3.3 Clustering using auxiliary data

The separate metric estimation stage enables the use of many data analysis techniques. On the other hand, it can be seen as a downside of the framework, since the estimation of the metric is not optimized with respect to the specific analysis goal, for example clustering. In order to overcome this, one has to embed the metric estimation principle into the data analysis method of interest. The clustering done by utilizing some relevant auxiliary information can be regarded as a clustering in transformed metrics. Some existing approaches presented in the literature are first reviewed, and then a clustering in learning metrics is introduced, together with an application to gene expression data.

Joint distribution modeling

To start from the obvious choice, it is possible to use joint distribution models for clustering. For example, a method called MDA2 (Hastie et al., 1995) is a mixture model for $p(c, \mathbf{x})$. As the name implies these methods model all the variation present in the original features and the classes. This means that, in principle, either data, primary or auxiliary, could dominate the other, depending on the preprocessing.

Semi-supervised clustering

Clustering with auxiliary data is related to a recent class of methods called *semi-supervised clustering* or *clustering with constraints*, see for example Basu et al. (2004); Chang and Cheung (2004); Bilenko et al. (2004). The general aim in these models is to improve the clustering using auxiliary data available for all or for a subset of the items. Usually the auxiliary data consists of pairwise (dis)similarities or some class information, and the aim is to put similar items, in the sense of the constraints, into the same cluster. Note that the nomenclature is not fully established yet since occasionally this type of methods are also referred to as *supervised clustering* (Daumé and Marcu, 2004; Eick et al., 2004). Care should be taken to avoid confusing them with supervised learning methods, in particular with semi-supervised classification, where the aim is to improve classification using unlabeled data.

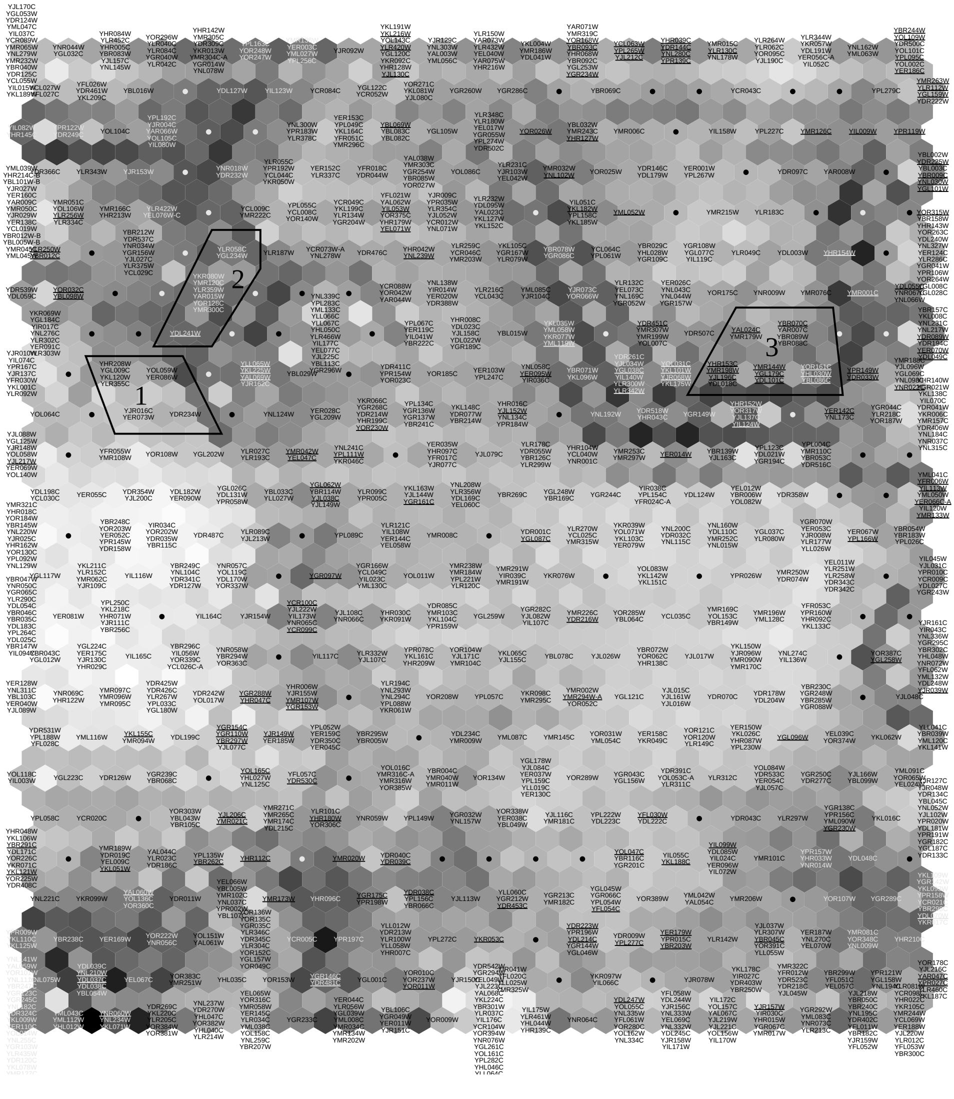


Figure 4.4: U-matrix visualization of the gene expression data measured from 300 knock-out mutations. The underlined genes are the ones for which the metric changed the most in comparison to the Euclidean one. The enumerated clusters are sample clusters: 1: A cluster associated with mitochondria, 2: Localization of purine biosynthesis pathway, and 3: An area where the metric has changed. The genes in area 3 largely have an unknown function; some are associated to transcription and DNA repair. The figure is taken from Publication 6.

One of the first attempts to supervise clustering is presented by Becker (1996). The motivations in that work stem from modeling the perceptual processing stages, but the abstract aim of the models is to maximize mutual information between the representations (outputs) of different inputs. The two variants called *discrete Imax* and *binary Imax* can be regarded as clustering methods, but the framework is more general.

Another early semi-supervised clustering is presented in Wagstaff et al. (2001), which presents an algorithm for constraining the standard K-means clustering. In the algorithm data is assigned to clusters that do not violate auxiliary data constraints. As a result, the clusters are not necessarily local in any sense in the primary data space. Another variation of a standard clustering, in this case of the hierarchical clustering, is introduced in Klein et al. (2002) to incorporate the similarity constraints into the distance matrix of the data.

A probabilistically motivated approach is presented in Shental et al. (2003) that proposes to fit a Gaussian mixture model to the data with equivalence constraints that can be both of similarity or of dissimilarity type.

In Basu et al. (2004) a probabilistic framework for semi-supervised clustering is introduced that unifies the two, alternative main principles of the previous approaches: i) auxiliary information-sensitive assignment of the data into clusters, and ii) metric learning from the data. The framework is based on so-called Markov random fields, and enables the use of both similarities and dissimilarities. A variation of the method is presented in Bilenko et al. (2004).

Applications of (semi)supervised clustering to genomic data include at least a supervised clustering of gene expression data measured from various cancer types (Dettling and Bühlmann, 2002). The aim in this approach was to cluster gene expression profiles of various cell lines and patients in a similar incremental way that is applied in (Ben-Dor et al., 1999) but with an additional goal to find sets of genes that discriminate maximally well the given cancer classes.

An application of a kind of supervised clustering to discover molecular pathways based on gene expression and protein interaction data was presented recently in Segal et al. (2003c). The core of their idea was to model the joint distribution of pathway indicators (clusters) and gene expression given all the available binary protein interactions. The model assumed that each gene belongs to one pathway and given that, assumed that the expression of a gene in each experiment was independent and normally distributed (so-called Naive Bayes model). The effect of the interaction data was modeled with so-called binary Markov networks that could easily be combined with the expression model, and in effect it induced a potential over gene pairs that favored interacting genes in the same pathways. The model was shown to be better than pure probabilistic clustering of expression data or pure graph-theoretic clustering of interaction data. Note that from the perspective of semi-supervised clustering this approach is close to a joint distribution modeling, and it does not change the metric of the primary (expression space), only the assignment of the data to the clusters (pathways).

In general, semi-supervised clustering methods can not be regarded as joint distribution models, because not all probability distributions are modeled in them, that is, the clustering is discriminative in some sense. For example, the uncertainty of pairwise constraints is usually not modeled.

Information bottleneck

As discussed in Subsubsection 4.2.2, the Information bottleneck (IB) principle (Tishby et al., 1999) is about clustering a discrete variable (for example words in documents) in such a way that the resulting clusters are maximally informative with respect to some discrete auxiliary variable (for example topics of the documents).

IB can be interpreted as a semi-supervised clustering of discrete data. The method is information-theoretically well-justified, but the requirement of the discrete data is a severe limitation in genomic data analysis. Simple discretization of a data matrix is likely to produce clusters that are not easily interpretable, since they are not local in the data space. So far no applications of basic IB to genomic data in the sense of semi-supervised clustering have been presented in the literature. For an application without auxiliary data, see Section 4.2.2

4.3.4 Clustering in learning metrics: discriminative clustering (DC)

Discriminative clustering (DC) (Sinkkonen and Kaski, 2000, 2002; Kaski et al., 2005) is a prototype-based clustering method for continuous data paired with discrete auxiliary data (class labels). It aims at producing clusters that are local in the primary data space and have as homogeneous class distribution as possible. It is strongly related to the learning metrics principle in the sense that if the number of clusters approaches infinity, the DC becomes clustering in learning metrics (Kaski and Sinkkonen, 2004).

Given paired data set $\{\mathbf{x}, c\}$, where $\mathbf{x} \in \mathbb{R}^n$ and c is multinomially distributed, a set of prototype vectors $\{\mathbf{m}\}$, $\mathbf{m} \in \mathbb{R}^n$, and a set of prototype class distributions $\{\psi\}$ associated to them are defined. The cost function of the DC can be be written as follows

$$E_{DC} = \sum_j \int y_j(\mathbf{x}; \boldsymbol{\theta}_j, \sigma) D_{KL}(p(c|\mathbf{x}), \psi_j)\, p(\mathbf{x})\, d\mathbf{x}\ , \tag{4.6}$$

where $D_{KL}(p(c|\mathbf{x}), \psi_j)$ is the Kullback-Leibler divergence between the observed class distribution $p(c|\mathbf{x})$ and the prototype distribution ψ_j, and $y_j(\mathbf{x}; \boldsymbol{\theta}_j, \sigma)$ are the unimodal membership functions of the clusters centered at $\boldsymbol{\theta}_j \in \mathbb{R}^n$ having the width parameter σ and fulfilling $0 < y_j(\mathbf{x}) < 1$, $\sum_j y_j(\mathbf{x}) = 1$. Note that if the distortion measure would be replaced by the conventional Euclidean distortion measure, one would obtain the cost function of the soft vector quantization (or soft K-means clustering).

The DC is optimized with respect to the parameters ψ_j, $\boldsymbol{\theta}_j$ and σ. The prototypes ψ_j parameterize multinomial distributions that describe the local class probabilities. Optimization of ψ_j and $\boldsymbol{\theta}_j$ takes place with stochastic approximation, for instance, and σ is chosen with a validation set. For more details on the method, see (Sinkkonen and Kaski, 2002; Sinkkonen, 2002; Kaski et al., 2005).

An important property of DC is that it searches for clusters that have a homogeneous auxiliary data distribution. This is in contrast to most semi-supervised clusterings that typically aim at clusters consisting of one class. This unique flexibility of DC in principle enables the discovery of unexpected combinations of class information, for example a group of functional classes in which genes are expressed similarly in specific experiments.

DC of yeast gene expressions and functional classes

Discriminative clustering has been applied to gene expression data and functional classes in Publication 3.

The gene expression data was taken from Eisen et al. (1998). It consisted of 8 time series experiments, altogether 79 dimensions. The functional classes were from MIPS (Mewes et al., 2002). Since the classification is hierarchical and the smaller classes included occasionally too few genes, only the main classes were utilized here.

DC was compared to MDA2 (Hastie et al., 1995) and found to outperform it when measured with empirical mutual information.

4.3.5 Discussion

This section reviewed the learning-metrics-based explorative clustering methods to gene expression data and related methods, from the perspective of applicability to gene expression data. It was argued that if suitable auxiliary information exists, it should be used in order to incorporate all the available prior information to the analysis. This is especially important in analyzing complex systems, such as the cell, and in presence of noisy data.

The use of conventional metrics that implies equal relevance of the dimensions is of course often very sensible. Its use is logical in situation where there naturally exists no prior information about the importance of the features, and the majority of analysis tasks are of this kind. Thus, as such, the use of the conventional metrics is not a bad idea, but it is important to realize that the choice of the metric affects the objective of the analysis, and that it actually may affect the results of the analysis crucially.

The section dealt with transforming the metrics based on discrete auxiliary information, but it is naturally possible to change the metric based on the primary data alone. Subspace/biclustering methods (see Section 4.2) are examples of that and in the metric estimation sense they are related to the methods presented in this section. These two types of metric transformations can also be integrated, for an example see (Liu et al., 2004).

The type of auxiliary information deserves some attention. The learning metrics methods, SOM-LM and DC, utilize multinomially distributed auxiliary information, usually class information. Similarly, all the reviewed (semi)supervised methods utilize discrete side information, often class labels, but some of them may also use pairwise similarity constraints. The pairwise constraints are more general than class labels, allowing more structured information, for example Gene Ontology graphs, to be taken into account more easily. The type of auxiliary data is one of the foci for future research to improve learning metrics-based methods. The first approaches have already been made, and are presented in Section 5, where the auxiliary data can be continuous.

As a conclusion, learning metrics approaches are distinguished from the other methods by the following characterization: they are designed for unsupervised, topography-preserving data analysis of continuous data under a supervised metric learned from relevant auxiliary data. The methods allow new unlabeled data to be incorporated into analysis after the model has been trained with the labeled data. They belong to the field of general machine learning, but are especially suitable for genomic data problems.

Chapter 5

Exploring dependencies between genomic data sets by clustering

The use of auxiliary data to guide the data analysis, described in Section 4.3, can be seen as a special case of combining two sources of information into the analysis. Being able to combine multiple data sources is extremely important especially in systems biology problems. Systems biology views the cell and organisms as systems where various components of the cell operate in a highly interactive manner. Subsequently, each separate information source for the cell, that is, gene expression, gene ontology, metabolic pathways, DNA sequence, metabolic state, protein concentrations, etc., offers only a limited perspective to the functionality of the cell. In order to understand the systemic behavior of the cell, all the available information sources need to be combined in the analysis. In practice, each source is represented by a set of features (variables) for the same set of objects (for example genes) of which there is a set of observations (the data).

Consider one set of objects, two sources of information both described with continuous multivariate features, and a data set obtained from both of them for the objects. Intuitively, it would be possible to search for at least the things that are i) *in common between the data sets*, or ii) the things that are *data-set-specific.* The problem with the latter goal is that it becomes ill-defined very easily when the data sets contain a lot of noise. The noise is usually assumed to be independent between the sources, hence, the effects extracted as data-set-specific would very easily consist mainly of noise. On the other hand, the first goal seems tempting precisely for this reason: the effects common to the several data sources would by definition be free from the data set specific noise. This is an important property especially with microarray data that is notoriously noisy.

The explicit search for common aspects between co-occurring data from two information sources, X and Y, can be formulated as an analysis of statistical dependencies between representations, $f_x(\mathbf{x})$ and $f_y(\mathbf{y})$, for the data sets. This is equivalent to stating that $p(f_x(\mathbf{x}), f_y(\mathbf{y})) \neq p(f_x(\mathbf{x}))p(f_y(\mathbf{y}))$. If the representations are chosen to be clusters, the setting provides a way to summarize the data as *dependent subgroups.* Other, more traditional, formulations include for example *canonical correlation analysis (CCA)*, (Hotelling (1936); see Timm (2002)) in which the representations are components in the feature spaces, or its recent generalizations (Fyfe and Lai, 2000; Akaho, 2001; Bach and Jordan, 2002; Klami and Kaski, 2005).

The key property of the modeling of dependency is that one is focusing on the dependencies *between* the feature sets X and Y, and not on the dependencies of variables inside X or Y. This differentiates the dependency modeling from joint distribution modeling, where all the dependencies present in the joint space (X, Y) are modeled. For example, given gene expressions measured from lung cancer and from breast cancer, with dependency modeling it would be possible to find dependencies in gene expression between the two cancers. In contrast, modeling the joint distribution of all the gene expression data would take into account all the dependencies between the individual variables (genes) in the data, and it would model the inter-dependencies of the cancers suboptimally.

This chapter presents the third main contribution of the thesis: new methods for explorative cluster analysis of the dependencies between genomic data sets and their applications. First, a set of existing methods applicable for dependency modeling are reviewed, and, second, the methods integrating two data sets into the clustering analysis are reviewed. Next, two new dependency analysis methods, that are extensions of discriminative clustering (see Section 4.3.4), are introduced: *maximum a posteriori*-DC and associative clustering. Their applications to genomic data sets, and related work in the literature are also summarized. Finally, an extension of associative clustering to multiple data sets is presented in Section 5.4 with an application to characterizing the yeast stress reaction and its regulation.

5.1 Existing approaches combining data sets

This section focuses on two groups of methods: classical methods used in dependency analysis, and methods combining data sets by clustering. The new methods described later are built on the concepts used or introduced in these two method groups.

5.1.1 Classical methods for measuring dependency

Classical dependency analysis here means statistical methods that were designed for estimating and/or finding the dependency between two (multivariate) variables.

Correlation coefficient

Correlation is one the most common distance and association measures used in data analysis. There exist many variants of correlation coefficients, both parametric and non-parametric, of which the most popular are perhaps Pearson correlation coefficient, Spearman's correlation coefficient, and Kendall's Tau (see Conover (1971)).

By far, the most widely used of these is the Pearson correlation coefficient for the linear association between the two real-valued, co-occurring random variables, $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ and $\mathbf{y} = [y_1, y_2, \ldots, y_n]$, where $x_i, y_i \in \mathbb{R}$:

$$\rho_P = \frac{(\mathbf{x} - \bar{\mathbf{x}})^T (\mathbf{y} - \bar{\mathbf{y}})}{\sigma_x \sigma_y}, \tag{5.1}$$

where $\bar{\mathbf{y}}$ and $\bar{\mathbf{x}}$ denote the means of the variables and the σ the respective standard deviations. Pearson correlation is proportional to the mutual information between the variables if they are assumed to be normally distributed (Kullback, 1959).

Spearman's correlation coefficient and Kendall's Tau are nonparametric association measures (see Conover (1971)). Both of them use the ranks of the data items instead of the actual values of x and y. As measures they are more robust due to their non-parametricness, but their interpretation may be less clear since they do not trivially translate to statistical dependency.

Canonical correlation analysis

Canonical correlation analysis (CCA Hotelling (1936); see Timm (2002)) is the basic method when measuring the dependency between two multivariate variables $X \in \mathbb{R}^n$ and $Y \in \mathbb{R}^m$. It finds linear components in both data spaces, in such a way that the correlation between the components is maximal in the sense of the Pearson correlation coefficient. This corresponds to maximization of mutual information if both variables are normally distributed. CCA can be presented as the following generalized eigenvalue problem:

$$\mathbf{C}\xi = \lambda\mathbf{D}\xi, \tag{5.2}$$

where $\mathbf{C}$ is the covariance matrix of the concatenated data $[\mathbf{x}^T\ \mathbf{y}^T]$, $\mathbf{D}$ is the block-diagonal matrix of the original covariance matrices, $\lambda = 1 + \rho$ are the eigenvalues (ρ being the canonical correlation), and ξ are the eigenvectors.

CCA is related to the mutual information between the variables X and Y as follows (Kullback (1959); see Bach and Jordan (2002)):

$$I(X, Y) = -\frac{1}{2}\log\prod_i(1 - \rho_i^2), \tag{5.3}$$

where ρ_i are the canonical correlations between $\mathbf{x}$ and $\mathbf{y}$.

There also exist a number of generalizations and extensions of CCA, see for example Timm (2002) for a partial CCA (which removes the effect of a third variable), and Kettenring (1971); Bach and Jordan (2002) for various extensions to multiple variables, often called with a common name *generalized CCA (gCCA)*.

A gCCA with a similar simple connection to mutual information was presented, for example, in Bach and Jordan (2002), and it can be expressed similarly to CCA as a generalized eigenvalue problem, see Eq. 5.2.

Also nonlinear kernel versions of CCA and gCCA have been developed, see for example (Fyfe and Lai, 2000; Bach and Jordan, 2002).

Linear discriminant analysis

Linear discriminant analysis (LDA) was summarized in Chapter 4.3. LDA, like other classification methods, has a connection to estimating the dependency between the class labels and the continuous feature variable (see Torkkola and Campbell (2000) and references therein).

Contingency table analysis

Given two discrete univariate features (categories or classifications) for the same set of objects, they can be represented as a cross-classification table called usually a *contingency table*.

In contingency table analysis the aim is to study the possible dependence of the two classifications with various tests. The tests are usually based on the comparison of a model where the two classifications are dependent versus a model for independent classifications. Since the tables can have multiple columns and rows, and multiple dimensions making them essentially hypercubes, the array of various tests and approaches to their analysis is large.

Even for the simplest case, a two-way contingency table with two columns and two rows, there exist several methods and test statistics (see Yates (1984); Agresti (1992)). However, the classical Fisher test ((Fisher, 1934), see Agresti (1992)) still seems a very reasonable tool and it has recently gained ground also in the analysis of the genomic data. Specifically, it has been implemented into numerous software packages analyzing the proportions of some known classes, like Gene Ontology classes, in a list of genes extracted for example from cluster analysis (see for example Hosack et al. (2003); Zeeberg et al. (2003)).

The Fisher test represents so-called *exact* inference for contingency tables, where the possible configurations of the data in the table can be explicitly enumerated. Another common class of inference uses approximations for the distributions of test statistics, which are often χ^2 -based. They are more suitable for large tables (with lots of data) (see Agresti (1992)). Bayesian approaches have also been applied extensively, see for example Good (1976); Albert (1997).

Correspondence analysis uses contingency tables but has a different aim (see Wickens (1998)). It is an explorative analysis method of contingency tables that can be used to visualize the relationships between the columns and rows of the table.

For a short overview and references of categorical data analysis and their connections, see for example (Wickens, 1998).

5.1.2 Methods combining data sets by clustering

The methods reviewed in the previous section were based on finding or measuring any global dependency between feature sets, either assuming normality of the distributions or operating with given categorizations (classes). If the normality assumptions are not valid, or one has to estimate the categories for continuous data, the methods are not trivially applicable.

The clustering methods reviewed here are capable of integrating multiple feature sets, without assumptions about normal distribution of data, or given categorizations. However, they do not model the dependencies between the data sets. In general, the methods are not exclusively designed for genomic data integration but can certainly be used for that purpose. In addition to the ones presented here, there exists a large array of other methods that have been used to integrate genomic information sources by other means than clustering but they are beyond the scope of this thesis.

Joint distribution models

There exist several joint distribution modeling algorithms for the simultaneous clustering of two feature sets. They range from simple clusterings of concatenated features through mixture models capable of integrating various feature types to rather elaborate graphical models. Such methods are briefly reviewed here, together with their applications to genomic data.

The simple concatenation of features enables the use of all methods applicable for a single data set. However, it usually assumes that the features are equally important and of a similar type. Both of these assumptions may have a dramatic effect on the analysis. In principle, this can be managed by appropriate preprocessing of the features, but this is usually a very difficult task to perform optimally. Example analyses of concatenated data are presented in Publication 2 or in Eisen et al. (1998).

Perhaps the first mixture models integrating two feature sets in the context of genomic data were introduced in Holmes and Bruno (2000). The aim in their work was to find effective regulatory elements using both expression and sequence information. Thereafter, the research focus has largely moved to even more flexible models, in particular, to graphical models.

Graphical models, and specifically Bayesian networks (Pearl, 1988), are a main trend in bioinformatics nowadays. Graphical models are, in short, models for the joint distribution of all the variables in data, and they attempt to explicitly express the conditional independences between the variables. They have been applied to numerous problems where integration of data sources has been needed, for example to gene prediction (Pavlovic et al., 2000), gene regulation modeling (Beer and Tavazoie, 2004; Friedman, 2004; Hartemink et al., 2002; Segal et al., 2003b), gene function prediction (Troyanskaya et al., 2003), and protein-protein interaction prediction (Janse et al., 2003).

An example of graphical-model-based-clustering can be found in so-called context-specific clustering application to gene expression data (Barash and Friedman, 2002). The method works in the framework of Bayesian networks, and proposes a joint distribution model for gene expression and putative transcription factor sites.

While graphical models are in principle an efficient and elegant approach to model the cell, they may suffer from the lack of prior data needed to determine the structure of the graph and/or computational difficulties in searching the correct structure and values for the other parameters. This is partly due to their aim to model the whole phenomenon, that is, all the variation that is present in the data, also the variation possibly irrelevant to the actual analysis task. This same drawback exists in principle in all joint distribution models.

However, the graphical models can also be used in a discriminative fashion (Friedman et al., 1997; Segal et al., 2003d; Segal and Sharan, 2004; Taskar et al., 2002). For example, in Segal and Sharan (2004) a graphical model is built to find the combination of transcription factor binding sites that best discriminates the promoters of a set of co-regulated genes from all the other promoters. These approaches are very promising since they combine the versatility of the probabilistic models to the power of discriminative learning.

Note that building and training of the graphical models must be adjusted rather carefully for the problem at hand. In a sense, while the framework of the graphical models is universally applicable, the individual models are practically always heavily hand-tuned, requiring a lot of methodological expertise.

Information bottleneck methods

The information bottleneck (IB) method was introduced in Section 4.2.2: IB maximizes the dependency between clusters of a discrete random variable and a discrete auxiliary variable. The difference is now that there can be several variables.

The *multivariate information bottleneck* (Friedman et al., 2001; Slonim, 2002) is an extension of IB to multiple variables. The extension is a general one, allowing various kinds of models and dependencies to be estimated. However, for M discrete variables the estimation of the dependency is essentially based on an M-dimensional contingency table. It thus suffers from the large number of data sets, because for the finite data the contingency table becomes sparse.

The requirement of discrete data of the IB-based methods is more restricting here, since now in principle all the data sets should be discrete or discretized. One possible solution to this, a combination of K-means and IB coined *K-IB*, is presented in Publication 8. In that approach two data sets are first discretized by K-means after which the multivariate IB can be readily applied.

In K-IB the vectorial margin spaces (different data sets), $\mathbf{x}$ and $\mathbf{y}$, are first quantized separately by K-means, without paying attention to possible dependencies between the two margins. This results in two sets of margin partitions which span a large, sparse contingency table that can be filled with frequencies of the training data pairs $(\mathbf{x}_k, \mathbf{y}_k)$. In the second phase, the large table is compressed with the symmetric sequential IB algorithm (Slonim, 2002) to explicitly maximize the dependency of margins in the resulting smaller contingency table.

The final partitions obtained by K-IB are of a very flexible form, and therefore the method models the dependencies of the margin variables well. As a drawback, the final margin clusters in the original data spaces will consist of many atomic Voronoi regions, and they are therefore not guaranteed to be particularly homogeneous with respect to the original continuous variables ($\mathbf{x}$ or $\mathbf{y}$). Interpretation of the clusters may then be difficult. K-IB is used as a reference method in Publication 8.

Ad hoc methods

Especially in genomic data analysis the data integration settings are occasionally so complicated or include so large feature sets that development of unified computational frameworks is tedious. In these cases algorithms with a less rigorous approach to technical details of statistical modeling can perform very well. An example of this kind of approach, with respect to gene expression data, is presented in Bar-Joseph et al. (2003), for integration of gene expression and transcription factor (TF) binding data.

5.2 MAP-DC: Integration of one discrete and one continuous data set

This section presents an improved discriminative clustering model based on maximum a posteriori estimation (MAP-DC), introduced originally in Publication 5. MAP-DC maximizes the dependencies between the given class information and the clusters extracted from continuous primary data. It is thus a combination of dependency maximization and clustering.

The discriminative clustering (DC) method presented in Section 4.3 can be interpreted as a piecewise constant model for the auxiliary data distribution $p(c|\mathbf{x})$

given the data $\mathbf{x}$, estimated by maximizing the log-likelihood

$$L = \sum_j \sum_{\mathbf{x} \in V_j} \log \psi_{j,c(\mathbf{x})}, \tag{5.4}$$

where $c(\mathbf{x})$ is the index of the class of the sample $\mathbf{x}$, and $\psi_{j,c(\mathbf{x})}$ is the prototype distribution associated to the cluster j, and V_j is the Voronoi region of the prototype j (for more details, see Publication 5).

The key idea of MAP-DC is to introduce a prior for the parameters of the distribution ψ_j. After that one can integrate the parameters ψ_j and their uncertainty out, and maximize the model only with respect to the prototype vectors $\mathbf{m}_j$ (MAP estimation). If a uniform prior is introduced for the prototypes, the maximization corresponds to a maximum likelihood estimate.

More formally, in MAP-DC presented in Publication 5, denoting the observed primary data by $D^{(x)}$ and the observed auxiliary data by $D^{(c)}$, we wish to maximize the posterior

$$p(\{\mathbf{m}\}|D^{(c)}, D^{(x)}) = \int_{\{\boldsymbol{\psi}\}} p(\{\mathbf{m}\}, \{\boldsymbol{\psi}\}|D^{(c)}, D^{(x)}) d\{\boldsymbol{\psi}\} \ , \tag{5.5}$$

or equivalently $\log p(\{\mathbf{m}\}|D^{(c)}, D^{(x)})$. Here the integration is over all the $\boldsymbol{\psi}_j$.

In practice, the prior for the $\{\boldsymbol{\psi}\}$ is chosen to be a conjugate (Dirichlet) prior, $p(\boldsymbol{\psi}_j) \propto \prod_i \psi_{ji}^{n_i^0 - 1}$, where the $\{n_i^0\}_i$ are the prior parameters common to all j, and $N^0 = \sum_i n_i^0$. The integration in Equation 5.5 can now be done analytically and the log of the posterior probability then is

$$\log p(\{\mathbf{m}\}|D^{(c)}, D^{(x)}) \propto \sum_{ij} \log \Gamma(n_i^0 + n_{ji}) - \sum_j \log \Gamma(N^0 + N_j) \ , \tag{5.6}$$

where $\Gamma()$ is gamma function. In MAP estimation this function is maximized.

The Bayesian formulation of MAP-DC opens up interesting connections to dependency maximization. First note that the clusters can be interpreted as one categorization for the data and the auxiliary data, the class c, as another. The dependency between two categorizations has been traditionally measured with contingency tables (see Section 5.1.1). A Bayesian version of a dependency test for contingency tables has been derived in (Good, 1976). The test was formulated as a Bayes factor of the null hypothesis M_I of independent margins against the alternative hypothesis M_D of dependent margins,

$$\frac{P(\{n_{ij}\}|M_D)}{P(\{n_{ij}\}|M_I)},$$

where the n_{ij} are the observed counts of data in row i and in column j.

It turns out that the cost function of MAP-DC in Equation 5.6 is proportional to the Bayes factor for the dependency in contingency table

$$\begin{aligned} &\frac{P(\{n_{ij}\}|\{n(c_i)\}, M_D)}{P(\{n_{ij}\}|\{n(c_i)\}, M_I)} \\ &\quad = \frac{\prod_{i,j} \Gamma(n_{ji} + n^0)}{\prod_j (N_j + N^0)} \times \frac{\Gamma(N^0)^k}{\Gamma(n^0)^{N_c k}} \frac{\Gamma(kn^0)^{N_c} \Gamma(N + kN^0)}{\prod_i \Gamma(n(c_i) + kn^0) \Gamma(kN^0)} \\ &\quad = p(\{\mathbf{m}\}|D^{(c)}, D^{(x)}) \times const. \ , \end{aligned} \tag{5.7}$$

where the constant does not depend on the amount of data in clusters N_j or table cells n_{ij}. Here $n(c_i)$ denotes the number of samples in the (auxiliary) class c_i, which is a constant. MAP estimation for discriminative clustering is thus equivalent to constructing a dependency table that results in a maximal Bayes factor, under the constraints of the model.

The Bayes factor has the advantage of properly taking into account the finite size of the data set while still being asymptotically equivalent to mutual information. In MAP-DC the Bayes factor is *optimized* instead of only being used to measure dependency in a fixed table. The categorical variable that defines either rows or columns of the contingency table is defined by the clusters, parameterized by prototypes $\{\mathbf{m}\}$. The prototypes are then tuned to make the dependent model describe the (contingency table) data better than the independent model, which can be interpreted as maximizing dependency. Note that the Voronoi regions, that is, the clusters, are local in the original data spaces, making the interpretation straightforward.

The actual optimization of the MAP-DC is done by a conjugate gradient method (for a textbook account, see Bazaraa et al. (1993)) after a smoothing trick which makes gradient-based optimization possible. See Publication 5 for details.

5.2.1 MAP-DC of yeast stress reaction and regulatory gene expression

Baker's yeast *Saccharomyces cerevisae* is widely used both in academic research and in industry. Its utilization is usually based on exposing the yeast to some external treatment that modifies its behavior in a desired way. As a unicellular organism, the change in the yeast's behavior is largely reflected in its gene expression. Any external treatment that is a change from the optimal growth conditions, is likely to induce some stress-like behaviour in yeast. Hence, it is of crucial importance to understand the yeast's stress reaction on the expression level.

Yeast stress has been actively studied during recent years, also with DNA microarray techniques (Gasch et al., 2000; Causton et al., 2001). One of the goals in the publications has been the definition of common *environmental stress response* (ESR) genes. However, the previous papers do not agree totally on the set of the ESR genes and their definition. This is because the definition of the stress itself is still not complete. Additionally, the regulation of these genes' expression is of great interest, but the dependency of the stress reaction on the transcription factors (TFs) is still somewhat unclear.

We explore the dependencies between the yeast gene expression under stress and the expression after the knockout of the two known stress transcription factors Msn2p/Msn4p with MAP-DC in Publication 7. The data from the publications by Gasch et al. (2000) and Causton et al. (2001) is publicly available, and it includes genome-wide time series measurements of yeast gene expression under various stress treatments like heat and acid shock. Additionally, the data set from (Causton et al., 2001) included the genome-wide expression measurements under acid shock after knockout of Msn2p/Msn4p (called *deletion strain* from now on), and the list of the genes hypothesized to be regulated by those TFs from (Gasch et al., 2000).

The setting of the problem is ideal for an exploratory analysis with MAP-DC since we are interested in groups of genes expressed in various ways in a normal strain, but especially of those that are dependent on the knocked-out stress regulators. Hence, the primary data was the gene expression in normal strain under stress, and the auxiliary data was discretized gene expression in the deletion strain under stress.

The results demonstrate the ability of the MAP-DC to make biologically significant findings. The two most important results are explained here. First, the dependency between the normal yeast strain gene expression under stress and the auxiliary information was reproducible, and statistically significantly higher than the one obtained with K-means clustering. This was measured with Equation 5.6 for both models in a 20-fold cross-validation with paired t-test, giving $p < 0.001$.

Second, MAP-DC found a cluster of genes, of which unexpectedly many were down-regulated in the deletion strain. Moreover, practically all the genes in this cluster were up-regulated in the normal strain during the stress treatments, suggesting that this cluster included genes that are regulated by MSN2p/MSN4p during environmental stress. This was confirmed with an environmental stress response (ESR) gene list from Gasch et al. (2000), that revealed that this cluster included statistically significantly many ESR genes. Hence, it may be concluded that MAP-DC manages to reveal biologically significant findings from the data.

5.3 Associative clustering (AC): Combining two continuous-valued data sets

So far the problem setting has been the integration of one real-valued multivariate data set with discrete class information. In a more general situation two real-valued multivariate data sets from different information sources need to be integrated. This is a common situation for example with gene expression data measured under various treatments, or with gene expression and protein interaction data about the same set of genes.

The abstract goal here is the same: to find what is in common between the data sets. A natural idea is to extend the MAP-DC by parameterizing both of the data spaces with prototypes, and to use the analogous dependency measure. This results in a symmetric dependency clustering method coined *associative clustering (AC)* presented in Publication 8 (technical details in Sinkkonen et al. (2005)).

The setting of AC is as follows: Assume a set of objects with two sets of features, or more generally, any samples coming in pairs $(\mathbf{x}, \mathbf{y})$ where $\mathbf{x}$ belongs to the first set and $\mathbf{y}$ to the second. We search for dependencies between the feature sets, expressible as clusters.

Both data sets, $\mathbf{x}$ and $\mathbf{y}$, are clustered separately, in such a way that (i) the clusterings will capture as much as possible of the dependencies between the pairs of data samples $(\mathbf{x}, \mathbf{y})$, and (ii) the clusters contain (relatively) similar data points. The latter roughly equals a definition of a cluster.

More formally, for paired data $\{(\mathbf{x}_k, \mathbf{y}_k)\}$ of real vectors $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$, we search for partitionings $\{V_i^{(x)}\}$ for $\mathbf{x}$ and $\{V_j^{(y)}\}$ for $\mathbf{y}$. The partitions can be interpreted as clusters in the same way as in K-means, DC, and MAP-DC; they are Voronoi regions parameterized by their prototype vectors $\mathbf{m}_i$. The $\mathbf{x}$ belongs to $V_i^{(x)}$ if $\|\mathbf{x} - \mathbf{m}_i\| \leq \|\mathbf{x} - \mathbf{m}_k\|$ for all k, and correspondingly for $\mathbf{y}$. Figure 5.1 presents AC in a nutshell.

The cost function of associative clustering is, analogously to MAP-DC, the

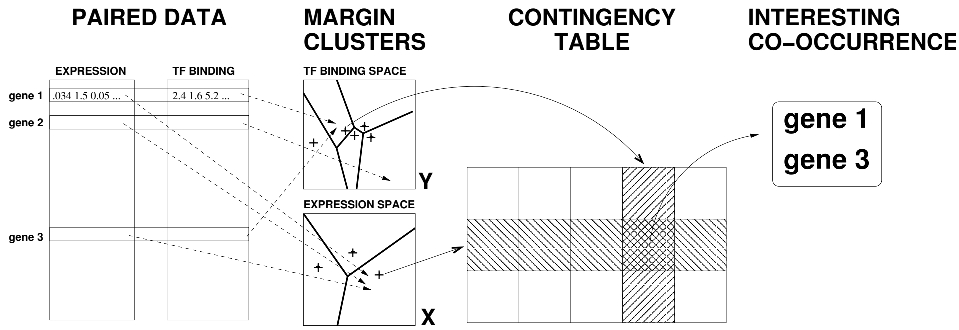


Figure 5.1: An overview of associative clustering (AC). Two data sets are clustered into Voronoi regions. The Voronoi regions are defined in the standard way as sets of points closest to prototype vectors. However, the prototypes are not optimized to minimize a quantization error but by the AC algorithm. In this example, the data sets are gene expression profiles and transcription factor (TF) binding profiles. A one-to-one correspondence between the sets exists: each gene has an expression profile and a TF binding profile. As each gene falls to a certain combination of TF cluster and an expression cluster, we get a contingency table by placing the two sets of clusters as rows and columns, and by counting genes falling to each combination. Rows and columns, that is, the Voronoi regions defined within each data set respectively, are called *margin clusters*, while the combinations corresponding to the cells of the contingency table are called *cross clusters*. *Associative clustering* by definition finds Voronoi prototypes that maximize the dependency seen in the contingency table. Voronoi regions are representations for the data sets just as the linear combinations are in canonical correlation analysis. In both cases, dependency between the two parametrized representations is maximized. Maximization of dependency in a contingency table results in a maximal amount of surprises, counts not explainable by the margin distributions. The most surprising cross clusters with a very high or low number of genes give rise to interesting interpretations. Reliability is assessed by bootstrap.

Bayes factor between the model having dependent clusters M_D and the model having independent clusters M_I,

$$BF = \frac{p(\{n_{ij}\}|M_D)}{p(\{n_{ij}\}|M_I)} = \frac{\prod_{ij}\Gamma(n_{ij}+n^{(d)})}{\prod_i \Gamma(n_{i\cdot}+n^{(x)})\prod_j \Gamma(n_{\cdot j}+n^{(y)})}\ , \tag{5.8}$$

where $n_{i\cdot} = \sum_j n_{ij}$ and $n_{\cdot j} = \sum_i n_{ij}$ express the contingency table margins. The hyperparameters $n^{(d)}$, $n^{(x)}$, and $n^{(y)}$ arise from Dirichlet priors. For large data set sizes the logarithmic Bayes factor approaches mutual information (see Sinkkonen et al. (2005)).

Similarly to MAP-DC, frequencies over the cells of a contingency table can be assumed to be multinomially distributed. The model M_I of *independent margins* assumes that the multinomial parameters over cells are outer products of posterior parameters at the margins: $\theta_{ij} = \theta_i\theta_j$. The model M_D of *dependent margins* ignores the structure of the cells as a two-dimensional table and samples cell-wise frequencies directly from a table-wide multinomial distribution θ_{ij}. Dirichlet priors are set for both the margin and the table-wide multinomials.

The Bayes factor of AC (5.8) will be maximized with respect to the Voronoi prototypes. Analogously to MAP-DC, the Voronoi regions must be smoothed in order for the gradient methods to be applicable. The smoothed *BF* is then optimized with respect to the cluster prototypes $\{\mathbf{m}\}$ by a conjugate-gradient algorithm.

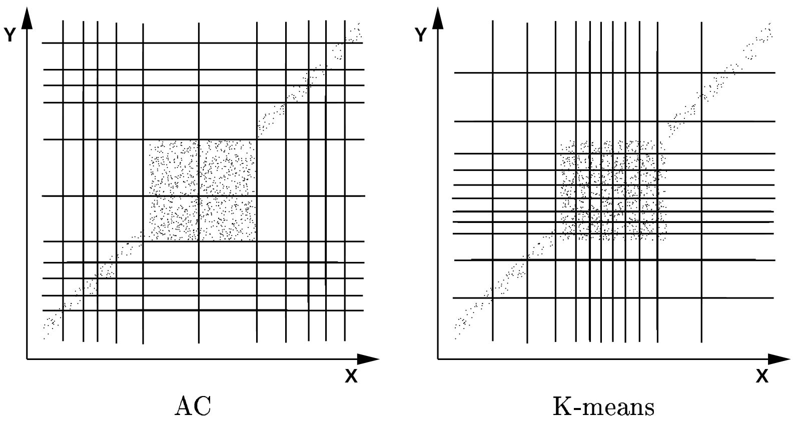


Figure 5.2: Associative clustering concentrates on dependent subsets of data. Here both margin spaces, denoted by **X** and **Y**, are 1-dimensional, and the figure shows a scatterplot of the data (dots on the plane where **X** and **Y** are the axes). Cluster borders in the **X**-space are shown with the vertical lines and cluster borders in the **Y**-space with horizontal lines. The resulting grid of so-called cross clusters then corresponds to the contingency table; the number of dots within each grid cell gives the amount of data in a contingency table cell. The AC cells are sparse in the bulk of independent data in the middle and denser on the sides where the **X** and **Y** are dependent. K-means, in contrast, focuses on modeling the bulk of the data in the middle.

Demonstration of AC with artificial data. Figure 5.2 demonstrates a key property of AC with as simple artificial data sets as possible.

The AC clusters focus on modeling those regions of the margin data spaces, that is, those subsets of data, where the co-occurring pairs $\mathbf{x}$ and $\mathbf{y}$ are dependent. This is clearly visible as the higher density of AC cross clusters on diagonal data areas in Figure 5.2.

Uncertainty in associative clustering. The use of Bayes factors in AC (and in MAP-DC) is different from their traditional use in hypothesis testing, cf. Good (1976); Kass and Raftery (1995). No hypotheses are tested in the methods but the Bayes factor is optimized to maximize the dependencies. However, with respect to the prototype vectors $\mathbf{m}$, the obtained solution in AC is a point estimate that may over-fit easily. This uncertainty is comparable to that of any standard point estimate-based prototype clustering: small changes in the prototype locations may change the clustering. In AC this in turn then changes the contingency table and the found dependencies.

A widely used 'light-weight' method to take into account the uncertainty in clustering is bootstrap (Efron and Tibshirani, 1993; Hastie et al., 2001). As in Kerr and Churchill (2001), the bootstrap is used to produce several perturbed clusterings in Publication 8. The aim is to find cross clusters (contingency table cells) that signify dependencies between the data sets and that are reproducible.

Reproducibility of the found dependencies can be estimated from the bootstrap clusterings as follows. First, *significantly dependent cross cluster* is defined within a given AC-clustering. The optimized AC model provides a way of estimating how unlikely a cross cluster is, given that the margins are independent. For this purpose several (1000 or more) data sets of the same size as the observed one are generated from the marginals of the contingency table (i.e. under the null hypothesis of independence). Those cross clusters with the observed amount of data more extreme than obtained by chance with probability 0.01 or less (Bonferroni corrected with the number of cross clusters), are defined to be *significantly dependent cross clusters.*

The two criteria, dependency and reproducibility, can be combined by evaluating, for every gene pair, how likely they are to occur within the same significantly dependent cross cluster in several bootstrap clusterings (this is analogous to Kerr and Churchill (2001)). This similarity matrix is finally summarized by hierarchical clustering in Publication 8.

The dependencies for all genes between data sets are not to be expected, since with noisy genomic data that would hardly be possible. The main focus is to find the most dependent, robust *subsets of the data.* This is exactly what the final gene clusters from bootstrapped, most dependent cross clusters of AC provide.

Interpretation of the cross clusters. It is often of interest to know which original variables have an exceptionally high or low average value in a cluster. The extremity of the mean profiles of the data in the AC cross clusters can be evaluated by random sampling, as in Publication 8. There 10,000 gene sets of the same size as the observed cluster are sampled at random from the data, and by checking which of the dimensions had their observed average value higher than in all random clusters. These average values are then considered *reliably extreme.*

Validation of bootstrapped AC with real data. Especially in bioinformatics it is often a real challenge to test new methods since there rarely exists any ground truth, that is, known correct answers. The (bootstrapped) AC approach was validated in Publication 8 by searching for dependencies between data sets containing known, real-world duplicate measurements that should be more dependent than random pairs.

A rank sum was used to test whether the similarity distribution of the known duplicates is different from the similarity distribution of all the other genes. In AC the known duplicates turned out to co-occur clearly more frequently in a dependent cross cluster than other genes.

The results were additionally compared to the similarity distribution obtained from bootstrapped K-means, using a sign test. AC detected connections of the duplicate measurements statistically significantly more often than K-means. These two results support the validity of AC in finding dependent subsets of data better than standard unsupervised clusterings.

5.3.1 Dependencies between human and mouse

The use of model organisms is one of the fundamental building blocks in biology. It enables the use of research methods that would be unethical, too expensive, or perhaps impossible when applied on the actual organism of interest. The concept of model organism relies on the assumption that the model and the target organism are similar on the level the research deals with, for example on the genetic level. The similarity in DNA sequence between the two genes from different species is also often used as a tool to hypothesize the functions of unknown genes in the target organism, or to infer evolutionary aspects of the organisms. The genetic similarity between organisms thus has also pure scientific value as itself. However, the gene sequence similarity is still only a hypothesis for actual similarity, that is, it is unconfirmed whether the genes with similar sequence really act in similar roles. Hence, the *functional similarity* of two organisms at genetic level is worth studying.

The genome-wide gene expression measurements made in the same tissues in two organisms open up a way to study the dependencies between the two species. For example, it is possible to find groups of orthologous genes that are similar by sequence but differ in their expression, or groups of genes that are similar both by sequence and by expression. The first kind of groups are perhaps the most interesting because they might suggest that whole functional groups of genes have altered their role in cells during evolution. The latter kind of groups are the groups one expects to find, since they are one kind of a validation for the sequence-based similarity of the genes. More complicated dependencies, like partial correspondence in expression, are naturally also among the most interesting aspects.

In Publication 8 the dependencies between the expression of human and mouse orthologous genes are studied with associative clustering, K-IB, and independent K-means clusterings, based on the data set derived from (Su et al., 2002).

AC produced significantly more dependent clusters than standard K-means clustering. However, K-IB produced significantly more dependent clusterings than AC and K-means. On the other hand, cross clusters from AC studies were significantly more homogeneous than those of K-IB and random clustering. This demonstrates the better interpretability of the AC cluster, since it is then possible to summarize clusters, for example, by mean expression profile, as in Figure 5.5. The measure of homogeneity (actually dispersion) was the sum of intra-cluster variances in Publication 8.

Bootstrapped AC produced a similarity matrix for the genes, computed from the co-occurrence frequencies of genes in the AC cross clusters. The matrix was in Publication 8 summarized with simple hierarchical clustering, and a set of most homogeneous gene clusters was extracted by cutting the dendrogram at the level of 80% co-occurrence, and discarding genes belonging to clusters smaller than 3 genes. This resulted in 139 orthologous gene pairs in 31 clusters, and some key findings about them are presented in the following.

First, the AC results were compared to the simplest dependency measure between the orthologous genes: correlation. A global trend existed in our data to some extent: the higher the correlation between the expression profiles of an orthologous gene pair, the more often the pair tended to be located in an unexpectedly large cross cluster. This suggested that AC is capable of detecting the simple tendency of the orthologs to depend linearly.

Second, the AC clusters were checked for an overall enrichment of Gene Ontology (GO) (Consortium, 2000) categories with EASE (Hosack et al., 2003). This might hint at the groups of orthologous genes with exceptional functional conservation, which could be expected to be of a specific importance for species survival. The most enriched GO categories in AC cross clusters were ribosomal categories, whose high conservation has been suggested also in earlier studies (see, for example, Jiménez et al. (2002)). Additional, more unexpected findings included the category of "transmission of nerve impulse," whose conservation is still unreported.

Third, preservation of function between human and mouse and the differentiation of the function were studied according to the highest correlation and the lowest correlation, respectively, in clusters. For example, testis-specific gene expression seemed to be preserved, whereas embryonic development was differentiated. The testis-related results also produced a novel potential functional link to the genes in the same cluster. In addition to these examples, a wide variety of other results were found, but their confirmation is a tedious process. Publication 8 presents some of these.

5.3.2 Dependencies between yeast gene expression and TF binding

Regulatory interactions between genes are nowadays studied by measuring genome-wide expression with microarrays in knock-out mutation experiments and in time series experiments. In the knock-out experiments, a mutation is targeted to a single gene in the yeast genome to modify (usually knock out) the normal function of that gene. It is then hoped for that by measuring the gene expression changes with microarrays after the mutation, the role of the mutated gene in cellular processes is revealed. Genes belonging to the same regulatory pathway as the mutated gene could be unveiled, for example. In time series experiments the goal is often to infer causality in the gene regulatory network based on the sequential changes in expression levels. However, since the interaction network between the genes is complicated, discerning the direct effects of the knock-out, or the change of expression in a time series from noise and the mass of second-order effects can be very difficult, if not impossible. At least a comprehensive, very expensive high resolution time-series experiment with numerous replications would be required. The same holds also for knock-out experiments. Thus alternative approaches are worth exploring.

Gene expression is not the only source of information about gene regulation. For instance, microarray-based chromatin immunoprecipitation (ChIP) allows measuring the binding strength of the transcription factor (TF) proteins on any gene's promoter region (Lee et al., 2002). This reveals which TFs are able to bind the specific gene's promoter, and are thus potential regulators. But many TFs bind numerous gene promoter regions and are still not operational regulators. The number of false positives can be very high, and thus inferring the regulatory relationships based on the binding information alone is not in general possible. However, searching genes with maximal dependency between ChIP data and gene expression data should improve the inference from either data source alone.

Associative clustering was applied to explore the dependencies between expression and TF binding data in two case studies in Publication 8. The difference between the cases are the expression data, which were chosen to represent both archetypes of data used in gene regulation studies: knockout data and time-series data. In both of the cases the aim was to find subsets of genes whose expression is maximally dependent on their transcription-factor-binding profiles. These sets of genes are then hypotheses for expression co-regulation candidates.

Knock-out gene expression and TF binding. The yeast gene expression data used in this analysis had been measured from 300 different mutation strains and medical treatments with cDNA microarrays (Hughes et al., 2000). Transcription factor binding data on genes for 113 transcription factors was obtained from Lee et al. (2002) . After preprocessing, we had two full data matrices, each with 6185 genes. The number of clusters in the margin spaces was chosen to produce roughly 10 data points in each cross cluster, resulting in 30 clusters in the expression space and 20 clusters in the TF binding space.

AC discovered dependencies in the data significantly better than the reference methods. Margin clusters produced by AC were statistically significantly less dispersed than those produced by IB, but for the cross clusters the differences were not significant.

A similarity matrix was generated for the genes from the bootstrap results, and summarized by hierarchical clustering. Clusters with average similarity higher than 20 (frequency of co-occurrence within the 100 sets) and with the minimum size of 3 genes were chosen for the final analysis, resulting in 20 clusters.

The clusters were again first screened with EASE, which found enriched gene ontology classes in 12 of the 20 clusters. The cluster types found by AC could be divided into three types:

1. Clusters with genes known to be expressed often very homogeneously in yeast, and also often found in conventional cluster analyzes, cf. ribosomal proteins (Beer and Tavazoie, 2004) and in Publication 2.

2. Clusters where some of the genes and their main regulator(s) had been previously identified in wet lab experiments. However, the groups also contained components not previously associated with the corresponding biological function. This provided new hypotheses for the functions of the genes.

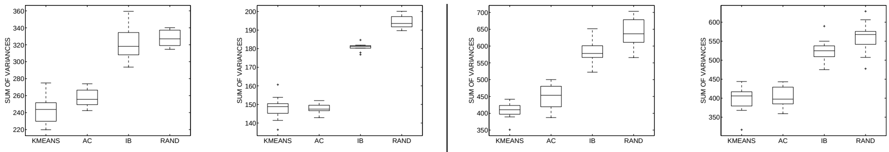
3. Clusters of genes with mostly unknown molecular function, and even with an unknown biological process. These clusters represent the most promising results, but are naturally extremely hard to interpret. However, since AC produces the suggestion for both the set of genes *and* the set of potential regulators, the future research concerning these should prove easier than starting from the single data source results, that is, from the plain expression, for example.

Cell cycle gene expression and TF binding. The expression data for this case study was measured during the yeast cell cycle and was originally published in two different papers (Spellman et al., 1998; Cho et al., 1998). The data consisted of 77 measurementst of all the yeast genes in total. The transcription factor (TF) binding data used here were the updated (2003) version of Lee et al. (2002) for 106 transcription factors. In this case study after preprocessing we had two matrices with 5618 genes. The chosen cluster numbers were 30 in the expression space and 20 in the TF-binding space.

The differences in dependency modeling between all the methods were statistically significant also for this data pair. The cluster dispersion was the same as in the previous cases: the sum of the component-wise variances. For this data pair, AC produced significantly less dispersed cross clusters and margin clusters than IB. Figure 5.3 visualizes the margin cluster dispersion and the cross cluster dispersions for all methods.

In a similar manner as in the previous cases, biological findings were sought from the bootstrapped AC clusters. The clusters with an average distance smaller than 60 (times in the same dependent cross cluster out of 100) and with more than 2 genes were chosen. This resulted in a total of 16 clusters.

Gene ontology classes were enriched statistically significantly in 13 of the 16 clusters (EASE).



Expression space TF binding space | Expression space TF binding space

Margin clusters | Cross clusters

Figure 5.3: Margin cluster dispersion and cross cluster dispersion for all methods in cell-cycle experiments, demonstrating that AC produces clusters that are almost as compact as K-means clusters, whereas the IB-clusters are significantly more dispersed. RAND is a kind of an upper limit for cluster dispersion, obtained by randomly assigning samples to clusters.

The closer biological analysis of the clusters revealed the same subtypes of cluster as were found in the knock-out and TF case (Section 5.3.2).

5.4 AC + gCCA: Multiple continuous-valued data sets

In some cases the dependencies between *multiple data sets* about the same set of objects, or more generally *data from multiple co-occurring variables*, are of interest. The basic principle of MAP-DC, AC, and K-IB, representing dependencies with hard clustering and a contingency table, becomes infeasible with multiple data sets. Specifically, the volume of a multi-way contingency table grows exponentially as a function of the data sets, while the amount of data items stays fixed. Other approaches are thus needed.

In Publication 9 a new approach for finding dependencies between N data sets is introduced. The key idea is to first try to drop the amount of the data sets down to two, trying to maximally preserve the dependencies between the $N-1$ data sets. Then the dependency maximizing methods for two feature sets are applicable (see Section 5.3). In Publication 9 a generalized canonical correlation analysis (gCCA) is first used as a preprocessing method to form a representation of $N-1$ data sets, and then the associative clustering (AC) is used to hunt for dependencies between the remaining data set and the gCCA representation. The approach is motivated by an information-theoretic interpretation of the gCCA, which justifies the use of the specific variant of gCCA and the creation of the new representation for $N-1$ data sets with it.

Information-theoretic interpretation of gCCA. The projection computed with gCCA can be interpreted from an information-theoretic point of view as explained in Publication 9. The interpretation starts from a standard assumption that the variables (individual data sets) $\{X_i\}$ are normally distributed, enabling the multi-information between the variables to be expressed as a function of covariance matrices (Kullback (1959); see also Bach and Jordan (2002)):

$$\begin{aligned} \mathrm{I}(X_1;\ldots;X_M) &= \sum_{i=1}^{M} H(X_i) - H(X_1,\ldots,X_M) \\ &= -\frac{1}{2}\ln\frac{\det\mathbf{C}}{\det\mathbf{C}_1\cdots\det\mathbf{C}_M}\,, \end{aligned} \tag{5.9}$$

where H denotes entropy, $\mathbf{C}$ is the covariance matrix of the concatenated data, and $\mathbf{C}_i$ are the original covariance matrices.

It is shown in Publication 9 that whitening, the removal of the covariances, of the original data sets preserves the mutual information between the data sets. Using this, the mutual information between original data sets can be expressed as:

$$I(X_1; \ldots ; X_M) \quad = \text{const.} - H(X'_1, \ldots , X'_M), \tag{5.10}$$

which intuitively means that all the mutual information between the original variables can be represented by the joint entropy of the whitened data sets plus a constant.

Dimensionality reduction and data integration. The information theoretic interpretation of gCCA makes possible its well-justified use in dimensionality reduction. Specifically, the aim is now to find the components of the joint data that maximally preserve the multi-information $I(X_1; \ldots ; X_M)$ between the original data sets, to discard the irrelevant variation.

According to Eq. (5.10) the optimal representation of $I(X_1; \ldots ; X_M)$ is the one that maximally preserves the entropy, $H(X'_1, \ldots , X'_M)$. Note that the joint entropy equals the entropy of the concatenated data sets. It turns out that the maximization of the entropy coincides with the maximization of the variation for Gaussian variables. The dimensionality can thus be reduced by sequentially searching for the one-dimensional projection that maximizes the variation of the whitened and concatenated data sets. Hence, the data can be integrated by performing a principal component analysis for this data.

5.4.1 Dependencies between yeast stress reaction and TF binding

The yeast's common reaction to environmental stress, see Subsection 5.2.1, is still largely undefined. It is hard to discern it in gene expression studies, since also the normal processes of yeast cells are in operation in the experiments, and the microarray measurements are noisy. However, it is *a priori* known which treatments are stressful for yeast.

The yeast stress reaction is modeled by extracting the variation that is common to a set of stress treatments in Publication 9. All the other variation is considered irrelevant. This is a data-driven way to define stress; no strong assumptions about the type of the stress reaction are made. Furthermore, in Publication 9 the regulation of stress is explored by searching for maximal dependencies between the extracted stress reaction and a transcription factor (TF) binding data.

Common stress response of yeast was sought from expression data of altogether 16 stress treatments (Causton et al., 2001; Gasch et al., 2000): heat (2), acid, alkali, peroxide, NaCl, sorbitol(2), H2O2, menadione, dtt(2), diamide, hypo-osmotic, aminoacid starvation, and nitrogen depletion. The gene expression data sets from stress experiments formed in total a 104-dimensional expression data for 5998 genes. In the integration and dimensionality reduction with gCCA, the number of components was chosen such that the same components could be found in left-out data reasonably well (measured with the angle between the components) in 20-fold cross-validation. This resulted in 12 generalized canonical components.

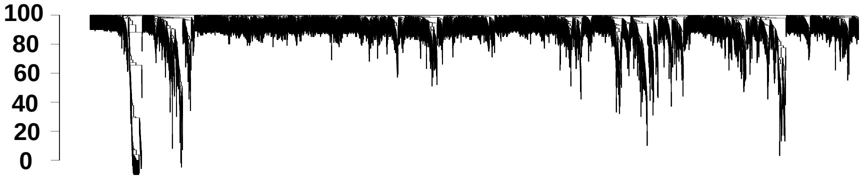


Figure 5.4: Dendrogram of the hierarchical clustering visualizing the similarities between all the genes clustered with 100 bootstrap AC models. The vertical axis represents the average dissimilarity of the genes: 100 meaning that a pair of genes occur never in the same significantly dependent cross-cluster in 100 bootstrap runs, and 0 that they always co-occur. Note how there is a mass of genes whose dissimilarity, or co-occurrences in the different cross-cluster, is over 80, which was the cutoff threshold to produce the final clusters. Several very reliable clusters can also be seen, as downward protruding peaks.

gCCA components were validated with the set of environmental stress genes (ESR) defined in the literature (Gasch et al., 2000). Of the 12 generalized canonical components 9 showed statistically significant association to ESR genes known to be either up-regulated or down-regulated.

The dependencies between the extracted stress response and TF data (Lee et al., 2002) were explored with the associative clustering, for motivation see Subsection 5.3.2. AC found statistically significantly higher dependency between the data sets than K-means.

A similarity matrix from bootstrapped AC clusterings was produced as described in Section 5.3, and summarized by hierarchical clustering. Figure 5.4 visualizes the dendrogram from hierarchical clustering, and shows a few clear clusters interspersed within a background that shows no apparent dependencies.

The clusters based on the dendrogram were analyzed first by investigating the distribution of ESR genes within them. The up-regulated ESR genes were enriched statistically significantly in 14 out of the 51 clusters, and down-regulated ESR genes in 12 of them. This confirms that the combination of gCCA and AC has succeeded in capturing stress-related genes in clusters.

For more detailed interpretation of the clusters, they were analyzed with EASE (Hosack et al., 2003) to find significant enrichments of gene ontology classes. In total we found 14 statistically significant enriched GO classes in our 51 clusters. Additionally the enrichments of ESR genes as well as interesting non-random TF bindings were used as indicators to select clusters for the analysis.

With the help of the known ESR genes, GO classes, and more detailed biological analysis, clusters can be characterized into the following categories:

1. Clusters consisting mainly of the ESR genes with known functions. For those, the analysis produced a set of potential regulators.

2. Clusters consisting partly of the ESR genes that are largely unknown, thus offering hypotheses for the function of the ESR genes based on the functions of the other genes in those clusters, and for their regulation.

3. Clusters consisting mainly of other genes than the known ESR genes, but having homogeneous gene expression under stress and homogeneous TF binding. These clusters thus suggest additions to the current ESR gene list, and to their regulators.

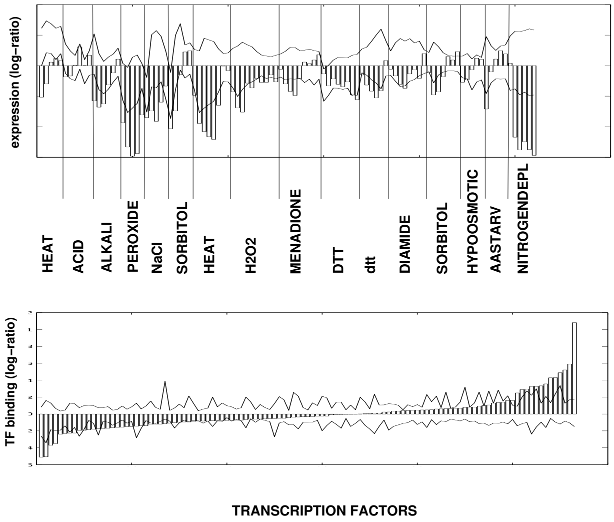


Figure 5.5: A gene cluster related to cell cycle revealing how cell-cycle machinery is driven down under stress, and which are putative regulators for that set of genes. The upper figure represents the mean expression profile (bars) of the genes with their confidence intervals (lines, computed by random sampling), revealing how the genes are down-regulated in practically every treatment, and thus conveying information about the shut down of the cell cycle machinery. The lower figure represents the mean TF binding profile (bars), with confidence intervals (lines), revealing several significant, strong TF bindings. The most interesting of these are analyzed in the text.

The cluster visualized in Figure 5.5 is an example of a set of genes that are not known to be specifically associated to stress, but obviously behave very homogeneously under stress. The cluster actually contains only two genes known to be ESR genes. Nevertheless, this relatively large cluster can be used to show two characteristic predictions obtained using AC and confirmed by biological observations. First, this cluster is enriched in genes involved in the process of cell cycle (12 out of 57, Bonferroni corrected p-value < 0.0001). This reflects the coordinated expression of also other genes than the ESR genes under stress. Second, AC proposes a set of transcription factors involved in the regulation of the member genes of the cluster. This is of special value in this case, because although co-ordinated interactions between different signal transduction pathways are essential in biological systems, inter-pathway connections are difficult to identify. The two most prominent transcription factors of this cluster are coded SWI4 (YER111C) and FKH2 (YNL068C), which both are known to be involved in cell cycle control. However, they operate on different parts of the process, as shown by Shapira and coworkers for the Forkhead factor (Shapira et al., 2004)

The significant TF bindings in the same cluster also include ASH1, which is not directly related to the cell cycle process but rather to mating type selection. However, mating-type switching in the yeast is a multi-step programme, which enables Ash1p to asymmetrically localize to the daughter cell nucleus at the end of cell division in order to prevent the daughter cell from switching mating type. Thereby, it is interesting to see that AC has grouped ASH1 together with SWI4 and FKH2.

5.5 Discussion

To summarize, all the methods discussed in this chapter were clustering methods integrating two or more data sets about the same set of objects. The four new clustering methods, MAP-DC, AC, K-IB, and gCCA/AC, proposed for the analysis of genomic data, were based on maximizing the dependencies between data sets. The clusters were parameterized and formed separately in each feature set, or were given as a pre-defined classification (MAP-DC). The dependencies between the clusters were represented in all cases as a contingency table.

Interpretations of the methods. The methods presented in this chapter can be viewed from several perspectives. Firstly, they are semi-supervised (by dependency maximization) clusterings in each continuous data space. This makes them related to the learning metrics principle (see Section 4.3). While this is obvious in the case of MAP-DC since it is an extension of clustering in learning metrics (DC), also AC is related to LM. In particular, AC maximizes the dependency by allowing the Voronoi regions (clusters) to become elongated in some directions in the data space, that is, some variables are estimated to be less relevant for the dependency. In contrast, K-IB maximizes the dependency by grouping together disjoint atomic Voronoi regions, mostly neglecting the topology of the original data space. It can be hypothesized that if the AC maximizes the dependency better for some data (yeast cases), the data is of more continuous nature and the models preserving the topology benefit from that. On the other hand, if K-IB performs better in the dependency maximization (human-mouse case), it can be assumed that the data is more discrete in nature. This assumption gets some support from the previous exploratory analyses of both the yeast data sets and the human-mouse data sets. Especially human-mouse expression data seems to include many small clusters that are formed by sets of genes that are active only in one specific tissue and nowhere else, and K-IB found higher dependency than AC for these data.

Secondly, all the methods are capable of integrating two data sets into the same analysis in a theoretically justified way. The integration here is tightly connected to dependency maximization and provides information about the subsets of the data items that have dependent feature sets. Integration in these methods is thus not an all-purpose methodology, but rather a strictly defined goal to find things shared by the sets. However, the methods themselves are general purpose. They can be used, for example, in feasibility studies to infer whether data sets of interest are dependent, before more elaborate and hand-tuned methods are applied for the problem.

Thirdly, the methods can be seen as special cases of a framework that by-passes incommensurability, and the problems caused by the different data types. The incommensurability here refers to different, unknown scales of various data sets, for example in the data from different microarray platforms. In this framework models are defined by using feature set-specific parameterizations with hidden variables, and then integrating the information sources on the hidden variable level. For example, in AC the models are clusters parameterized with the prototypes in the data space, and the hidden variables are the memberships to the clusters that are represented in the contingency table. This can be seen as a way to circumvent the full Bayesian treatment that would require specifying proper probability distributions everywhere.

The prior. In AC and MAP-DC the probabilistic formulation of the problem extends earlier mutual information-based approaches in earlier versions of DC. In particular, AC and MAP-DC are better-justified for finite (small) data sets. The methodology requires the choice and use of specific prior distributions for the data in the contingency table. The prior used here was an uninformative one using $n_{ij} = n_j = n_i = 1$.

The choice of the prior has some effect on the model, and improving the prior is one possible research direction in the future. It would be possible to move towards the more complicated mixture priors investigated for example in (Good, 1976). However, while the prior certainly has a large effect when *testing* the dependency of an observed fixed contingency table, it is probably not that crucial when the aim is to *maximize* the dependency in a contingency table. This assumption is related to the fact that, both in MAP-DC and in AC, it is not necessary to find global dependency between the information sources, but any subset of data with dependency is interesting. Hence, over-simplifying slightly, as long as the dependency is significantly and reproducibly larger than in independent clusterings, the results of MAP-DC and AC are well justified, and the actual value of the Bayes factor is not that crucial.

Parametrization of clusters. While the proposed method was shown to already be viable as such, it can be further improved. So far, the problem of choosing an optimal number of clusters was not addressed. If clustering is interpreted as a partitioning or quantization of data to compress its presentation, then the exact number of clusters is not a crucial parameter, but nevertheless the results could be improved by optimizing it. Since the task is formulated in Bayesian terms, Bayesian complexity control methods are applicable in principle. The setting is not conventional, however, because of the non-standard use of the Bayes factors.

Regularization. Another direction of improvement is regularization of the clustering solutions. Dependency-searching methods may potentially over-fit the data, which is well-known from canonical correlation analysis and can be avoided by regularization. Two different kinds of regularizations for MAP-DC have been developed earlier (Kaski et al., 2003): "entropy regularization" and regularization by mixture model. The first was used here in both MAP-DC and AC simulations, because it is easier in practice and has not been shown to be worse than the alternative (Kaski et al., 2003, 2005).

Multiple data sets. The gCCA as a preprocessing that integrates multiple data sets was shown viable. However, the assumptions made by gCCA can be unrealistic. If it is suspected that the linearity of the components, or the assumption of Gaussian data distributions, could have a large effect on the analysis, it is possible to use non-linear versions of gCCA, for example kernel CCA (see Bach and Jordan (2002) and the references therein).

The definition and analysis of the stress. An important point in the application of gCCA and AC was the data-driven determination of the yeast stress reaction. In principle, it would have been possible to use ready-made gene lists from the literature, but they are not optimally constructed in the sense that they are based on joint distribution modeling. The use of gCCA in Publication 9 can be held as a step towards the data-driven determination of the stress reaction of the yeast. However, it is clear that the full understanding of the environmental stress still requires much work.

Note that different analyses of yeast stress response and its regulation were presented, in Publication 7 and in Publication 9. In the former the setting was to study the applicability of MAP-DC to explore the dependencies between the normal yeast strain gene expression and the expression of the yeast strain with disabled stress regulator. However, the role of the other possible stress regulator genes and the groups of genes regulated by them was still left open. This was addressed in the latter.

Biological compatibility. Another biologically relevant aspect is that the TF-binding data are measured in the optimal growth conditions, and it is possible that under different experimental conditions TF-binding is altered. This may be one reason that the stress regulators Msn2p and Msn4p analyzed in Section 5.3.2 are not saliently regulating any cluster in the analysis prsented in Publication 9. More precise results would be obtained by using data sets gathered in similar conditions.

Chapter 6

Conclusion

In this thesis exploratory cluster analysis methods have been developed and applied to genomic high-throughput data sets. The motivation for the work have been the new measurement techniques, in particular microarrays, that produce biological data for which no established hypotheses exist yet. The aim of all the methods is to provide insights into data sets, and to prepare the way for the actual formulation of the biological questions.

Several gene expression data sets have been clustered and visualized with the self-organizing map (SOM) in the thesis. New methods for interpreting the mapping of SOM have been introduced and used to analyze a patent abstract data set and gene expression data sets. SOM was demonstrated to be easy to use and intuitive to interpret in analyzing a single genomic data set.

The learning metrics principle provides a way to focus the analysis to the relevant aspects of the data, derived from auxiliary information. For example, if the biological process associated to genes is of primary interest, gene expression data can be visualized and clustered in a way that optimally reveals the biological processes. The specific algorithms of learning metrics applied in this thesis were the SOM in learning metrics and discriminative clustering. They outperformed the reference methods in applications.

Dependency-maximizing clustering methods, *maximum a posteriori* discriminative clustering (MAP-DC), and associative clustering (AC), that have been introduced in the publications included in this thesis, open up a new framework for fusing two feature sets. This fusing is of primary importance in genomic data analysis. In this thesis AC has also been extended to multiple data sets using generalized canonical correlation analysis as a preprocessing. In all case studies, the methods performed better in their tasks than their alternatives.

All the methods used in the thesis have been shown to produce biologically relevant results, thus demonstrating their applicability to genomic data analysis. Moreover, they are all of general purpose: although their motivation lies in genomic data analysis, they can be readily applied to any application domain.

The most important area for future research is opened by the general-purpose dependency exploration methods. At least two distinct research strategies can be seen: i) to create methods that are general-purpose, but not optimal for a specific problem, or ii) to construct customized methods to specific problems. This is of course one version of the dilemma of whether one should search for the best application for an elegant method, or develop an efficient method for an important application. Without committing oneself on either side, it can be argued that it seems recommendable to integrate the two views in genomic data analysis. The rationale is that systems biology needs both biologically relevant results *and* new perspectives. The latter are given by abstract approaches, but the former cannot be achieved without an effort to take into account the special requirements and existing knowledge in biology. Hence, the most fruitful research in computational biology is likely to include a large but modular, partially *ad hoc* computational machinery that can both include the essential biological knowledge and be simultaneously used to introduce new methodological advances, based, among others, on dependency exploration.

Bibliography

R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of ACM SIGMOD*, pages 94–105, 1998.

A. Agresti. A survey of exact inference for contingency tables. *Statistical Science*, 7(1): 131–153, 1992.

S. Akaho. A kernel method for canonical correlation analysis. In *Proceedings of the International Meeting of the Psychometric Society (IMPS2001)*. Springer-Verlag, 2001.

J. H. Albert. Bayesian testing and estimation of association in a two-way contingency table. *Journal of the American Statistical Association*, 92(438):685–693, 1997.

B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. Watson. *Molecular biology of the cell.* Garlan Publishing, 1994.

E. Alhoniemi. *Unsupervised Pattern Recognition Methods for Exploratory Analysis of Industrial Process Data.* PhD thesis, Helsinki University of Technology, 2002.

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *Proc. of International Conference on Machine Learning 2003*, 2003.

Z. Bar-Joseph, G. Gerber, T. Lee, N. Rinald, J. Yoo, F. Robert, B. Gordon, E. Fraenkel, T. Jaakkola, R. Young, and D. Gifford. Computational discovery of gene modules and regulatory networks. *Nature Biotechnology*, 21(11):1337–1342, 2003.

Y. Barash and N. Friedman. Context-specific Bayesian clustering for gene expression data. *Journal of Computational Biology*, 9:169–191, 2002.

T. Barrett, T. O. Suzek, D. B. Troup, S. E. Wilhite, W.-C. Ngau, P. Ledoux, D. Rudnev, A. E. Lash, W. Fujibuchi, and R. Edgar. NCBI GEO: mining millions of expression profiles—database and tools. *Nucleic Acids Research*, 33:D562–D566, 2005.

S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, pages 59–68, 2004.

M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms.* Wiley, New York, 1993.

S. Becker. Mutual information maximization: models of cortical self-organization. *Network: Computation in Neural Systems*, 7:7–31, 1996.

M. Beer and S. Tavazoie. Predicting gene expression from sequence. *Cell*, 117:185–198, 2004.

A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3-4):281–97, 1999.

T. D. Bie, N. Cristianini, and R. Rosipal. *Handbook of Computational Geometry for Pattern Recognition, Computer Vision, Neurocomputing and Robotics*, chapter Eigenproblems in Pattern Recognition. 2005. To appear.

M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the 21st International Conference on Machine Learning*, pages 81–88, 2004.

C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, 1995.

A. Brazma, P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, C. Stoeckert, J. Aach, W. Ansorge, C. A. Ball, H. Causton, T. Gaasterland, P. Glenisson11, F. Holstege, I. Kim, V. Markowitz, J. Matese, H. Parkinson, A. Robinson, U. Sarkans, S. Schulze-Kremer, J. Stewart, R. Taylor, and J. V. . M. Vingron. Minimum information about a microarray experiment (miame) - toward standards for microarray data. *Nature Genetics*, 29(4):365–371, 2001.

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

N. A. Campbell, J. B. Reece, and L. G. Mitchell. *Biology.* Benjamin Cummings, San Francisco, 6th edition, 2001.

H. C. Causton, B. Ren, S. S. Koh, C. T. Harbison, E. Kanin, E. G. Jennings, T. I. Lee, H. L. True, E. S. Lander, and R. A. Young. Remodeling of yeast genome expression in response to environmental changes. *Molecular Biology of the Cell*, 12:323–337, February 2001.

H. Chang and D.-Y. Cheung. Locally linear metric adaptation for semi-supervised clustering. In *Proc. of International Conference on Machine Learning 2004*, 2004.

V. Cherkassky and F. Mulier. *Learning from Data.* John Wiley and Sons, 1998.

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

W. J. Conover. *Practical nonparametric statistics.* John Wiley & Sons Inc., New York, 1971.

T. G. O. Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley, New York, 1991.

S. Dasgupta. Performance guarantees for hierarchical clustering. In *Proc. of International Conference on Learning Theory*, 2002.

H. I. Daumé and D. Marcu. Supervised clustering with the Dirichlet process. In *NIPS04 workshop on Learning with Structural Outputs*, 2004.

M. Dettling and P. Bühlmann. Supervised clustering of genes. *Genome Biology*, 3(12): research0069.1–0069.15, 2002.

I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of KDD'03, The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 89–98. ACM Press, New York, NY, USA, 2003.

C. H. Ding. Analysis of gene expression profiles: class discovery and leaf ordering. In *Proc. RECOMB 2002*, pages 127–136, 2002.

D. Dykxhoorn, C. Novina, and P. Sharp. Killing the messenger: Short RNAs that silence gene expression. *Nature Reviews: Molecular Cell Biology*, 4:457–467, June 2003.

B. Efron. Bootstrap methods: another look at the jackknife. *Annals of Statistics*, 7:1–26, 1979.

B. Efron and R. Tibshirani. *An Introduction to the Bootstrap.* Chapman&Hall, New York, 1993.

C. F. Eick, N. Zeidat, and Z. Zhao. Supervised clustering - algorithms and benefits. In *Proceedings of International Conference on Tools on AI*, pages 775–776, 2004.

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

R. Fisher. *Statistical Methods for Research Workers.* Oliver and Boyd, Edinburgh, 1934. Originally published 1925, 14th ed. 1970.

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

N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

N. Friedman, O. Mosenzon, N. Slonim, and N. Tishby. Multivariate information bottleneck. In *Proceedings of UAI'01, The Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 152–161. Morgan Kaufmann Publishers, San Francisco, CA, 2001.

C. Fyfe and P. Lai. ICA using kernel canonical correlation analysis. In *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2000)*, 2000.

M. Galperin. The molecular biology database collection: 2005 update. *Nucleic Acids Research*, 33:D5–D24, 2005.

A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, 11:4241–4257, 2000.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis (2nd edition).* Chapman & Hall/CRC, Boca Raton, FL, 2003.

S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias-variance dilemma. *Neural Computation*, 4:1–58, 1992.

G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proceedings of National Academy of Sciences*, 97(22):12079–12084, 2000.

T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

I. J. Good. On the application of symmetric Dirichlet distributions and their mixtures to contingency tables. *Annals of Statistics*, 4(6):1159–1189, 1976.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning*, 2003.

A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, and R. A. Young. Combining location and expression data for principled discovery of genetic regulatory netwrok models. In *Proc. of Pacific Symposium on Biocomputing*, volume 7, pages 462–473, 2002.

T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84:502–516, 1989.

T. Hastie, R. Tibshirani, and A. Buja. Flexible discriminant and mixture models. In J. Kay and D. Titterington, editors, *Neural Networks and Statistics.* Oxford University Press, Oxford, 1995.

T. Hastie, R. Tibshirani, M. B. Eisen, A. Alizadeh, R. Levy, L. Staudt, W. C. Chan, D. Botstein, and P. Brown. 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, 1:3.1–3.21, 2000.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer, New York, 2001.

S. Hautaniemi, O. Yli-harja, J. Astola, P. Kauraniemi, A. Kallioniemi, M. Wolf, J. Ruiz, S. Mousses, and O.-P. Kallioniemi. Analysis and visualization of gene expression microarray data in human cancer using self-organizing maps. *Machine Learning*, 52: 45–66, 2003.

S. Haykin. *Neural Networks: A Comprehensive Foundation.* Prentice Hall, Upper Saddle River, New Jersey, second edition, 1999.

T. Heskes. Bias/variance decompositions for likelihood-based estimators. *Neural Computation*, 10:1425–1433, 1998.

I. Holmes and W. J. Bruno. Finding regulatory elements using likelihoods for sequence and expression profile data. In *Proceedings of International Conference on Intelligent Systems in Biology*, pages 202–210, 2000.

R. Hooke. *Micrographia.* 1665.

D. A. Hosack, G. D. Jr., B. T. Sherman, H. C. Lane, and R. A. Lempicki. Identifying biological themes within lists of genes with ease. *Genome Biology*, 4(R70), 2003.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

T. R. Hughes, M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Coffrey, H. Dai, Y. D. He, M. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard, and S. H. Friend. Functional discovery via a compendium of expression profiles. *Cell*, 102:109–126, 2000.