Integrating Machine Learning Approaches into Network Science: Exemplary Applications and Novel Algorithms

Dissertation zur Erlangung des Doktorgrades der Naturwissenschaften (Dr.rer.nat.) der Naturwissenschaftlichen Fakultät II – Physik der Universität Regensburg

vorgelegt von

Florian Blöchl

aus

Regensburg

November 2010

Die vorliegende Dissertation entstand während einer dreijährigen Beschäftigung in der Arbeitsgruppe Computational Modeling in Biology am Institut für Bioinformatik und Systembiologie des Helmoltz Zentrums München.

Wissenschaftliche Betreuer:

Prof. Dr. Elmar W. Lang
Computational Intelligence and Machine Learning Group,
Institut für Biophysik und physikalische Biochemie,
Universität Regensburg

Prof. Dr. Dr. Fabian J. Theis
Lehrstuhl M12: Biomathematik,
Department of Mathematics,
Technische Universität München

Promotionsgesuch eingereicht am: 02.11.2010.
Das Kolloquium fand am 14.02.2011 statt.

Prüfungsausschuß:

Prof. Dr. Christian Schüller	(Vorsitzender)
Prof. Dr. Elmar W. Lang	(1. Gutachter)
Prof. Dr. Dr. Fabian J. Theis	(2. Gutachter)
Prof. Dr. Ingo Morgenstern	(3. Gutachter, weiterer Prüfer)

Sailors fighting in the dance hall
Oh man! Look at those cavemen go
It's the freakiest show
Take a look at the Lawman
Beating up the wrong guy
Oh man! Wonder if he'll ever know
He's in the best selling show
Is there life on Mars?

D. Bowie, Life on Mars

Summary

The most exciting questions in science share a common property: they are related to systems of highest complexity. Consider, for instance, a cell with its genetic and metabolic machinery. What makes it differentiate? How can a single cell give rise to a whole organism? In almost the same manner, neuroscientists ask how the human brain forms and works. A much deeper understanding of economic systems is required, e.g. to learn what makes economy run into bubbles or world-wide crisis and how to overcome them. What are the rules that form social structures ranging from families to nations? Are we finally even to predict the behavior of human beings, as recently approached by Barabási (2010)?

Physics has a long history in searching for answers to such questions. We only want to mention the works of von Neuman and Wiener on cybernetics, Haken (1977, 1983) on synergetics or the development of socio-dynamics by Weidlich (2000). Despite the studied problems being of tremendous complexity, physicists are always interested in searching for basic principles, unraveling the underlying organizing laws that explain certain systems on a high level of abstraction (Barabási, 2010). Biologists, social scientists or psychologists in contrast often object to this ultimate quest, arguing that fundamental laws just don't exist in their fields. Consequently, these sciences seem to be characterized by a strong emphasis on data accumulation and descriptive results, in other words by some intrinsic lack of abstraction. These differences tempted Rutherford to his famous provocation that "in science there is only physics; all the rest is stamp collecting"(Barabási, 2010).

Indeed, missing of fundamental laws may be too fast a prediction: It has been in the past years that people from many different disciplines began to recognize common organizing principles in the systems they were interested in. The key to this was that the objects of study were no longer analyzed as the full dynamical systems they are, but were rather reduced to a pure topology of interactions. This means that the functional form of the interactions was completely ignored, and instead one only analyzed which of the different players in a system are interacting with each other and which are not at all. In mathematics, this abstraction level is called a *graph* or *network*. It turned out that on this level fundamental insights and organizing principles shared by a huge variety of complex systems could be deciphered. Barabási

(2002) gives a popular introduction. Subsequently, over the last decade a new field *complex network science* has rapidly flourished as a new branch of statistical physics and now offers us a novel language to tackle such challenging questions.

Moreover, the last two decades have also witnessed an explosion of computational resources. This, for instance, allowed to observe large-scale real-world data of complex social systems for the first time in history. In addition, experimental techniques have advanced dramatically in many disciplines, such as microarray technology and metabolite measurements in biology or imaging techniques in neuroscience. All these developments now allow researchers to generate, store and analyze data in a way that they could not even imagine one generation ago. Reflecting the complexity of the systems under study, these data however are usually very hard to interpret which makes it necessary to employ or even develop specific tools that go beyond classical statistical approaches. Here, automated approaches are crucial, on the one side to cope with the high dimensionality of data, on the other side to provide objectivity of the analyses. To this end, the different disciplines need to get in close contact to machine learning, which is traditionally the scientific field dealing with the analysis of such data. Likewise, the machine learning community can profit from these contacts, first from the adoption of the complex network perspective, but also from the novel interdisciplinary application areas.

Overview

The goal of this thesis is therefore to exemplify how one can bring together methods for describing complex systems, mainly the language of complex network science, and machine learning approaches. Thereby it deals with several projects that arose from concrete questions to different complex systems. These systems stem from multiple fields of science. Considering the different applications as well as methodological approaches treated in this work, each Chapter is written as self-contained as possible. Having read the basic Chapter 1 that provides the necessary prerequisites, the reader should be able to directly skip to the subject he is interested in. Consequently, each Chapter contains a separate introduction which formulates the project's motivation and introduces the application details. Similarly, it gives separate conclusions and suggestions for future research.

The thesis is organized as follows. Chapter 1 provides the necessary background. Its first part is devoted to unsupervised learning approaches for data analysis. Techniques employed in the course of this thesis are introduced. We concentrate on clustering techniques and linear latent variable models. Regarding the first, we introduce hierarchical and k-means clustering, the two most prominent clustering

strategies. Focusing on the latter, we can essentially summarize related methods as matrix factorization algorithms. We demonstrate how constraining the factorization using (delayed) correlation, statistical independence or non-negativity leads to second-order algorithms (e.g. principal component analysis), independent component analysis and non-negative matrix factorization. Moreover, methods to evaluate cluster stability and algorithm performance are covered.

The second part of the Preliminaries gives a primer on complex networks. After an overview of prominent examples from various disciplines, we introduce basic definitions from graph theory. We then develop important indices for measuring a graph's topological properties; on a large-scale, we employ degree distributions and shortest-path lengths. Local structure is described by clustering coefficients and motifs. Subsequently, we give a detailed overview of the literature on vertex centrality measures which allow to quantify the importance of individual nodes. One of the most striking features of complex systems is their community structure. Detecting such communities is so far the key application of machine learning in complex networks. Section 5 recapitulates the development from the basic approaches to graph partitioning over the divisive Newman-Girvan algorithm to modularity optimization. Finally, we outline the ideas behind the Potts spin and the clique percolation method, two strategies to detect overlapping communities. We conclude with a short description of the seminal generative models and their salient properties.

A trend in the field of network science goes beyond the well studied binary graphs, towards more complex objects like weighted, directed, but also colored graphs. In Chapter 2 we focus on network analysis in the presence of self-loops, which are rarely taken into account in current approaches. We develop two measures of node centrality that can be applied in weighted, directed graphs and explicitly incorporate the role of such self-loops. Then, using these indices as similarity measures, we show that applying a clustering technique enables the automatized comparison of different graphs connecting a common set of nodes. Our application stems from empirical economics: In order to weaken the impact of the recent financial crisis to their local economies, many governments initiated support programs like the German car scrappage program. We ask the question how the effects of such programs on the different business sectors within a national economy can be quantified.

To this end, we take input-output tables which aggregate the flows of goods and services between the sectors as the adjacency matrices of corresponding weighted directed networks with self-loops. The first Section of the Chapter explains how we interpret the upper task as the search for a suitable node centrality measure for input-output networks. Then we derive our two centrality measures. Both are based upon random walks and have a direct economic interpretation as the propagation

of supply shocks. We further describe that the two measures differ in how they treat the self-loops. In the remainder of Chapter 2 we apply both measures to data from a wide set of countries and show that they uncover salient characteristics of the structures of these national economies. Clustering countries according to their sectors' centralities reveals geographical proximity and similar developmental status. Finally, we discuss the impact of our approach to the complex network as well as to the econophysics community.

Chapter 3 then goes beyond network analysis and exemplifies how to bridge the gap between interaction topology and system dynamics. It investigates the connection between the topology of hierarchical networks and the solutions of according dynamical systems, when the dynamics are modeled by two special types of differential equations. The first Section shows that such models are of high relevance for mathematical biology because they are mimicking signal transduction in biological systems in a generic way. Subsequently, we study the combined effect of the various kinetic parameters on the dynamics within hierarchical complex systems. For given topology these dynamics are determined by an interplay of the single parameters. We describe this by algebraic expressions which we call *effective parameters*.

In Section 3.2, we model switch-like interactions by Heaviside step functions. We show how to obtain the effective parameters recursively from the interaction graph. Their visualization as directed trees allows to determine the global effect of single kinetic parameters on the system's behavior. We provide evidence that our results partially generalize to sigmoidal Hill kinetics. Section 3.3 treats the case of linear activation functions. We show that effective parameters can be directly inferred from the interaction topology, which allows us to transform time-consuming analytic solutions of differential equations into a graph-theoretic problem. Finally, we focus on the connection of the effective parameters to learning. When fitting complex systems to actual data, commonly a large number of parameters has to be estimated. Moreover, it is often impossible to measure all species in the network, which makes these problems even more ill-determined. A toy example demonstrates how the effective parameters can be used to stabilize and speed up the estimation process. We conclude by discussing the domain of applicability for our methods.

Besides the social sciences, biology is by far the richest source for complex systems. The aim to understand phenomena like population dynamics, pattern formation or brain development was a driving force for research over the last decades. Recently, large-scale biological and biomedical data sets start to provide detailed information on some of these topics. However, the analysis of such data is still a field of extensive research because usually only few samples are available, whereas the dimensionality of data and the noise level are high. Chapter 4 deals with a prominent example

from bioinformatics, namely gene expression levels obtained from microarray chips. Besides standard clustering approaches or statistical tools, matrix factorization techniques are new and efficient tools in this area. Related methods successfully applied in the field differ significantly in concepts, but share the fact that they do not take prior knowledge into account. On the other hand, in signal processing strategies that incorporate intrinsic data properties like spatial and temporal structure have been shown to perform fast and robust. These approaches are commonly based on delayed correlations. However, large-scale biological data rarely imply a natural order that allows to define such a delayed correlation function.

Chapter 4 proposes to solve this issue by employing prior knowledge encoded in a weighted directed graph model. Linking features along this underlying graph introduces a partial ordering that allows us to define a graph-delayed correlation function. The first Section of the Chapter defines this concept. Using our framework as constraint to the matrix factorization task then allows us to set up the fast and robust graph-decorrelation algorithm GraDe. We also analyze identifiability in our situation. Then, after defining the novel concept of graph-moving average processes that allow generation of signals exhibiting graph-delayed correlation, we compare the performance of GraDe to other common methods. Subsequently, we show how our approach naturally can be derived from ordinary differential equation models for gene regulatory networks. The Section concludes with a toy example demonstrating the application of GraDe to gene expression data.

Section 4.2 then deals with the interpretation of a time-course microarray experiment on alterations in the gene response in *IL-6* stimulated primary mouse hepatocytes. First, the biological motivation and the data generation scheme is described. The following Subsection discusses in detail the time-resolved gene expression profiles extracted by GraDe. Subsequently, we validate these expression profiles via a pathway enrichment index and functional enrichments and compare our results to those obtained by standard methods. Finally, we demonstrate the robustness of our approach towards errors in the prior knowledge as well as biological noise.

Large-scale interaction networks derived from such experiments, but also from automated text mining approaches are increasingly available. Subsequent data integration in bioinformatics facilitates the construction of large biological networks whose particularity lies in their k-partiteness. The challenge is to analyze and interpret these networks in a comprehensive fashion. Community detection has received considerable attention over the last years, however only few researchers have focused on this generalized situation. Recently, Long et al. (2006) proposed a method for jointly clustering such a network and at the same time estimating a weighted graph connecting the communities. This allows simple interpretation of the resulting de-

composition. Chapter 5 extends this work by allowing fuzzy communities, which is crucial for the successful application to real-world data since biological networks consist of highly overlapping cohesive groups of vertices.

Section 5.2 starts with an illustration of the idea of graph approximation and then derives the fuzzy clustering algorithm. We propose an extended cost function for graph clustering and a novel efficient minimization procedure, mimicking the multiplicative update rules employed in non-negative matrix factorization. We validate our algorithm on graphs with known modular structure and also analyze the stability of the results towards initialization and the chosen number of clusters. In the rest of the Chapter, we decompose a tripartite disease-gene-protein complex graph. We begin with a description of the data and justify our choice of parameters. A feature of our algorithm is that it allows to work on different resolution levels. First, we evaluate whether we are able to structure this graph into biologically meaningful large-scale clusters. This is carried out by including functional annotations. Finally, focusing on the small-scale architecture, we exemplify how overlapping communities allow for reclassification or annotation of elements on a local level.

Despite the recent advances, our knowledge of most complex systems is still only partial. Hence, large-scale networks as analyzed before are usually far from complete. Chapter 6 is a first attempt to ask whether it is possible to predict the missing nodes and how they connect to the known parts of a network. Solutions to this task however seem out of reach without access to the dynamical properties of the system. Hence, we focus on the estimation of latent causes coupling to a dynamical system for which we have an incomplete model in the form of ordinary differential equations. In particular, we address the applicability of blind source separation methods to identifying such latent causes in biological systems. We focus on metabolic networks in Section 6.1, and analyze gene regulation in Section 6.2. First, we demonstrate how linear mixture models emerge in simple metabolic processes obeying first order mass action kinetics. Section 6.1.2 gives a proof of principle that latent causes can indeed be estimated using standard techniques in such a situation. However, more complex situations lead to new classes of blind source separation problems. In gene regulatory systems, interactions have switch-like character which we model by Hill functions and a neural network approach. Again, we provide proof of principles that latent causes can be estimated in special situations, where we can reduce the problem to a linear mixing model. The general situation of gene regulation leads us to novel non-linear blind source separation problems. Finally, Section 6.3 proposes strategies which may allow to cope with such non-linear situations in the future.

The last Chapter 7 concludes the thesis and summarizes its main contributions.

Publications

The results presented in the course of this thesis have led to the following papers that are already published or in the publication process (sorted by the corresponding Chapter):

- Chapter 2:
  - **Blöchl F**, Theis FJ, Vega-Redondo F, and Fisher E. Vertex Centralities in Input-Output Networks Reveal the Structure of Modern Economies. *Physical Review E, in press.*
  - **Blöchl F**, Theis FJ, Vega-Redondo F, and Fisher E. Which sectors of a modern economy are most central? *CESifo Working Paper Series No. 3175*, 2010.
- Chapter 3:
  - **Blöchl F**, Wittmann DM, and Theis FJ. Effective parameters determining the information flow in hierarchical biological systems. *Bulletin of Mathematical Biology, in press.*
- Chapter 4:
  - **Blöchl F**, Kowarsch A (equal contributors), and Theis FJ. Second-order source separation based on prior knowledge realized in a graph model. *In Proc. LVA/ICA 2010, volume 6365 of Springer LNCS, pages 434–441, St. Malo, France*, 2010. Springer.
  - Kowarsch A, **Blöchl F** (equal contributors), Bohl S, Saile M, Gretz N, Klingmüller U, and Theis FJ. Knowledge-based matrix factorization temporally resolves the cellular responses to *IL-6* stimulation. *BMC Bioinformatics, 11:585*, 2010.
- Chapter 5:
  - **Blöchl F**, Hartsperger ML (equal contributors), Stümpflen V, and Theis FJ. Uncovering the structure of heterogeneous biological data: fuzzy graph partitioning in the k-partite setting. *In Proc. GCB 2010, volume 173 of LNI, pages 31–40, Braunschweig, Germany*, 2010. GI.
  - Hartsperger ML, **Blöchl F** (equal contributors), Stümpflen V, and Theis FJ. Structuring biological data using fuzzy clustering of k-partite graphs. *BMC Bioinformatics, 11:522*, 2010.
- Chapter 6:

– **Blöchl F** and Theis FJ. Estimating hidden influences in metabolic and gene regulatory networks. *In Proc. ICA 2009, volume 5441 of Springer LNCS, pages 387–394, Paraty, Brasil*, 2009. Springer.

Besides these contributions, I have worked on various projects that are not described in this thesis since they are either not exactly within its scope, or collaboration partners had the main project lead. These resulted in the following publications:

- Wong P, Althammer S, Hildebrand A, Kirschner A, Pagel P, Geissler B, Smialowski P, **Blöchl F**, Oesterheld M, Schmidt T, Strack N, Theis F, Ruepp A, and Frishman D. An evolutionary and structural characterization of mammalian protein complex organization. *BMC Genomics, 9:629*, 2008.

- Wittmann DM, Schmidl D, **Blöchl F**, and Theis FJ. Reconstruction of graphs based on random walks. *Journal of Theoretical Computer Science, 410(38-40):3826–3838*, 2009.

- Wittmann DM, **Blöchl F**, Trümbach D, Wurst W, Prakash N, and Theis FJ. Spatial analysis of expression patterns predicts genetic interactions at the mid-hindbrain boundary. *PLoS Computational Biology 5:11*, 2009.

- Ansorg M, **Blöchl F**, zu Castell W, Theis FJ, and Wittmann DM. Gene regulation at the mid-hindbrain boundary: Study of a mathematical model in the stationary limit. *International Journal of Biomathematics and Biostatistics, 1(1):9-21*, 2010.

- Baskaran T, **Blöchl F** (equal contributors), Brück T, and Theis FJ. The Heckscher-Ohlin Model and the Network Structure of International Trade. *International Review of Economics and Finance, 20(2):135-145.*

- **Blöchl F**, Rascle A, Kastner J, Witzgall R, Lang EW, and Theis FJ. Are we to integrate previous information into microarray analyses? Interpretation of a Lmx1b-knockout experiment. *In Recent Progress in Biomedical Signal Processing, Edited by Górriz JM, Lang EW, Ramírez J. Bentham Science Publishers, in press.*

- Lang EW, Schachtner R, Lutter D, Herold D, Kodewitz A, **Blöchl F**, Theis FJ, Keck IR, Górriz Sáez JM, Gómez Vilda P, and Tome AM. Exploratory Matrix Factorization Techniques For Large Scale Biomedical Data Sets. *In Recent Progress in Biomedical Signal Processing, Edited by Górriz JM, Lang EW, Ramírez J. Bentham Science Publishers, in press.*

Acknowledgements

Here, I would like to thank the following people that accompanied and supported me during the last years.

First, my two supervisors who nicely complemented each other. Elmar Lang always had the time for inspiring and encouraging discussions. He willingly agreed to be at first a member of my thesis committee and finally even the first supervisor of this multi-disciplinary work. Fabian Theis gave me the opportunity to become an early member in his fast-growing group. I enjoyed the possibility to be part of various exciting projects and to work together with many excellent colleagues and collaborators. He also gave me the freedom to follow my own projects, no matter how far away from his own research, and even promoted this with additional funding. Thanks also for enabling my visits to Imperial College and all the travels to retreats, workshops, and conferences.

The entire CMB group and the rest of the IBIS for providing a fantastic working atmosphere, thank you all for the discussions, coffee breaks, barbecues, tabletop soccer games and conference trips.

Thushyanthan Baskaran for the unbelievably efficient collaboration when we wrote our Heckscher-Ohlin network paper. We had pleasant Skype discussions about economics, science in general, and girlfriends.

Eric Fisher, with whom I had a very intense collaboration on centralities in input-output graphs. He introduced me into the world of economics, and it also seems I needed an American tourist guide to show me the beautiful sides of Munich. Thanks for the lessons in life and related things during our walks in the English garden.

Mara Hartsperger for our fruitful collaboration on the fuzzy clustering algorithm and its applications. Thanks also for undertaking the cumbersome journey to Braunschweig and giving my talk there.

Andreas Kowarsch 'who wrote at least 2 papers for me'. We had a good collaboration in the GraDe project and many hard soccer duels. Big thanks to you for taking over large parts of the paper revision when I had to finish this thesis.

Dominik Wittmann, my room mate. In extensive collaborations, leading to four papers, but even more over our countless coffee discussions about science, our shared problems, and the rest he became a good friend. Thanks also for proof-reading and helpful comments on almost every publication I wrote.

My further collaboration partners in successful, failed, and unfinished projects, especially Marcus Ansorg, Tilman Brück, Sabine Dietmann, Harold Gutch, Jan Krumsieck, Carsten Marr, Nilima Pakash, Andreas Ruepp, Daniel Schmidl, Martin Sturm, Fernando Vega-Redondo, Ralph Witzgall, and Philip Wong.

Finally, my parents and the rest of the family, I want to apologize for the continuous disregard. I know that you are always there when you are needed.

Above all I want to thank Miriam for sharing her life with me. She's the one.

Thank you all

Florian

Contents

1 Preliminaries

This thesis addresses novel approaches for the understanding of complex systems, with a focus on the actual use in different application domains. The systems analyzed will be primarily treated within the language of complex network science, a branch of statistical physics that has emerged in the last decade. In particular, we ask how this new discipline and the field of machine learning can profit from each other. This first Chapter lays the basis for the topics addressed later and introduces both the necessary machine learning techniques and the key concepts for analyzing and modeling of complex networks.

1.1 Preliminaries from machine learning

Providing the necessary background in machine learning, we focus on two classes of unsupervised learning for data analysis: clustering techniques and latent variable models. The methods and algorithms employed in the course of this thesis are introduced, and the concepts our novel approaches build on are explained. For broader introductions to the field we refer the reader to the various textbooks, e.g. by Hastie et al. (2001) or Bishop (2006) which we follow in parts.

1.1.1 Clustering

Clustering techniques are a common approach to unsupervised data analysis. As *cluster analysis* or simply *clustering* we subsume techniques that allow the assignment of a set of observations into subsets – the clusters – such that observations belonging to the same cluster are similar in some sense.

Clustering methods can be divided into two classes: *hierarchical* and *partitional* clustering. Hierarchical algorithms successively identify clusters using previously established ones. Partitional clustering, on the other hand, attempt to directly decompose the data set into a set of clusters.

The following Sections introduce the two classical algorithms for both classes: hierarchical clustering and k-means clustering in its fuzzy and hard version. We further discuss a method for the evaluation of algorithm stability. These techniques will be applied in the remainder of this thesis.

1.1.1.1 Hierarchical clustering

Perhaps the easiest and most commonly used clustering method is hierarchical clustering, for a detailed mathematical treatment see e.g. (Hastie et al., 2001). It creates a hierarchy of clusters which is commonly represented in a tree structure called a *dendrogram*: at the highest level, it consists of a single cluster containing all observations, while the leaves of the tree correspond to individual observations. The hierarchy has $n - 1$ levels, where n is the number of data points to cluster.

Strategies for hierarchical clustering are either divisive ("top-down") or agglomerative ("bottom-up"). Divisive algorithms begin with the whole data set and divide it into successively smaller clusters. We will later employ an agglomerative algorithm, which begins with each data point as a separate cluster. Then, at each step, the two most similar clusters are merged into a single one, producing one less cluster. The amount of similarity between merged clusters can be encoded in the branch heights of the dendrogram. The algorithm stops after $n - 1$ steps, where it ends up with one single cluster.

This procedure requires to determine the similarity between groups based on the similarity of the elements they contain, which however involves some arbitrariness. The commonly used strategies are

- *Single-linkage*: the distance between two clusters is the distance between their two closest members.
- *Complete-linkage*: the distance between two clusters is defined as the distance between their two farthest members.
- *Average-linkage*: the distance between two clusters is calculated based on the average values using all elements of each cluster.
- *Ward's method*: it aims to minimize the increase of the within-cluster distances. At each step, the union of every possible cluster pair is considered and the two clusters whose fusion leads to the minimum increase are combined.

However, in order to cluster data points we first of all need to define a precise measure for the similarity of two data points $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^m$. Commonly used (dis-)similarity measures are the Euclidean distance, the maximum distance or the Manhattan distance, where $d(\mathbf{x}_1, \mathbf{x}_2) := \sum_{i=1}^{m} |x_{1i} - x_{2i}|$. Likewise, Pearson's or Spearman's correlation coefficients are useful to quantify similarity.

Hierarchical clustering has the advantage that it does not require preliminary knowledge on the number of clusters. However, it does not provide a direct way to discriminate between the many partitions obtained by the procedure. Moreover,

the results of the method depend on the used similarity measure and the linkage strategy.

1.1.1.2 k-means clustering

The most popular partitional approach is k-means clustering. Here, the number of clusters has to be preassigned, say we want to estimate m clusters.

Our data points $\mathbf{x}_1 \dots \mathbf{x}_n$ are embedded in a metric space, and we again have to choose a distance measure d between them, for instance one of those mentioned in the last Section. Denoting the partition of the data set into m disjunct clusters by $C = \{C_1, \dots, C_m\}$, this algorithm represents clusters by *centroids* $\mathbf{y}_j, j = 1, \dots, m$, which are the cluster centers

$$\mathbf{y}_j = \frac{\sum_{\mathbf{x}_i \in C_j} x_i}{\sum_{\mathbf{x}_i \in C_j}} . \tag{1.1}$$

Each data point is then assigned to the cluster with the nearest centroid. Note that centroids do not have to be actual data points. The goal of the k-means algorithm is to partition the data points into clusters in a way that the within-cluster sum of distances is minimized. We can formulate a corresponding cost function

$$f(C) = \sum_{j=1}^{m} \sum_{\mathbf{x}_i \in C_j} d(\mathbf{x}_i, \mathbf{y}_j) . \tag{1.2}$$

The common algorithm to optimize this cost function is an iterative refinement technique called *Lloyd's algorithm* (Lloyd, 1982). It starts with randomly chosen centroids. Then, each vertex is assigned to the nearest centroid, and new cluster centers can be estimated. This new set of centroids allows for a new classification of the observations, and so on. Typically, after a small number of iterations the positions of the centroids are essentially stable, and the algorithm has converged.

Analyzing real-world data, there is often no sharp boundary between clusters and *fuzzy* techniques are better suited: Instead of crisp assignments of data points to clusters, one introduces continuous *degrees of membership* between zero and one. These can be aggregated in a degree of membership matrix $\mathbf{C}$, where the entry $\mathbf{C}_{ij}$ quantifies the assignment of observation i to cluster j. We normalize $\mathbf{C}$ to be right-stochastic, i.e. the degrees of membership of each data point sum up to one.

A prominent fuzzy clustering algorithm is the fuzzy version of the k-means (Bezdek, 1981, Dunn, 1973). It is based on the minimization of the following objective function that generalizes Equation (1.2):

$$f(\mathbf{C}) = \sum_{j=1}^{m} \sum_{i=1}^{n} (\mathbf{C}_{ij})^{\mu} \, d(\mathbf{x}_i, \mathbf{y}_j) , \tag{1.3}$$

where we additionally introduce a *fuzzification factor* $\mu \leq 1$. Such fuzzification factors are a common strategy to extend cost functions to include fuzzy clusters. The fuzzy centroids in Equation (1.3) are calculated as

$$\mathbf{y}_j = \frac{\sum_{i=1}^{n} (\mathbf{C}_{ij})^{\mu} \mathbf{x}_i}{\sum_{i=1}^{n} (\mathbf{C}_{ij})^{\mu}} .$$

Then, one employs the same incremental procedure as in the hard clustering, except the update rule of the degrees of membership being modified to

$$\mathbf{C}_{ij} = \frac{1}{\sum_{l=1}^{m} \left(\frac{d(\mathbf{x}_i, \mathbf{y}_j)}{d(\mathbf{x}_i, \mathbf{y}_l)} \right)^{\frac{1}{\mu - 1}}} . \tag{1.4}$$

When μ is close to 1, the cluster center closest to the observation is given much more weight than the others and the algorithm is similar to k-means.

However, the solutions found by these two algorithms are not optimal, and strongly depend on the initial choice for the centroids. Therefore, the results are usually improved by performing multiple runs starting from different initializations, and picking the best solution obtained. Besides its dependency on the random initialization, a limitation of the discussed approaches is that the number of clusters must be specified at the beginning instead of being derived by the algorithm.

1.1.1.3 Evaluating cluster stability

In case of a non-deterministic algorithm like k-means clustering it is crucial to understand the stability of the cluster assignments towards the random initialization.

One possible approach to quantify the stability or replicability of estimated hard clusters is *Cramer's* v^2 (Agresti, 1996, Garge et al., 2005). This index employs the χ^2 statistics to measure the degree of association in contingency tables larger than 2×2. Clustering a data set twice with two different initializations, we obtain a two-way contingency table. Cramer's v^2 then measures the squared canonical correlation between the two sets of nominal variables, indicating the proportion of variance of one clustering run that can be explained by the other one:

$$\text{Cramer's } v^2 = \frac{\chi^2}{n\,(m-1)} . \tag{1.5}$$

Here, χ^2 is the usual χ^2 test statistic for testing independence in the contingency tables, n the number of elements to be clustered, and m the number of clusters extracted. The index ranges from 0 to 1, with 1 indicating a perfect reproducibility.

One can in principle also use Cramer's v^2 in the case of fuzzy clustering. However, then crisp assignment of data points to clusters – every data point is assigned to the

cluster showing maximum degree of membership – is required. As this may affect the stability score, we will later use a different similarity measure, the so-called *fuzzy Rand index* (FRI) recently proposed by Hüllermeier and Rifqi (2009).

Let $\mathbf{C}$ be the matrix of degrees of memberships from a fuzzy clustering of a data set. A fuzzy equivalence relation on $\mathbf{C}$ is then defined in terms of a similarity measure on the degrees of membership vectors. Generally, this relation can be defined via any distance measure on $[0,1]^m$ that yields values in $[0,1]$. We will employ the maximum norm and define the distance $d_{\mathbf{C}}(x_r, x_s)$ between the degrees of memberships of two data points x_r, x_s as

$$d_{\mathbf{C}}(x_r, x_s) := \max_t |\mathbf{C}_{st} - \mathbf{C}_{rt}| \,. \tag{1.6}$$

Now, given two different fuzzy clusterings $\mathbf{C}$ and $\mathbf{C}'$ resulting from two random initializations we calculate the FRI that is defined as the degree of concordance

$$FRI(\mathbf{C}, \mathbf{C}') = 1 - \frac{\sum_{r<s} |d_{\mathbf{C}}(x_r, x_s) - d_{\mathbf{C}'}(x_r, x_s)|}{n\,(n-1)\,/2} \,, \tag{1.7}$$

where n is the number of data points. The FRI ranges from 0 to 1, with 0 indicating no relationship and 1 indicating a perfect reproducibility. For a detailed mathematical analysis of this measure we refer to Hüllermeier and Rifqi (2009).

1.1.2 Latent variable models

Latent variable models, commonly in the context of *blind source separation* (BSS), have raised much interest in the signal processing community over the last decades. In short, here the aim is to recover latent variables that are the underlying sources of observed mixtures. The term *blind* is used because neither the original signals nor the mixing process are known. Related techniques that we discuss in this Section have a multitude of relevant applications, e.g. in telecommunications, the analysis of financial data, or biological and biomedical signal processing (Blöchl et al., 2010, Hyvärinen et al., 2001, Theis and Meyer-Bäse, 2010).

The standard example for the visualization of BSS is the "cocktail-party problem": Imagine n people talking at a cocktail party, acting as sound sources. Their conversation is recorded by m microphones which are positioned in different places around the room. Due to the different distances between speakers and microphones every recorded signal is a weighted mixture of the talks. Now, we want to extract both the individual speakers (source signals) and the mixing process from the recorded (mixed) signals only.

Assuming this mixing process to be instantaneous and linear, the blind source separation model can be formulated more precisely: denoting the source signals at

time t by $s_1(t) \dots s_n(t)$ and the recorded signals by $x_1(t), \dots x_m(t)$, their functional relation is given by the linear combination

$$\begin{aligned} x_1(t) &= a_{11}s_1(t) + a_{12}s_2(t) + \cdots + a_{1n}s_n(t) \\ &\vdots \\ x_m(t) &= a_{m1}s_1(t) + a_{2m}s_2(t) + \cdots + a_{mn}s_n(t). \end{aligned}$$

The mixing coefficients a_{ij} quantify the weight of speaker s_j in the signal x_i. Now, we aggregate the m measured observations of these mixtures in a data matrix $\mathbf{X} \in \mathbb{R}^{m \times l}$. We then may write the upper equations in matrix representation and arrive at the common formulation of the *linear mixing model*:

$$\mathbf{X} = \mathbf{AS}\,. \tag{1.8}$$

In the following, we always assume the *mixing matrix* $\mathbf{A}$ to be of full rank. Moreover, for simplicity we center all data denoted by $\mathbf{X}$ to mean zero.

Of course the matrix decomposition (1.8) has an infinite number of solutions, so further assumptions have to be made. The following Sections introduce common matrix factorization techniques that are based on different constraints imposed on the sources' properties.

We will employ matrix factorization techniques mainly for the unsupervised extraction of overlapping clusters. To this end, we threshold the obtained sources; all data points within one source that are above this threshold are then assigned to the same cluster. Since a data point may have a strong contribution to more than one sources, the obtained clusters may be overlapping. The determination of the threshold however is an important issue of this approach which is, for instance, discussed by Lutter et al. (2008).

1.1.2.1 Principal component analysis

One possible approach is *whitening* of the data. Here, we first assume that the underlying sources are *decorrelated*, i.e. the cross-correlation matrix $\mathbf{C_S}$ of $\mathbf{S}$ is diagonal. This matrix can be easily estimated using the unbiased variance estimator

$$\mathbf{C_S} = E(\mathbf{SS}^T) = \frac{1}{l-1}\,\mathbf{SS}^T\,. \tag{1.9}$$

Second, the scaling indeterminacy of the matrix factorization can be fixed by requiring that the sources have unit variance, hence $\mathbf{C_S} = \mathbf{I}$. A whitening matrix can be easily calculated from the eigenvalue decomposition of the data covariance matrix $\mathbf{C_X}$, which always exists since this matrix is symmetric: let $\mathbf{VC_XV}^T = \mathbf{D}$ with orthogonal $\mathbf{V}$ and the diagonal matrix $\mathbf{D}$ of eigenvalues $|\mathbf{D}_{11}| \geq |\mathbf{D}_{22}| \geq \cdots \geq |\mathbf{D}_{nn}|$.

In the following, we assume that these eigenvalues are pairwise different, which is no severe restriction as this is always the case when working with numerical data. We define $\mathbf{U} = \mathbf{D}^{-1/2}\mathbf{V}$, then

$$\begin{aligned}\mathbf{C_{UX}} &= E(\mathbf{U}\mathbf{X}\mathbf{X}^T\mathbf{U}^T) \\ &= \mathbf{U}\mathbf{C_X}\mathbf{U}^T \\ &= \mathbf{D}^{-1/2}\mathbf{V}\mathbf{C}_X\mathbf{V}^T\mathbf{D}^{-1/2} \\ &= \mathbf{D}^{-1/2}\mathbf{D}\mathbf{D}^{-1/2} = \mathbf{I}\end{aligned}$$

Hence, $\mathbf{U}$ is an *unmixing* matrix for the mixed data $\mathbf{X}$ under the requested constraints. Correlation as the basic measure to indicate a relation between two signals is a relatively weak statistical requirement to the sources. Hence, these assumptions leave a large indeterminacy. Let $\mathbf{X}$ be white, i.e. centered and with $\mathbf{C_X} = \mathbf{I}$, and $\mathbf{G}$ be an arbitrary $n \times n$ matrix with full rank. Then,

$$\mathbf{C_{GX}} = E(\mathbf{G}\mathbf{X}\mathbf{X}^T\mathbf{G}^T) = \mathbf{G}\mathbf{C_X}\mathbf{G}^T = \mathbf{G}\mathbf{G}^T\,.$$

The whitening transformation $\mathbf{U}$ is therefore unique only up to an orthogonal transformation.

A method that allows for the unique decomposition of several correlated signals into an equal or smaller number of uncorrelated random variables is *principle component analysis* (PCA), as already introduced by Pearson (1901). This widely used technique transforms multivariate data into a new orthogonal basis, where the first new basis vector – the *first principal component* (PC) – refers to the direction with the largest data variance. Mathematically, we search for a vector $\mathbf{y_1}$ such that the linear combination $\mathbf{s_1} := \mathbf{y_1}^T\mathbf{X}$ has maximum variance. We are only interested in the direction of $\mathbf{y_1}$ and therefore may require $|\mathbf{y_1}| = 1$. Then

$$\mathbf{C_{s_1}} = \mathbf{C}_{\mathbf{y_1}^T\mathbf{X}} = \mathbf{y_1}^T\mathbf{C_X}\mathbf{y_1} = \mathbf{y_1}^T\mathbf{V}^T\mathbf{D}\mathbf{V}\mathbf{y_1}\,,$$

where again $\mathbf{V}^T\mathbf{D}\mathbf{V} = \mathbf{C_X}$ is the eigenvalue decomposition of the data covariance. With $\mathbf{V}$ being orthogonal, also $|\mathbf{V}\mathbf{y_1}| = 1$ and we see that we achieve the maximum variance when $\mathbf{y_1}$ is the first unit vector. Hence, the desired $\mathbf{s_1}$ is the eigenvector to the largest eigenvalue $\mathbf{D}_{11}$ of $\mathbf{C_X}$. The second PC is orthogonal to the first one and carries the largest amount of variance remaining. Analogously, we find it to be the second eigenvector of $\mathbf{C_X}$ and so on (remember that we assumed pairwise different eigenvalues). The orthogonality of eigenvectors implies the decorrelation of the different principal components.

The decomposition into PCs is unique except for scaling; by choosing the directions of maximum variance the rotational invariance of the whitening transformation is

broken. Since PCA takes only the mean and variance of a data set into account, it is called a *second-order technique*.

There are no model restrictions but the existence of the first two moments that are estimated from the data. Taking data variance as a measure for information content, PCA can be used for dimension reduction via projection onto the space spanned the first PCs. This dimension reduction is a common pre-processing step in more elaborate approaches.

1.1.2.2 Second-order methods using time structure

A frequent interest in signal processing is to find repeating patterns in a data set, such as the presence of a periodic signal. This can be achieved using *time-delayed correlations*, which quantify the similarity of a signal with itself after a time shift. In addition to the delayed correlation of one signal, we can define delayed cross-correlations between two signals. For instance, the *time-delayed correlation matrix* of a centered, wide-sense stationary multivariate random process $\mathbf{x}(t)$ is

$$\left(\mathbf{C}_{\mathbf{x}}(\tau)\right)_{ij} := E\left(\mathbf{x}_i(t+\tau)\mathbf{x}_j(t)^\top\right), \tag{1.10}$$

where E denotes expectation. Here, off-diagonal elements detect time-shifted correlations between different data dimensions. For a given data matrix $\mathbf{X}$, the time-delayed correlation matrix can be easily estimated with the unbiased variance estimator. In the following, we use a slightly modified version, the symmetrized time-delayed covariance matrix:

$$\bar{\mathbf{C}}_{\mathbf{X}}(\tau) = \frac{1}{2}\left(\mathbf{C}_{\mathbf{X}}(\tau) + \mathbf{C}_{\mathbf{X}}^T(\tau)\right).$$

For $\tau = 0$, this reduces to the common cross-correlation.

The so far discussed approaches to solve the matrix factorization problem have considered independent random variables, where the samples in particular have no intrinsic order. In many applications, however, we observe mixtures of temporal signals, or images where a well defined ordering is obviously present. In the following, we introduce a technique which makes assumptions on the temporal structure of the sources instead of taking into account higher-order moments. It allows for the estimation of the model when observing time-resolved data.

Now, we will use the information in a time-delayed covariance matrix as constraint to the BSS problem and try to find a factorization such that not only the instantaneous cross-covariances of the sources as in the PCA case, but also all (symmetrized) time-delayed cross-covariances vanish. In other words, $\bar{\mathbf{C}}_{\mathbf{S}}(\tau)$ has to be diagonal for all τ. We will see that this extra information is enough to estimate the model, under the conditions specified below. No higher-order information is needed.

Under the above assumption, the time-delayed correlation matrices of the observations have the following structure:

$$\bar{\mathbf{C}}_{\mathbf{X}}(\tau) = \mathbf{A}\bar{\mathbf{C}}_{\mathbf{S}}(\tau)\mathbf{A}^{\top} . \tag{1.11}$$

A full identification of $\mathbf{A}$ and $\mathbf{S}$ is not possible because we always can exchange a scalar factor between a source and the corresponding column of the mixing matrix. Hence, without any loss of generality, we assume the sources have unit variance. This normalization turns out to be extremely convenient: Since the sources are assumed to be uncorrelated, we have $\bar{\mathbf{C}}_{\mathbf{S}}(0) = \mathbf{I}$ and Equation (1.11) simplifies to

$$\bar{\mathbf{C}}_{\mathbf{X}}(0) = \mathbf{A}\mathbf{A}^{T} . \tag{1.12}$$

After whitening our observations, $\bar{\mathbf{C}}_{\mathbf{X}}(0) = \mathbf{I}$ and therefore the normalization makes $\mathbf{A}$ orthogonal. Thus, Equation (1.11) describes the eigenvalue decomposition of the symmetric matrix $\bar{\mathbf{C}}_{\mathbf{X}}(\tau)$. This was the reason to work with the symmetrized instead of the simple time-delayed correlations: The spectral theorem guarantees that the eigenvalue decomposition exists and, moreover, is unique, if all eigenvalues are pairwise different. In addition to this uniqueness result, we see that the unmixing matrix $\mathbf{U}$ for a fixed choice of τ can be easily obtained by calculating the eigenvalue decomposition of $\bar{\mathbf{C}}_{\mathbf{X}}(\tau)$.

Altogether we have derived the simple AMUSE (Algorithm for Multiple Unknown Signals Extraction) algorithm (Molgedey and Schuster, 1994, Tong et al., 1991). In summary, it performs the following steps:

1. Whiten the data.
2. Choose a time lag τ and compute the eigenvalue decomposition of the time-delayed covariance matrix. These eigenvectors form the rows of the desired separating matrix.

In practice, if the eigenvalue decomposition turns out to be problematic, choosing a different τ may often resolve this problem. Nonetheless, there may still be sources with equal time-delayed correlation spectrum. Moreover, the performance of AMUSE is known to be relatively sensitive to additive noise and the numerical estimation by a finite amount of samples may lead to a badly estimated autocorrelation matrix (Theis et al., 2004).

A strategy that considerably improves the performance of AMUSE is the use of *several* time lags instead of a single one, as for instance in SOBI (Belouchrani et al., 1997), TDSEP (Ziehe and Mueller, 1998), or TFBSS (Févotte and Doncarli, 2004). Then, it can be shown that it is enough when the delayed correlations for one of

these time lags are different (Hyvärinen et al., 2001). Thus the choice of τ is a less serious problem.

In principle, using several time lags, we have to simultaneously diagonalize the corresponding time-delayed correlation matrices. This diagonalization is probably not exact, since the eigenvectors of the different covariance matrices are unlikely to be identical. So, one has to formulate and optimize functions expressing the degree of diagonalization obtained. LSDIAG, for instance, is an iterative linear least-squares algorithm based on a multiplicative update rule: it performs gradient descent on the sum of the off-diagonal terms (Ziehe et al., 2003). Yeredor (2002) proposes an iterative alternating-directions algorithm called AC-DC that minimizes the weighted least squares criterion with respect to a general – not necessarily orthogonal – diagonalizing matrix. We will always employ the freely available Jacobi-type algorithm proposed by Cardoso and Souloumiac (1995), which iteratively constructs the solution by Givens rotation in two coordinates. A Givens rotation is a rotation matrix that only acts in the plane spanned by two coordinate axes, rotating by a chosen angle. Any orthogonal matrix can be built up as a product of such elementary rotations. The Jacobi idea consists of successively applying Givens rotations to the $\bar{\mathbf{C}}_{\mathbf{X}}(\tau)$ in order to minimize the total sum of the off-diagonal elements. The interesting aspect of this method is that the minimization step can be done *algebraically*. For the technical details, we refer to the review by Févotte and Theis (2007) who also discuss implementation strategies.

1.1.2.3 Independent component analysis

In *independent component analysis* (ICA), one assumes that the underlying sources, the so-called *independent components*, are statistically independent. Statistical independence is a much stronger requirement than decorrelation as discussed in Section 1.1.2.1. In fact, all ICA algorithms employ data whitening as the first step to independence. It can be shown that by assuming independence we may achieve a unique solution of the matrix decomposition, if at most one of the sources has a Gaussian distribution and the mixing matrix has full column rank (Comon, 1994, Theis, 2004). The latter implies that the number of mixtures is at least as large as the number of sources. Unique in this context means unique modulo scaling and permutation; performing these operations on $\mathbf{S}$ can always be compensated by corresponding operations on the columns of $\mathbf{A}$.

In practice, it is not straightforward to measure statistical independence, which therefore has to be approximated. To this end, a common approach is *non-gaussianity* (Hyvärinen et al., 2001): from a heuristic interpretation of the central limit theorem it follows that any weighted sum of independent sources is "more Gaussian" than

the sources themselves. So maximizing non-gaussianity is a way to reveal the independent underlying sources. This property can be quantified by the fourth-order cumulants, the *kurtosis*. The kurtosis $\text{kurt}(x) := E(x^4) - 3(E(x^2)^2)$ is a measure for the peakedness of a probability distribution x and vanishes for a Gaussian.

A second measure is the negentropy, which is based on the information-theoretic concept of entropy. The entropy of a random variable is related to the information that its observation gives. The more random, i.e. unpredictable and unstructured, the variable is, the larger is its entropy. Its largest value among all random variables of equal variance is found for a Gaussian. Robust approximations of negentropy instead of kurtosis may enhance the statistical properties of the resulting estimator.

Further approximations exist, however the two widely used algorithms are based on the outlined ideas: JADE (Cardoso, 1999) carries out an approximate joint diagonalization (as discussed in the last Section 1.1.2.2) of the fourth-order cumulants. FastICA developed by Hyvärinen (1999) is based on a fixed-point iteration scheme that maximizes negentropy. We will later use the MATLAB implementation that is freely available at `http://www.cis.hut.fi/projects/ica/fastica/`.

1.1.2.4 Non-negative matrix factorization

Non-negative matrix factorization (NMF) is a further method to solve the BSS problem. Here, the constraint is that no negative entry is allowed in both the mixing matrix and the extracted sources. Therefore the statistical requirements to the sources are omitted or at least weakened. By not allowing negative entries, NMF enables a purely additive combination of parts that together reconstruct the original data. A classical example is the parts-based decomposition of face images from Lee and Seung (1999). Recently, NMF has gained attention in a variety of applications in computational biology, including the discovery of molecular patterns in 'omics' data via unsupervised clustering (Devarajan, 2008, Schachtner et al., 2008).

So, we want to decompose a data set $\mathbf{X}$ as a product of two non-negative matrices $\mathbf{A}$ and $\mathbf{S}$ such that $\mathbf{X} \approx \mathbf{AS}$, where this factorization may only be an approximation. Of course this decomposition is far from being unique. Obviously, the inner dimension has to be reduced, otherwise the factorization is completely arbitrary.

Algorithmically, a direct approach to NMF is the constrained minimization of the quadratic reconstruction error

$$f(\mathbf{A}, \mathbf{S}) = \|\mathbf{X} - \mathbf{AS}\|_2^2 = \sum_i \sum_j \left(\mathbf{X}_{ij} - \sum_k \mathbf{A}_{ik}\mathbf{S}_{kj} \right)^2 . \tag{1.13}$$

From this expression, different learning techniques can be obtained from different constraints on the sources $\mathbf{S}$. For instance, one can show that PCA corresponds to

an unconstrained minimization. Requiring non-negativity of all entries, a seminal paper by Lee and Seung (1999) proposes a gradient descent technique to minimize this cost function. In the following, we review their approach. Its key ideas will be adopted in Chapter 5 to develop a fuzzy graph-partitioning algorithm. We derive the update rules only for the matrix $\mathbf{A}$, the corresponding expressions for $\mathbf{S}$ follow from symmetry arguments by considering the transposed BSS problem $\mathbf{X}^T \approx \mathbf{S}^T\mathbf{A}^T$. Taking the derivative of the cost function with respect to an element $\mathbf{A}_{rs}$, we find

$$\begin{aligned} \frac{\partial f(\mathbf{A},\mathbf{S})}{\partial \mathbf{A}_{rs}} &= -2\sum_j \left(\mathbf{X}_{rj} - \sum_k \mathbf{A}_{rk}\mathbf{S}_{kj} \right) \cdot \mathbf{S}_{sj} \\ &= -2\left(\mathbf{X}\mathbf{S}^T - \mathbf{A}\mathbf{S}\mathbf{S}^T\right)_{rs} . \end{aligned} \tag{1.14}$$

Now, we could minimize f by alternating gradient descent: starting from initial guesses for $\mathbf{A}$ and $\mathbf{S}$, we alternate between updates of the $\mathbf{A}_{rs}$ and the $\mathbf{S}_{rs}$ with learning rates $\eta^{\mathbf{A}}_{rs}$ and $\eta^{\mathbf{S}}_{rs}$, respectively. The update rule for $\mathbf{A}_{rs}$ then reads

$$\mathbf{A}_{rs} \leftarrow \mathbf{A}_{rs} - \eta^{\mathbf{A}}_{rs} \frac{\partial f(\mathbf{A},\mathbf{S})}{\partial \mathbf{A}_{rs}} = \mathbf{A}_{rs} + 2\eta^{\mathbf{A}}_{rs} \left(\mathbf{X}\mathbf{S}^T - \mathbf{A}\mathbf{S}\mathbf{S}^T\right)_{rs} . \tag{1.15}$$

However, such update rules have two disadvantages: first, the choice of the update rates (possibly different for $\mathbf{A}$ and $\mathbf{S}$) is unclear; in particular, for too small η convergence may take too long or may not be achieved at all, whereas for too large η we may easily overshoot the minimum which may lead to negative entries. Hence, Lee and Seung propose to use *multiplicative* update rules and define

$$\eta^{\mathbf{A}}_{rs} := \frac{\mathbf{A}_{rs}}{2(\mathbf{A}\mathbf{S}\mathbf{S}^T)_{rs}} . \tag{1.16}$$

With this choice, from Equation (1.15) we obtain the update rule

$$\mathbf{A}_{rs} \leftarrow \mathbf{A}_{rs} \frac{(\mathbf{X}\mathbf{S}^T)_{rs}}{(\mathbf{A}\mathbf{S}\mathbf{S}^T)_{rs}} . \tag{1.17}$$

Since the update rates are not small, one may wonder why such a multiplicative gradient descent should cause the cost function to decrease. Surprisingly, this is indeed the case as Lee and Seung could show via auxiliary functions. Multiplicative update rules incorporate the non-negativity constraint automatically in an elegant way, since all factors on the right hand side are positive. However, a new possible drawback arises: once a matrix entry has been set to zero, which may happen due to zeros in the mixing matrix or to numerics, the coefficient will never then be able to become positive again during learning.

Several other methods for NMF have been proposed, Berry et al. (2007) give a recent survey. In particular, other cost functions like the generalized Kullback-Leibler

divergence are often employed. In contrast to the upper interior-point optimization, where no entry ever becomes negative, *projected gradient methods* perform a step into the descent direction and then project the result back onto the non-negative orthant. Here, an issue is that the projection actually may increase the cost function. A third class of NMF algorithms employs *alternating least squares* (Paatero and Tapper, 1994). Alternating least squares algorithms exploit the fact that the cost function from Equation (1.13) is convex in either $\mathbf{A}$ or $\mathbf{S}$, it is not convex in both together. Thus, given one of these matrices, the other one can be found with a simple least squares computation. A very successful concept is the integration of an additional condition: requiring sparseness of the two matrices results in better localized features (Hoyer, 2004) and moreover allows for theoretical results on uniqueness of the factorization, see e.g. (Theis et al., 2005). Finally, in the case of noisy mixtures, integrating the explicit form of this noise into the algorithmic solution has been shown to enhance performance, e.g. by Neher et al. (2009).

1.1.2.5 Performance indices

The performance of any matrix factorization technique has to be evaluated on artificially generated data with known mixing matrices. The most common BSS situation is the square case where there are as many mixtures as sources. Then, the most widely used measure for assessing the accuracy of this estimation is the *Amari (performance) index* (Cichocki and Amari, 2002):

$$\mathrm{AI}(\mathbf{E}) = \sum_{i=1}^{n}\left(\sum_{j=1}^{n}\frac{|\mathbf{E}_{ij}|}{\max_k|\mathbf{E}_{ik}|}\right) + \sum_{j=1}^{n}\left(\sum_{i=1}^{n}\frac{|\mathbf{E}_{ij}|}{\max_k|\mathbf{E}_{kj}|}\right). \tag{1.18}$$

The Amari index quantifies the deviation of $\mathbf{E} := \mathbf{UA}$, i.e. the product of the estimated unmixing matrix $\mathbf{U}$ and the known mixing matrix $\mathbf{A}$, from a permutation matrix. A value of zero indicates perfect separation. The larger the index is, the poorer is the performance of a separation algorithm.

Likewise, we will also measure the recovery quality of sources with the *signal-to-noise ratio*. It measures which portion of original signal $\mathbf{s_1}$ has been corrupted in the estimated source $\mathbf{s_2}$:

$$\mathrm{SNR}(\mathbf{s_1}, \mathbf{s_2}) = 20\log\frac{|\mathbf{s_1}|}{|\mathbf{s_1} - \mathbf{s_2}|}. \tag{1.19}$$

This logarithmic index is measured in the unit dB.

1.2 Introduction to complex networks

This Section gives an introduction into the recently emerged field of complex network science, partially following the reviews by Newman (2003) and Boccaletti et al. (2006). After an overview of the manifold examples of networks that arose in different disciplines, it provides the necessary basics from graph theory and reviews important graph properties. The following two Subsections are devoted to vertex centrality measures and community detection methods. It concludes with a short description of the seminal generative models.

1.2.1 Complex networks in nature

Once one adopts the language of complex networks to describe the world around us, these structures seem to emerge everywhere, which makes the field quite interdisciplinary. Here, we review some prominent examples from different sciences:

- **Social sciences:** The social sciences have a long history in what they call Social Network Analysis, starting already in the 1920s (Freeman, 2006). Many fundamental concepts presented in the following Sections have their origin in sociometry. Social networks link a set of people or groups with a pattern of contacts or interactions. Here, for instance, one can investigate friendship, sexual contacts or opinion patterns, but also business relationships between companies, intermarriages between families, and many more (Vega-Redondo, 2007). Social network analysis traditionally suffered from data inaccuracy and small sample size. It was only in the last few years that large scale data became available, as for example the collaboration networks of movie actors in the Internet Movie Database. Other examples are scientific authorship graphs or networks of company directors (Grassi, 2010, Newman, 2003).

- **Communication:** A rich source for recovering social interactions are communication networks like the networks of phone calls, e-mail messages or mail (Diesner et al., 2005, Wang et al., 2009).

- **Economics:** Schweitzer et al. (2009) recently emphasized on the critical need for an understanding of the complex networks underlying national economies on a systemic level. Also, all kinds of trade relations are subject to intense research, for instance in the work of Baskaran et al. (2010) or Fagiolo et al. (2009).

- **Information networks:** Besides the World Wide Web (Albert et al., 1999), the classical example for information networks are citations between academic

papers, first studied by Newman (2001). The structure of a citation network reflects the structure of the information present at the vertices, but contains social aspects, too.

- **Biology:** In order to understand a cell's functional organization, one has to understand the topology of gene regulatory networks, the interaction network of the resulting proteins as well as finally the metabolic network (Barabási and Oltvai, 2004). The construction of such networks from experiments and the interpretation of their properties is a major challenge in current biology. Neural networks are of similar importance towards understanding brain function (White et al., 1986). Further examples include food webs (Garlaschelli et al., 2003) or protein folding networks (Rao and Caflisch, 2004).

- **Technical networks:** The internet, power grids or delivery networks, but also properties of complex electronic circuits have been analyzed (Ferrer-i-Cancho et al., 2001, Watts and Strogatz, 1998). In software engineering, complex dependency structures between components arise which are analyzed by different authors including Blöchl (2007) and Myers (2003).

- **Traffic:** All kinds of traffic networks have been studied, e.g. roads on different scales, flight nets and shipping routes (Barabási, 2002). The network of human traveling has been analyzed by different methods, for example with the help of the movement of marked dollar bills in the USA (Brockmann and Theis, 2008, Brockmann et al., 2006).

- **Linguistics:** Language is built of a dictionary plus a set of rules (grammar). The first network approach to language by Ferrer-i-Cancho and Solé (2001) and its extension by Dorogovtsev and Mendes (2001) analyze co-occurrence and interaction of distinct words in sentences.

- **Physics:** Besides complex networks being of major interest to statistical physics by themselves, concrete examples from physics show interesting topologies: Scala et al. (2001) studied the conformation space of polymers, where links correspond to transitions. Other examples are the networks of free energy minima and saddle points in glasses (Newman, 2003) or atomic clusters (Doye and Massen, 2004).

One of the original, and still primary, motivation for studying complex networks is the understanding of dynamics taking place on graphs. For instance, one tries to find answers on how diseases, rumors, information, innovations or computer viruses spread over the corresponding networks (Malmgren et al., 2009, Wang et al., 2009).

However, let us bring this enumeration of examples to an end. The overwhelming literature is collected, among others, in the reviews by Barabási and Albert (2002), Newman (2003), or Boccaletti et al. (2006). Now, we instead focus on the various tools put forward for their analysis.

1.2.2 Basic definitions

Networks are objects consisting of dots representing a set of items, and lines between them indicating some relation. In the language of mathematics such objects are called *graphs*. In its formal definition, a graph G is a pair of sets (V, E), where V is non-empty, and E is a subset of $V \times V$ (Bollobas, 1998). The elements of V are called *vertices* or *nodes*, where elements of E are called *edges* or *links*. The *size* of the graph is the number of vertices it contains, in the following denoted by n. An edge (i, j) is called *self-loop* if $i = j$.

The vertex pairs in E can be either unordered or ordered. In the first case, we are talking of an *undirected* graph. Examples are collaboration graphs, protein interactions or train routes. In undirected graphs, two vertices i and j are *adjacent* if $(i, j) \in E$. The set of all neighbors of node i is its *neighborhood* $\mathcal{N}(i)$. In the second case, the graph is *directed*, as for instance are citation graphs, food webs or trade networks. For any vertex $i \in V$ of a directed graph we denote by $\mathcal{S}(i) := \{j|(i, j) \in E\}$ the set of *successors* or *outputs* of i, by $\mathcal{I}(i) := \{j|(j, i) \in E\}$ its *predecessors* or *inputs*.

The *degree* or *connectivity* k_i of vertex i in an undirected graph is the number of neighbors it has. In a directed network, we differentiate between the *i–degree* k^{in} counting the number of predecessors of a node, and the *out-degree* k^{out} determined by the number of its successors. The most basic topological characterization of an undirected graph can be obtained by its *degree distribution* $p(\mathsf{k})$, giving the probability that a node chosen at random has the degree k. Equivalently, we can define and easily calculate it as the fraction of nodes in the graph having degree k. In the case of directed networks one commonly considers two degree separate distributions, $p(\mathsf{k}^{in})$ and $p(\mathsf{k}^{out})$.

A sequence of incident edges of the form $(i_1, i_2), (i_2, i_3) \ldots (i_{m-1}, i_m)$ is called a *walk* from i_1 to i_m. The *length* ℓ of a walk is the number of edges it contains. A walk with pairwise distinct edges is a *trail*, and a trail, in which all vertices – possibly except the start and end node – are visited only once, is a *path*. A path with identical start and end nodes is a *loop* or *cycle*. A graph is called *acyclic* or *hierarchical* if it contains no cycles.

However, in a trade network, for instance, the amount of goods traded between different pairs of countries may be quite different. Likewise, speaking of social

networks, there are different levels of acquaintance. Hence, we will also analyze *weighted* graphs: a weighted graph $G(V, E, w)$ is a graph $G(V, E)$ together with real edge weights $w : E \to \mathbb{R}$.

An unweighted graph can be represented by an *adjacency matrix* $\mathbf{A}$ with elements

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if there is an edge from } i \text{ to } j \\ 0 & \text{otherwise} \end{cases} . \tag{1.20}$$

This matrix is of size $n \times n$. For undirected graphs, $\mathbf{A}$ is symmetric. In the case of weighted graphs, the adjacency matrix can be extended to a *weight matrix* by taking the weights of the edges as entries of $\mathbf{A}$. The diagonal of $\mathbf{A}$ contains the self-loops.

In *colored* graphs, vertices may have attached attributes that represent different categories. For example, a person in the social network can be assigned characteristics like gender, race, or nationality. Some graphs are naturally partitioned into different node types. Imagine an affiliation network, where people are joined to some group, say sports teams, families, or the companies they are working for. Here, we have two types of vertices (people and groups), with edges only between the two types and not within them. This is an example of a *bipartite graph*. Networks consisting of k different types of vertices with edges allowed only between different types are called *k-partite*.

One can also imagine situations, where edges connect more than two vertices. For instance, imagine proteins, where a natural grouping is by common membership in a protein complex. Such graphs are called *hypergraphs* and gain more and more importance over the last years (Klamt et al., 2009, Latapy et al., 2008). Each hypergraph defines an equivalent bipartite graph, where one node type corresponds to the hyperedges that are connected to the original nodes they group together.

In this thesis, many different kinds of graphs introduced will be used. Hence, in any of the following Chapters we will specify precisely what type we are currently focusing on. Note that standard notations from two different fields collide: $\mathbf{A}$ denotes both adjacency matrices and the mixing matrices in BSS problems. Hence, if there is danger of confusion we represent graphs as (weight matrices) $\mathbf{W}$ instead.

1.2.3 Properties of networks

This Section provides the basic tools for complex network analysis in the undirected and unweighted case. For a more detailed introduction into the field we refer to the reviews by Newman (2003), Boccaletti et al. (2006), or Barabási and Albert (2002) or the textbooks by Dorogovtsev and Mendes (2003) and Vega-Redondo (2007).

1.2.3.1 Shortest paths and the small world effect

One can define a geodesic distance d_{ij} between two vertices i and j of a graph by the length of the shortest path from i to j. In an undirected network, $d_{ij} = d_{ji}$, while in a directed graph these may be different. The calculation of shortest-path lengths is a classical problem in graph theory. There exist many algorithms that solve it, including Dijkstra's algorithm and breadth-first or depth-first search. In weighted graphs the Floyd-Warshall algorithm can be used.

We can then introduce the distribution of the shortest-path lengths and the *mean shortest-path length* or *characteristic path length* $\bar{d}$. With this definition every non-connected graph has a characteristic path length of infinity, hence one often restricts the average to the largest connected component. Alternatively, the harmonic mean of the geodesic path lengths called *efficiency* can be considered.

It turns out that in most complex networks the characteristic path length is quite small, even if the number of nodes is large. This fact is commonly referred to as the *small world effect*. The best known example is the "six degrees of separation" found by Milgram (1967). He showed that there is an average number of six acquaintances that links two arbitrary people from the United States. However, the small world effect is not an indication of an intricate underlying organizing principle. Erdös and Rényi (1959) demonstrated that the characteristic path length in random graphs scales logarithmically with the number of nodes: $\bar{d} \propto \log(n)$ (see Section 1.2.6.1).

1.2.3.2 Clustering and transitivity

A typical characteristic of social networks is that two persons with a common friend are likely to be friends, too. This property is reflected in a large number of triangles in the corresponding acquaintance graph. Such *clustering* or *transitivity* has been found to be an apparent local topological feature of most real-world complex networks (Amaral et al., 2000).

A quantitative measure for the "cliquishness" of the neighborhood of a node i is its *clustering coefficient* $c(i)$. If the node has $\mathsf{k}(i)$ neighbors, then at most $\mathsf{k}(i)\left[\mathsf{k}(i)-1\right]/2$ edges can be present between these nodes. Watts and Strogatz (1998) define the clustering coefficient of i as the ratio between the total number y of the edges connecting its nearest neighbors and this upper bound. This is equivalent to comparing the number of triangles connected to i with the number of connected triples centered at it:

$$c(i) = \frac{2y}{\mathsf{k}(i)\left[\mathsf{k}(i)-1\right]} = \frac{\text{number of triangles connected to } i}{\text{number of connected triples centered at } i}\,. \tag{1.21}$$

The clustering coefficient c of a graph is then defined as the average over all

vertices:

$$c = \frac{1}{n} \sum_{i \in \mathcal{V}} c(i) \,. \tag{1.22}$$

By definition, c and $c\,(i)$ are normalized to $[0, 1]$.

A more convenient measure for the clustering of a graph that has a long history in social sciences is *transitivity* (Barrat and Weigt, 2000). It is directly defined as the fraction of connected triples of nodes in the graph that also form a triangle:

$$T = \frac{3 \times \text{number of triangles in G}}{\text{number of connected triples of vertices}} \,. \tag{1.23}$$

The factor three arises from the fact that each triangle contributes to three connected triples, one centered on each of the three nodes. So it is ensured that $T \in [0, 1]$.

Note that there is a big difference between the transitivity and the clustering coefficient of a graph, since the order of calculating mean and ratio is different in Equations (1.22) and (1.23). Latora and Marchiori (2003) even give a construction scheme for a graph in which $c = 1$, but $T = 0$. Commonly, the clustering coefficient tends to weight contributions from low-degree vertices more strongly than it is the case for transitivity (Newman, 2003).

1.2.3.3 Degree correlations

An interesting question to ask is whether vertices with certain properties connect preferentially to similar ones, or to unlike ones. For the node degrees, the most basic property, this question can be answered by calculating *degree correlations*.

The degree distribution completely determines the statistical properties of *uncorrelated* networks, where the probability that a node of degree k is connected to another node of degree k' is independent of k (Boccaletti et al., 2006). In other words, the joint probability $p\,(\mathsf{k}, \mathsf{k}')$ that an edge connects vertices of degrees k and k' factorizes.

In a large number of real networks such degree correlations have been found, reflecting their intricate local topological structure. However, measuring the joint degree distribution gives very noisy results because of the finite size of the networks. To overcome this, Pastor-Satorras et al. (2001) introduce a more coarse, but less fluctuating measure. They calculate the mean degree

$$\bar{\mathsf{k}}_{nn}\,(\mathsf{k}) = \sum_{\mathsf{k}'} \mathsf{k}' p\,(\mathsf{k}' \mid \mathsf{k}) \tag{1.24}$$

of the nearest neighbors of all vertices with degree k. If there are no correlations present in the network, $\bar{\mathsf{k}}_{nn}\,(\mathsf{k})$ is independent of k. Newman (2002) classified correlated graphs as *assortative*, if $\bar{\mathsf{k}}_{nn}\,(\mathsf{k})$ is an increasing function of k, whereas they

are called *disassortative*, if it is a decreasing function of k. In other words, in assortative networks vertices of high degrees tend to have neighbors of high degrees, while in disassortative networks nodes with high degrees tend to have neighbors of low degrees.

Newman (2002) reduces the measurement of degree correlations still further. He simply calculates the Pearson correlation coefficient of the degrees at either ends of an edge. This gives a single number that is positive for assortatively mixed networks and negative for disassortative ones.

Almost all social networks have been found to be assortative, while other types of networks such as information networks, technological networks or biological networks are disassortative (Newman, 2002).

1.2.3.4 Motifs

A *motif* M is a pattern of interconnections that occurs at a significantly higher rate in a graph G than in randomized versions of the graph (Milo et al., 2002). By randomized versions we mean graphs with the same number of nodes and edges, and also the same degree distribution as the original one, but with all links distributed at random.

A motif is usually meant as a connected n-node graph which is a subgraph of G. An example is the triangle we used in the last Section to define the clustering coefficient. The search for significant motifs in a graph G is based on matching algorithms which count the total number of occurrences of each n-node subgraph in the original graph and in the randomized ones. The statistical significance of a motif M is thereby described by the Z-score:

$$Z(M) = \frac{|M|^G - |M|^{rand}}{\sigma_M^{rand}} .$$

Here $|M|^G$ denotes the number of occurrences of M in G, $|M|^{rand}$ and σ_M^{rand} are the mean and the standard deviation of its appearances in the randomized versions.

The concept of motifs was first introduced by Milo et al. (2002), who studied small n motifs in biological and other networks. They found the motifs shared by ecological food webs to be distinct from the motifs shared by transcription factor networks or the World Wide Web. Motifs may thus be used to define universal classes of networks. Moreover, they may uncover the basic building blocks of networks, having certain functions within the network.

In gene regulatory networks, for instance, Shen-Orr et al. (2002) identified motifs with particular switching functions in the system, such as gates, and feed-forward loops. Subsequently, it was the new field of Systems Biology that analyzed the

dynamics arising from these motifs by means of ordinary differential equations. Experiments on the dynamics generated by motifs in living cells indicate that they indeed have characteristic dynamical functions. Thus, in this context we could witness a successful transfer from an analysis of the network topology to the networked system's dynamics. Alon (2006) summarizes the typical motifs in biological networks and the information-processing functions carried out.

1.2.4 Vertex centrality measures

An important question that has been studied extensively by many researchers in the field is to determine the relative importance of a vertex within a graph. For example, one might want to determine who is the most influential person in a social network, or how critical a certain router in an Internet network is to the flow of traffic. The amount of importance of a particular vertex is conventionally referred to as its *centrality* (Freeman, 1977). Various ways of defining centrality have been developed, depending on the type of network, and the type of question one is interested in. Borgatti (2006) or Newman (2010) provide detailed reviews.

The currently used centrality concepts essentially arise from two quite different structural intuitions. First, one may view a node as important in a network to the extent that it is somehow *close* to all other nodes in the network. This view is motivated in the social sciences where high closeness of people to others may imply access to more information (Leavitt, 1951), or correspond to higher status (Katz, 1953) and more power (Bonacich, 1987). The second intuition grows out of the idea that a node's centrality is based on the degree it lies *between* other nodes. For instance, people central on the paths of communication can facilitate or inhibit the communication of others. Hence, they can mediate the access of others to information (Bavelas, 1948).

The simplest centrality measure indicating the closeness of a node to others is the *degree*, the number of edges incident on a vertex. Jeong et al. (2001) used the node degrees in a protein interaction network to identify proteins essential for surviving. The workhorse measure in social sciences is *closeness centrality*, first introduced by Sabidussi (1966). It was originally defined as the inverse of the mean shortest-path distance between a given vertex and all other vertices reachable from it. As the geodesic distance is symmetric, it also can be interpreted as the (inverse) average distance to reach a certain node starting from an arbitrary node in the graph. We adopt the second view and define

$$C_{close}(i) = \frac{n}{\sum_{t \in V} d_{ti}} . \tag{1.25}$$

Closeness can also be regarded as a measure of how long it will take information spreading in the network to arrive at a certain node.

Freeman (1977) defined the *betweenness centrality* of a node as the fraction of shortest paths between pairs of other nodes that pass through it. If there is more than one shortest path between a given pair of vertices, then each such path is given equal weight summing to unity. To be precise, let $\sigma_{st}(i)$ denote the number of geodesic paths from node s to t that pass through i, and σ_{st} the total number of geodesic paths between s and t. Then, the shortest-path betweenness centrality C_{sp} of node i is given by

$$C_{sp}(i) = \sum_{s \neq i \neq t \in V} \frac{\sigma_{st}(i)}{\sigma_{st}}. \tag{1.26}$$

A fast algorithm for calculating this measure has been developed by Brandes (2001).

However, commonly information (or anything else) does not travel along geodesic paths. Thus, a more realistic measure should also include contributions from non-geodesic paths. To this end, Freeman et al. (1991) introduced *flow betweenness*, which is based upon the maximum capacity of flows between nodes. Imagine the edges in a graph as pipes that can carry a unit flow of some fluid. Then, by making simultaneous use of different paths, the maximum possible flow between two vertices in this network will be more than a single unit. The flow betweenness of a vertex is defined as the amount of flow through it when the maximum possible flow is transmitted from source s to target t, averaged over all pairs of vertices. It can be calculated by extending the augmenting path algorithm (Ahuja et al., 1993).

These measures require flows in the network to know an ideal route from a source to a target, either in order to find a shortest path or to maximize flow. Addressing this potential deficiency, Newman (2005) developed *random walk betweenness*. A random walker starts at a certain vertex and repeatedly chooses an edge incident to the current position (Bollobas, 1998). These choices are made according to a probability distribution determined by the edge weights. We consider *absorbing* random walks, i.e. the walker has a prescribed goal he never leaves once it is reached. Let $\bar{N}^{st}(i)$ be the expected number of times node i is visited on an absorbing random walk from source s to target t. Newman's random walk betweenness of node i is then defined as the mean of this quantity, averaged over all pairs of vertices:

$$C_{Nrw}(i) = \frac{\sum_{s \in V} \sum_{t \in (V - \{s\})} \bar{N}^{st}(i)}{n(n-1)}. \tag{1.27}$$

Here, only *effective visits* are taken into account, i.e. if a walk passes through a vertex and then later passes back through it in the opposite direction, the two contributions cancel out. Additionally, if it is equally likely to pass through a vertex

in either direction, the two directions cancel when averaging over the realizations of a certain walk. Newman (2005) uses the natural analogy between random walks in graphs and electrical currents in resistor networks to calculate his measure from Kirchhoff's laws of current conservation.

Recently, Estrada et al. (2009) introduced *communicability betweenness* which takes into account all walks up to a certain length. Therefore, this measure interpolates between the extreme cases of shortest path and random walk betweenness. Latora and Marchiori (2007) introduced *delta centralities* as a unifying framework for the discussed measures.

1.2.5 Community detection

A key question in network analysis is how to detect and interpret the internal organization of networks. A possible answer may be a modular decomposition, which implies the coexistence of structural sub-units associated with more highly interconnected parts that are present on a scale between the scale of the whole network and the scale of the motifs. We regard the identification of these a priori unknown building blocks – such as for instance functional modules in protein-protein interaction (PPI) networks – as *community detection*. The communities, as well as their interconnections, are essential for understanding underlying functional properties.

A community is built of a set of nodes that are highly interconnected – in other words, it consists of a subset of nodes that is quite similar in a topological sense. Such a grouping of elements according to a similarity measure however was exactly our definition for a data cluster in the last Section. Hence, in the following we also refer to community detection as *(graph) clustering* to indicate its tight relation to classical learning. Consequently, community detection was the main application area of machine learning techniques in the complex network field over the last years. This Section presents the main classes of methods to solve the graph clustering task, developed either by adoption of various machine learning approaches, but also from analogies to statistical physics. It follows in parts the recent reviews by Fortunato (2010) and Porter et al. (2009).

1.2.5.1 Traditional methods

Community detection has a long tradition, and it has appeared in several forms in multiple disciplines. Probably the first example stems from Weiss and Jacobson (1955) who searched for work groups within a government agency. The mathematical formalization of community detection is *graph partitioning*, a classical problem in computer science for which algorithmic approaches were proposed already in the

1970's. It consists of partitioning a graph's vertices into subsets, such that these are of about the same size and there are minimal connections between them. Most variants of this problem are known to be NP-hard, however several heuristics perform well, giving solutions that are not necessarily optimal. Still widely used is, among others, the *Kernighan-Lin algorithm*, a greedy approach to graph bisection (Kernighan and Lin, 1970).

Spectral graph partitioning, see e.g. (von Luxburg, 2007), usually gives better global results than such heuristic approaches. It is commonly based on the properties of the spectrum of the graph Laplacian $\mathbf{L}$, which is defined as

$$\mathbf{L} = \mathbf{A} - \mathsf{K}\,, \tag{1.28}$$

where K is the diagonal matrix of node degrees. Any partition of a graph into two groups can be represented by an index vector $\mathbf{s}$ where $\mathbf{s}_i = \pm 1$ depending on the group membership of node i. The total edge weight between the two groups of nodes can then be written as

$$R = \frac{1}{4}\mathbf{s}^T\mathbf{L}\mathbf{s}\,.$$

The best partition of the graph will also minimize R. Now, if we write $\mathbf{s}$ in the basis given by the eigenvectors $\mathbf{v_i}$ of $\mathbf{L}$, say $\mathbf{s} = \sum_i a_i\mathbf{v_i}$, we obtain

$$R = \sum_i a_i^2\lambda_i\,.$$

Here, λ_i are the eigenvalues of $\mathbf{L}$. Note that $\mathbf{L}$ has at least one eigenvalue zero.

Finding the true global minimum of R is hard. However, the second-lowest eigenvector, say $\mathbf{v_2}$, may provide a good approximation, if the corresponding λ_2 is close to zero. Choosing $\mathbf{s}$ parallel to this direction, we find $R \sim \lambda_2$. Since the index vector contains only entries ± 1, it can obviously be not perfectly parallel. The best choice turns out to match the signs of the entries: set $\mathbf{s}_i = 1(-1)$ if $(\mathbf{v_2})_i > 0 (< 0)$. This procedure yields two partitions, from which one could iteratively extract a predefined number of communities. This iterative bi-sectioning however is not a reliable strategy for community detection.

The Laplacian is the common target for spectral clustering; for alternatives as well as for more elaborate spectral strategies we refer to (von Luxburg, 2007) and (Fortunato, 2010).

The natural connection between community detection and standard data clustering as outlined in Section 1.1.1 is a rich source for algorithmic solutions. One can employ the discussed techniques like hierarchical or k-means clustering to group nodes based on some *graph-theoretic similarity* measure. Common measures are:

- Various graph-theoretic node distance measures like shortest path or random-walk distance (Pons and Latapy, 2005, Rosvall and Bergstrom, 2008). Brockmann and Theis (2008) propose inverse fluxes as distance measure in weighted, directed networks.
- Neighborhood similarity indices, e.g. pairwise clustering-coefficients (Newman, 2003, Radicchi et al., 2004) or topological overlap measures (Maslov and Sneppen, 2002, Müller-Linow et al., 2008).
- Standard distance measures between the columns or rows of the adjacency matrix, like Euclidean or Manhattan distance (Burt, 1976), or Pearson's or Spearman's correlation (Boccaletti et al., 2006, Vega-Redondo, 2007).

These methods are easy to implement and show a fast performance, and extensions to the detection of overlapping clusters are straightforward. A further advantage is that one can use a similarity measure that is adjusted to the graph under consideration. Obviously, they share the pros and cons of the clustering technique employed.

1.2.5.2 From divisive algorithms to modularity

The first algorithm for community detection that emerged from within the complex network community was the algorithm developed by Girvan and Newman (2002). It is based on the *edge betweenness* of a link, which is defined as the number of shortest paths that runs through it. Clearly, links that lie between communities will have higher edge betweenness than other ones. Other strategies to quantify an edge's centrality can be derived from the various node centrality measures from the last Section. Edge centrality is then iteratively used as an indicator of where to divide the network, so that the communities become disconnected from each other. The general form of the algorithm is:

1. Calculate the centrality scores of all edges.
2. Identify the edge with the highest score and remove it from the graph.
3. Recalculate centralities of all remaining edges.
4. Repeat from step 2 until a prescribed number of communities is achieved.

This algorithm turned out to perform well in many applications (Lancichinetti and Fortunato, 2009). As nodes are removed, the algorithm breaks the network into components which then can be connected hierarchically by a dendrogram. As a variant, Holme et al. (2003) modify this algorithm and remove vertices instead of edges, based on a centrality measure chosen to identify boundary vertices.

As with all clustering algorithms, an issue is what is the optimal point to stop removing links. In order to answer this question, Newman and Girvan (2004) introduce the *modularity* Q, an index that measures how well a network breaks into some detected communities. They argue that networks are modular not when there are few edges between communities, but rather when the number of intra-community links are *larger than expected*. The goodness of a partitioning into communities can then be defined by

$$Q = (\text{fraction of intra-communities edges}) \; - \; (\text{expected fraction of such edges}) \, .$$

This expected fraction of edges depends on the chosen null model, which should keep some of the structural properties of the graph to be clustered but lacks community structure. Modularity can then be written as

$$Q = \frac{1}{2m} \sum_{i,j=1}^{n} \left(\mathbf{A}_{ij} - \mathbf{P}_{ij} \right) \delta_{C_i C_j} \, . \tag{1.29}$$

Here, $\mathbf{P}_{ij}$ is the expected number of edges between vertices i and j in the null model. With the Kronecker δ, only pairs of vertices that are in the same community give a contribution. The choice of the null model graph is in principle arbitrary, for instance one of the generative models discussed in the following Section can be used. However, usually one takes randomized versions of the graph of interest, as in the definition of motifs before.

The more the number of internal edges of the clusters exceeds the expected number, the better defined are the communities. Hence, a large value of Q indicates a good partition. If the nodes are assigned to communities at random, we get $Q = 0$. Note that the modularity is always smaller than one, and can be negative as well.

Among the results obtained by the Newman-Girvan algorithm, we can now pick those with the highest values of Q. However, this algorithm is very slow. A possible alternative is to directly take the modularity as a cost function for the community clustering and to try maximize it algorithmically. Indeed, this modularity optimization became a popular method for community detection over the last years. Newman (2004) was the first to realize this approach; he maximized Q on very large networks using a simple greedy optimization. Subsequently, other approaches employ simulated annealing, genetic algorithms, or spectral optimization, which allow manifold adjustments of time complexity and accuracy. Relevant papers are reviewed in Fortunato (2010).

Despite its elegance, modularity optimization is somewhat problematic: first, it has been shown that modularity optimization often fails to detect clusters smaller

than some scale, depending on the size of the network. Fortunato and Barthélemy (2007) analyze this well-defined resolution limit in detail. Second, the landscape of modularities has many local optima very close to the absolute maximum, which show quite different cluster structures. Hence, convergence to these local minima has to be avoided.

1.2.5.3 Detecting overlapping communities

Recently, Reichardt and Bornholdt (2006) have developed an elegant method based on a *Potts spin-glass model* associated to the graph. They find communities to coincide with the domains of equal spin value in the ground state of a modified m-state Potts spin-glass Hamiltonian:

$$H(\{\sigma\}) = -\sum_{ij} \mathbf{J}_{ij}\delta_{\sigma_i\sigma_j} .$$

Here, σ_i is the spin of particle i and $\{\sigma\}$ the configuration of all spins. The interaction energy between two spins is set to $-\mathbf{J}_{ij}$ if the spins are in the same state and 0 if they are not.

They assign a spin to each of the n nodes and set $m = n$. The interaction energy $-\mathbf{J}_{ij}$ is only added to H if i and j are in the same community. Then two adjacent nodes are set to interact ferromagnetically, i.e. $\mathbf{J}_{ij} > 0$, if the weight of the edge is greater than expected by a specific null model and interact antiferromagnetically ($\mathbf{J}_{ij} < 0$) when it is less than expected. Hence, two nodes want to be in the same community in the first case, while in the second case they tend to be in a different one. Otherwise, the spins do not interact at all. Now Reichardt and Bornholdt (2006) minimize H globally to find the ground state of the system. The spin-glass analogy thereby enables the application of many powerful optimization techniques from statistical physics.

Their Hamiltonian is essentially designed as a measure for the number of links and non-links both inside and between communities. However, with the choice of $\mathbf{J}_{ij} \propto \mathbf{A}_{ij} - \mathbf{P}_{ij}$ one obtains $H = -Q$ and recovers the modularity optimization. An interesting feature of the spin-glass approach is that by combining global and local extrema it allows to detect fuzzy communities, assigning each node a degree of membership to multiple communities (Reichardt and Bornholdt, 2004). Moreover, no prior knowledge of the number of communities has to be assumed.

In real-world complex networks, vertices are often part of multiple communities, hence the detection of overlapping clusters has become quite popular recently. Besides the algorithms of Reichardt and Bornholdt (2004), the *clique percolation method*

developed by Palla et al. (2005) is the most popular approach to such non-crisp community assignment.

This local algorithm is based on the idea that the internal edges of a community are likely to form k-cliques (fully connected subgraphs of k nodes), a consequence of the high edge density within such a cluster. On the other hand, it should be unlikely that edges between different communities form cliques. Then, if k-cliques were able to move around the graph, in some way, they usually would be trapped inside their original community, which is hard to leave through the bottleneck of few outgoing edges. In other words, they percolate in the communities. Palla et al. (2005) call two k-cliques adjacent if they share $k-1$ nodes. They define a k-community as the maximal union of k-cliques that can be reached from each other through a series of adjacent k-cliques. Then, single nodes can belong to several k-communities. In order to find the k-clique communities, in practice they locate maximal cliques rather than the individual k-cliques. Detecting maximal cliques is computationally demanding, however, Palla and coworkers found that for the real networks analyzed the procedure is quite fast.

1.2.6 Generative models

On a fundamental level, the most important distinction between different types of networks is whether they are structured or random. That is, if a network exhibits some kind of regularity, it is called structured; if its topology has evolved through a process of uncoordinated actions, it is referred to as a random network.

In a structured network such as a crystal lattice each node has the same number of neighbors, forming a tightly connected local pattern. Hence, such networks are highly clustered, but compared to random networks, they have large average path length. While structured networks are useful for analyzing homogeneous systems – for instance in solid-state physics – they are inappropriate for understanding heterogeneous and complex phenomena such as social or biological interactions.

1.2.6.1 Erdös-Rényi random graphs

The classical mathematical model for the generation of random networks has been formulated by Erdös and Rényi (1959). In their model, a network is constructed by connecting nodes completely at random: They start with a set of n nodes and connect each pair of nodes with a fixed probability p_{ER}. The presence or absence of any two distinct edges is independent of the connectivity of other nodes.

This construction leads to a mean connectivity of $\bar{\mathsf{k}} = p_{ER}(n-1) \approx p_{ER}n$. The structure of ER random graphs strongly depends on the value of p_{ER}, showing a

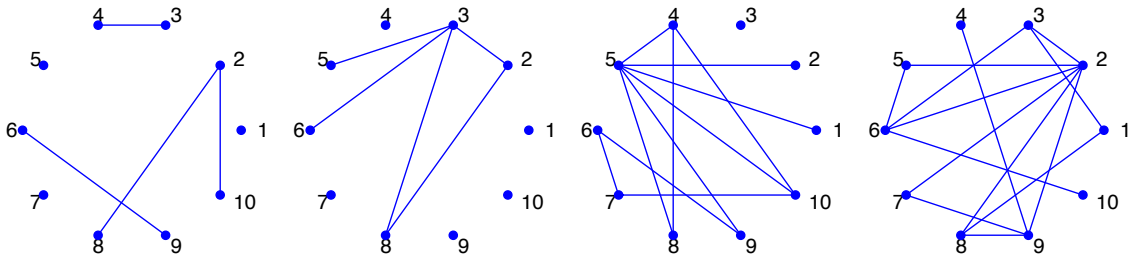


Figure 1.1: Realizations of ER random graphs with $p_{ER} = 0.09,\ 0.11,\ 0.21,\ 0.25$. The graphs contain ten nodes, corresponding to $p_c = 0.1$. Note the emergence of a giant component for $p_{ER} > p_c$. Above $p_{ER} > 0.1 \log 10 \sim 0.23$, the graph is strongly connected.

dramatic change at a critical probability $p_c = \frac{1}{n}$, which corresponds to a mean degree of $\bar{\mathsf{k}} = 1$. If one considers the size of the largest component as order parameter, the transition at p_c shows the typical features of a second order phase transition, falling in the same universality class as that of the mean-field percolation transition (Boccaletti et al., 2006). Erdös and Rényi (1959) proved that:

- If $p_{ER} < p_c$, then almost surely the graph has no component of size greater than $\mathcal{O}(\ln n)$, and no component has more than one cycle. So the resulting graphs are locally tree-like.
- If $p_{ER} = p_c$, then almost surely the largest component has size $\mathcal{O}\left(n^{2/3}\right)$.
- If $p_{ER} > p_c$, then almost surely the graph has a giant component of size $\mathcal{O}(n)$, containing a number $\mathcal{O}(n)$ of cycles. The other components are smaller than $\mathcal{O}(\log n)$ and tree-like.
- If $p_{ER} > p_c \log n$, the graph is almost surely strongly connected, i.e. for any pair of nodes there exists a path connecting them.

The ER model in the different regimes is illustrated in Figure 1.1.

Node degrees in an Erdös-Rényi (ER) graph are binomially distributed, i.e. the probability that a node has k neighbors is given by

$$p(\mathsf{k}) = \binom{n-1}{\mathsf{k}} p_{ER}^{\mathsf{k}} (1 - p_{ER})^{n-1-\mathsf{k}} \ . \tag{1.30}$$

This can be easily understood: p_{ER}^{k} gives the probability for the existence of k edges, $(1 - p_{ER})^{n-1-\mathsf{k}}$ the probability for the absence of the remaining $n-1-\mathsf{k}$ edges. The factor $\binom{n-1}{\mathsf{k}}$ accounts for the possible number of ways to select the endpoints of the k edges. For large n, the binomial distribution is well approximated by a

Poisson distribution

$$p(\mathsf{k}) \approx \mathrm{e}^{-\bar{\mathsf{k}}} \frac{\bar{\mathsf{k}}^{\mathsf{k}}}{\mathsf{k}!} \,. \tag{1.31}$$

The mathematical properties of the Poisson distribution imply that most of the nodes in such networks exhibit the same characteristic number of connections. Thus, despite the fact that the emergence of edges is random, a typical ER graph tends to be rather homogeneous, the majority of nodes having similar connectivity. Path lengths between nodes are small: Erdös and Rényi (1959) show that the characteristic path length $\bar{d}$ scales only logarithmically with the number of nodes, i.e.

$$\bar{d} \propto \log(n) \,.$$

However, with the edges being present independently, random networks completely lack a local topological structure and degree correlations. Considering a node and its nearest neighbors, the probability that two of these neighbors are connected equals the probability that two randomly chosen vertices are connected. Hence, we expect a clustering coefficient $c = p_{ER} = \bar{\mathsf{k}}/n$. This implies that the random graphs have vanishing clustering coefficient in the limit of large network size.

In conclusion, the ER model describes very few real-world networks adequately. However, as Newman (2003) points out, most of our basic intuition about the way networks behave arose from the study of Erdös-Rényi graphs.

1.2.6.2 Watts-Strogatz small-world model

A more realistic model of network evolution has been proposed in a seminal paper by Watts and Strogatz (1998). Their model leads to *small-world networks* which can be understood as an interpolation between random and structured networks.

The Watts-Strogatz (WS) or small-world network model is based on an edge rewiring procedure: initially, they start with a regular ring lattice of n nodes, where each node is connected to its first κ neighbors. Then they randomly rewire each edge of the lattice with a fixed probability p_{WS} such that self-loops and multiple edges are excluded. This process introduces in expectation $p_{WS} \cdot n \cdot \kappa$ long-range connections that connect nodes that were initially far apart. By varying p_{WS}, they can control the degree of randomness between zero (completely ordered) and one (completely random). This generative model is illustrated in Figure 1.2.

The degree distribution in WS graphs has been calculated first by Barrat and Weigt (2000). For $p_{WS} = 0$, this distribution is a delta function centered at the mean connectivity $\bar{\mathsf{k}} = \kappa$. A non-zero p_{WS} introduces disorder in the lattice. Then, the shape of the degree distribution looks similar to that of random graphs: It has a pronounced peak at κ degrees and decays exponentially for large degrees.

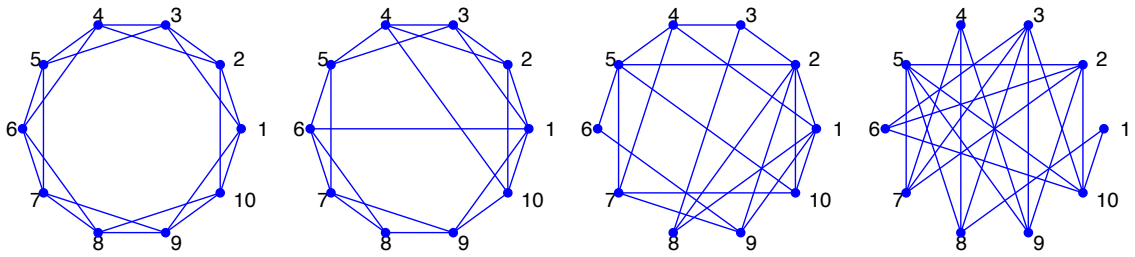


Figure 1.2: Realizations of WS graphs with 10 nodes, $\kappa = 2$ and rewiring probabilities $p_{WS} = 0,\ 0.25,\ 0.5,\ 1$.

The WS model was the first network model that is able to reproduce two key properties regularly observed in real-world networks: it preserves the local neighborhood (high clustering coefficient), and incorporates the small-world effect.

The WS model was of seminal importance for development of complex network science, but as there are no real-world graphs with sharply peaked degree distributions, we now proceed to the type of graphs that is the dominating one in current research as well as in nature: the scale-free networks.

1.2.6.3 Scale-free networks and the Barabási-Albert model

In lack of real-world data and computational resources, most research in the field of networks has been traditionally devoted to single-scale networks because researches believed that networks should display this property. In a groundbreaking article, Albert et al. (1999) analyze the topological structure of the World Wide Web. Their findings lead them to introduce the concept of scale-free networks.

The degree distribution of scale-free networks follows a power law, that is

$$p(\mathsf{k}) \sim \mathsf{k}^{-\alpha}\,.$$

Scale-free networks derive their name from the fact that power laws do not exhibit a "characteristic" connectivity and they do not have a peak value around which the distribution is centered. If the degree distribution follows a power law, there exist a significant number of nodes with a large number of connections. In fact, this is intuitive: for example, most of the web sites on the Internet have only a few outgoing and incoming links. However, a small number of sites, such as Google or Yahoo, act as *hubs* and tend to be extremely well connected.

Clearly, modeling a process leading to a scale-free network has to differ in important aspects from the previous discussed approaches. Barabási et al. (1999) identify two mechanisms that are responsible for the emergence of power-law distributions:

preferential attachment and *growth*. With this observation, the structure and the evolution of a network become inseparable. They define a generative model (BA) in two steps:

1. *Growth*: Starting with a small number m_0 of vertices, at every time step a new vertex with $m \leq m_0$ edges is added to the graph.

2. *Preferential attachment*: When choosing the vertices to which the new edges are connected, the probability $\Pi(i)$ that the new vertex will be connected to node i is proportional to the degree $\mathsf{k}(i)$ of that vertex:

$$\Pi(i) = \frac{\mathsf{k}(i)}{\sum_{j=1}^{n} \mathsf{k}(j)} \,. \tag{1.32}$$

After t periods the network has $n = t + m_0$ nodes and mt edges and it can be shown that for $t \to \infty$ the degree distribution follows a power law with an exponent of 3 (Barabási et al., 1999).

For the further properties of BA networks analytical results are difficult to obtain, hence we rely on numerical simulations. One finds that the characteristic path length in the BA model is even smaller than in ER networks with the same number of nodes and edges. Unfortunately, the clustering coefficient vanishes as $c \propto n^{-0.75}$ and with this the BA model fails in modeling highly clustered graphs. However, the discovery of scale-free networks and their modeling by the BA model have helped to catalyze the emergence of complex network science as a new field of physics.

There is a huge variety of variations of the BA growth mechanism, yielding flexible values of the power-law exponent and a reinforcement of the clustering property. A detailed overview and a collection of references can be found in (Boccaletti et al., 2006), where also further recent generative models are reviewed. These, for instance, incorporate node fitness concepts, or are based on vertex copying strategies or spatial embeddings. Also weighted and directed approaches are discussed.

Generalizations to hypergraphs or bipartite graphs are studied by many authors including Guillaume and Latapy (2004) or Mashaghi et al. (2004). Recently, Ghoshal et al. (2009) proposed a model to understand *folksonomies*. These are tripartite structures of users, resources, and tag-labels that are collaboratively applied by the users of otherwise undifferentiated databases, as e.g. Flickr.

We will not explicitly refer to these generative models in the rest of this work, however it is important to keep in mind that there is a tight relation between structure and evolution of a given network. We saw in this Section that a trend in the field goes from simple binary graphs to more complex objects like weighted, directed and recently also colored graphs.

2 Vertex centralities in input-output networks reveal the structure of modern economy

Within a few weeks of the onset of the financial crisis in 2008, the world economy had plunged into a severe global recession. The volume of international trade contracted sharply, and the world economy did not grow in 2009 for the first time since World War II. Many governments reacted with programs to mitigate the effects of the global downturn on their local economies. The United States spent $3 billion on the Car Allowance Rebate System (CARS). Germany spent an even larger fraction of its national economy for a car scrappage program. What effect did these programs have? How did the supply of new cars work its way through the rest of the local economy?

Input-output analysis was designed to explore this kind of effect (Miller and Blair, 2009, ten Raa, 2006). An *input-output table* is the matrix of the sales of goods and services between the different business sectors of an economy. A sector is a fairly coarse level of aggregation; an industry is composed of many firms making an identical product, and a sector is composed of several industries making similar products. "Agriculture" and "Pharmaceuticals" are two typical sectors.

The techniques of input-output analysis have had ready applications in economic planning. It is alleged that Leontief developed aspects of input-output analysis during the Second World War partly as an attempt to help identify strategic weaknesses in the German economy (Leontief, 1986). Ranking the influences of single sectors on national economic activity allows the identification of "key" sectors. For example, there has been much discussion about firms that are "too big to fail", and there was an implicit understanding that the bail-out of General Motors was necessary because of the importance of the automotive sector in the American economy.

2.1 Problem formulation

In order to formalize such intuitive aspects, a deeper understanding of the structures of national economies seems to be warranted. Any national economy is a complex

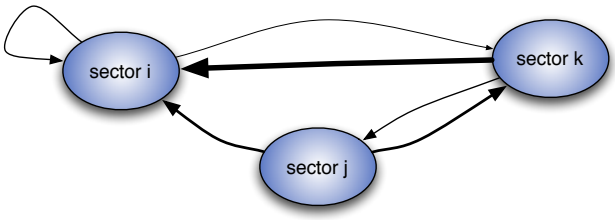


Figure 2.1: Schematic picture of an input-output network: Each sector corresponds to a vertex, and the flow of economic activity from one sector to another constitutes a weighted directed edge.

system in which many agents of different sizes interact by buying and selling goods and services. Schweitzer et al. (2009) suggest that an understanding of these interactions on a systemic level may be achieved by analyzing the underlying complex networks. We have already discussed in the Preliminaries that network analysis has been applied successfully to many problems from physics, biology and the social sciences during the last decade.

The literature on economic networks is growing rapidly. Several authors have studied international trade networks. The early work used binary approaches (Garlaschelli and Loffredo, 2005, Serrano and Boguñá, 2003), but it soon became evident that these graphs ought to be analyzed as weighted networks (Bhattacharya et al., 2008, Fagiolo et al., 2009). Interpreting the gross domestic product (GDP) as a country's fitness, Garlaschelli and Loffredo (2004) proposed a model to reproduce the network topology of bilateral trade links. A gravity model of trade has been used to understand weighted trade networks (Bhattacharya et al., 2008). While these approaches did not look at trade on a disaggregated level, we used an extended gravity model that incorporates properties of product-specific trade networks to test the Heckscher-Ohlin hypothesis in (Baskaran et al., 2010). The statistical properties of the network of world investment have been analyzed by Song et al. (2009). There are many other economic networks, such as the system of exchange rate arrangements (Li et al., 2004), the kinds of products made by different nations (Hidalgo et al., 2007), or connections between banks (Iori et al., 2008). Using network analysis and classical economic modeling, Grassi (2010) studied the information flow across board members of different firms; she focused mainly on node centralities.

In fact, it is natural to interpret an input-output table as a network, see Figure 2.1 for an illustration. Each sector corresponds to a vertex, and the flow of economic activity from one sector to another constitutes a weighted directed edge. In the language of complex network theory, identifying "key" sectors and ranking the sectors' roles in an economy is the task of applying an appropriate measure of node

centrality to this input-output graph.

Vertex centrality measures have been studied extensively for quite some time, compare our review in Chapter 1.2.4. Three properties of input-output graphs however make it hard to apply the discussed currently used centrality measures. First, at the usual level of aggregation, these networks are dense, typically almost completely connected. Thus applying measures based on shortest paths makes little sense. As the topology is nearly trivial, one needs to analyze edge weights. Second, they are directed; for example, in the United States in 2000, \$13.5 billion of rubber and plastic products were used in the production of motor vehicles, but only \$53 million of the output of the motor vehicle sector was used in the production of rubber and plastic products. Third, self-loops play a central role; in the same case, more than 60% of the total output of the cars sector was used as its own input. Some authors including White and Borgatti (1994) have extended centrality concepts to the directed case, but, to the best of our knowledge, no one until now has examined node centralities that incorporate self-loops. We derive two measures that are suited for such networks. Both rely on random walks and each has an economic interpretation.

The rest of this Chapter is structured as follows. The next Section provides the basic concepts. The third Section derives two centrality measures and shows their relation to economic theory. We contrast our two approaches using a small example. The forth Section shows our empirical results using input-output data from a wide range of countries. The proposed measures reveal important aspects of different national economies. Moreover, the consistency of the data allows us to develop a strategy to compare the centralities of the nodes across countries: We structure the whole data set by performing hierarchical clusterings of countries based on the similarity of their sectors' centralities. The obtained results are intuitive. Finally, we present some brief conclusions and suggestions for future research.

2.2 Basic definitions

Let $G = (V, E)$ be a graph with a set of vertices V and edges $E \subset V \times V$. In this Chapter, each edge $(i, j) \in E$ is directed and assigned a non-negative real weight a_{ij}. Moreover, for now the graph may contain self-loops. The number of vertices is n. We consider strongly connected graphs only; for any pair of nodes, there exists a directed path connecting them.

The graph is represented by its $n \times n$ adjacency matrix $\mathbf{A} = (a_{ij})$. Note that the element (i, j) represents the weight a_{ij} of the edge from node i to node j. To keep notation simple, we name the vertices by natural numbers, and we can identify them

with according indices in the adjacency matrix. Missing edges correspond to zero weights in the adjacency matrix. Then, the out-degree of node i, i.e. the total edge weight going out from it, is $\mathsf{k}_i^{out} = \sum_{j=1}^{n} a_{ij}$.

2.2.1 Input-output networks

We interpret the input-output table $\mathbf{A}$ directly as an adjacency matrix of a network whose vertices are the sectors of an economy. Its weighted directed edges quantify the flow of economic activity between sectors, as illustrated in Figure 2.1.

We focus on the input-output table of *intermediate inputs*. It records only sales of goods and services by firms to other firms that are directly consumed or used up as inputs in the production process. It is not a closed system; the row and column sums are not equal. In national accounts, the total value of the gross output of a sector also includes sales for final demand: consumption, investment, government purchases, and net exports. The total value of gross inputs into a sector also includes payments to the factors of production: gross operating surplus, compensation to employees, and indirect business taxes (ten Raa, 2006).

2.2.2 Random walks

The movement of goods between the sectors of an economy is best modeled as a random walk (Borgatti, 2005). Recall that in graph theory, a random walker starts out at a given position and repeatedly chooses an edge incident to the current position (Bollobás, 2001). These choices are made according to a probability distribution determined by the edge weights. The random walker proceeds for an arbitrarily long time or until a prescribed goal is reached.

An input-output table keeps track of the goods circulating through an economy, consisting of the outputs of a large number of firms in each sector. Hence, each entry is the statistical aggregation of many individual sales. We are interested in the transition probabilities of outputs produced by a sector. These can be obtained by normalizing the input-output matrix by its row sums. Hence in the following we work with the transition matrix

$$\mathbf{M} = \mathsf{K}^{out\,-1}\mathbf{A}, \tag{2.1}$$

where K^{out} is the diagonal matrix of the out-degrees k_i^{out}. Note that while $\mathbf{A}$ is not a closed system, $\mathbf{M}$ is.

2.3 Two measures of vertex centrality

This Section derives two centrality measures that are suited for weighted directed networks with self-loops. First, we explain the economic intuition behind their definitions. Following their derivations, we also relate the measures to other commonly used ones and also give a small example that contrasts them.

2.3.1 Economic intuition

Following ideas of Fischer Black (1987), we design both our centrality measures to quantify the response of sectors to an *economic shock*. Such a shock is a change in an exogenous variable that has repercussions on the endogenous variables under analysis (ten Raa, 2006). In input-output accounts, prices, technologies, firms, the distribution of profits, government policy, and the vector of final demands are exogenous, and the flows of commodities and corresponding payments between sectors are endogenous. Fischer Black (1987) hypothesized that the business cycle might arise because of the propagation of such shocks between the sectors of an economy. Long and Plosser (1983) developed an elegant analysis of the United States economy based on this idea.

We trace supply shocks as they flow as intermediate inputs through the business sectors of an economy. Their random journeys end at the sector from which the extra output eventually satisfies final demand, which we interpret as the target of some random walk. Consider an extra dollar of production in the car sector – perhaps as a result of a government program – and the target "Food products". The initial output will be sold randomly to another sector, according to the pattern of sales in the input-output table. The original dollar of extra revenue will be paid to capital, labor, or indirect business taxes in "Motor vehicles". The supply shock becomes an input into some sector, and it will increase economic activity there by one dollar, akin to the conservation of current in a circuit. The new output again will be sold to some sector. Eventually this process will hit the target "Food products", where the extra dollar of output exits the system to satisfy final demand. Averaging over all initial shocks or over all pairs of shocks and targets, we define a node's centrality by how quickly or how frequently it is visited during this process.

Every economic transaction consists of a real and a monetary counterpart; thus when keeping track of the flow of goods and services from a source to a destination at the same time we monitor the flow of a dollar in payments from the destination back to the source.

2.3.2 Random walk centrality

Closeness centrality as defined in Equation (1.25) is widely used in social network analysis (Freeman, 1979). We defined it as the inverse of the mean geodesic distance from all nodes to a given one. However, shortest paths make little sense in densely connected networks like input-output graphs. Moreover, they completely ignore self-loops.

In order to generalize the concept of closeness, distance between nodes has to be measured in a different way. We propose using the mean first passage time (MFPT). This distance is the measure of choice when dealing with random walk processes (Bollobás, 2001). The MFPT $H(s,t)$ from node s to t is the expected number of steps a random walker who starts at s needs to reach t for the first time:

$$H(s,t) := \sum_{r=1}^{\infty} r \cdot p(s \xrightarrow{r} t) \; . \tag{2.2}$$

Here $p(s \xrightarrow{r} t)$ is the probability that it takes a random walker exactly r steps before its first arrival at t. Note that $H(t,t) = 0$ since $p(t \xrightarrow{r} t) = 0$ for $r \geq 1$. The MFPT is not symmetric, even for undirected graphs. This property reflects the fact that it is much more probable to travel from the periphery to the central nodes of a graph than to go the other way around.

We are interested in the first visit of the target node t. For calculations we can consider an absorbing random walk that by definition never leaves node t once it is reached. It is thus appropriate to modify the transition matrix $\mathbf{M}$ by deleting its $t-$th row and column. This $(n-1) \times (n-1)$ matrix we denote by $\mathbf{M}_{-t}$. To keep notation simple, we denote the entries in $\mathbf{M}_{-t}$ with the indices of their original positions in $\mathbf{M}$. Hence, the row and column indices of $\mathbf{M}_{-t}$ are not $i, j \in \{1, \ldots, n-1\}$, but $i, j = \{1, \ldots, t-1, t+1, \ldots, n\}$.

The element (s,i) of the matrix $(\mathbf{M}_{-t})^{r-1}$ gives the probability of starting at s and being at i in $r-1$ steps, without ever having passed through the target node t. Consider a walk of exactly r steps from s that first arrives at t. Its probability is

$$p(s \xrightarrow{r} t) = \sum_{i \neq t} ((\mathbf{M}_{-t})^{r-1})_{si} m_{it} \; .$$

Plugging this into Equation (2.2), we find

$$H(s,t) = \sum_{r=1}^{\infty} r \sum_{i \neq t} ((\mathbf{M}_{-t})^{r-1})_{si} m_{it} \; .$$

The infinite sum over r is essentially the sum of the geometric series for matrices

$$\sum_{r=1}^{\infty} r(\mathbf{M}_{-t})^{r-1} = (\mathbf{I} - \mathbf{M}_{-t})^{-2} \ , \tag{2.3}$$

where $\mathbf{I}$ is the $n-1$ dimensional identity matrix. Making this inversion is the reason for having deleted one row and column from the original transition matrix $\mathbf{M}$. Lovász (1993) shows that $(\mathbf{I} - \mathbf{M}_{-t})$ is invertible as long as there are no absorbing states, whereas $(\mathbf{I} - \mathbf{M})$ is not. So

$$H(s,t) = \sum_{i \neq t} \left((\mathbf{I} - \mathbf{M}_{-t})^{-2} \right)_{si} m_{it} \ .$$

For fast calculation, this can be easily vectorized as $\mathbf{H}(.,t) = (\mathbf{I} - \mathbf{M}_{-t})^{-2} \mathbf{m}_{-t}$. Here $\mathbf{H}(.,t)$ is the vector of mean first passage times for a walk that ends at target t and $\mathbf{m}_{-t} = (m_{1t}, ..., m_{t-1,t}, \ m_{t+1,t}, ..., m_{nt})'$ is the $t-th$ column of $\mathbf{M}$ with the element m_{tt} deleted. Further, let $\mathbf{e}$ be an $n-1$ dimensional vector of ones. Then $\mathbf{m}_{-t} = (\mathbf{I} - \mathbf{M}_{-t})\mathbf{e}$, and we finally obtain

$$\mathbf{H}(.,t) = (\mathbf{I} - \mathbf{M}_{-t})^{-1} \mathbf{e} \ . \tag{2.4}$$

This equation allows calculation of the MFPT matrix row-by-row with basic matrix operations only. Using the Sherman-Morrison formula (Golub and Van Loan, 1996), we can speed up the n matrix inversions further.

Using the natural analogy with closeness centrality, we define *random walk centrality* as the inverse of the average mean first passage time to a given node:

$$C_{rw}(i) = \frac{n}{\sum_{j \in V} H(j,i)} \ . \tag{2.5}$$

This measure is similar to the one mentioned by Noh and Rieger (2004). Random walk centrality incorporates self-loops only indirectly as they slow down the traffic between other nodes.

The economic interpretation of this measure is straightforward. Consider a supply shock that occurs with equal probability in any sector. Then a high random walk centrality of a sector means that it is very sensitive to supply conditions anywhere in the economy. Hence, if one could predict sectoral shocks accurately, one would go short equity in a central sector and long equity in a remote sector during an economic downturn.

2.3.3 Counting betweeness

Our second approach is inspired by Newman (2005)'s random walk betweenness. We modify his concept and generalize it to directed networks with self-loops. The

proposed measure denoted as counting betweenness keeps track of how often a given node is visited on first-passage walks, averaged over all source-target pairs.

For source node s and target $t \neq s$, the probability of being at node $i \neq t$ after r steps is $((\mathbf{M}_{-t})^r)_{si}$. Then, the probability of going from i to j is m_{ij}. So the probability that a walker uses the edge (i,j) immediately after r steps is $((\mathbf{M}_{-t})^r)_{si} m_{ij}$. Summing over r, we can calculate how often the walker is expected to use this edge:

$$\begin{aligned} N_{ij}^{st} : &= \sum_r \left((\mathbf{M}_{-t})^r\right)_{si} m_{ij} = m_{ij} \sum_r ((\mathbf{M}_{-t})^r)_{si} \\ &= m_{ij} \left((\mathbf{I} - \mathbf{M}_{-t})^{-1}\right)_{si} . \end{aligned}$$

Notice that a walker never uses an edge (i,j) if j is not a neighbor of i since the according transition probability is zero. The total number of times we go from i to j and back to i is $N_{ij}^{st} + N_{ji}^{st}$. Here we differ from Newman (2005), who excludes walks that oscillate and thus counts only the net number of visits. On any walk from s to t, we enter node $i \neq s,t$ as often as we leave it. Hence, on a path from s to t, vertex i is visited $\sum_{j \neq t}(N_{ij}^{st} + N_{ji}^{st})/2$ times. For source s, target t and vertex $i \neq s,t$, we define:

$$N^{st}(i) = \sum_{j \neq t} (N_{ij}^{st} + N_{ji}^{st})/2 . \tag{2.6}$$

We allow for self-loops, hence a random walker may follow the edge (i,i), in which case the vertex i is visited twice consecutively. Since it is possible that $i = j \neq t$, we have to divide by 2 in all cases.

There are two special cases. If $i = s$, then the walker visits node s one extra time when it starts

$$N^{st}(s) = \sum_{j \neq t} (N_{sj}^{st} + N_{js}^{st})/2 + 1.$$

Also, if $i = t$, then the walker is absorbed by vertex t the first time it arrives there and

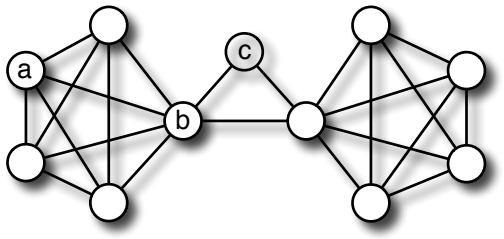
$$N^{st}(t) = 1. \tag{2.7}$$

We define the *counting betweenness* of node i as the average of this quantity across all source-target pairs:

$$C_c(i) = \frac{\sum_{s \in V} \sum_{t \in (V - \{s\})} N^{st}(i)}{n(n-1)} . \tag{2.8}$$

Counting betweenness can be used as a micro-foundation for the *velocity of money* (Mishkin, 2007), which associates the amount of economic activity with a given money supply. Consider a dollar of final demand that is spent with equal probability on the output of any sector, and assume that all transactions must be paid for with

A



B

	a	b	c
Shortest-path betweenness C_{sp}	0.2	0.64	0.2
Newman's betweenness C_{Nrw}	0.27	0.67	0.33
Random walk centrality C_{rw}	0.048	0.094	0.044
Counting betweenness C_c	1.93	2.80	1.03

Figure 2.2: The network in (A) is taken from Newman (2005). (B) contrasts different centrality measures calculated for selected nodes. Even though 'c' is topologically central, our measures do not rank it highly, in contrast to Newman's betweenness. Instead, they focus on how quickly or how frequently traffic within the network reaches a node. In a graph with two completely connected subcomponents, a slightly remote bridge-like node is not crossed over frequently.

cash, not credit. Then the counting betweenness of sector i is the expected number of periods that this dollar will spend there. If it is a high number, then that sector requires many transactions before the money is eventually returned to the household sector as a payment to some factor of production. If each transaction takes a fixed amount of time, then a sector with a high counting betweenness is a drag on the velocity of money in the economy.

2.3.4 Illustrative examples

Before applying our measures to actual data, we demonstrate their behavior in small artificial examples. Figure 2.2(A) shows a graph introduced by Newman (2005) to illustrate different centrality concepts. Here, all useful measures should obviously rank nodes of type 'b' most central. While concepts based on shortest paths, like the betweenness centrality from Equation (1.26), do not account for the topologically central position of node 'c', Newman's betweenness gives a high centrality to 'c'. In contrast, our measures both rank nodes of type 'a' higher than node 'c'. A random walker spends a lot of time within the fully connected subgraph on the left and seldom crosses over the bridge-like node 'c'. The former is why counting betweenness ranks node 'a' highly, the latter is why random walk centrality gives it

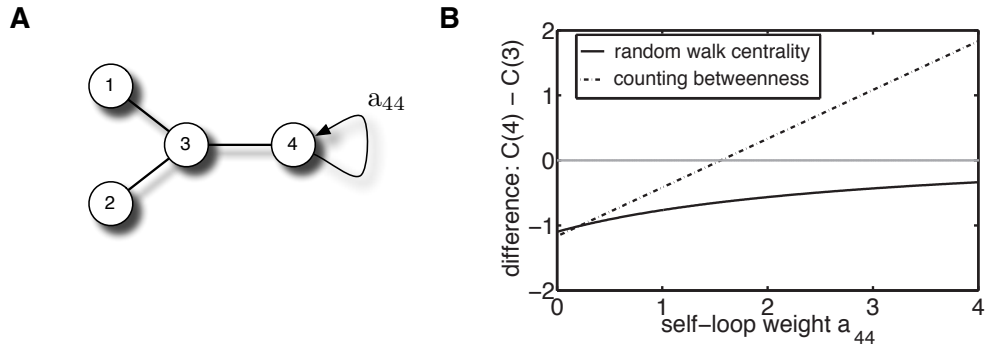


Figure 2.3: (A) This small network illustrates the importance of a self-loop. (B) shows the difference between the centrality of node 4 and 3 as a function of the self-loop weight a_{44}. All other links have unit weight. Random walk centrality always ranks node 3 highest. Counting betweenness ranks node 4 higher when a_{44} exceeds a threshold near 1.6. If the self-loop has a large weight, it takes a long time before a random walk leaves node 4 and enters the rest of the network.

a high ranking.

Figure 2.3(A) shows a small network that illustrates the difference between our two measures. It emphasizes the role of a self-loop. Depending on the self-loop weight a_{44} attached to node 4, either node 3 or 4 has the highest counting betweenness. In contrast, random walk centrality ranks node 3 highest, no matter the value of a_{44} is. Counting betweenness strongly emphasizes on the importance of self-loops which are considered only indirectly by random walk centrality.

2.4 The central sectors of modern economies

Our data are the input-output accounts from the STAN database at the Organization for Economic Co-operation and Development (OECD), which are freely available at `http://www.oecd.org/sti/inputoutput/`. They consist of 47 sectors and are benchmarked for 37 countries near the year 2000 (see Table 2.1 for a list). Each country's input-output table is taken as one input-output graph. The analyzed countries account for more than 85% of world GDP.

The used data are consistent on three important dimensions. First, they are designed to be consistent across countries. Second, they are consistent with macro-economic accounts; indeed, they maintain the national income accounting identities. Third, they are consistent across time; so we can compare Germany and the United States against themselves in two different benchmark years. The input-output accounts are reported in local currencies, but we have no need to use exchange rates

or GDP deflators because we are only considering the unit-free transition matrices.

Some countries have sectors with no input or output. These arise because of data limitations in the local national accounts. The most serious case is the Russian Federation, where the OECD records output in only 22 sectors. Such sectors hinder the matrix inversion in Equation (2.3). We therefore assign zero centrality to these nodes and remove them from the adjacency matrix.

2.4.1 Results for individual countries

Table 2.1 presents each country's most central sector with respect to our two measures. The complete results are available at `http://hmgu.de/cmb/ionetworks`. It is striking that "Wholesale and retail trade" is most frequently the sector with highest random walk centrality. In many economies, this sector has the highest share of final demand. Still, it is noteworthy that our normalization does not depend upon this fact. For example, in Germany in 2000, this sector accounts for 12% of final demand. "Real estate activities" is the second most important sector accounting for 9.6% of final demand, but its random walk centrality is ranked only eighth.

Counting betweenness reveals the importance of Nokia in Finland and the "Motor vehicles" sector in several advanced industrialized economies like Germany, France or Japan. Textiles play an important role in China, Indonesia, and Turkey, showing the significance of that manufacturing sector in countries with low wages. "Finance and insurance" is most central for Luxembourg. Finally, we note that "Public administration, defence, and compulsory social security" is most central in Israel, South Africa, and the US in 2000.

2.4.2 Comparison of different countries

The consistency of the data across countries allows us to immediately compare the centralities of sectors over different countries. We use a clustering technique to visualize our results. The adjacency matrices are of dimension $2209 = 47 * 47$, but our focus on centrality reduces each economy to a vector of length 47. Reducing the complex networks to a list of centrality values, we compress dramatically the relevant information. Moreover, we do not want to attach too much importance to the actual centrality numbers themselves, since we removed sectors without output in some countries. Instead, we are concerned with rankings. Thus, for us two economies are similar if their Spearman rank correlation of centralities across the sectors is high.

An easy and commonly used clustering technique is hierarchical clustering, introduced in Chapter 1.1.1.1. The agglomerative algorithm groups economies starting with the closest pair. In Figure 2.4(A), Belgium and Spain are the two most similar

Table 2.1: The most central sectors in the economies benchmarked by the OECD.

Country	Random Walk Centrality	Counting Betweenness
Argentina	Food products	Health and social work
Australia	Wholesale and retail trade	Wholesale and retail trade
Austria	Wholesale and retail trade	Wholesale and retail trade
Belgium	Wholesale and retail trade	Motor vehicles
Brazil	Wholesale and retail trade	Food products
Canada	Wholesale and retail trade	Motor vehicles
China	Construction	Textiles
Czech Republic	Wholesale and retail trade	Construction
Denmark	Wholesale and retail trade	Food products
Finland	Wholesale and retail trade	Communication equipment
France	Construction	Motor vehicles
Germany 1995	Wholesale and retail trade	Motor vehicles
Germany 2000	Wholesale and retail trade	Motor vehicles
Great Britain	Wholesale and retail trade	Health and social work
Greece	Wholesale and retail trade	Wholesale and retail trade
Hungary	Wholesale and retail trade	Motor vehicles
Indonesia	Wholesale and retail trade	Textiles
India	Land transport	Food products
Ireland	Construction	Office machinery
Israel	Public admin. & defence social security	Health and social work
Italy	Wholesale and retail trade	Wholesale and retail trade
Japan	Other business activities	Motor vehicles
Korea	Construction	Motor vehicles
Luxembourg	Finance and insurance	Finance and insurance
Netherlands	Wholesale and retail trade	Food products
Norway	Wholesale and retail trade	Food products
New Zealand	Wholesale and retail trade	Food products
Poland	Wholesale and retail trade	Wholesale and retail trade
Portugal	Wholesale and retail trade	Health and social work
Russia	Wholesale and retail trade	Food products
Slovakia	Wholesale and retail trade	Motor vehicles
South Africa	Public admin. & defence social security	Public admin. & defence social security
Spain	Wholesale and retail trade	Construction
Sweden	Other business activities	Motor vehicles
Switzerland	Wholesale and retail trade	Chemicals
Turkey	Food products	Textiles
Taiwan	Wholesale and retail trade	Office machinery
USA 1995	Wholesale and retail trade	Health and social work
USA 2000	Public admin. & defence social security	Public admin. & defence social security

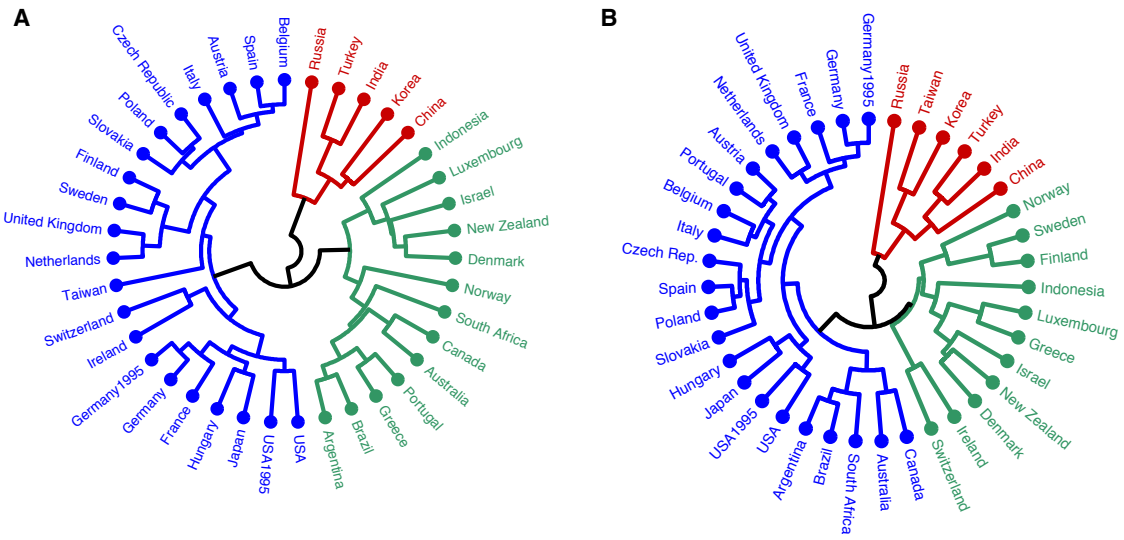


Figure 2.4: (A) gives a hierarchical clustering according to random walk centrality. Colors indicate the three important clusters: (1) the industrial countries from Belgium through the USA; (2) a mixed group from Argentina through Indonesia, where agriculture and primary products are important; and (3) a group of emerging economies from China through Russia. (B) shows clusterings according to counting betweenness. The clusterings according to the two measures are in large parts stable.

countries; hence, they are on the lowest linked branches. Similarity is measured by Spearman rank correlation of the centralities across the sectors. Then, we use complete linkage clustering to construct the rest of the dendrogram.

Cutting a dendrogram at a predefined threshold gives a clustering at the selected precision. At the threshold of 0.65, we find three clusters in Figure 2.4(A): (1) a group of advanced industrial economies ranging from Belgium through the United States; (2) a mixed group of countries where agriculture may be important; and (3) a group of rapidly emerging economies ranging from China through Russia.

Figure 2.4(B) shows a clustering of economies based upon the similarity according to counting betweenness. Note that Taiwan is grouped quite differently in the two dendrograms. According to random walk centrality, it is in the middle of the advanced industrial economies. But in the clustering according to counting betweenness, it is a close neighbor of Korea, in the "Asian Tigers" sub-group of the emerging economies. An important reason for this difference is that Korea and Taiwan have food products and textiles sectors, both of which have strong self-loops. The clusterings capture the remnants of the historical development process in which both economies were based on manufacturing sectors just one generation ago.

It is reassuring that the clusterings are in large parts stable across the two measures. The groupings are natural; it is appropriate that the American and German

Table 2.2: Two advanced economies that are similar in their nodes' rankings according to random walk centrality.

Rank	Sector in Belgium	Sector in Spain
1	Wholesale & retail trade	Wholesale & retail trade
2	Construction	Construction
3	Other business activities	Hotels and restaurants
4	Food products	Other business activities
5	Chemicals	Food products
6	Hotels and restaurants	Real estate activities
7	Travel agencies	Travel agencies
8	Motor vehicles	Other social services
9	Agriculture	Motor vehicles
10	Health and social work	Agriculture

Table 2.3: Two emerging economies that are similar in their nodes' rankings according to random walk centrality.

Rank	Sector in India	Sector in Turkey
1	Land transport	Food products
2	Food products	Wholesale & retail trade
3	Agriculture	Construction
4	Construction	Hotels and restaurants
5	Hotels and restaurants	Agriculture
6	Textiles	Finance & insurance
7	Health and social work	Textiles
8	Wholesale & retail trade	Land transport
9	Chemicals	Travel agencies
10	Production	Machinery and equipment

economies, each sampled five years apart, are most closely related to their former selves. Leontief argued that the stability of input-output relations across time was a good empirical justification for using a fixed-coefficients technology in his original work (Leontief, 1986). These clusterings support his assertion.

2.4.3 Two detailed comparisons

Focusing on random walk centrality, we turn briefly to a detailed study of two different pairs of similar economies. Tables 2.2 and 2.3 look into the details inherent in the sector's rankings that arise from that measure.

The two nearest neighbors in Figure 2.4(A) are Belgium and Spain. Both are advanced economies. Table 2.2 reports the ten most central sectors in each country.

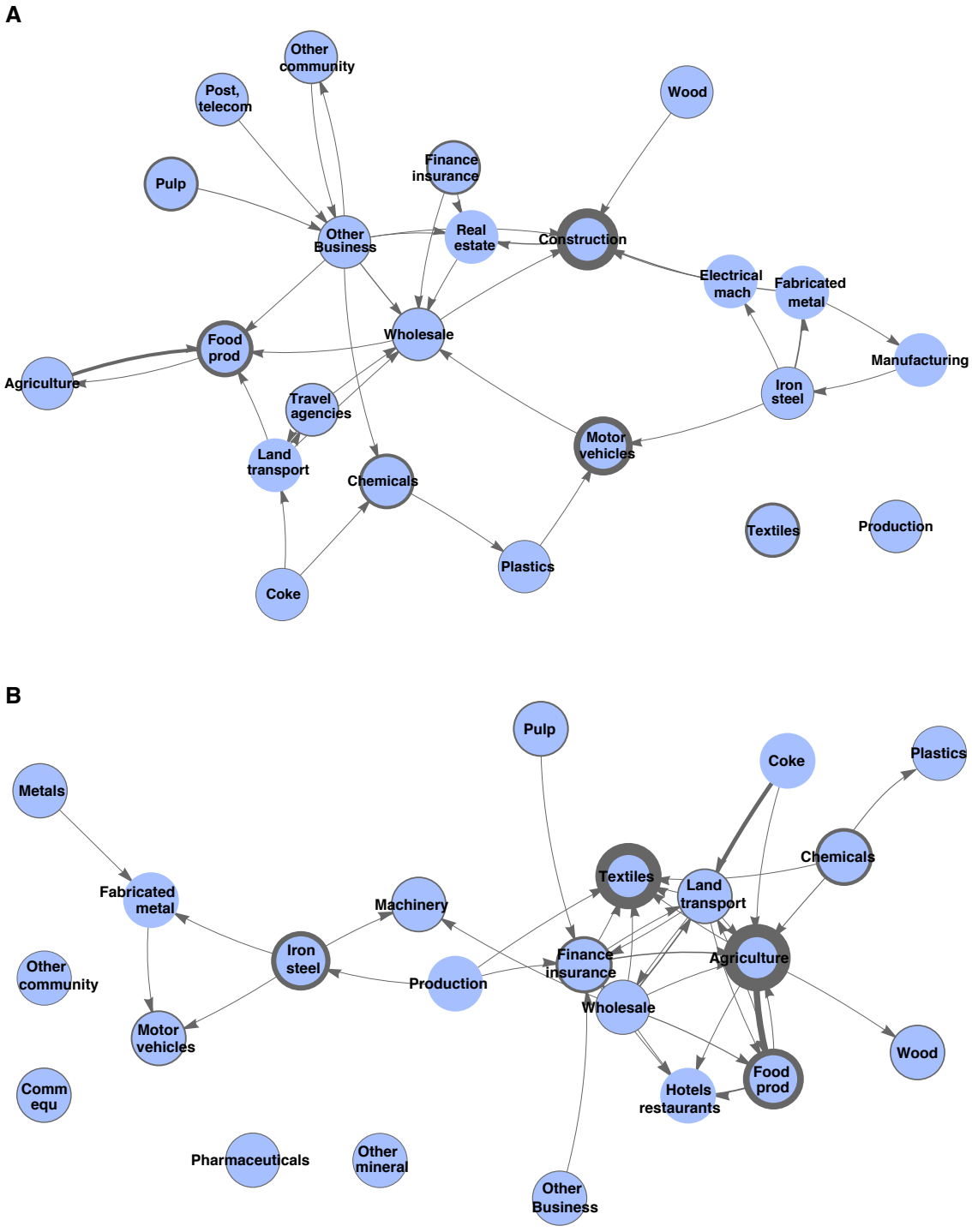


Figure 2.5: The core of the input-output networks of (A) Spain and (B) Turkey. For this illustration, the graphs were thresholded. Edge thickness corresponds to the observed commodity flows, the thickness of the nodes' strokes encodes self-loop weight.

There is a remarkable similarity between the flow of intermediate inputs in these economies. The most central sectors in both countries are "Retail trade" and "Construction". These sectors are notoriously pro-cyclical, and random walk centrality shows that fact clearly.

India and Turkey are two developing countries that cluster together. This pair is somewhat less similar than Belgium and Spain; in Figure 2.4(B), the length of the branch that brings them together is twice as high as that for Belgium and Spain. "Food products", "Construction", and "Hotels and restaurants" all have high centrality rankings. These rankings seem to indicate that the sectoral composition of business cycles is somewhat different in an emerging economy.

Figure 2.5, where we have drawn the core of the input-output networks of Spain and Turkey, additionally visualizes the differences between the structure of an advanced and an emerging economy.

2.5 Conclusions and outlook

This Chapter developed two vertex centrality measures that are based on random walks. A node's random walk centrality is the inverse of the mean number of steps it takes a random walker to reach it, averaged over all starting nodes. Counting betweenness measures the expected number of times that a random walk passes a certain node before it reaches its target, averaged over all pairs of sources and targets. Both measures allow the analysis of weighted directed networks with self-loops.

The need for such measures arose from interpreting an economic question within a graph-theoretic framework. We expect that our techniques will be useful for analyzing payment networks and other financial systems. Moreover, any coarsely grained network – such as one describing clubs or teams, not just individuals themselves – will have important self-loops. Our measures will serve well to describe this kind of network architecture. We agree with Estrada et al. (2009) that there is no best measure of centrality, and we followed their advice and developed two measures that are based in economic theory.

We directed our attention to the flow of economic activity as intermediate inputs before they exited the system for use in final demand. Our measures identifed a central node as a sector that is affected most immediately or most strongly by a random supply shock. Applying these measures to OECD data revealed important aspects of different national economies. We took full advantage of the consistency of the data across countries and gave clusterings of the sector' rankings in these networks. These were intuitive, grouping countries with similar levels of development.

To the best of our knowledge, our hierarchical clustering based on node centralities

was the first attempt to quantitatively compare properties of individual nodes which are linked differently in multiple instances of connecting graphs. This was possible because we had the same sectors trading goods and services in many countries. Hence, we believe this data set is a rich source for other researchers in the field.

There is a lot more work to be done in this area. The theory of networks has flourished in the last decade, and consistent international data have also become widely available during this time. These data have a time dimension, and one may also begin to study the temporal evolution of economic networks. This may well enable researchers to connect generative models of networks with observations from the real world. Further, comparisons of extended versions of these network architectures may shed light on the oldest question in all of economics: Why are some countries poor, while others are prosperous?

3 From topology to dynamics: effective parameters in hierarchical systems

This Chapter shows the connection between the graph topology of hierarchical interaction networks and the structure of the according dynamical systems when the dynamic is modeled by two special types of systems of differential equations. These models are of high relevance for theoretical biology because they are mimicking signal transduction in biological systems in a generic way. We describe the interplay of the different model parameters by algebraic expressions called *effective parameters* and derive how these can be obtained directly from the topology of the interaction graph.

3.1 Hierarchical systems as generic models of cell signaling

In higher organisms, complex regulatory networks are responsible for the correct processing and transduction of information: Developmental processes are coordinated by transcriptional regulatory cascades (Bolouri and Davidson, 2003). Protein signaling networks enable cells to react to environmental conditions and external stimuli. When foreign antigens bind to the receptor of T-cells, for example, a signaling cascade is triggered within the cells controlling its activation (Saez-Rodriguez et al., 2007). Malfunctions within this cascade may cause autoimmune diseases or cancer.

3.1.1 Problem formulation

Due to their importance, a thorough understanding of how information propagates through signaling networks is a major goal in theoretical biology. To this end, models of these cascades are constructed. Typically, biological processes are modeled as a *dynamical system*, i.e. a set of variables

$$x = (x_1, x_2, \ldots, x_n)$$

that represent biological quantities, such as mRNA or proteins, and a 'rule' describing the time-dependence of these variables. When modeling macroscopic chemical systems one commonly neglects the discrete nature of the participating reactants and

their reactions. The variables x_i then assume continuous values and their temporal development is determined by a system of *ordinary differential equations (ODE)*.

The dynamics within a signaling network are the result of a concerted action of interaction topology and interaction parameters. We study the combined effect of the various kinetic parameters on signal transduction. To this end, we consider hierarchical complex systems as generic prototypes of signaling networks. Admittedly signaling pathways contain feedback loops. These interactions, however, typically operate on a much slower time-scale. Moreover, they start later in time, causally following the initial 'forward-signaling' components. A clear separation of time-scales in mammalian signaling networks into a fast signal reception and transduction phase (mediated by constitutively expressed proteins) and a slower feedback phase regulated by de novo induced genes has been recently demonstrated by Legewie et al. (2008). Hence, if we concentrate on the initial phase of signal transduction, our focus on hierarchical networks is no severe restriction.

Depending on the network's topology, the kinetic parameters have synergistic as well as antagonistic effects on the system's output. In the following, we show that for different kinetics one is able to describe this interplay by algebraic expressions which we call the effective parameters. We derive for two commonly employed types of activations functions how these effective parameters can be immediately inferred from the topology of the interaction network. We also demonstrate how an interpretation of these effective parameters may yield insights into the biological properties of the system under study.

Closely related to the task of determining effective parameters is the estimation of numeric values for the parameters. For some parameters we may be able to find estimates in the literature or in databases (Sharova et al., 2008, Yen et al., 2008), but usually a large number of parameters have to be determined by fitting them to experimental data. Since in larger systems it is unfeasible to experimentally observe all quantities in the network, these optimization problems are often ill-determined. In analytic as well as heuristic studies many criterions and algorithms have been proposed that allow detection of unidentifiable parameters (Davidescu and Jŏrgensen, 2008, Denis-Vidal et al., 2003, Hengl et al., 2007). Algebraic knowledge of effective parameters can be used to stabilize and speed up parameter fitting processes. We will confirm this experimentally by simulations.

3.1.2 A mathematical model of signal transduction

In the following, we consider a connected directed hierarchical (acyclic) graph $G = (V, E)$ of size n with set of vertices $V = \{x_1, \ldots, x_n\}$ and set of l edges $E \subset V \times V$. Our graph has a set of source nodes Σ, representing e.g. receptors, and a set of

observable nodes Ω. Each node x_i, $i = 1, 2, \ldots, n$, represents a biological entity such as a protein, mRNA, etc. Node i has its inputs $\mathfrak{I}_i \subset \{1, 2, \ldots, n\}$ and the in-degree $\mathsf{k}_i^{in} := |\mathfrak{I}_i|$. The temporal development of x_i is described by a variable, which — by abuse of notation — is also denoted by $x_i(t)$, and governed by the ODE

$$\dot{x}_i(t) = P_i\left(x_j(t) \mid j \in \mathfrak{I}_i\right) - \tau_i x_i(t) \; . \tag{3.1}$$

The function P_i represents the level of activation of x_i depending on its inputs. Without loss of generality we assume that P_i depends non-trivially on all its inputs. Decay of x_i is assumed to be proportional to its concentration with rate τ_i. Note that we consider information-flow and not mass-flow networks. In particular, for $i \in \Sigma$ we have $\mathfrak{I}_i = \emptyset$ and P_i is a constant function, $P_i = k_i$.

The edges E typically represent enzymatic molecular reactions. In the following, we consider the two extreme cases of models for this type of interaction: We choose linear as well as Heaviside activation functions P_i. For both kinetics we develop general methods to analytically derive the effective parameters determining the dynamics of the system from its interaction graph. To this end, we study the following idealized situation mimicking signal transduction in biological systems.

> At time $t = 0$, we set all variables to their steady state value for $x_i = 0$, $i \in \Sigma$, except for the external inputs x_i, $i \in \Sigma$, which we set to a positive activation level $x_i(0) = a_i$. Subsequently, information will propagate through the network causing nodes to toggle. We observe the time-course of the observable nodes Ω. (S)

3.2 Heaviside step activation functions

Many molecular interactions show a switch-like behavior. They rapidly switch from the deactivated state to the activated state, where they reach a saturation level. In this Section we model these interactions by idealized Heaviside step functions. Here, the transition phase is completely neglected. Consequently, each activation function P_i assumes only two values, 0 and a maximal activation level k_i. This implies that each variable x_i has an upper bound $m_i := k_i/\tau_i$, as for larger values its derivative (3.1) will be negative.

For each input x_j, $j \in \mathfrak{I}_i$, of node x_i we introduce a switching threshold θ_{ji}. In order to ensure its effectiveness, we require $0 < \theta_{ji} < m_j$. Let $0 < \overline{\theta}_{ji} := \theta_{ji}/m_j < 1$ denote the thresholds normalized by the maximal expression level. The hyperplanes $x_j = \theta_{ji}$, $j \in \mathfrak{I}_i$, subdivide the k_i^{in}-dimensional input-space of P_i into $2^{\mathsf{k}_i^{in}}$ rectangular domains. In each of these regions, we assign P_i a constant value, either 0 or k_i.

The resulting piecewise constant activation functions P_i were first introduced by Glass and Kauffman (1973) as simple extension of Boolean functions. With this choice of activation function, (3.1) becomes a system of piecewise linear ODEs. This type of model has been extensively used to study regulatory and signaling networks in a qualitative and semi-quantitative way (de Jong et al., 2004, Öktem, 2005, Snoussi, 1989). Of particular interest is the relation between the thresholds and many algorithms have been proposed to infer them from experimental data (Drulhe et al., 2006).

Each P_i is uniquely determined by k_i, the underlying Boolean function, denoted by B_i, and the threshold vector $(\theta_{ji}, j \in \mathfrak{I}_i)$. In the rest of this Section, we assume that we are given the logics B_i and study the combined effect of the parameters on the dynamics of the network. We always represent the B_i in a disjunctive normal form

$$B_i = \left(\bigwedge_{j \in U_{i1}} x_j \right) \vee \ldots \vee \left(\bigwedge_{j \in U_{ik_i}} x_j \right) \tag{3.2}$$

with subsets $U_{i1}, \ldots, U_{ik_i} \subset \mathfrak{I}_i$.

3.2.1 Systematic substitution of inhibitory interactions

We restrict ourselves to activating influences which implies that no negations will appear in (3.2). We point out that this restriction is no severe loss of generality, as we can substitute inhibitions by activations, provided the network is consistent in the sense that there are no ambiguous effects. Therefore, all B_i are monotonous in their arguments. Consistent networks are biologically plausible assumptions, as we already focus only on the initial phase of signaling. The practical implementation of this substitution is described and exemplified in the following.

Now, let $G = (V, E)$ be an acyclic interaction graph of some monotonous Boolean network with source nodes $\Sigma \subset V$, a target node t. Note that if B_i is monotonously increasing (decreasing) in an input x_i, we can write B_i such that only literals x_i ($\neg x_i$) appear in the propositional formula.

The sign of a path p in G is defined as $(-1)^i$ where i is the number of negative edges along p. In particular, the sign of an "empty" path consisting of only one node is 1 and the sign of a path consisting of one edge agrees with the sign of this edge. Our graph is consistent, hence for any two nodes $x_i, x_j \in V$ each path between x_i and x_j has the same sign $\sigma(x_i, x_j)$. We divide the nodes into two classes by setting $c(x) = \sigma(x, t)$. Note that there is always a (directed) path from x to t as t is the only target node.

We define new variables y_i, $i = 1 \dots n$ by

$$\begin{aligned} y_i &:= x_i \qquad \text{if} \quad c(x_i) = 1 \ \text{ and} \\ y_i &:= \neg x_i \qquad \text{if} \quad c(x_i) = -1 \end{aligned}$$

and write the Boolean update functions in these new variables. In order to keep clear whether we look at the original or the transformed system, we add x or y as superscript to the Boolean functions. We will see that the functions B_i^y of the new variables (written in terms of the new variables) are monotonously increasing in each argument. To explain this in detail, let us fix some node $x_\alpha \in G$ with inputs (immediate predecessors) $x_1, x_2, \dots, x_k$. Note that $c(x_i) = \sigma(x_i, x_\alpha)c(x_\alpha)$. There are two possibilities:

- $c(x_\alpha) = 1$: An edge (x_i, x_α) is inhibiting, i.e. $\sigma(x_i, x_\alpha) = -1$, if and only if $c(x_i) = -1$. Hence, in B_α^y all inhibitions $\neg x_i$ from B_α^x are replaced by activations y_i. If B_α^x is in disjunctive normal form, then so is B_α^y.
- $c(x_\alpha) = -1$: Now, an edge (x_i, x_α) is activating, $\sigma(x_i, x_\alpha) = 1$, if and only if $c(x_i) = -1$. Hence, replacing all x_i satisfying $c(x_i) = -1$ by $\neg y_i$ eliminates all activations from B_α^x. Consequently, $B_\alpha^y = \neg B_\alpha^x$ is purely activating. Note that after application of de Morgan's laws, B_α^y will in general not have the same form as B_α^x. However, it can be brought into disjunctive normal form by standard methods such as the Quine-McCluskey algorithm (McCluskey and Bartee, 1962, Quine, 1952).

The substitution rule for the Boolean update functions generalizes to the according piecewise linear ODE systems. This is a consequence of the transformation properties of the Heaviside function (denoted by H): for $x \neq 0$, it holds that $H(-x) = 1 - H(x)$, as is in the Boolean case. By convention, $H(0) = 0$.

A simple inhibition $x_j \dashv x_i$ is described by the equation

$$\dot{x}_i = k_i \left(1 - H\left(x_j - \theta_{ji}\right)\right) - \tau_i x_i \,.$$

Now, we can decide between substituting either the initial node or the target. A substitution $y_i := m_i - x_i = k_i/\tau_i - x_i$ of the target species gives us

$$\begin{aligned} \dot{y}_i &= -k_i \left(1 - H\left(x_j - \theta_{ji}\right)\right) + k_i - \tau_i y_i \\ &= H\left(x_j - \theta_{ji}\right) - \tau_i y_i \,, \end{aligned}$$

which is exactly the equation for an activation $x_j \to y_i$ with the same threshold. If

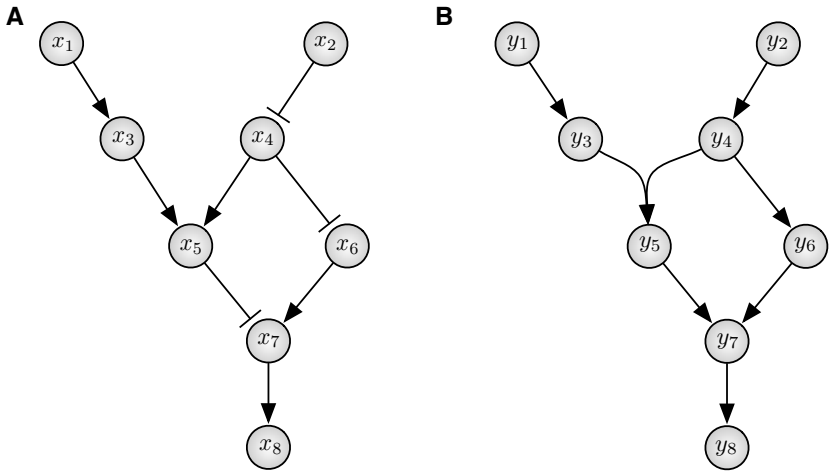


Figure 3.1: (A) shows a hierarchical network with inhibiting and activating edges. Of the eight nodes, two are inputs and we have one observed output x_8. At each node the Boolean function B_i is encoded in the structure of the incoming edges (Klamt et al., 2006). Hyperedges represent coupling by AND logic. Multiple incoming (hyper)edges at a node are linked by OR gates. In (B) we show the same network, with all inhibitions substituted. Note the change of logical circuitry in the input of node x_5.

we substitute the source node $y_j := m_j - x_j$, we find

$$\begin{aligned} \dot{x_i} &= k_i \left(1 - H\left(m_j - y_j - \theta_{ji}\right)\right) - \tau_i x_i \\ &= k_i \left(1 - \left(1 - H\left(-m_j + y_j + \theta_{ji}\right)\right)\right) - \tau_i x_i \\ &= k_i H\left(y_j - \left(m_j - \theta_{ji}\right)\right) - \tau_i x_i \,. \end{aligned}$$

This is the ODE for an activation $y_j \to x_i$ with a shifted threshold $(m_j - \theta_{ji})$. Thus we can change inhibitions to activations and vice versa, where eventually the threshold has to be shifted, depending on which variable is substituted. We illustrate this procedure in the example system shown in Figure 3.1.

3.2.2 Algorithmic determination of the effective parameters

We note that $P_i\left(0, 0, \ldots, 0\right) = 0$, $i = 1, 2, \ldots, n$. This is due to our restriction to purely activating P_i that non-trivially depend on all inputs. Inductively it follows that in situation (S) at $t = 0$ we have $x_i(t) = 0$, $i \notin \Sigma$. Now assume without loss of generality that x_n is an observable node and that we stimulate the external inputs x_i, $i \in \Sigma$, by their maximal expression level, i.e. $a_i = m_i$. Then the only effect of an interaction $x_j \to x_i$ on the activation of x_n is a time delay δ_{ji} between the time-point T_j where P_j switches to 1 and the time-point where $x_j(t) = \theta_{ji}$. For fixed variables x_i and $x_j \in \mathfrak{I}_i$, $j \notin \Sigma$, we have

$$x_j(t) = m_j \left(1 - e^{-\tau_j (t - T_j)}\right) \quad \text{for } t > T_j$$

Algorithm 1: ConstructEffectiveParameter(A, D, i)

```
D ⟵ D ∪ {i};
Create node Y_i labeled 'min';
if A ≠ ∅ then
  | Connect Y_i to the graph through A;
end
foreach U_ik do
  | Create node Y_{i,k} labeled 'max' and connect it to the graph through Y_i;
  | foreach j ∈ U_ik do
  |   | if j ∉ Σ then
  |   |   | Create node Y_{i,k,j} labeled 'δ_ji+' and connect it to the graph through Y_{i,k};
  |   |   | if j ∈ D then
  |   |   |   | Insert edge Y_{i,k,j} ⟶ Y_j;
  |   |   | else
  |   |   |   | Call ConstructEffectiveParameter(Y_{i,k,j}, D, j);
  |   |   | end
  |   | else
  |   |   | Return;
  |   | end
  | end
end
```

and it follows that

$$\delta_{ji} = -\frac{\log\left(1-\overline{\theta}_{ji}\right)}{\tau_j} \,. \tag{3.3}$$

We see that the delay caused by each interaction is determined by the relative thresholds as well as by the decay rates of the variables.

We now compute the effective parameter determining the time-point of activation of $x_n \in \Omega$. Algorithm 2 constructs this effective parameter as a directed tree $\mathcal{T}$ by recursively calling the function `ConstructEffectiveParameter`. This function takes three inputs: the current position A in $\mathcal{T}$, the set D of already processed nodes of G and the current position i in G. The output is a directed tree describing the time delay between the activation of the inputs of x_i and the activation of node x_i. It is linked to $\mathcal{T}$ through A. The activation function P_i of x_i is switched on as soon as the first Boolean monomial $\wedge_{j\in U_{ik}} x_j$, $1 \leq k \leq k_i$, becomes true. Hence the function `ConstructEffectiveParameter` first inserts a 'min'-node. The monomial, in turn, becomes true only after its last input has switched, which is why a 'max'-node is inserted next. Naturally, the time delays caused by the interactions simply add up to the total delay of the signal. For each non-external input x_j of x_i, $j \notin \Sigma$, that has not yet been processed ($j \notin D$), `ConstructEffectiveParameter` is called recursively. The postprocessing steps in Algorithm 2 simplify the resulting graph $\mathcal{T}$ by eliminating min- and max-operators with only one argument.

Algorithm 2: Construction of the effective parameter.

```
T ⟵ Call ConstructEffectiveParameter(∅, ∅, n);
foreach node A with label L of T do
    if L ∈ {min, max} then
        switch outdegree of A do
            case 0
                Delete A;
                Delete all incoming edges;
            end
            case 1
                Delete A;
                Connect all incoming edges directly to the successor of A;
            end
        end
    end
end
foreach leaf A do
    Delete the + sign from the label
end
```

Algorithm 1 creates one node for every node, hyperedge and edge in the interaction graph. Thus, as the number of hyperedges is bounded by the number of edges l, it requires $\mathcal{O}(n + l)$ steps to build the preliminary graph. The postprocessing in Algorithm 2 runs over all nodes of this intermediate tree, hence it also has runtime of $\mathcal{O}(n + l)$. Altogether, the effective parameter is constructed in $\mathcal{O}(n + l)$ steps.

Due to the saturating activation functions the total level of expression of the variables upstream of x_n does not influence its expression. Besides the effective activation threshold computed above, $x_n(t)$ is only affected by the parameters k_n and τ_n, which, in particular, determine its steady state m_n.

3.2.3 Effective parameters in a toy example

Consider, for example, the hierarchical graph shown in Figure 3.2(A). The effective parameter for the activation of node x_8 can be computed by Algorithm 2. In Figure 3.2(B) it is visualized as a directed tree. We see that the effect of a single parameter on the system's dynamics crucially depends on the values of the other parameters. The delaying effect of θ_{24} and θ_{34}, for instance, becomes completely irrelevant as soon as e.g. θ_{48} becomes sufficiently large; then the activation of x_8 will occur by the path corresponding to the right-hand argument of the min-source node.

Another way of interpreting the output of Algorithm 2 is to relate the effective parameters to a decomposition of the interaction graph into elementary motifs (Ravasz et al., 2002). Consider, for instance, the two feed-forward loops from Figures 3.3(A)

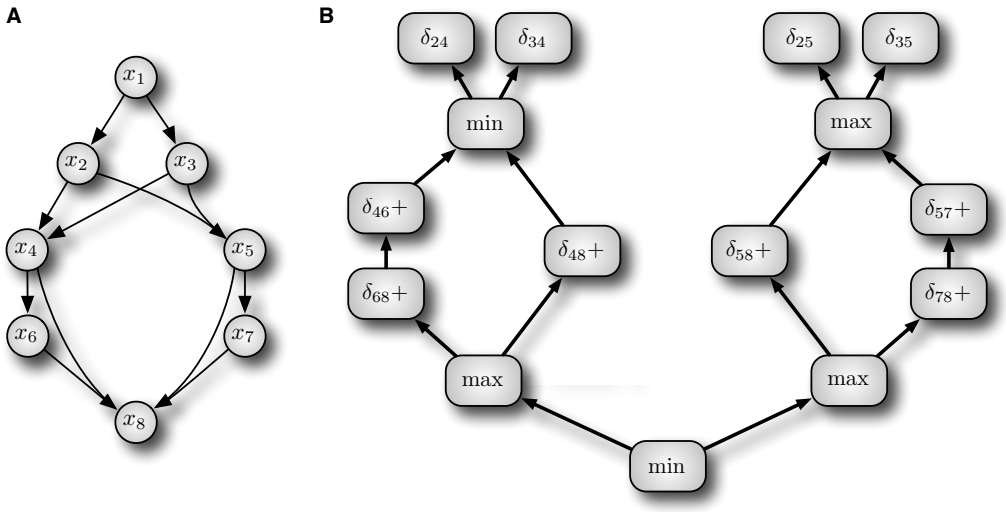


Figure 3.2: (A) shows a hierarchical network. At each node a disjunctive normal form of the Boolean function B_i is encoded in the structure of the incoming edges (Klamt et al., 2006). Hyperedges represent the Boolean monomials U_{ik}. Multiple incoming (hyper)edges at a node are linked by OR gates. (B) shows the output from Algorithm 2. The directed tree shows the effective parameter determining the activation of node x_8 in (A).

and 3.3(B) where the two incoming interactions at x_3 are linked by an OR and AND gate, respectively. For illustration purposes, we set all $\tau_i = k_i = 1$. Then the dynamic of these systems is determined only by the three thresholds θ_1, θ_2 and θ_3. We can now decompose the networks into a linear three-node cascade and an OR respectively AND gate, as shown in Figures 3.3(C-E). Each of these smaller networks contains only two parameters θ_a and θ_b. From Equation (3.3) we obtain corresponding time delays $\delta_a = -\log(1-\theta_a)$ and $\delta_b = -\log(1-\theta_b)$. Consequently, the effective parameter of the OR gate, for instance, is $\min(\delta_a, \delta_b)$. In terms of the parameters θ_a and θ_b this implies a relation $\max((1-\theta_a),(1-\theta_b)) = const.$ Similar relations can be obtained also for the linear cascade and the AND gate. They are visualized in the θ_a-θ_b parameter planes, cf. Figures 3.3(C-E). By glueing the parameter space of 3.3(D) to the one of 3.3(C) and 3.3(E), respectively, along one dimension, we obtain the three dimensional parameter spaces with the one-codimensional effective parameters of 3.3(A) and 3.3(B). This is shown in Figures 3.3(F) and 3.3(G).

3.2.4 Generalization to Hill kinetics

Heaviside step functions might, of course, be an unrealistic idealization of switch-like interactions. Biologically more realistic activation functions are sigmoid Hill functions of the form

$$H_{\alpha,\theta}(x) = \frac{x^\alpha}{x^\alpha + \theta^\alpha} \ .$$

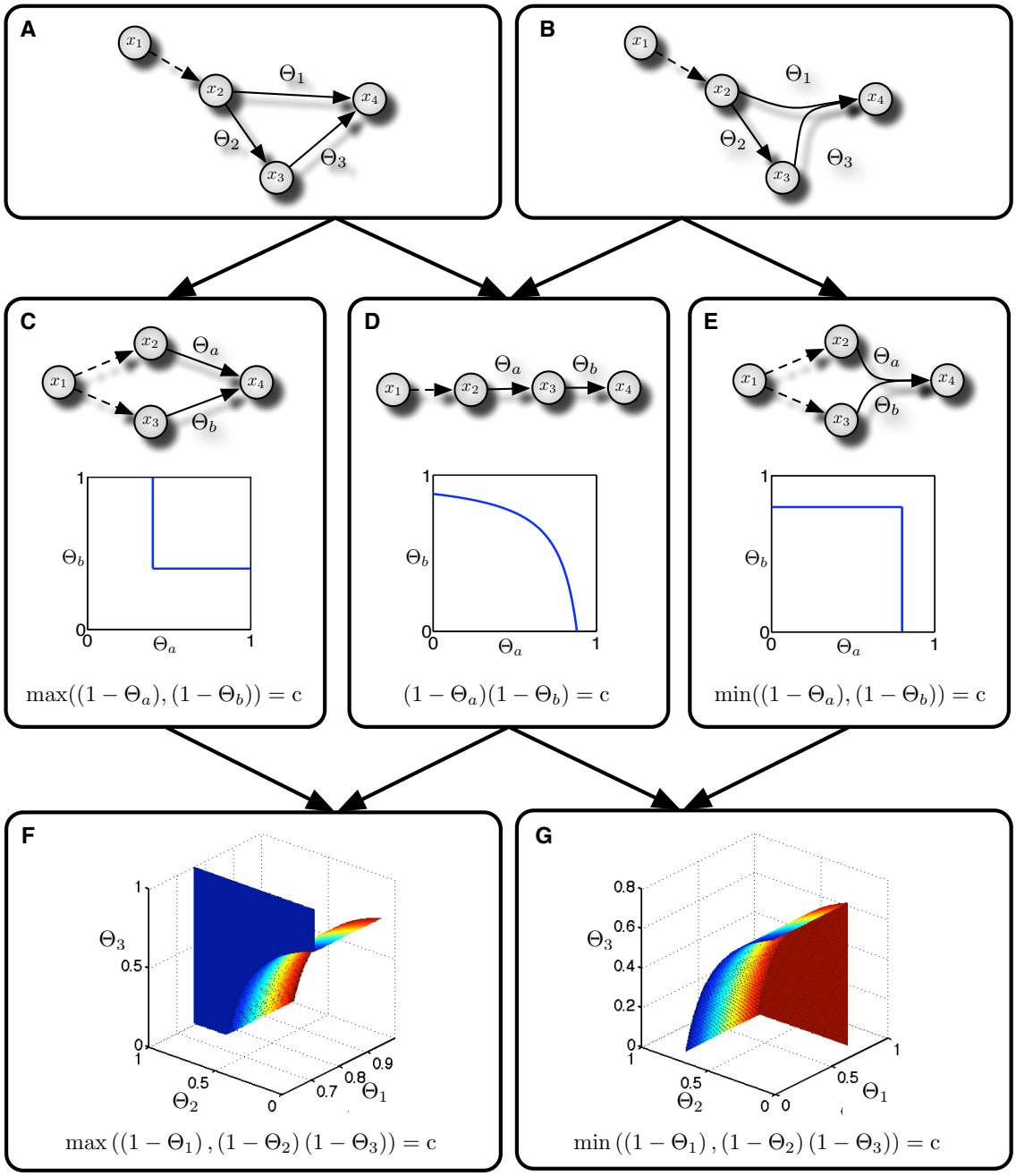


Figure 3.3: Interpretation of effective parameters by a decomposition of the interaction graph. Dashed lines indicate interactions, whose thresholds are irrelevant as at $t = 0$ we set x_1 to its maximal expression level. For the Boolean functions the same hypergraph representation as in Figure 3.3 is used. (A) and (B) illustrate feed-forward loops with OR and AND gates. We set all $\tau_i = k_i = 1$ and illustrate the combined influence of the threshold parameters on the system. We decompose the networks into a linear cascade (shown in (D)) and OR (C) respectively AND (E) gates. The corresponding effective parameters are stated and visualized as relations between θ_a and θ_b. Similarly, in (F) and (G) the effective parameters of the feed-forward loops are stated and visualized as relations between θ_1, θ_2 and θ_3. We see that both, the analytic expressions as well as the graphical representations in (F) and (G) are hierarchical combinations of their counterparts in (C), (D) and (E).

The parameter θ plays the same role as in step functions. It fixes the position of the switch in the sense that it determines the x-value for half-maximal activation. The Hill parameter α describes the cooperativity of the interaction and determines the slope of the curve. This allows modeling of interactions with rapid yet smooth switches. In the limit $\alpha \longrightarrow \infty$ Hill functions approach Heaviside step functions, cf. Figure 3.4(A). For fixed thresholds $\theta_1 = 0.8$ and $\theta_2 = 0.4$ the AND- and OR-feed-forward loop motifs as well as the linear cascade from Figure 3.3 were numerically simulated; we denote the corresponding time-courses of x_3 by $x_{3,\mathrm{AND}}(t)$, $x_{3,\mathrm{OR}}(t)$ and $x_{3,\mathrm{casc}}(t)$, respectively. Then for thresholds θ_1', θ_2' and corresponding time-courses $x'_{3,\mathrm{AND}}(t)$, $x'_{3,\mathrm{OR}}(t)$ and $x'_{3,\mathrm{casc}}(t)$ we can compute the square errors

$$\begin{aligned} \epsilon_{\mathrm{AND}} &= \log\left(\sum_t \left(x_{3,\mathrm{AND}}(t) - x'_{3,\mathrm{AND}}(t)\right)^2\right) \\ \epsilon_{\mathrm{OR}} &= \log\left(\sum_t \left(x_{3,\mathrm{OR}}(t) - x'_{3,\mathrm{OR}}(t)\right)^2\right) \\ \epsilon_{\mathrm{casc}} &= \log\left(\sum_t \left(x_{3,\mathrm{casc}}(t) - x'_{3,\mathrm{casc}}(t)\right)^2\right) \end{aligned} \quad . \tag{3.4}$$

Clearly, the minima of these functions correspond to the effective parameters, as can be seen from Figure 3.4(B), second row. We repeat these *in silico* experiments replacing the Heaviside step functions by sigmoid activation functions as suggested by Plahte et al. (1998) or Wittmann et al. (2009). The resulting error functions are shown in Figure 3.4(B), third and fourth row. We see that the results obtained for idealized step functions, in principle, generalize to smooth sigmoidal functions.

3.3 Linear activation functions

Choosing Heaviside step functions, we so far neglected the transition between the deactivated and activated state of an interaction. Here we focus on this phase of signal transduction. To this end, we approximate the transition by a linear function. Hence our activation functions P_i become

$$P_i\left(x_j(t) \mid j \in \mathcal{I}_i\right) = \sum_{j \in \mathcal{I}_i} k_{ji} x_j(t)$$

with rate constants k_{ji}.

With this choice, Equation (3.1) now becomes a linear system of ODEs

$$\dot{\mathbf{x}} = \mathbf{K}\mathbf{x} \;, \quad \mathbf{x}(0) = \mathbf{x_0} \;, \tag{3.5}$$

where

$$\mathbf{K}(i,j) = \begin{cases} -\tau_i & \text{for } i = j \notin \Sigma \\ k_{ji} & \text{for } (j,i) \in E \\ 0 & \text{elsewhere} \end{cases} \quad .$$

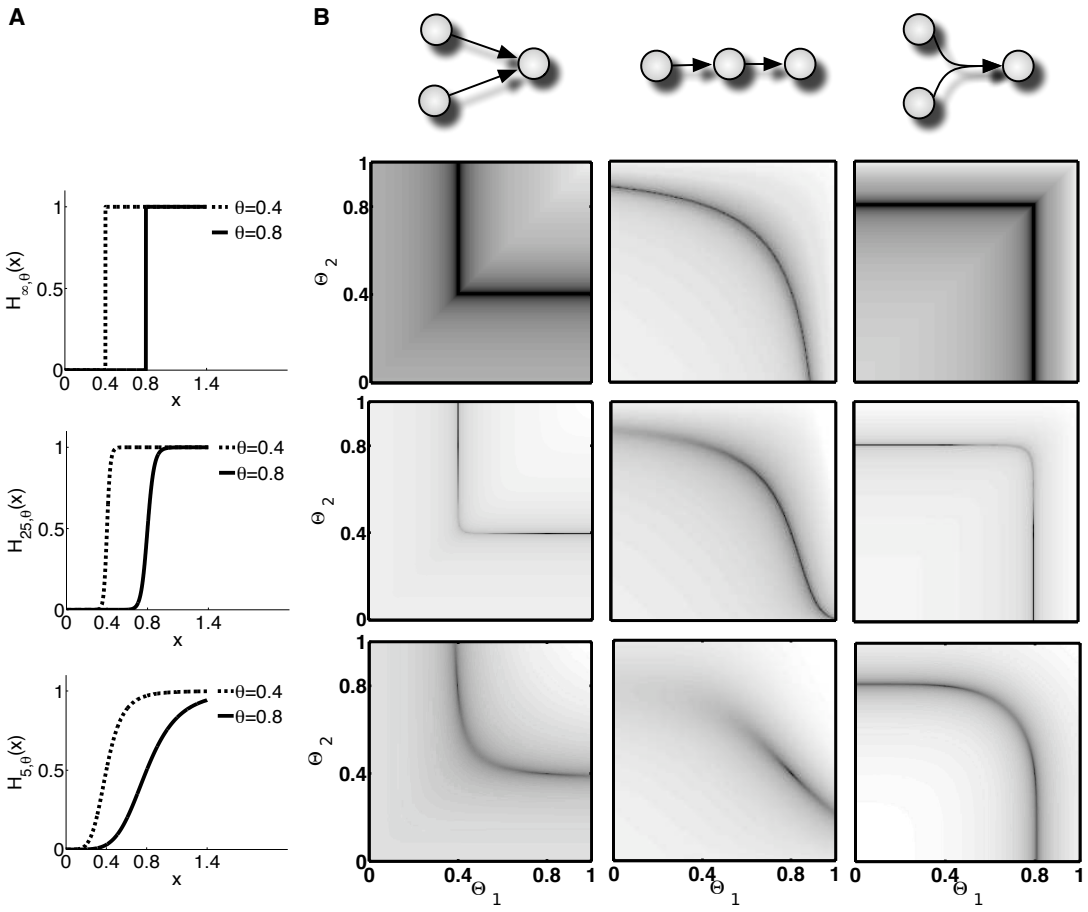


Figure 3.4: Comparison of Heaviside and Hill activation functions. (A) illustrates Hill functions with varying exponents $\alpha = 5, 25, \infty$ and $\theta_1 = 0.8$, $\theta_2 = 0.4$. The case $\alpha = \infty$ are Heaviside step functions. For growing exponents the Hill functions approach the step function. In (B), each row shows the error functions (3.4) for the corresponding Hill functions from (A) as heat-maps (gray scale linearly interpolated between black=minimum and white=maximum). The respective networks are shown at the top. The dark regions agree well with the curves from Figures 3.3(C-E). We see that, in principle, the results from the step function case generalize to sigmoid activation functions. However, Hill functions only asymptotically approach but never assume the value 1. This effect becomes manifest in the case of the cascade (middle column). Especially for large θ and small α it is non-negligible.

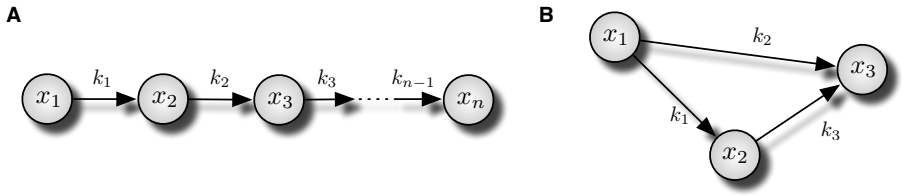


Figure 3.5: (A) gives the linear cascade, (B) a feed-forward loop.

As we restricted ourselves to hierarchical networks, the matrix $\mathbf{K}$ is lower-triangular after possible index permutation.

3.3.1 Illustration: linear cascades

We first study an illustrative special case of hierarchical networks — a linear cascade between the species $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, cf. Figure 3.5(A). The simple structure of this network allows us to algebraically investigate the arising effective parameters for general n. However, this example is also interesting from a biological point of view as linear cascades are frequently found in real-world systems.

We model the cascade by the linear system of ODEs

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 & 0 & \cdots & 0 \\ k_1 & \lambda_2 & 0 & \cdots & 0 \\ 0 & k_2 & \lambda_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & k_{n-1} & \lambda_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} \tag{3.6}$$

with initial conditions $\mathbf{x}(0) = \mathbf{x_0}$.

3.3.1.1 Analytic solution

The ODE system (3.6) is similar to the *Bateman equations* (Bateman, 1910) and can be solved analytically using the formalism of Laplace transforms. For a real function $f(t)$, the Laplace transform is defined as $\mathcal{L}\{f\}(s) := \int_0^\infty e^{-st} f(t)\mathrm{d}t$. To keep notation simple, we set the initial concentrations to $\mathbf{x_0} = (x_{0,1}, 0, \ldots, 0)$. In the general case of arbitrary initial conditions one can inductively combine the solutions for the subchains starting at non-zero reactants.

Applying the Laplace operator $\mathcal{L}$ to (3.6) and using the linearity and the differentiation property of the transform, we find that

$$\begin{aligned} \mathcal{L}\{\dot{x_1}\} &= s\mathcal{L}\{x_1\} - x_{0,1} = \lambda_1 \mathcal{L}\{x_1\} , \\ \mathcal{L}\{\dot{x_i}\} &= s\mathcal{L}\{x_i\} = \lambda_i \mathcal{L}\{x_i\} + k_{i-1}\mathcal{L}\{x_{i-1}\} \qquad \text{for } i = 2, 3, \ldots, n . \end{aligned}$$

Rearranging these equations we get

$$\mathcal{L}\{x_1\} = \frac{x_{0,1}}{s-\lambda_1} \quad \text{and} \quad \mathcal{L}\{x_i\} = \frac{k_{i-1}}{s-\lambda_i}\mathcal{L}\{x_{i-1}\}$$

and arrive at an algebraic expression for the solution of the ODE system in the s-domain:

$$\mathcal{L}\{x_i\} = \frac{k_{i-1}k_{i-2}\dots k_1}{(s-\lambda_i)(s-\lambda_{i-1})\dots(s-\lambda_1)}x_{0,1}\,.$$

The inverse transform of this equation can be obtained by solving the Bromwich integral. Alternatively, instead of solving the complex integrals directly, we apply Heaviside's Expansion Theorem (Davies, 2002): Let $P(s)$ and $Q(s)$ be polynomials of degree m and n, respectively, and $n > m$. If Q has n distinct simple zeros at the points $s_1, s_2, \dots, s_n$, then $P(s)/Q(s)$ is the Laplace transform of $\sum_{k=1}^{n}\left(P(s_k)/Q'(s_k)\right)e^{s_k t}$.

Hence, if we make the reasonable assumption that all λ_j are different, we obtain in our situation

$$x_i(t) = x_{0,1}\,k_{i-1}k_{i-2}\dots k_1\sum_{j=1}^{i}\frac{e^{\lambda_j t}}{\prod_{m=1,m\neq j}^{i}(\lambda_m-\lambda_j)}\,. \tag{3.7}$$

From this equation it can easily be deduced that the effective parameter for signal transduction in linear cascades is the product of the rate constants k_i. Roughly speaking, this result shows that the elements in a linear cascade have equal influence on the output regardless of their position. This is interesting from a biological point of view, as one reason for the widespread occurrence of cascades in biological systems is that each of their nodes constitutes a potential control point (Thattai and van Oudenaarden, 2002). The above results now suggest that each of these control points is equally effective, whereby an exquisite regulation of the output signal is achieved.

3.3.1.2 Example simulation

In order to visualize the effective parameter in linear cascades, let us now relate our previous result to the problem of parameter estimation. We choose $n = 3$ and $\lambda_i = -\tau_i = -1$, $i = 1, 2, 3$. Model (3.6) is numerically simulated for initial conditions $x_1(0) = 1$, $x_2(0) = x_3(0) = 0$ and rate constants $k_1 = 3/2$, $k_2 = 2/3$. Then Gaussian noise with a standard deviation of 0.1 is added to the values $x_3(t)$, $t = 0, 1, \dots, 8$. These data are used as toy data for the fitting process.

As we want to illustrate the effective parameter combined from the rate constants in this example, we assume that we know the decay rates and only have to estimate k_1 and k_2. We determine a best-fit parameter set by minimizing the least-square error between the model simulation and the toy-data with a non-deterministic *simulated annealing* algorithm (Kirkpatrick et al., 1983). The optimization is performed 100

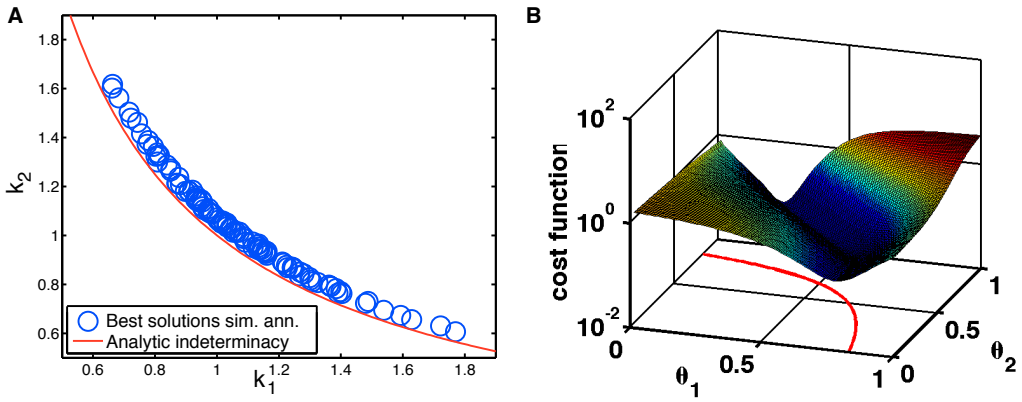


Figure 3.6: Parameter fitting in a three-node cascade modeled according to (3.6). In each figure, the curve representing the parameter indeterminacy, $k_1 k_2 = 1$, is shown for comparison. (A) gives the results of 100 fits of the rate constants k_1 and k_2 using *simulated annealing*. (B) is the cost function (least-square error) depending on the reaction rates k_1 and k_2.

times using random initial conditions. Figure 3.6(A) shows the best-fit parameters as well as the curve $k_1 k_2 = 1$ representing the parameter indeterminacy. Obviously, the optimization algorithm is unable to distinguish between the points along this curve. In Figure 3.6(B) the cost-function (log least-square error) is shown. For comparison the curve $k_1 k_2 = 1$ is plotted in the $k_1 - k_2$ plane and one clearly sees that the cost function is minimal along this curve. The idea of multiple random restarts was also used by Hengl et al. (2007) to empirically describe parameter indeterminacies.

3.3.2 Analytic solution in the general case

We now consider the general case of hierarchical networks as modeled by Equation (3.5). To calculate algebraic expressions for the effective parameters in this setting, we diagonalize $\mathbf{K} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$. Since the network is hierarchical, all eigenvalues are real. This reflects the fact that in such systems no oscillations occur. Then after substitution, (3.5) reads

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{V}^{-1}\mathbf{x}\right) = \mathbf{D}\mathbf{V}^{-1}\mathbf{x}\,, \quad \mathbf{x}(0) = \mathbf{x_0}\,.$$

The solution of this system of ODEs is clearly given by $\mathbf{V}^{-1}\mathbf{x} = e^{t\mathbf{D}}\mathbf{V}^{-1}\mathbf{x}_0$ and we obtain the solution of (3.5):

$$\mathbf{x} = \mathbf{V}e^{t\mathbf{D}}\mathbf{V}^{-1}\mathbf{x_0}\,. \tag{3.8}$$

Hence, each x_i can be written as a linear combination of exponential functions

$$x_i(t) = \sum_{j=1}^{n} a_{ij} e^{\lambda_j t}\,. \tag{3.9}$$

In principle, so far the calculation also holds for non-hierarchical systems, but in practice the applicability of this solution technique is – even for small systems – hampered by the time-consuming or even impossible symbolic computations. However, in the case of hierarchical interaction networks the matrix $\mathbf{K}$ is lower-triangular which implies $\lambda_j = -\tau_j$. The coefficients $a_{ij} = a_{ij}(\mathbf{K}, x_0)$ are rational functions of the entries of $\mathbf{K}$, i.e. of the rate constants k_{ji} and the decay rates τ_i, as well as of the initial conditions $\mathbf{x_0}$.

Let us assume again without loss of generality that x_n is observable and we start with the initial condition $\mathbf{x_0} = (1, 0, \ldots, 0)$. Then, if we assume all τ_j to be different, the time-course $x_n(t)$ uniquely determines the eigenvalues λ_j and the coefficients a_{nj} in (3.9) up to permutation, since the exponential functions are linearly independent over $\mathbb{C}$. The effective parameters are therefore implicitly given by

$$\{(\lambda_j(\mathbf{K}), a_{nj}(\mathbf{K}, \mathbf{x_0})) \mid j = 1, 2, \ldots, n\} . \tag{3.10}$$

The lower-triangular structure of $\mathbf{K}$ implies that it is possible to calculate the j-th column of the eigenvector matrix $\mathbf{V}$ by forward-substitution in the linear system of equations for calculating the kernel of $(\mathbf{K} - \lambda_j \mathbf{I})$. Note that with $\mathbf{K}$ also the eigenvector matrix $\mathbf{V}$ has to be lower-triangular. Graphically speaking, this substitution follows the directed paths in the interaction graph. After choosing $\mathbf{V}(i,i) = 1$ we obtain

$$\mathbf{V}(j,i) = \begin{cases} 0 \text{ for } j < i \\ 1 \text{ for } j = i \\ \displaystyle\sum_{p \in \mathcal{P}(x_i \to x_j)} (-1)^{\ell(p)} \prod_{\text{edges } (e_1,e_2) \text{ along } p} \frac{\mathbf{K}(e_2,e_1)}{\lambda_{e_2} - \lambda_i} & \text{for } i < j \end{cases} . \tag{3.11}$$

Here $\mathcal{P}(x_i \to x_j)$ denotes the set of all possible paths from x_i to x_j, $\ell(p)$ the length of a path p. We define $\mathcal{P}(x_i \to x_i) = \emptyset$ and formally set the empty sum equal to one. For each $p \in \mathcal{P}(x_i \to x_j)$, $\mathbf{V}(j,i)$ contains the squarefree monomial

$$m(p) := \prod_{\text{edges } (e_1,e_2) \text{ along } p} \mathbf{K}(e_2,e_1)$$

as a summand, weighted by a coefficient

$$c_p := (-1)^{\ell(p)} \prod_{\text{nodes } v \neq i \text{ along } p} \frac{1}{\lambda_v - \lambda_i} .$$

Reaction rates always appear divided by corresponding lifetimes, an interplay that determines the actual scale of a species' concentration.

Let us now compute $\mathbf{w} = (w_i) := \mathbf{V}^{-1}\mathbf{x_0}$. We show by induction on i that w_i is of the form $w_i = \sum_{p \in P(x_1 \to x_i)} c'_p m(p)$. From Equation (3.11) it immediately follows that $w_1 = \mathbf{V}(1,1)$. For $i > 1$ forward-substitution gives us

$$w_i = -\sum_{j=1}^{i-1} \mathbf{V}(i,j) w_j = \sum_{j=1}^{i-1} \left(\sum_{p \in \mathcal{P}(x_j \to x_i)} c_p m(p) \right) \left(\sum_{p \in \mathcal{P}(x_1 \to x_j)} c'_p m(p) \right) .$$

We can merge two consecutive paths $p_1 \in \mathcal{P}(x_1 \to x_j)$ and $p_2 \in \mathcal{P}(x_j \to x_i)$ to a single path $p \in \mathcal{P}(x_1 \to x_j \to x_i)$ and it holds that $m(p) = m(p_1) \cdot m(p_2)$. Hence,

$$w_i = \sum_{j=1}^{i-1} \sum_{p \in \mathcal{P}(x_1 \to x_j \to x_i)} c''_p m(p) .$$

As $\bigcup_{j=1}^{i-1} \mathcal{P}(x_1 \to x_j \to x_i) = \mathcal{P}(x_1 \to x_i)$, we have $w_i = \sum_{p \in \mathcal{P}(x_1 \to x_i)} c'''_p m(p)$.

From Equation (3.8) we finally obtain:

$$\begin{aligned} x_n(t) &= \sum_{i=1}^{n} \mathbf{V}(n,i) \cdot w_i \cdot e^{\lambda_i t} = \sum_{i=1}^{n} \left(\sum_{p \in \mathcal{P}(x_i \to x_n)} c_p m(p) \right) \left(\sum_{p \in \mathcal{P}(x_1 \to x_i)} c'_p m(p) \right) e^{\lambda_i t} \\ &= \sum_{i=1}^{n} \sum_{p \in \mathcal{P}(x_1 \to x_i \to x_n)} \gamma_{p,i} m(p) e^{\lambda_i t} \end{aligned}$$

We further find

$$\gamma_{p,i} = (-1)^{\ell(p)} \prod_{\text{nodes } v \neq i \text{ along } p} \frac{1}{\lambda_v - \lambda_i} .$$

Hence, the coefficients a_{ni} in Equation (3.9) are given by

$$a_{ni} = \sum_{p \in \mathcal{P}(x_1 \to x_i \to x_n)} \gamma_{p,i} m(p) . \tag{3.12}$$

This result allows us to give a general solution of Equation (3.5) without performing any time-consuming symbolic calculations: Instead we can deduce it from the interaction graph by finding the paths between the input and the output node. This enumeration of all possible paths between two nodes is a classical problem in graph theory, see e.g. (Klamt and von Kamp, 2009) for a review. It can be easily solved by performing a breadth-first or depth-first traversal starting from the input node, but also more sophisticated algorithms have been developed recently, for instance by Klamt et al. (2006).

The effective parameters of general systems with linear activation functions show a remarkable structure. Comparing our result (3.12) to the solution for the linear cascade (3.7), we see that the effective parameters decompose the hierarchical network into a superposition of all the cascades it contains. Thus, in a general network,

the influence of different nodes varies, depending on the different cascades they are part in. Having a set of control points with different overall effects on the output signal, external regulators may achieve an exquisite regulation of the output signal. In the following, we demonstrate how the explicit expressions for the effective parameters can be utilized for stabilizing parameter estimation in a given network.

3.3.3 Implications on parameter estimation

We now investigate the role of effective parameters in parameter fitting processes. Assume that in situation (S) we experimentally observe node x_n. Theoretically, this information allows us to estimate the eigenvalues λ_j and the coefficients a_{nj} in (3.9) up to permutation. In practice, however, this problem is ill-conditioned. In fact, it has been a long-standing numerical challenge for more than 200 years (O'Leary, 2004, Prony, 1795). It appears, for example, when one wants to determine the single decay rates of the radioactive substances in a mixture but is only able to observe the decay of the whole mixture. The classical approach is to determine the longest half-life by the behavior at long times as the mixture will eventually assume the slope of its most slowly decaying component. After subtracting the corresponding curve from the mixture's decay curve one can iteratively determine the remaining decay rates (Soete, 1972). More advanced algorithms have been proposed recently, which are rather time-consuming but fairly robust despite the ill-conditioned problem (Kaufmann, 2003, Nielsen, 2000a,b, Pedersen et al., 2002, Windig and Antalek, 1997).

Of course, these algorithms are still only able to determine the λ_i and a_i up to permutation. In hierarchical signaling networks, however, the eigenvalues λ_j of $\mathbf{K}$ are the negatives of the decay rates τ_j. For many biological quantities, the latter are well studied and usually vary greatly (Sharova et al., 2008, Yen et al., 2008). Therefore, at least the relation between the decay rates should be known in many cases and (3.10) becomes

$$\begin{aligned} a_{nj}\left(\mathbf{K}, \mathbf{x_0}\right) &= \hat{a}_{nj} \\ \lambda_j\left(\mathbf{K}\right) &= \hat{\lambda}_j \end{aligned} \qquad j = 1, 2, \ldots, n \;, \tag{3.13}$$

where $\hat{a}_{nj}$ and $\hat{\lambda}_j$ are best-fit estimates. As long as we do not have less nodes in the network than paths from the source to the observed node, we can solve the linear system of equations $a_{ni}(\mathbf{K}, \mathbf{x_0}) = \hat{a}_{ni}$ for the monomials corresponding to all possible cascades from x_1 to x_n. Thus the values of the system's effective parameters can be calculated easily.

The reason for the numerical difficulties in fitting sums of exponents mentioned above can easily be seen. Let us, for example, consider the cascade from Figure

3.5(A). In situation (S) this system converges to the steady state

$$\begin{aligned}\overline{\mathbf{x}} &= (\overline{x}_i)_{i=1}^n \begin{pmatrix} x_{0,1} \\ -(k_1\overline{x}_i)/\lambda_2 \\ \vdots \\ -(k_{n-1}\overline{x}_{n-1})/\lambda_n \end{pmatrix} \\ &= x_{0,1} \begin{pmatrix} 1 \\ -k_1/\lambda_2 \\ \vdots \\ (-1)^{(n-1)}(k_1k_2\cdots k_{n-1})/(\lambda_2\lambda_3\cdots\lambda_n) \end{pmatrix},\end{aligned}$$

and it follows that for $t \gg 0$ we can approximate

$$x_n(t) \approx (-1)^{(n-1)} x_{0,1} \frac{k_1k_2\cdots k_{n-1}}{\lambda_2\lambda_3\cdots\lambda_n} .$$

Now, if the majority of the measurements lie in the region $t \gg 0$, i.e. in the region where the system is near its steady state, most optimization algorithms will in fact only be able to determine the product $\lambda_2\lambda_3\cdots\lambda_n$.

In fact, for the previous steady-state analysis, we do not have to impose any restrictions on the activation functions P_i. The steady-state equations of model (3.1) read

$$P_i(\overline{x}_j \mid j \in \mathcal{I}_i) - \tau_i\overline{x}_i \stackrel{!}{=} 0 .$$

These equations can easily be solved for $\overline{x}_i$, yielding the steady-state concentration of each node in terms of the steady-state concentrations of its inputs. As G is a hierarchical graph this iteratively allows to state the $\overline{x}_i$ in terms of $\overline{x}_1$.

3.3.4 Application to a feed-forward loop motif

We conclude this Section by applying the presented method to an illustrative example. We consider again the feed-forward loop motif shown in Figure 3.5(B), which we model by

$$\begin{aligned}\dot{x}_1 &= -\tau_1x_1 \\ \dot{x}_2 &= k_1x_1 - \tau_2x_2 \\ \dot{x}_3 &= k_2x_1 + k_3x_2 - \tau_3x_3 ,\end{aligned} \tag{3.14}$$

where $\tau = (2, 1, 0.1)$ and $k = (0.2, 0.5, 0.3)$ are chosen. Assume that in situation (S) we stimulate x_1 by an impulse signal of $a_1 = 1$ and experimentally observe $x_3(t)$. According to the graphical solution method developed in Section 3.3.2, we then have

$$x_3(t) = a_1e^{\lambda_1t} + a_2e^{\lambda_2t} + a_3e^{\lambda_3t}$$

Table 3.1: Numerical results of the fit of the exponential sum in (3.3.4) using Nielsen's algorithm (Nielsen, 2000a,b).

coefficients $\hat{a}_i$	exponents $\hat{\lambda}_i$
0.2736	-0.0994
-0.0687	-1.0643
-0.2048	-2.0782

with

$$\begin{aligned} a_1 &= -\frac{k_2}{\lambda_3-\lambda_1}+\frac{k_1k_3}{(\lambda_2-\lambda_1)(\lambda_3-\lambda_1)} & \lambda_1 &= -\tau_1 \\ a_2 &= \frac{k_1k_3}{(\lambda_1-\lambda_2)(\lambda_3-\lambda_2)} & \lambda_2 &= -\tau_2 \\ a_3 &= -\frac{k_2}{\lambda_2-\lambda_3}+\frac{k_1k_3}{(\lambda_1-\lambda_3)(\lambda_2-\lambda_3)} & \lambda_3 &= -\tau_3\,. \end{aligned} \tag{3.15}$$

Let us consider the condition for the effective parameter (3.10). We see that up to permutation each parameter has an effect that is discernible from the dynamics of x_3 with the only exception of the parameters k_1 and k_3 which only appear as a product in the expressions from (3.15). As we expect, we can identify the two monomials corresponding to the two possible paths from x_1 to x_3.

We now use the algorithm developed by Nielsen (2000a,b) to fit the a_i and λ_i, $i = 1, 2, 3$, to the time-course of x_3, cf. Table 3.1. Already we see that the algorithm is well able to fit the decay rates up to permutation. Different initial guesses produce comparable results.

Assuming that we can assign the exponents to their estimated values, (3.13) becomes

$$\begin{aligned} -\frac{k_2}{\lambda_3-\lambda_1}+\frac{k_1k_3}{(\lambda_2-\lambda_1)(\lambda_3-\lambda_1)} &= -0.2048 & -\tau_1 &= -2.0782 \\ \frac{k_1k_3}{(\lambda_1-\lambda_2)(\lambda_3-\lambda_2)} &= -0.0687 & -\tau_2 &= -1.0643 \\ -\frac{k_2}{\lambda_2-\lambda_3}+\frac{k_1k_3}{(\lambda_1-\lambda_3)(\lambda_2-\lambda_3)} &= 0.2736 & -\tau_3 &= -0.0994\,. \end{aligned}$$

Solving this for k_1k_3 and k_2 yields estimates $\widehat{k_1k_3} = 0.0672$ and $\hat{k}_2 = 0.4717$, which agree well with the true values $k_1k_3 = 0.06$ and $k_2 = 0.5$. So, all parameters can be estimated up to the indeterminacy $k_1k_3 = \widehat{k_1k_3}$.

3.4 Domain of applicability

We presented our theoretical results within a semi-quantitative framework, located in between qualitative (phenomenological) and quantitative (mechanistic) models. We started with the interaction graph describing the qualitative behavior of the system. The chosen activation functions then introduced kinetic information with only one parameter per interaction. Commonly, a fully mechanistic description of biological systems is hampered by the low number of available experimental observations which do not allow estimation of all necessary model parameters. Thus the domain of applicability for our models are systems for which quantitative data are accessible, but insufficient to determine a fully quantitative model. We have demonstrated how in this situation maximal information about kinetic parameters can be extracted even from scarce data.

We have shown that both our methods scale well and can — from a theoretical perspective — be readily applied to large systems. However, growing network size might limit the interpretability of the obtained effective parameters. Using small examples, we could substantiate our methods' applicability for the understanding of smaller systems. The presented results are valuable, since real-world biological systems are modularly built up of such small subnetworks (del Vecchio et al., 2008, Ravasz et al., 2002).

The main restriction that may hamper applicability of our approaches to real-world systems is the assumption of hierarchical systems. Legewie et al. (2008) show that usually signaling pathways can be divided into a fast (hierarchical) signal reception and transduction part and a negative feedback operating on slower time-scale. Hence, the approximation of networks by acyclic subgraphs is often feasible and valid in the time domain of initial responses. However, for a given system there are typically several possibilities how to cut the feedback structure. This problem has been regularly addressed in the context of logical steady-state analysis. For instance, Klamt et al. (2006) develop the concept of minimal intervention sets which in many examples has been successfully employed to cut the feedback structure in models of signaling networks (Franke et al., 2008, Saez-Rodriguez et al., 2007).

We argue that algebraic expressions for effective parameters are valuable for several reasons. First, an interpretation of these expressions yields insights into the biological properties of the system under study. From a theoretical perspective, the presented results show the connection between the interaction graphs of reaction networks and the according ODE systems. They provide graphical solutions for differential equations. Finally, algebraic expressions for parameter indeterminacies can be used to stabilize and speed up parameter fitting processes. For example,

the whole theory of optimization on manifolds is now applicable (Absil et al., 2008, Edelman et al., 1998, Gruber and Theis, 2006).

3.5 Conclusions and outlook

Many biological systems, such as gene regulatory networks or signaling pathways, can be thought of as information flow networks. In this contribution we asked the question of how the kinetic parameters determine the information flow in the initial response of biological systems. To this end, we algebraically described the effective parameters governing signal transduction in two modeling frameworks.

When modeling switch-like molecular interactions by (idealized) Heaviside step functions, these effective parameters can be constructed recursively from the interaction graph. We presented an algorithm for this task that returns the effective parameters visualized as a directed tree. This allows to assess the effect of single parameters on the system's global behavior. These effects are not absolute but rather depend on the values of the other parameters as well as on the network structure. We also outlined that our results may partially generalize to the case of non-linear Hill activation functions, which offers an interesting direction for further studies.

Subsequently, we studied models with linear activation functions. In this situation, we were able to transform the symbolic solution of the ODE describing the time-course of the observed node into the well-known graph-theoretic problem of path enumeration. This allows to easily find the algebraic structure of the effective parameters. We showed that the networks of this type can be interpreted as superposition of the cascades they contain. In this context we also addressed the problem of numerically ill-conditioned parameter fitting problems and showed how robust algorithms can be applied. The performance of these algorithms still needs to be thoroughly compared to conventional optimization strategies.

In conclusion, this Chapter took a novel approach in the study of hierarchical networks and analyzed the role of parameters therein when taking the big step from network topology to network dynamics. We outlined general strategies as well as presented first results, opening up new avenues of research on these complex systems.

4 Knowledge-based matrix factorization with an application to microarray data analysis

With the availability of high-throughput 'omics' data, coupled with increasingly detailed knowledge about underlying regulatory processes, more and more methods from statistics and signal processing are applied in the field of bioinformatics (Tarca et al., 2007). Direct application of such methods to biological and biomedical data sets however is essentially complicated by three issues, namely

(i) the large-dimensionality of observed variables (e.g. transcripts or metabolites),

(ii) the small number of independent experiments and

(iii) the necessity to take into account prior information in the form of e.g. interaction networks or chemical reactions.

While *(i)* may be tackled by targeted analysis, feature selection or efficient dimension reduction methods, the issue of low number of samples may hinder the transfer of methods. Quantitative data from experiments are often classified as 'small-n-large-p' problems and algorithms that are currently being developed are tailored for such kind of data. Detailed prior information is in general best handled by Bayesian methods (Gelman et al., 2004, Wilkinson, 2007), which are however not straightforward to formulate in small-n-large-p problems.

In this Chapter, we focus on the unsupervised extraction of overlapping clusters in data sets exhibiting properties *(i-iii)*. If applied to gene expression profiles acquired by microarrays or metabolic profiles from mass spectrometry, we can interpret these clusters as jointly acting species (cellular processes). While partitioned clustering based on k-means (Tavazoie et al., 1999) or hierarchical clustering (Eisen et al., 1998) has been successful in some domains and is often the initial tool of choice for data grouping, overlapping clusters are better described by fuzzy techniques (Gasch and Eisen, 2002) or linear latent variable models (Kerr and Churchill, 2001). We focus on the latter, which can be solved by matrix factorization algorithms. Constraining the factorization by decorrelation, statistical independence or non-negativity leads to PCA, ICA and NMF, respectively, as discussed in the Preliminaries 1.1.2. Although these methods are successfully applied in bioinformatics (Blöchl et al., 2010,

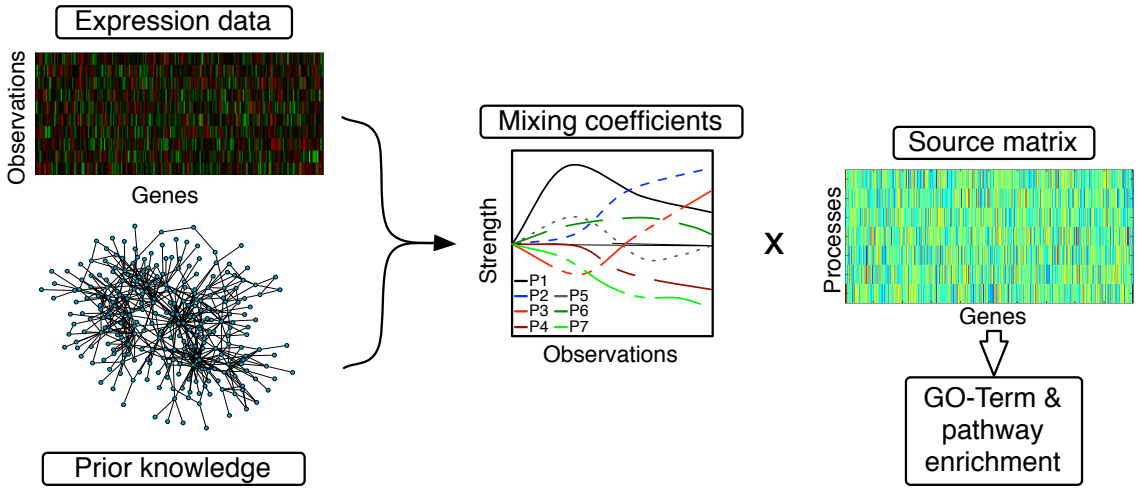


Figure 4.1: In cells, various biological processes are taking place simultaneously. Each of these processes has its own characteristic gene expression pattern, but different processes may overlap. A cell's total gene expression is then the sum of the expression patterns of all active processes, weighted by their current activation level. Matrix factorization techniques decompose observed expression data into underlying sources and their mixing coefficients. The idea behind the GraDe algorithm is to combine a matrix factorization approach with prior knowledge in form of an underlying regulatory network. Analyzing time-course microarray data, we interpret these sources as the biological processes and the mixing coefficients as their time-dependent activities. We further filter process-related genes by taking the genes with the strongest contribution in each source. Finally, we test for enrichment of cellular processes (GO) and biological pathways (KEGG).

Liebermeister, 2002, Schachtner et al., 2008), they partially run into issues *(i-iii)* as described above. In particular, it is not clear how to include prior knowledge, which has been a quite successful strategy in other contexts (Nachman et al., 2004, Subramanian et al., 2005).

A first step towards this direction is Network Component Analysis (NCA) (Boscolo et al., 2005, Liao et al., 2003). It integrates prior knowledge in form of a multiple-input motif to uncover hidden regulatory signals from the outputs of networked systems. Hence, it focuses on the estimation of single gene's expression profiles, not in a linear decomposition of a data set into overlapping clusters. NCA poses strict assumptions on the topology of the predefined network, which makes it hardly applicable to mammalian high-throughput 'omics' data. Moreover, feedbacks from the regulated species back to the regulators are treated only as 'closed-loops', without explicitly modeling the feedback structure.

To overcome these constraints, this Chapter provides a novel framework for the linear decomposition of data sets into expression profiles. It develops a new matrix factorization method that is computationally efficient *(i)*, able to deal with the low

number of samples *(ii)* and includes as much prior information as possible *(iii)*. In order to achieve robust estimation, we use delayed correlations instead of higher-order statistics. We have discussed in Chapter 1 that this strategy is advantageous for two reasons: such methods use more information from the data without overfitting it, and they are second order and therefore computationally efficient.

However, delayed correlations cannot be computed in the case of independent and identically-distributed random variables such as in microarray samples. While time-resolved experiments may provide correlations, the number of temporal samples are commonly too small (<10) for the estimation of time-delayed correlations. Hence, we instead pose factorization conditions along the set of genes or other biological variables. We link these variables using prior knowledge, e.g. in the form of a large-scale transcription factor or protein-protein interaction (PPI) network, metabolic pathways or via explicitly given detailed models. Using this information enables us to define a graph-decorrelation algorithm that combines prior knowledge with source-separation techniques. Hereby, the extracted sources group the samples' expression that can be explained by the underlying regulatory network, e.g. different responses of a cell to an external stimulus. Figure 4.1 visualizes the concept.

4.1 Source separation based on a graph model

In signal processing, various matrix factorization techniques have been developed that employ intrinsic properties of data to decompose them into underlying sources. Examples are the classic algorithms AMUSE (Tong et al., 1991) or SOBI (Belouchrani et al., 1997) which we have introduced in Chapter 1. These methods are based on *delayed correlations* that we have defined for data having a temporal or spatial structure.

However, as mentioned above, the data sets we obtain in biological experiments rarely imply a natural order that allows to define a generic kind of delayed correlation. We therefore generalize this concept by introducing prior knowledge that links samples along a pre-defined underlying network. This network may be large-scale, but can be also an explicitly given small-scale process. Moreover, integrated information may be of qualitative (e.g. interaction) as well as quantitative nature (e.g. interaction strength, reaction rates).

4.1.1 Graph-delayed correlation

We encode prior knowledge in a directed, weighted graph $G := (V, E, w)$ defined on vertices $V \in \{1, \ldots, l\}$ corresponding to our samples. The edges E are weighted with weights $w : E \to \mathbb{R}$. These are collected in the *weight matrix* $\mathbf{W} \in \mathbb{R}^{l \times l}$, where

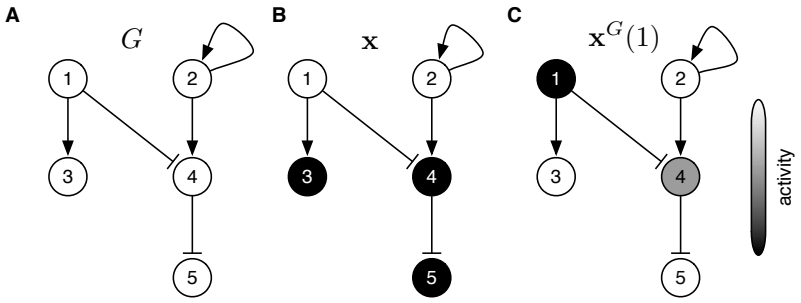


Figure 4.2: Illustration of the G-shift in the unweighted graph G shown in (A). We start with an initial node activity $\mathbf{x}$ depicted in (B). Then, the graph is used as a propagator for the time evolution of this pattern: after one positive shift we achieve the activity pattern $x^G(1)$ in (C).

w_{ij} specifies the weight of edge $i \to j$. Note that our weights may be negative, and G may contain self-loops. Recall that for $i \in V$, we denote the set of its inputs by $\mathfrak{I}_i := \{j|(j,i) \in E\}$.

We want to shift our samples along the partial ordering implied by the graph G. However, it is important to note that besides the network topology also the logical wiring chosen to link different parents of a node is crucial. In the following, we will focus on a linear superposition of all inputs, which seems appropriate for many applications. In Section 4.1.5 we will show that this choice can be directly derived from first principles in the case of gene regulation, which justifies the later interpretation of a microarray experiment. We would like to point out that any other wiring which is better adapted to the type of data under consideration can be used instead.

Hence, using the weight matrix $\mathbf{W}$ as a propagator for an activity pattern $\mathbf{x} \in \mathbb{R}^l$ of our samples we define the *G-shift* $\mathbf{x}^G$ of $\mathbf{x}$ as the vector with components

$$x_i^G := \sum_{j \in \mathfrak{I}_i} \mathbf{W}_{ji}\, x_j\,. \tag{4.1}$$

Recursively, we define any positive shift $\mathbf{x}^G(\tau)$. This is illustrated in Figure 4.2. For negative shifts we replace predecessors $\mathfrak{I}$ by successors, which formally corresponds to a transposition of the weight matrix $\mathbf{W}$. Using the convention of trivial weights for non-existing edges of G, we can extend the above sum to all vertices. Gathering all available observations (rows) into a data matrix $\mathbf{X}$ we obtain the simple, convenient formulation of a G-shifted data set

$$\mathbf{X}^G(\tau) = \begin{cases} \mathbf{X}\mathbf{W}^\tau & \tau \geq 0 \\ \mathbf{X}(\mathbf{W}^\top)^\tau & \tau < 0 \end{cases}\,. \tag{4.2}$$

After mean removal, we may assume that each row of $\mathbf{X}$ is centered. Then, in analogy to the unbiased estimator for cross-correlations in Equation (1.9), we define the *graph-delayed (cross)-correlation* matrix of $\mathbf{X}$ as

$$\mathbf{C}_{\mathbf{X}}^{G}(\tau) := \frac{1}{l-1}\,\mathbf{X}^{G}(\tau)\mathbf{X}^{\top}\,. \tag{4.3}$$

This definition is independent of the choice of logic, generalizing the time-delayed correlation in Equation (1.10) to correlations calculated with a delay based on the partial ordering implied by the graph G. With our choice of logic (4.2) we have

$$\mathbf{C}_{\mathbf{X}}^{G}(\tau) = \frac{1}{l-1}(\mathbf{X}\mathbf{W}^{\tau}\mathbf{X}^{\top})\,. \tag{4.4}$$

Note that with the chosen logic our definition includes the standard time-delayed correlation by shifting along the line graph $1 \to 2 \to \ldots \to l-1 \to l$.

The graph-delayed correlation is only symmetric if the used graph shows this feature which is, for instance in regulatory networks, rarely the case. For our following derivations, a symmetric generalized correlation measure however will turn out to be very convenient. In the remainder of this work, we will therefore use the *symmetrized graph-delayed correlation*

$$\bar{\mathbf{C}}_{\mathbf{X}}^{G}(\tau) = \frac{1}{2}\left(\mathbf{C}_{\mathbf{X}}^{G}(\tau) + \mathbf{C}_{\mathbf{X}}^{G}(\tau)^{\top}\right)\,. \tag{4.5}$$

Enforcing the symmetry property is a strategy has been often applied in the case of temporally or spatially delayed correlations, where it also stabilizes the estimation of the cross-correlations from data (Theis et al., 2004).

From (4.2) we see that $\mathbf{C}_{\mathbf{X}}^{G}(\tau)^{\top} = \mathbf{C}_{\mathbf{X}}^{G}(-\tau)$, and we therefore focus on the mean correlation shifting both in positive and negative direction.

4.1.2 The factorization model

In the following, we focus on the linear mixing model for a data matrix $\mathbf{X} \in \mathbb{R}^{m\times l}$. Now, extending the ideal situation discussed in the Preliminaries, we additionally introduce an additive noise term ϵ. Equation (1.8) then becomes

$$\mathbf{X} = \mathbf{A}\mathbf{S} + \epsilon\,. \tag{4.6}$$

Here, the matrix of source contributions $\mathbf{A} \in \mathbb{R}^{m\times n}\,(m \geq n)$ is assumed to have full column rank. The sources $\mathbf{S} \in \mathbb{R}^{n\times l}$ are uncorrelated, zero-mean stationary processes with non-singular covariance matrix. The noise term $\epsilon \in \mathbb{R}^{m\times l}$ is modeled by a stationary, white zero-mean process with variance σ^2. We assume white unperturbed data $\tilde{\mathbf{X}} := \mathbf{A}\mathbf{S}$ (possibly after whitening transformation). In other words, we

interpret each row of $\mathbf{X}$ as linear mixture of the n sources (rows of $\mathbf{S}$), weighted by mixing coefficients stored in $\mathbf{A}$. Without additional restrictions, this general linear blind source-separation problem is underdetermined.

Here, we assume that the sources have vanishing graph-delayed cross-correlation with respect to some given graph G and all shifts τ. Formally, this means that $\bar{\mathbf{C}}^G_{\mathbf{S}}(\tau)$ is diagonal. We observe

$$\bar{\mathbf{C}}^G_{\mathbf{X}}(\tau) = \begin{cases} \mathbf{A}\bar{\mathbf{C}}^G_{\mathbf{S}}(\tau)\mathbf{A}^\top + \sigma^2\mathbf{I}, & \tau = 0 \\ \mathbf{A}\bar{\mathbf{C}}^G_{\mathbf{S}}(\tau)\mathbf{A}^\top & \tau \neq 0 \end{cases} . \tag{4.7}$$

Clearly, a full identification of $\mathbf{A}$ and $\mathbf{S}$ is not possible, because Equation (4.6) defines them only up to scaling and permutation of columns: Multiplication of a source by a constant scalar can be compensated by dividing the corresponding row of the mixing matrix by the scalar. Similarly, the factorization implies no natural order of the sources. We can take advantage of the scaling indeterminacy by requiring our sources to have unit variance, i.e. $\bar{\mathbf{C}}^G_{\mathbf{S}}(0) = \mathbf{I}$. With this, as we assumed white data $\tilde{\mathbf{X}}$, we see that $\mathbf{A}\mathbf{A}^\top = \mathbf{I}$, i.e. $\mathbf{A}$ is orthogonal. Thus, the factorization in Equation (4.7) represents an eigenvalue decomposition of the symmetric matrix $\bar{\mathbf{C}}^G_{\mathbf{X}}(\tau)$. If additionally we assume that $\bar{\mathbf{C}}^G_{\mathbf{S}}(\tau)$ has pairwise different eigenvalues, the spectral theorem guarantees that $\mathbf{A}$ – and with it $\mathbf{S}$ – is uniquely determined by $\mathbf{X}$ except for permutation.

However, we have to be careful, because we cannot expect $\bar{\mathbf{C}}^G_{\mathbf{X}}(\tau)$ to be of full rank. Obviously, we require more samples than obtained sources ($l \gg m$), hence in general $\operatorname{rank}(\mathbf{X}) = m$. If G contains an adequate amount of information, $\operatorname{rank}(\mathbf{W})$ is of order l and since $l \gg m$, $\operatorname{rank}(\bar{\mathbf{C}}^G_{\mathbf{X}}(\tau))$ is essentially determined by (the upper bound) m. Hence, when analyzing high-throughput biological data linked by underlying large-scale networks, we can aim at extracting as many sources as observations are available.

4.1.3 The GraDe algorithm

Equation (4.7) also gives an indication of how to solve the matrix factorization task in our setting. The first step consists of whitening the no-noise term $\tilde{\mathbf{X}} = \mathbf{A}\mathbf{S}$ of the observed mixtures $\mathbf{X}$. The whitening matrix can be estimated from $\mathbf{X}$ by diagonalization of the symmetric correlation matrix

$$\bar{\mathbf{C}}^G_{\tilde{\mathbf{X}}}(0) = \bar{\mathbf{C}}^G_{\mathbf{X}}(0) - \sigma^2\mathbf{I}\,,$$

provided that the noise variance σ^2 is known or can be reasonably estimated. If more signals than sources are observed, dimension reduction can be performed in

this step. Insignificant eigenvalues then allow the estimation of the noise variance, as described by Belouchrani et al. (1997).

Now, we may estimate the sources by simple diagonalization of a single, symmetric graph-delayed correlation matrix $\bar{\mathbf{C}}^G_{\mathbf{X}}(\tau)$. This procedure generalizes the AMUSE algorithm discussed in Section 1.1.2.2, which employs the standard time-delayed correlation, to the extended definition of graph-correlation.

The performance of AMUSE is known to be relatively sensitive to additive noise. Moreover, we already discussed that an estimation by a finite amount of samples may lead to a badly estimated time-delayed correlation matrix (Theis et al., 2004). To alleviate these problems, algorithms like SOBI or TDSEP (see Section 1.1.2.2) extend this approach and perform a *joint diagonalization* of *multiple* time-delayed correlation matrices, calculated with a set of different delays τ.

In a similar manner, we may jointly diagonalize multiple graph-delayed correlation matrices obtained from G-shifts with different lags. This approximative joint diagonalization can be achieved by a variety of methods, compare Section 1.1.2.2. We use the Jacobi-type algorithm proposed by Cardoso and Souloumiac (1995), since we later compare GraDe's performance to the classic SOBI algorithm.

Altogether, we subsume this procedure in the GraDe (graph-decorrelation) algorithm. When shifting along the line graph, GraDe with a single lag reduces to AMUSE, and GraDe with multiple shifts corresponds to SOBI.

4.1.4 Comparison with other methods

In order to evaluate GraDe's performance, we generated random mixtures of artificial G-decorrelated signals. A common way to create standard-autocorrelated signals are *moving average* (MA) models (Hyvärinen et al., 2001): For a white noise process ϵ and real coefficients $\theta_1 \dots \theta_q$, a q-th order MA model $\mathbf{x}$ is defined by

$$\mathbf{x}_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q} \,. \tag{4.8}$$

In our notation, we interpret this MA signal $\mathbf{x}$ as a weighted sum of G-shifted versions of ϵ, shifted q times along the line graph G. Therefore, for an arbitrary graph G we define a q-th order *G-MA(q) model* as

$$\mathbf{x} = \epsilon + \theta_1 \epsilon^G(1) + \theta_2 \epsilon^G(2) + \dots + \theta_q \epsilon^G(q) \,. \tag{4.9}$$

Any G-MA(q) process is equivalent to a G-MA(1) process with a modified graph.

In a first simulation, we used directed Erdös-Rényi random graphs (see Chapter 1.2.6.1) with a mean connectivity of 17.5 and random weights in $(-1, 1)$ to generate $m = 2$ G-decorrelated G-MA(1) signals with $l = 5000$ samples. These data were normalized to unit variance and mixed with a random mixing matrix. We added

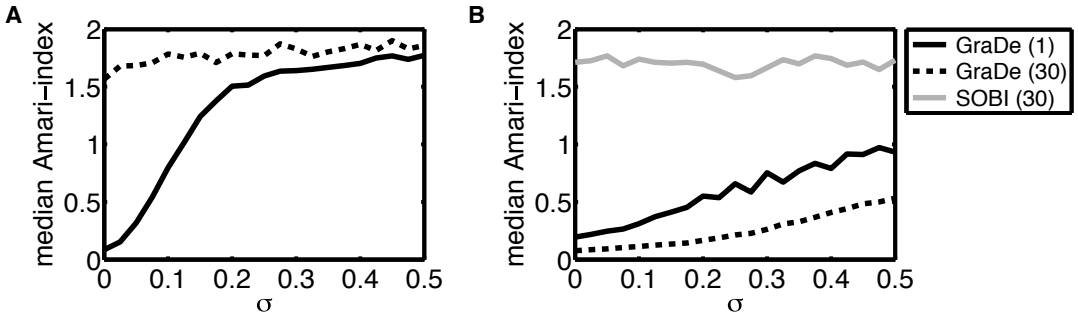


Figure 4.3: Performance on artificial data: mixtures of (A) two G-MA(1) processes with random graphs G, (B) mixtures of two G-MA(20) processes with signed line graphs. The plots show the dependence of median Amari-indices on the noise level σ over 1000 runs. We compare GraDe with one and 30 shifts, in (B) in addition SOBI with 30 shifts.

Gaussian uncorrelated noise of variable strength σ and applied GraDe (without noise estimation) with one and 30 shifts, respectively. Reconstruction quality was estimated using the Amari-index from Equation (1.18) which quantifies the deviation between the correct and the estimated mixing matrix.

From Figure 4.3(A) we see that for G-MA(1) processes GraDe with a single-shift performs well in the low-noise setting, in contrast to GraDe with multiple shifts. This is a consequence of the complex short-distance, but vanishing long-distance delayed correlation structures. When performing multiple shifts, each lag is weighted equally strong, which deteriorates the algorithm's performance.

Accordingly, as shown in Figure 4.3(B), GraDe with multiple shifts outperformed single-shift GraDe when applied to higher order G-MA processes. In this simulation, we generated G-MA(20) processes of sample size $l = 1500$ with a signed line graph, where the edges had weights ± 1 with equal probability. The unsigned line graph used by SOBI was not sufficient to reconstruct these signals in a proper way, whereas GraDe with the true graph separated them even using a single shift only. However, similar to the behavior in the standard time-delayed case, here multiple shifts dramatically enhance GraDe's robustness against additive noise.

4.1.5 G-shifts in gene regulation

As main application, we will employ GraDe to interpret an experiment on gene expression in the next Section. In this case, we have to link genes along an underlying gene regulatory network. The time-dependent expression level $x_i(t)$ of a gene i depends on multiple transcription factors. Regulatory control is thereby provided by cooperative binding of transcription factors to the promoter binding sites of a gene. To logically interconnect the set of gene regulators, we follow Snoussi and

Thomas (1993), who use the sum over all, positive or negative, regulators. The linear superposition of inputs ensures robustness to errors in the underlying graph and in addition no detailed knowledge on single interactions is necessary.

As already discussed in Section 3.1.2, a gene regulatory system can then be described by a set of ordinary differential equations (ODE) of the form:

$$\frac{\mathrm{d}x_i(t)}{\mathrm{d}t} = -\gamma_i x_i(t) + \sum_{j \in \mathcal{I}_i} f_{ji}\left(x_j(t)\right). \tag{4.10}$$

Here, the activation or inhibition function f_{ji} quantifies the influence of gene j on gene i. Again, we introduce a linear decay with rate $\gamma_i \geq 0$. By approximating the differential in (4.10) by a finite difference, we then find

$$x_i\left(t + \Delta t\right) = -\left(\gamma_i \Delta t - 1\right) x_i(t) + \sum_{j \in \mathcal{I}_i} \Delta t f_{ji}(x_j(t)). \tag{4.11}$$

For a large-scale regulatory network detailed activation or inhibition functions are unknown. However, assuming that the non-linear system operates near a steady state, the dynamics can be approximated by a linear system (Gardner et al., 2003). Then, we can formally identify the decay term with negative auto-regulation. Summarizing all parameters in Equation (4.11) into a single weight matrix $\mathbf{W}$, which also depends on Δt, we obtain

$$x_i(t + \Delta t) = \sum_{j \in \mathcal{I}_i \cup i} w_{ji} x_j(t). \tag{4.12}$$

Thus, in the steady state limit, general gene regulation on a time-scale Δt equals the G-shift from Equation (4.1) with an appropriate network $\mathbf{W}$. This theoretical consideration further corroborates the validity of the assumptions made by our approach. However, we see that Δt has to be chosen carefully when performing experiments to be analyzed by GraDe.

Information on gene regulation is available on different levels of granularity. Whenever dealing with large-scale biological data, regulatory control is commonly available in form of regulatory interactions only. In this case, the weight matrix $\mathbf{W}$ contains only the character of regulation, where positive interaction is denoted as 1 and negative as -1, respectively.

4.1.6 Illustrative examples

In order to illustrate GraDe's applicability to gene expression data, we analyze two examples. The first one mimics a time-course experiment: For the bifan structure shown in Figure 4.4(A), we assume to have six samples from the time-courses of

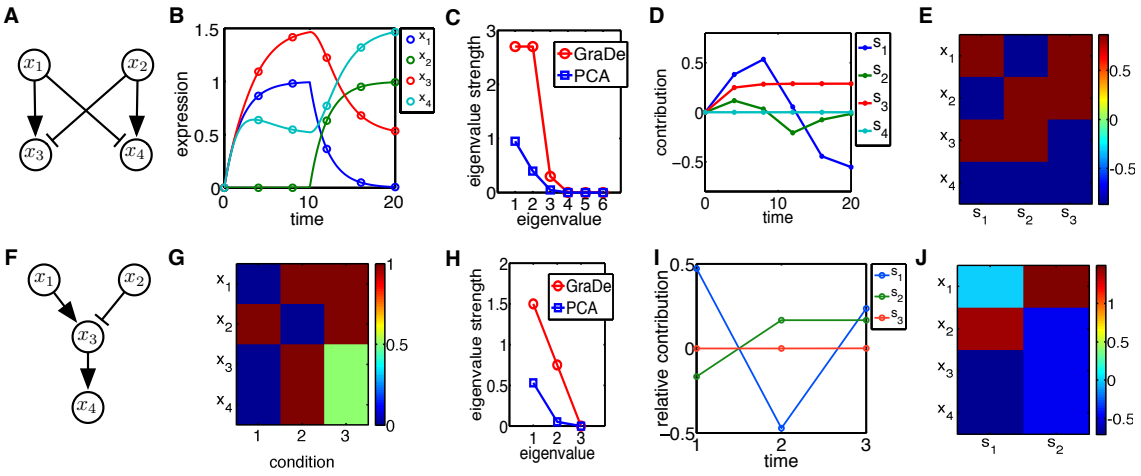


Figure 4.4: Toy examples: For the bifan motif (A) we take 6 samples (dots) from the simulated time-courses in (B) and apply GraDe. For simplicity we choose all thresholds equal to 1 and all Hill-exponents equal to 3, lifetimes are set to 1/2. (C) compares the eigenvalues of the two decompositions. In (D) we plot the time-courses of the extracted sources $s_1 ... s_6$, the curves are the columns of the mixing matrix. From (C) we see that only the first three sources are relevant, which are visualized as heat-map (E). For our second example (F) we assume to know expressions in different conditions as shown in (G). The factorization by GraDe is visualized in (H) to (J).

expression levels depicted in Figure 4.4(B). For data generation, the system is simulated by ordinary differential equations as introduced in Equation (4.10) where we model interactions by sigmoidal Hill functions. In this case, one input x_1 is active until time-point 10, when it is turned off and instead production of x_2 is switched on. Consequently, x_3 peaks at time 10, but also x_4 shows an early activation due to low expression of its inhibitor. Applying GraDe (with the known bifan topology, but without access to the underlying ODE system), we find that three sources are sufficient to explain the data. From the extracted sources and their time-courses (shown in Figure 4.4(E) and (D)) we see that the strongest source s_1 represents the externally controlled inputs and the network topology: the source couples x_1 and x_3, and in opposite direction x_2 and x_4. Source s_2 has the lowest contribution to the total expression values and is needed for fine-tuning the combined dynamics. Consequently, it is active at time-points 2 and 4, i.e. immediately after the switching operations. Source s_3 again reflects the crossover inhibitions, accordingly its time-course is flat.

Our algorithm is not limited to time-course data, it can just as well be applied to samples obtained in different conditions. For instance, let us assume that in the funnel structure in Figure 4.4(F) we know the expression values for the three different input conditions in Figure 4.4(G). Source s_1 again reflects the network

topology, while s_2 allows construction of the last condition. As we expect, GraDe recovers the two independent inputs. In contrast, PCA analysis only gives conditions 2 and 3 as sources. However, these two conditions are strongly negative G-cross-correlated, as one can see by shifting condition 2 by one step.

4.2 A microarray experiment on IL-6 mediated responses in primary hepatocytes

The regulation of gene expression is essential to proper cell functioning. The first step of gene expression is mRNA transcription. Here, a copy of a gene from the DNA to messenger RNA (mRNA) is made, encoding a chemical "blueprint" for a protein product. Microarrays are the state-of-the-art technology for the genome-wide measurement of these transcript levels. They are known to be quite noisy, and the still high costs keep the number of replicates small. This makes gene expression analysis a particular challenge for machine learning. Matrix factorization techniques are currently explored as unsupervised approaches to such data (Schachtner et al., 2008). The ability to identify patterns of genes whose linear combination explains the observed expression values overcomes the limitations inherent to the widely employed single gene statistics. As we illustrated in Figure 4.1, the extracted gene expression sources (GES) are interpreted as distinct biological processes, which are active on a level quantified in the mixing matrix. Applying GraDe, we require that biological processes that can be explained by the underlying network are not split up between different GES.

4.2.1 IL-6 stimulated mouse hepatocytes

In liver, the cytokine interleukin *IL-6* mediates two major responses. First, it induces hepatocytes to produce acute phase proteins upon infection-associated inflammation. These proteins include complement factors to destroy or inhibit growth of microbes. In addition, *IL-6* promotes liver regeneration and protects against liver injury (Fausto, 2000). *IL-6* regulates several cellular processes such as proliferation, differentiation and the synthesis of acute phase proteins (Gauldie et al., 1987). Upon binding to its cell surface receptor, *IL-6* activates the receptor associated Janus tyrosine kinase (*JAK1*) signal transducer and activator of transcription (*STAT3*) signal transduction pathway. The latent transcription factor *STAT3* is translocated to the nucleus after activation and subsequently alters gene expression. In a collaboration with the group of Ursula Klingmüller at DKFZ Heidelberg, we tried to identify the biological responses to *IL-6* in a time-resolved manner. To this end, they stimulated primary mouse hepatocytes with 1 nM *IL-6* up to 4 hours and analyzed the changes

in gene expression by microarray analysis.

RNA samples from primary mouse hepatocytes were thereby assessed with the Bioanalyzer 2100 (Agilent) to ensure that 28S/18S rRNA ratios were in the range of 1.5 to 2.0 and concentrations were comparable between samples. GeneChip Mouse Genome 430 2.0 Arrays (Affymetrix) were used to analyze changes in mRNA concentration. For each time point, 4 μg of total RNA were used for the hybridization procedure using the One-Cycle Target Labeling Kit (Affymetrix). Fluorescence intensities were acquired with the GeneChip Scanner 3000 and the GCOS software (Affymetrix). GeneChip Mouse Genome 430 2.0 Arrays (Affymetrix) were used in the analysis comprising stimulations with 1 nM *IL-6* for 1 h, 2 h, 4 h and an unstimulated control (0 h) each performed in triplicates. As a probe level model (PLM) for microarray data an additive-multiplicative error model was used. Data processing was performed using the Limma toolbox (Smyth et al., 2005) provided by Bioconductor (Gentleman et al., 2004). The RMA approach was used for normalization and background correction. Probe sets were filtered out by the genefilter package. A gene was considered as expressed if the signal was above 6.64 ($\log_2$ data) for at least one time point. Finally, we obtained a data set of 5709 genes. Significantly regulated genes compared to time point 0 h were determined by using the LIMMA (Linear Models for Microarray Data) method (Smyth, 2004). A gene was determined as significantly regulated if the p-value was < 0.05 after multiple testing correction by the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995). Raw data are available at GEO with accession number GSE21031.

4.2.2 Time-dependent biological processes upon IL-6 stimulation

In a first approach, we extracted all genes that were significantly regulated compared to time point 0 h. In total, we obtained 121 genes and applied k-means clustering to detect groups within this set. Based on this approach, we could not identify any time-resolved responses upon *IL-6* stimulation. Due to the small number of significantly regulated genes, we decided to employ a genome-wide approach using GraDe to resolve the cellular responses upon *IL-6* in more detail.

4.2.2.1 Application of GraDe

In order to link the genes along an underlying gene regulatory network we used the TRANSPATH database (Krull et al., 2006). This database provides detailed knowledge of intracellular signaling information based on changes in transcription factor activity. We filtered all genes that showed an expression during the time-course and searched for direct gene or protein interactions using the terms: transactivation, increase of abundance, expression, activation, DNA binding, increase of DNA binding,

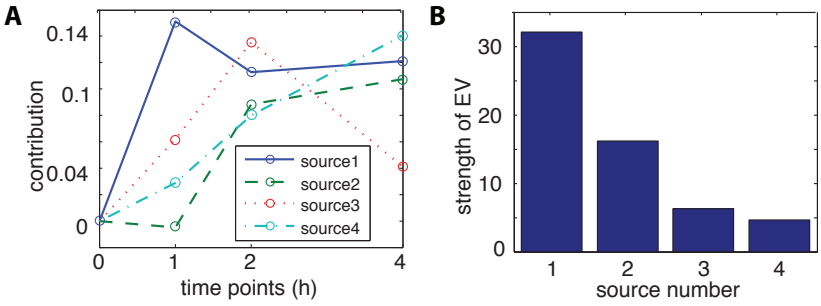


Figure 4.5: The decomposition of the time-course microarray experiment on *IL-6* stimulated hepatocytes with GraDe. As underlying network we used interactions from the TRANSPATH database. (A) shows the time-courses of the four extracted sources, centered to time point 0 h. The x-axis shows the measured time-points and the y-axis the contribution of the mixing matrix. In (B), we plot the strength of the eigenvalues (EV) of the resulting sources. All four extracted sources have significant contributions.

transrepression, decrease of abundance, decrease of DNA binding, and inhibition.

We applied GraDe with this network and obtained four graph-decorrelated gene expression sources (GES), which we labeled from 1 to 4 according to their decreasing eigenvalues (Figure 4.5(B)). We see that dimension reduction and with it noise level estimation were not possible in our case. The estimated mixing matrix is shown in Figure 4.5(A). The matrix of source contributions contains positive and negative components. We partitioned a source into submodes that contain either the negative signals or the positive signals, respectively. We selected all genes in the positive submodes by choosing a threshold ≥ 2 as well as all genes in the negative submodes with a threshold $\leq$ -2, respectively.

These sets were subsequently used for Gene Ontology (GO) (Camon et al., 2004) term enrichment analysis, which was carried out with the R package GOstats (Gentleman et al., 2004). We also performed a pathway enrichment analysis, where we tested for all non-metabolic pathways from the manually curated Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa et al., 2008). In both enrichment analyses we used the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995) to correct for multiple testing and called an enrichment significant if the p-value was less than 0.05.

4.2.2.2 Analysis of the obtained gene expression sources

Differentially expressed genes within GES 1 display an immediate strong increase in expression following *IL-6* stimulation. After peaking at one hour, expression decreases to elevated levels compared to untreated samples. Significantly enriched

Table 4.1: Summary of the main biological processes in hepatocytes regulated as response to *IL-6*. Mode indicates genes with significant positive (≥ 2) or negative (≤ -2) contribution to the source.

Source	Mode	Biological process
1	positive	(external) stimulus, inflammatory response
	negative	(fructose) metabolic process
2	positive	early cell cycle and division
	negative	metabolic process, apoptosis
3	positive	late cell cycle and division
	negative	-
4	positive	translation, coagulation
	negative	(protein) metabolic process

GO-Terms within this GES correspond to responses triggered by external stimuli and inflammation (see Table 4.1). In liver, upon infection- or injury-associated inflammation *IL-6* mediates production of acute phase proteins (APP) by hepatocytes as represented by the GO-Term "(acute) inflammatory response" (e.g. *Saa4*, *Fgg*, *Pai1*). The GO-Term "(external) stimulus" includes genes of the JAK-STAT signaling pathway like *STAT3* as well as several genes encoding for signaling components such as *Hamp*, *Cepbd* and *Osmr*. These entities represent regulatory processes like negative feedbacks as well as secondary signaling events. Genes with negative contribution in GES 1 were associated with metabolic processes like "L-serine biosynthesis" or "fructose metabolic processes". This is in line with the function of *IL-6* as a priming factor, mediating the conversion of quiescent hepatocytes from G0 to G1 phase of the cell cycle during liver regeneration (Fausto, 2000). It can be argued that down-regulation of genes associated with metabolic processes is due to the transformation of differentiated metabolically active hepatocytes into proliferative cells. The down-regulated metabolic functions at least partially take place in mitochondria. Accordingly, parts of the glycolysis pathway were down-regulated in primary hepatocytes.

GES 2 shows a slight decrease after stimulation followed by a late-phase increase in expression. We identify several biological processes associated with "cell cycle and division" within this GES. A representative gene of GES 2 is the cell cycle inhibitor *Cdkn1b*. Its reduction of expression corresponds to the induction of cell cycle progression and in particular to the transfer from G0 to G1. These characteristics are further supported by the negative contribution of *Cdkn1b* in GES 3. Analyzing genes with a positive contribution in GES 2 only, we found, in addition to involve-

ment in early cell cycle events, genes showing an association with (programmed) cell death and apoptosis. It was already indicated that *IL-6* promotes liver regeneration and protects against liver injury by inducing anti-apoptotic and survival genes (Fausto, 2000, Streetx et al., 2000). GO-Terms corresponding to genes found in GES 2 having a negative contribution are more heterogeneous. Within the top GO-Terms we identified several biological functions associated with the *IL-6* stimulus. Based on the induction of the acute inflammatory response, coagulation factors were activated. Moreover, several genes associated with gene translation were found. In addition, genes associated with metabolic processes are represented by this GES.

The time course behavior of GES 3 shows a delayed activation subsequent to stimulation with *IL-6*. We identified several GO-Terms associated with "cell cycle" and "cell division" similar to GES 2. However, GES 3 includes mainly genes related to late events in the cell cycle, i.e. during G2 and M phase (e.g. *Gmnn*, *Mcm2*, *Plk2*). *Wee1* as a main regulator of *Cdc2* displays a negative contribution to GES 3, hence indicating *Wee1* down-regulation and subsequent progression through the G2-M check point. The *IL-6*-induced priming phase is characterized by the activation of the latent transcription factor *STAT3*. This immediate response induces the expression of early responsive genes like the transcription factor *AP-1* (Westwick et al., 1994) subsequently inducing a secondary gene response leading to transcription of cyclins *A-E*, *p53*, and the cyclin dependent kinase *P34-cdc2* (Albrecht and Hansen, 1999). Applying KEGG pathway enrichment, we found the cell cycle, with DNA replication in particular, and *p53* pathway enriched within this GES. Interestingly, *IL-6* stimulation alone is not sufficient to efficiently induce proliferation of primary mouse hepatocytes *in vitro*. Hence, despite the persistent re-organization of the induced gene expression profile and the induction of early cell cycle players such as cyclin *A*, additional stimuli may be necessary to initiate a strong proliferative response of primary mouse hepatocytes.

Finally, GES 4 shows the lowest eigenvalue. It has a strong increase in expression following the *IL-6* stimulus. GO-Term enrichment reveals several biological processes found in GES 1-3 like coagulation, translation, acute phase, and response of the stimulus. Genes having a negative contribution in GES 4, indicating a decrease in expression after the stimulus, are again associated with metabolic processes. Both, GES 3 and 4 imply that hepatocytes stimulated with *IL-6* show affection for division causing a down-regulation of genes associated with the metabolic processes.

4.2.3 Validation of the time-dependent signals

In order to evaluate our findings we compared the outcome of GraDe with standard methods. As there is no established matrix factorization technique that incorporates

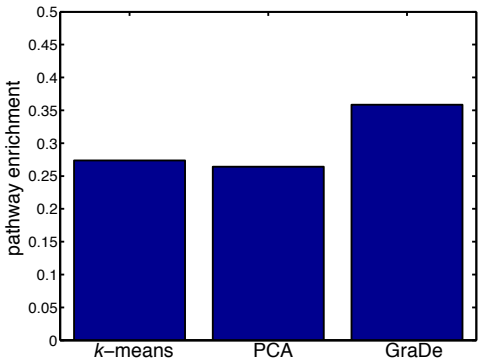


Figure 4.6: Result of the pathway enrichment analysis. For each method applied to our data set, we plotted the pathway enrichment index (PEI). This index gives the fraction of KEGG pathways found enriched in at least one submode or cluster. GraDe obtained a much higher PEI than PCA or k-means clustering. This indicates that sources obtained by GraDe map much closer to biological pathways.

prior knowledge, we employed PCA and k-means clustering as introduced in the first Chapter. Both techniques are standard solutions for the analysis of microarray data (Kaufman and Rousseew, 2005, Ringnér, 2008). For GraDe and PCA, we selected in each submode the genes having an absolute source contribution above two standard-deviations. The average number of selected genes in each submode ranges from 75 to 280. For k-means clustering, we infer eight clusters on a subset of the top 15% most variable genes to ensure that the average number of selected genes is comparable to GraDe and PCA, as proposed by Teschendorff et al. (2007).

4.2.3.1 The pathway enrichment index

To test whether our results are biologically reasonable, we first asked how well biological pathways can be mapped to the inferred submodes or clusters. To this end, we employed the *pathway enrichment index (PEI)* introduced by Teschendorff et al. (2007): For each submode or cluster we evaluated significantly enriched pathways by using a hypergeometric test. A pathway association was considered as significant if the p-value was below 0.05 after multiple testing correction using the Benjamini-Hochberg procedure. The PEI is then defined as the fraction of significant pathways mapped to at least one submode or cluster. Again, we tested for all non-metabolic KEGG pathways.

The PEI for each method is shown in Figure 4.6. We find that the PEI is higher for GraDe compared to PCA or k-means clustering indicating that GraDe maps submodes closer to biological pathways.

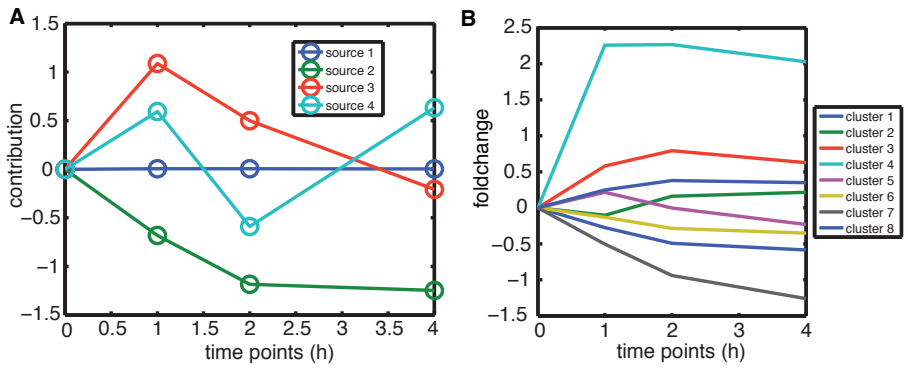


Figure 4.7: (A) illustrates the result of PCA for the time-course data of *IL-6* stimulated hepatocytes. The x-axis corresponds to the measured time-points and the y-axis gives the centered (to time point 0 h) contributions of the mixing matrix. The result of the k-means clustering is shown in (B). The x-axis shows the measured time-points and the y-axis shows the fold-change values of the centroids at that time-points.

4.2.3.2 Detailed analysis of the k-means and PCA results

In addition, we validated the time-dependent responses upon *IL-6* stimulation in more detail by searching for enriched GO-Terms (as described in the last Section). Applying PCA, we found that the first principle component (PC) contains 99% of data variance. GO-Term enrichment analysis revealed that PC 1 contains genes linked to blood coagulation and hemostasis. A second major response after *IL-6* treatment is the activation of cell cycle or cell division. We found an enrichment of these biological processes in PC 2 and PC 4. With GraDe we identified several genes that are associated with metabolic processes showing a down-regulation after stimulus. PCA covers these biological processes by two components PC 2 and PC 3, where PC 3 shows a strong increase and PC 2 a decrease of expression after the stimulus (see Figure 4.7(A)). The direct response of *IL-6* was found in PC 4, but we identified only acute inflammatory response. Moreover, PCA grouped cell cycle (negative mode) and the direct response (positive mode) into PC 4 and was not able to separate the cell cycle processes into the early (e.g. *Cdkn1b*) and late (e.g. *Mcm2*) responses after *IL-6* stimulation.

Focusing on the results of the k-means clustering, we obtained an enrichment of cell cycle processes in cluster 3. This cluster shows only a marginal increase in expression after the stimulus (see Figure 4.7(B)) and therefore does not reflect the strong activation of cell cycle found by GraDe and PCA. Genes associated with metabolic processes are grouped in cluster 5, which has a constant expression level after *IL-6* stimulus. Hence, k-means clustering failed to infer a cluster associated to the downregulation of metabolic processes upon *IL-6*. Cluster 4 shows a character-

istic time-course pattern after *IL-6* stimulation, but we were not able to reveal any significant biological processes associated to *IL-6*. Altogether, k-means clustering neither identifies the direct response upon *IL-6* nor the separation between early and late cell cycle genes.

These results show that the decomposition obtained by GraDe provided much more detailed biological insights than PCA or k-means clustering. PCA was able to identify three main biological processes upon *IL-6* stimulus. However, it failed to give a correct time-resolved pattern of these biological processes, whereas sources from GraDe reproduce the characteristic time-course behavior of the *IL-6* response. Moreover, GraDe reveals a much more structured and time-resolved result which allows assigning each source to a different main process.

4.2.4 Robustness analysis

Detailed knowledge about gene regulation is often not available and far from complete. Hence, the quality of a large-scale gene regulatory network is not perfect. In order to test the effect of network errors on the output of GraDe, we performed two robustness analyses. Starting with our TRANSPATH network, we generated randomized versions by either shuffling the network content or adding random information.

By shuffling edge information of the gene regulatory network between 0.1 and 100% of all original edges, we simulated a loss of information. In each step we shuffled 10^5 times the corresponding amount of edges using a degree-preserving rewiring (Maslov and Sneppen, 2002, Wong et al., 2008). Applying GraDe with the resulting networks we obtained new factorizations. To compare the original and new results in a quantitative way, we again used the Amari-index (Cichocki and Amari, 2002). For each step we took the 95% quantile of the random sampling and calculated a p-value by comparing this quantile to Amari-indices obtained comparing normally distributed random separating matrices and the original mixing matrix.

We obtained significantly low Amari-indices for up to 3% reshuffled edges within the gene regulatory networks (mean Amari-index = 3.83, $p = 0.034$), whereas a complete randomization of the network results in an Amari-index of 9.63 (see Figure 4.8). This shows that the quality of the regulatory network has of course a strong influence on the output of the GraDe algorithm. It is obvious that GraDe depends on the regulatory network, and replacing gene interaction through random information will lead to loss of the signals.

For a second robustness analysis we added random information to the gene regulatory network. This test is important because we expect large-scale networks extracted from literature to contain many false-positives. For each randomization

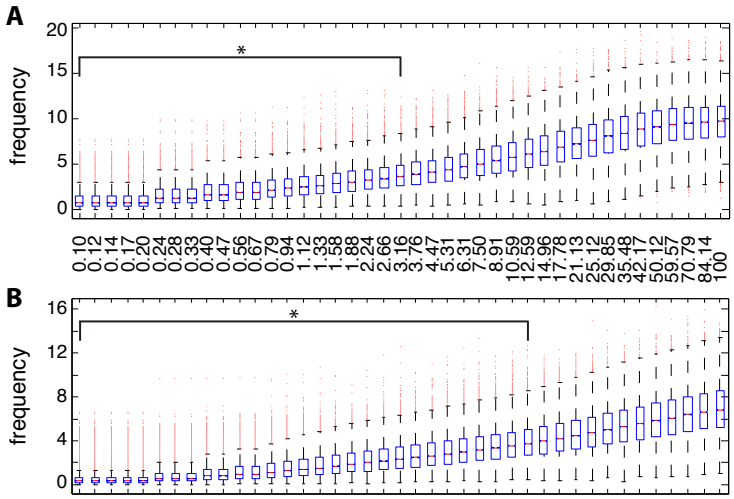


Figure 4.8: Robustness analysis: We compared the mixing matrix that we extracted with the TRANSPATH network with those obtained based on perturbed versions. For this comparison we use the Amari index. The boxplots show Amari-indices obtained with (A) a network rewiring approach and (B) when adding random information to the network. The x-axis shows the amount of information randomized (in %), the y-axis gives the obtained Amari-index. $*$ indicates significant 95% quantiles compared to a random sampling (p-value ≤ 0.05). GraDe is robust against a reasonable amount of wrong information.

step we added 10^5 times new information (edges) between 0.1 and 100% of the original network content and calculated the 95% quantile of the resulting Amari-indices. Again, a p-value was calculated by comparing each quantile with a random sampling.

Significantly low Amari-indices were obtained by adding up to 13% random information (mean Amari-index = 3.94, $p = 0.046$) to the network (see Figure 4.8). Hence, GraDe is able to detect the signals even after adding a large amount of probably wrong information to the network. The tolerance of the algorithm to the second randomization strategy is much higher, as here no correct information is destroyed. Overall, with both randomization procedures we were able to prove that GraDe is robust against a reasonable amount of both, false positives and missing information and is therefore applicable to large-scale expression studies.

In addition, we analyzed the noise effect of gene expression data by randomly choosing between one and three replicates for each time point. We found significantly low Amari-indices (mean Amari-index = 4.16, $p = 0.026$) by comparing the 95% quantile of the resulting Amari-index with a random sampling. Thus, GraDe is also robust against biological noise.

4.3 Discussion

Including prior knowledge into the source-separation task may obviously introduce bias from the pre-defined patterns that affects the analysis and results obtained. It is important to note that annotation of biological knowledge is always biased and also under permanent change. Therefore, when using gene regulatory networks as prior knowledge one has to keep in mind that this information is subject to annotation bias: The density of interactions in certain regions of the network might be higher due to the fact that these parts are better explored.

In the case of classification problems, recent studies have shown that classification accuracy in microarray data analysis can be improved by including prior knowledge into the classification process (Johannes et al., 2010). The applied methods mainly benefit from the fact that samples are not treated as independent. They are mainly based on the hypothesis that genes which are connected to each other should have similar expression profiles.

Applying standard methods like PCA or ICA, but likewise basic statistical tests, implies the assumption that all data points, i.e. in our setting the expression levels of different genes, are sampled i.i.d. from an underlying probability density. This assumption is obviously not fulfilled, since the genes' expression values are the readouts of different states of a complex dynamical system: Genes obey dynamics along a transcription factor network. Instead of ignoring these dependencies, we here proposed to explicitly model them using prior knowledge given within a gene-regulatory network. Therefore, one of the key advantages of GraDe is to overcome the problematic assumption of i.i.d. samples.

A further methodological strength of the proposed machine-learning approach is that it is based on theoretical considerations. The concept of G-shifts and with it GraDe could be derived from a network approximation of the general ODE model of gene regulation in Section 4.1.5.

4.4 Conclusions and outlook

IL-6 promotes liver regeneration and protects against liver injury. In order to understand these effects in a time-resolved manner, we performed a time-course microarray experiment of *IL-6* stimulated primary mouse hepatocytes. Standard techniques applied to this data set only partly revealed temporal gene expression patterns following the stimulation. To resolve the interaction of *IL-6* and the corresponding cellular responses in more detail, we developed GraDe. It extracts overlapping clusters from large-scale biological data by combining a matrix factorization approach with the integration of prior knowledge. Applying GraDe to our experiment, we

identified the activation of acute phase proteins, which are known to be one of the primary response upon infection based inflammation. Moreover, we observed that *IL-6* activates cell cycle progression, as well as the down-regulation of genes associated with metabolic processes and programmed cell death. Therefore, *IL-6* mediated priming renders hepatocytes more responsive towards cell proliferation and reduces expenditures for the energy metabolism.

The application to the *IL-6* microarray data showed that GraDe is a useful tool in this field. However, GraDe has much more potential beyond the simple exploratory clustering of 'omics' data. For instance, it offers a natural way to integrate different kinds of data available, e.g. to simultaneously analyze microRNA-mRNA expression data linked by predictions from the diverse tools. In future work it should be also investigated whether GraDe can be used for model selection when given different alternative underlying graphs of small-scale models. Before using multi-shift GraDe on large-scale data, its performance has to be studied when using different approximate joint diagonalization methods. Finally, the introduced G-MA processes which we defined to evaluate algorithm performance are of theoretical interest by themselves and merit further investigations.

5 Fuzzy clustering of k-partite graphs: the structural organization of biological data

Large-scale biological information can be derived from high-throughput experiments like in the last Chapter, but also from automated text mining approaches that are searching through the enormous amount of textual data accessible from the biomedical literature. Subsequent data integration in bioinformatics facilitates the construction of large biological networks. However, biological networks are complex and highly diverse and therefore often involve objects of multiple types, forming k-partite graphs consisting of different kinds of vertices. Methods able to structure these heterogeneous data and to extract new knowledge from them gain more and more importance.

Learning approaches commonly focus on the analysis of homogeneous data sets, i.e. graphs having vertices of a single kind only. However, taking into account the different node types provides a more comprehensive picture of the underlying structure compared to the widely used graph transformations. These so-called projections – e.g. of a bipartite network into an unipartite version – are known to discard important information (Guillaume and Latapy, 2004, Klamt et al., 2009). Montanez et al. (2010), for instance, show that in the case of metabolism the use of projections leads to wrong interpretations of some of the most relevant graph attributes, whereas the bipartite view offers a cleaner interpretation of its topological features.

The human disease network presented by Goh et al. (2007) is an example for a bipartite graph having two disjoint sets of vertices. Here, structural questions need to be addressed outside of the unipartite graph setting. One set of nodes represents all known genetic disorders, the vertices of the other partition correspond to all known disease genes in the human genome. A disorder and a gene are connected if mutations in that gene are implicated in that disorder. Other examples of bipartite networks in biology are protein complex or gene-localization, gene-function or microRNA-target networks. The integration of such bipartite network data then leads to the complex k-partite graphs we are interested in.

5.1 Modular decomposition by graph clustering

We want to interpret the internal organization of k-partite networks by performing a modular decomposition using a community detection/clustering method. The clusters and their interconnections are essential for understanding underlying functional properties, and they structure the biological data by compressing the contained information into a condensed form.

Most available community detection methods, as discussed in Section 1.2.5, do not treat the single node types (partitions) separately and therefore do not represent the global cluster structure of k-partite networks correctly. While this has been addressed in terms of algorithms for some time now (Barber, 2007, Karypis et al., 1997, Zhou et al., 2007), not many applications were successfully implemented in bioinformatics yet, although the field commonly deals with such networks (Klamt et al., 2009). A particular issue that may hamper application to biological data is that most existing algorithms identify separated, disjoint clusters by assigning each point to exactly one cluster (Jain and Dubes, 1988, MacQueen, 1967). This is unrealistic for biological systems as e.g. genes or proteins commonly participate in multiple processes or pathways (Pereira-Leal et al., 2004). So far, only a few approaches exist that allow the detection of overlapping clusters in the unipartite setting. We have discussed the prominent methods in Section 1.2.5.3. So far, however, such fuzzy clustering approaches have not been generalized to the common biological case of k-partite graphs.

To overcome these difficulties, this Chapter develops a novel fuzzy clustering algorithm based on a non-negative matrix factorization (NMF) model. Our algorithm extends a hard clustering algorithm recently put forward by Long et al. (2006). This algorithm clusters each node type of the graph separately and then connects clusters via a smaller, weighted k-partite graph in an alternating minimization procedure. Thereby, the cluster assignment in the first step is made in a binary fashion. However, it can be easily seen that the cost function proposed is not fully minimized. Our algorithm avoids this problem. It is similar in structure to multiplicative algorithms for NMF, with the difference that we address a three-matrix factorization problem (see e.g. (Dhillon and Sra, 2006)), and have to deal with a multi-summand cost function. As our cost function is monotonous with respect to the number of clusters, our algorithm allows detection of clusters on different scales. Hence, we are able to decompose the network on different resolution levels. The remainder of this Chapter is organized as follows. In the next Section, we develop the fuzzy clustering algorithm. Then, we validate it on a toy example and graphs with known modular structure. Our main application is a tripartite disease-gene-protein com-

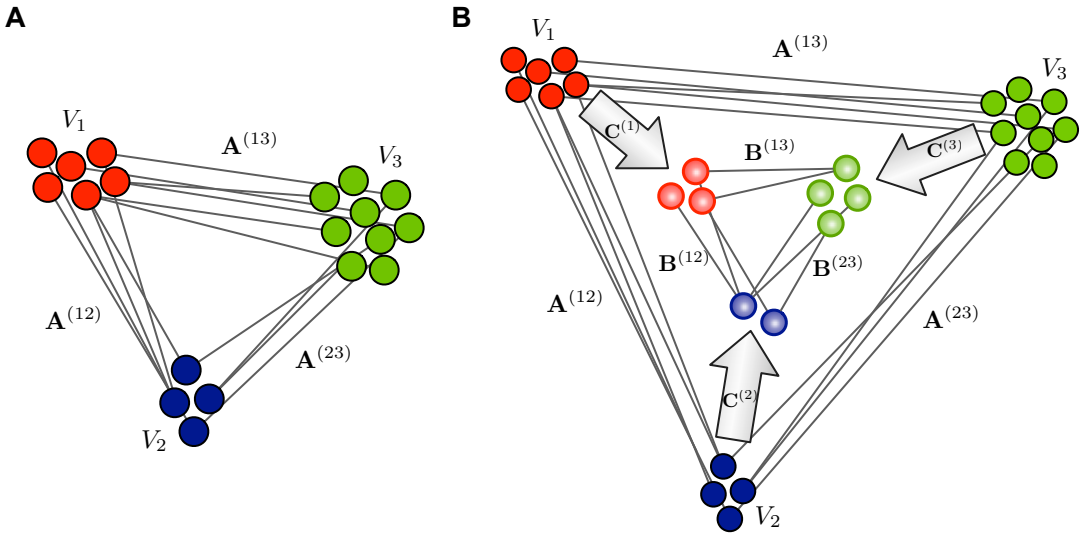


Figure 5.1: We want to approximate the tripartite example graph G in (A) by a smaller tripartite cluster network H, the so-called *backbone graph*, shown in (B). The decomposition into fuzzy clusters connected by this backbone must explain the original connectivity as well as possible. The edges of G are collected in adjacency matrices $\mathbf{A}^{(ij)}$ connecting the elements of the partitions i and j. The approximation of G by the backbone graph is encoded in the adjacency matrices $\mathbf{B}^{(ij)}$ connecting the fuzzy node clusters $\mathbf{C}^{(i)}$. These matrices $\mathbf{C}^{(i)}$ collect the degrees of membership of each node of partition V_i to each cluster of this type. Its (k, l)-th element $\mathbf{C}^{(i)}_{kl}$ specifies to which extent node k belongs to the backbone node l.

plex graph representing an expanded view of the human disease network from Goh et al. (2007) extended by protein complexes (Ruepp et al., 2008). By integrating functional annotation we demonstrate that we are able to structure this complex graph into biologically meaningful clusters on a large scale. Finally, focusing on the small-scale architecture, we identify overlapping clusters that give a more comprehensive picture about gene-disease connections rather than looking at disjoint clusters alone.

5.2 A NMF-type community detection algorithm

Let $G = (V, E)$ be a k-partite graph with edges E, vertices V and a partition of the vertices into k disjoint subsets V_i. Recall that in a k-partite graph no two vertices in the same partition are allowed to be adjacent. Let $n_i := |V_i|$ be the number of vertices in the partition i, $i = 1 \dots k$. We represent the graph as a set of $n_i \times n_j$-dimensional adjacency matrices $\mathbf{A}^{(ij)}$ for all i, j with $1 \leq i < j \leq k$. Typically, each matrix element is either 0 or 1, but we only restrict the matrices to have non-negative coefficients thereby allowing weighted graphs as well.

5.2.1 Graph approximation

We want to approximate G by a smaller k-partite cluster network H which we call *backbone network*. It is defined on the fuzzy clusters of each partition V_i. We fix the number of clusters in V_i to m_i and denote a non-negative $n_i \times m_i$-dimensional matrix $\mathbf{C}^{(i)}$ to be a *fuzzy clustering* of V_i, if it is right-stochastic, i.e. $\sum_{l=1}^{m_i} \mathbf{C}_{kl}^{(i)} = 1$ for all k. Its (k,l)-th element $\mathbf{C}_{kl}^{(i)}$ gives the degree of membership of the original node k to backbone node (cluster) l.

Then we search for a k-partite graph H with $m_i \times m_j$ adjacency matrices $\mathbf{B}^{(ij)}$ and a fuzzy clustering $C := (\mathbf{C}^{(i)})_{i=1,\dots,k}$ such that the connectivity explained by H is as close as possible to G after clustering according to C. Figure 5.1 shows an example graph and its approximation by a backbone network.

From the approximation, we can easily reconstruct an edge $\mathbf{A}_{uv}^{(ij)}$ between two nodes u and v from partitions i and j in the original graph. To this end, we have to sum up all edge weights $\mathbf{B}^{(ij)}$ in the backbone graph that connect the communities u and v are assigned to. Of course, in a fuzzy environment these contributions have to be weighted by the nodes' degrees of membership $\mathbf{C}^{(i)}$ and $\mathbf{C}^{(j)}$, respectively. Taken together, the entry of the adjacency matrix can be reconstructed as the double sum

$$\mathbf{A}_{uv}^{(ij)} \approx \sum_{x=1}^{m_i} \sum_{y=1}^{m_j} \mathbf{C}_{ux}^{(i)} \mathbf{B}_{xy}^{(ij)} \mathbf{C}_{vy}^{(j)} \,.$$

Writing this in matrix notation, we see that the requirement of explaining maximum possible connectivity means that the adjacency matrices $\mathbf{A}^{(ij)}$ are best possible approximated by factorizations of the form

$$\mathbf{A}^{(ij)} \approx \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^\top \,.$$

We can measure the difference between the two graphs H and G in a variety of ways. In (Long et al., 2006), this choice has been circumvented by focusing on arbitrary Bregman divergences, see e.g. (Banerjee et al., 2005), which still allow efficient reformulation of gradient-type algorithms without knowing the specific formula. This is also possible in our case of multiplicative update rules, as has been shown for NMF by Dhillon and Sra (2006). Here, we choose the minimum square distance as the cost function. This implies minimization of

$$f(H,C) := \sum_{i<j} \left\| \mathbf{A}^{(ij)} - \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^\top \right\|_F^2 \,, \tag{5.1}$$

where $\|.\|_F^2$ denotes the squared Frobenius norm, i.e. the square sum of the matrix elements. This cost function is obviously monotonous with respect to the number of clusters in each partition.

5.2.2 Derivation of the update rules

We want to minimize the cost function $f(H, C)$ in Equation (5.1) using a local algorithm extending gradient descent. Let $\mathbf{D}^{(ij)} := \mathbf{A}^{(ij)} - \mathbf{C}^{(i)}\mathbf{B}^{(ij)}(\mathbf{C}^{(j)})^\top$ denote the residuals, then $f = \sum_{i<j,k,l} (d_{kl}^{(ij)})^2$. Hence

$$\begin{aligned}
\frac{\partial f}{\partial b_{rs}^{(ij)}} &= -2\sum_{kl} d_{kl}^{(ij)} c_{kr}^{(i)} c_{ls}^{(j)} = -2\left((\mathbf{C}^{(i)})^\top \mathbf{D}^{(ij)} \mathbf{C}^{(j)}\right)_{rs} \quad \text{and} \\
\frac{\partial f}{\partial c_{rs}^{(i)}} &= -2\sum_{j>i,k,l} d_{rl}^{(ij)} b_{sk}^{(ij)} c_{lk}^{(j)} - 2\sum_{j<i,k,l} d_{kr}^{(ji)} c_{kl}^{(j)} b_{ls}^{(ji)} \\
&= -2\sum_{j>i} \left(\mathbf{D}^{(ij)} \mathbf{C}^{(j)} (\mathbf{B}^{(ij)})^\top\right)_{rs} \quad - 2\sum_{j<i} \left((\mathbf{D}^{(ji)})^\top \mathbf{C}^{(j)} \mathbf{B}^{(ji)}\right)_{rs}.
\end{aligned}$$

We assume an undirected k-partite graph, so $\mathbf{A}^{(ij)}$ is undefined for $i > j$. For simplicity of notation, we now set $\mathbf{A}^{(ij)} := (\mathbf{A}^{(ji)})^\top$ for $i > j$ (and similarly for the k-partite graph H). Then $\mathbf{D}^{(ij)} = (\mathbf{D}^{(ji)})^\top$, and the differential simplifies to

$$\frac{\partial f}{\partial c_{rs}^{(i)}} = -2\sum_{j\neq i} \left(\mathbf{D}^{(ij)} \mathbf{C}^{(j)} (\mathbf{B}^{(ij)})^\top\right)_{rs}.$$

Altogether, by replacing the residuals, we have shown that

$$\begin{aligned}
\frac{\partial f}{\partial b_{rs}^{(ij)}} &= -2\left((\mathbf{C}^{(i)})^\top \mathbf{A}^{(ij)} \mathbf{C}^{(j)} - (\mathbf{C}^{(i)})^\top \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^\top \mathbf{C}^{(j)}\right)_{rs} \quad \text{and} \\
\frac{\partial f}{\partial c_{rs}^{(i)}} &= -2\sum_{j\neq i} \left(\mathbf{A}^{(ij)} \mathbf{C}^{(j)} (\mathbf{B}^{(ij)})^\top - \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^\top \mathbf{C}^{(j)} (\mathbf{B}^{(ij)})^\top\right)_{rs}.
\end{aligned}$$

If we are to minimize f by alternating gradient descent, we start from an initial guess of $\mathbf{B}^{(ij)}, \mathbf{C}^{(i)}$. Then, we alternate between updates of the $\mathbf{B}^{(ij)}$ and the $\mathbf{C}^{(i)}$ with learning rates $\eta_{rs}^{(ij)}$ and $\eta_{rs}^{(i)}$, respectively:

$$\begin{aligned}
b_{rs}^{(ij)} &\leftarrow b_{rs}^{(ij)} - \eta_{rs}^{(ij)} \frac{\partial f}{\partial b_{rs}^{(ij)}} \qquad \forall i, j : i < j \\
c_{rs}^{(i)} &\leftarrow c_{rs}^{(i)} - \eta_{rs}^{(i)} \frac{\partial f}{\partial c_{rs}^{(i)}} \qquad \forall i
\end{aligned}$$

As discussed for NMF in the Preliminaries, these update rules have two disadvantages: first, the choice of update rate η (possibly different for $\mathbf{B}$, $\mathbf{C}$ and i, j) is unclear; in particular, for too small η convergence may take too long or may not be achieved at all, whereas for too large η we may easily overshoot the minimum. Moreover, the resulting matrices may become negative. Hence we follow Lee and Seung's idea for NMF from Section 1.1.2.4 and rewrite this into multiplicative update

rules. We therefore choose the update rates

$$\eta_{rs}^{(ij)} := \frac{b_{rs}^{(ij)}}{2\left((\mathbf{C}^{(i)})^\top \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^\top \mathbf{C}^{(j)}\right)_{rs}} \quad \text{and}$$

$$\eta_{rs}^{(i)} := \frac{c_{rs}^{(i)}}{2\left(\sum_{j \neq i} \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^\top \mathbf{C}^{(j)} (\mathbf{B}^{(ij)})^\top\right)_{rs}} \quad .$$

Plugging this into the gradient descent equations, we finally get:

$$b_{rs}^{(ij)} \leftarrow b_{rs}^{(ij)} \frac{\left((\mathbf{C}^{(i)})^\top \mathbf{A}^{(ij)} \mathbf{C}^{(j)}\right)_{rs}}{\left((\mathbf{C}^{(i)})^\top \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^\top \mathbf{C}^{(j)}\right)_{rs}} \quad \text{and}$$

$$c_{rs}^{(i)} \leftarrow c_{rs}^{(i)} \frac{\left(\sum_{j \neq i} \mathbf{A}^{(ij)} \mathbf{C}^{(j)} (\mathbf{B}^{(ij)})^\top\right)_{rs}}{\left(\sum_{j \neq i} \mathbf{C}^{(i)} \mathbf{B}^{(ij)} (\mathbf{C}^{(j)})^\top \mathbf{C}^{(j)} (\mathbf{B}^{(ij)})^\top\right)_{rs}} \quad .$$

From the Preliminaries, we know that fuzzification factors like in Equation (1.4) are a common strategy to extend cost functions in unipartite clustering to the fuzzy case. Instead of squared norm minimization of the residuals $\mathbf{D}^{(ij)}$, we could minimize a higher residual power, which results in overlapping non-trivial cluster assignments. In our setting, this implies an extended cost function $f_\mu = \sum_{i<j,k,l} = (d_{kl}^{(ij)})^{2/mu}$. However, we see that in our examples, already the standard case is sufficient. This is because we are interested in co-clustering, which is different from standard data clustering where only a unipartite graph and hence $\mathbf{C}^{(i)} = \mathbf{C}^{(1)}$ is assumed.

5.2.3 Algorithm formulation and complexity analysis

We note that, as in the case of NMF and multi-factor NMF (Dhillon and Sra, 2006), our update rules do not increase the cost function from Equation (5.1). This theoretical result implies convergences of the update rules. However, in contrast to early statements in NMF (Lee and Seung, 2001), it does not necessarily imply convergence to stationary points of the Euclidean norm (zero of the differential from (5.1)), since the update steps may be too small to reach those points. Another possible drawback of such multiplicative updates is again the fact that once a matrix entry has been set to zero (which may happen due to zeros in $\mathbf{A}^{(ij)}$ or to numerics), the coefficient will never then be able to become positive again during learning.

We have not yet taken into account the constraint that the fuzzy clusterings $\mathbf{C}^{(i)}$ are required to be right-stochastic. We force this constraint by regularly projecting each row of $\mathbf{C}^{(i)}$ onto the sphere of the 1-norm. The final fuzzy k-partite clustering algorithm is summarized in Algorithm 3.

The algorithm has two nested loops over the number of partitions k, hence its runtime depends quadratically on this number. The update rules for $\mathbf{C}^{(i)}$ and $\mathbf{B}^{(ij)}$,

Algorithm 3: fuzzy k-partite clustering

Input: k-partite graph G with possibly non-negatively weighted edge matrices $\mathbf{A}^{(ij)}$, $i < j$, number of clusters $m_1, \ldots, m_k$

Output: fuzzy clustering $\mathbf{C}^{(i)}$ and k-partite cluster graph H given by matrices $\mathbf{B}^{(ij)}$

1 Initialize $\mathbf{C}^{(i)}, \mathbf{B}^{(ij)}$ to random non-negative matrices.

2 Normalize $c_{rs}^{(i)} \leftarrow c_{rs}^{(i)} / (\sum_t c_{rt}^{(i)})$ for all i, r, s

repeat

update fuzzy clusters

for $i \leftarrow 1, \ldots, k$ **do**

3 $\quad \mathbf{C}^{(i)} \leftarrow \mathbf{C}^{(i)} \otimes (\sum_{j \neq i} \mathbf{A}^{(ij)} \mathbf{C}^{(j)} \mathbf{B}^{(ij)\top}) \oslash (\sum_{j \neq i} \mathbf{C}^{(i)} \mathbf{B}^{(ij)} \mathbf{C}^{(j)\top} \mathbf{C}^{(j)} \mathbf{B}^{(ij)\top})$

$\quad$ Normalize $c_{rs}^{(i)} \leftarrow c_{rs}^{(i)} / (\sum_t c_{rt}^{(i)})$ for all r, s

end

update k-partite cluster graph H

for $i \leftarrow 1, \ldots, k-1$ **do**

$\quad$ **for** $j \leftarrow i+1, \ldots, k$ **do**

4 $\quad\quad \mathbf{B}^{(ij)} \leftarrow \mathbf{B}^{(ij)} \otimes (\mathbf{C}^{(i)\top} \mathbf{A}^{(ij)} \mathbf{C}^{(j)}) \oslash (\mathbf{C}^{(i)\top} \mathbf{C}^{(i)} \mathbf{B}^{(ij)} \mathbf{C}^{(j)\top} \mathbf{C}^{(j)})$

$\quad$ **end**

end

until *convergence* ;

Note: $\otimes$ and $\oslash$ symbolize element-wise multiplication and division, respectively.

however, are fully vectorized and contain only matrix operations with non-square matrices. Their time complexity is dominated by the occurring matrix products: multiplying two matrices of sizes $s_1 \times s_2$ and $s_2 \times s_3$ is of complexity $\mathcal{O}(s_1 s_2 s_3)$. Assuming that the cluster numbers m_i are smaller than the largest two partition sizes, the total time complexity of the fuzzy clustering algorithm can then be estimated as

$$\#\text{iterations} \times \mathcal{O}(k^2 \, n_{\max 1} n_{\max 2} \, m_{\max}) \, .$$

Here, $n_{\max 1}$ and $n_{\max 2}$ denote the sizes of the largest and the second-largest partition, $m_{\max}$ is the maximum number of clusters within any partition. Hence, the runtime grows only quadratically in the total number of nodes in the case of graphs with similarly large partitions. In general, the runtime is linear in each partition's size n_i and cluster number m_i.

5.3 Algorithm evaluation

For illustration, we applied our algorithm to a bipartite graph having several vertices connected with all vertices of the other partition (e.g. nodes 1 and 10). Figure 5.2 shows that these vertices are assigned to two clusters with distinct degrees of

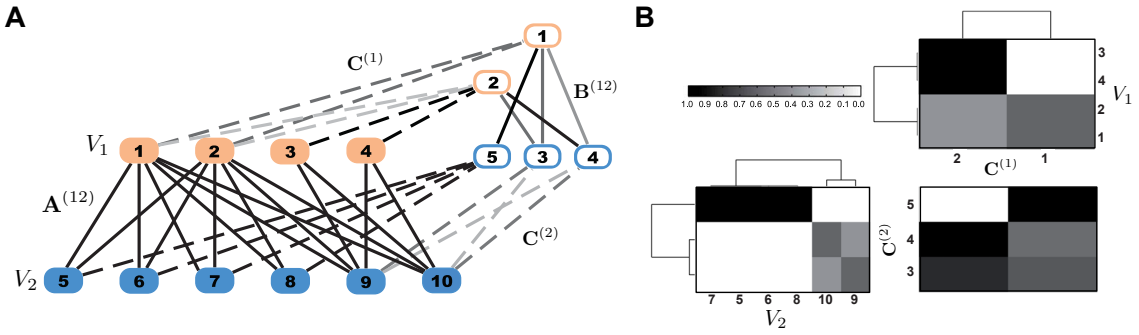


Figure 5.2: (A) demonstrates the graph decomposition with our algorithm on a small bipartite graph with overlapping cluster structure. The original graph consists of partitions $V_1 = \{1 \dots 4\}$ (red filled nodes) and $V_2 = \{5 \dots 10\}$ (blue filled nodes) connected by edges $\mathbf{A}^{(12)}$ colored in black. We decomposed it into two clusters for partition V_1 and three clusters for partition V_2. The resulting fuzzy clustering is illustrated as a weighted graph connecting original nodes to cluster nodes (framed red and blue). The cluster assignments $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$ are indicated by dashed edges, where edge color corresponds to the degree of cluster membership. The interconnections of the clusters form the *backbone graph*, encoded in the adjacency matrix $\mathbf{B}^{(12)}$ which we again indicate by full lines with color coding the edge weights.
Another way of illustrating the graph decomposition, clearer especially for larger graphs, is shown in (B). First, we plot hierarchical clusterings of the nodes' degrees of membership in partitions V_1 and V_2 (encoded by $\mathbf{C}^{(1)}$ and $\mathbf{C}^{(2)}$). This facilitates the identification of overlapping clusters (e.g. nodes 1 and 10 are assigned to more than one cluster) or hard cluster assignments (e.g. node 5). The backbone graph $\mathbf{B}^{(12)}$ is shown bottom right. This backbone graph is densely connected in our example.

membership, whereas vertices partially connected are element of a single cluster only (e.g. node 3). This demonstrates the importance of using a fuzzy approach that allows for overlapping clusters.

5.3.1 Performance analysis

Before applying our algorithm to real-world data, we tested its behavior on simulated data with controlled cluster structure. In particular, we compared it to the hard clustering algorithm from Long et al. (2006). We used exactly the same stopping criteria for both algorithms.

We built a random, modularly structured k-partite network as follows: We fix the number of clusters m_i of nodes with color i, $i = 1, \dots k$. The backbone graph was initialized by $m_i \times m_j$-matrices $\mathbf{B}^{(ij)}$ filled with zeros. We added uniformly random ones in each column according to a set percentage α (here on average $\alpha \geq 1$ ones are in each column) such that each row had at least a single non-zero entry. In order to construct the actual network $\mathbf{A}$, we split up $\mathbf{A}^{(ij)}$ into $m_i \cdot m_j$ blocks of a fixed chosen clustersize (here 10). We fixed a cluster connectivity β and a random

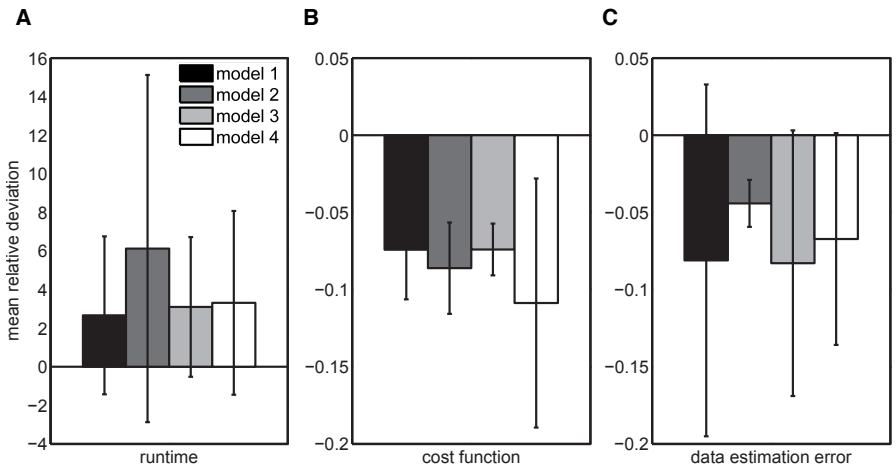


Figure 5.3: We validated our algorithm on graphs with predefined cluster structure. To this end, we compared it with the hard clustering method by Long et al. (2006) on four different random toy models, see Table 5.1. The plot shows the mean relative deviation between the two algorithms relative to the results of the hard clustering. Error bars denote standard deviations over 1000 runs. We see that the fuzzy cluster assignments of our method require more runtime, but both cost function and data estimation error are significantly smaller. The large standard deviations show the dependency of the decomposition on the random initialization. Therefore, by default we perform multiple restarts with different initializations.

connectivity $\gamma < \beta$. Now, for each non-zero entry in $\mathbf{B}^{(ij)}$, we set the corresponding block of $\mathbf{A}^{(ij)}$ to an Erdös-Rényi graph as introduced in Section 1.2.6.1 with density β. Finally the clusters were connected by replacing each zero block of $\mathbf{A}^{(ij)}$ with an Erdös-Renyi graph of the lower connectivity γ.

Table 5.1: Parameters for the simulated data models. k denotes the number of partitions of the network, $\mathbf{m}$ is a vector with the number of clusters in each partition, α the backbone connectivity, β the cluster and γ the noise connectivity.

model	k	$\mathbf{m}$	α	β	γ	description
1	2	$(3,3)$	1	0.7	0.2	equal-sized, no overlap
2	2	$(3,4)$	1	0.7	0.2	no cluster overlap
3	3	$(3,4,5)$	1.2	0.6	0.1	3-partite, low-noise
4	3	$(3,4,5)$	1.2	0.8	0.2	3-partite, noisy

In order to compare algorithm performance, we determined final cost function value, runtime, and the quality of cluster estimation. Cluster estimation quality was measured by the summed up Frobenius norms of the difference between the true $\mathbf{C}^{(i)}$ and the estimated $\hat{\mathbf{C}}^{(i)}$, where clusters have been permuted such as to give minimal difference (permutation indeterminacy). We analyzed 1000 realizations of

four network prototypes with increasing complexity (parameters are given in Table 5.1). We restricted ourselves to bipartite and layered tripartite graphs with two different noise settings because Long et al. (2006) provided code for analyzing these special cases only.

We found that while the method of Long et al. (2006) performed around two times faster, our algorithm produced around 10% lower cost function and was able to estimate the cluster structure better (see Figure 5.3). This difference in algorithm runtime originates from the much more fine-tuning of the continuous degrees of membership compared to hard cluster assignments. These require less update steps until convergence.

5.3.2 Stability of clusters against the random initialization

In contrast to deterministic methods like for instance PCA, NMF-based methods have problems concerning robust computation. Even for standard uni-partite NMF there is no unique global minimum of the cost function (Langville et al., 2006). Our algorithm aims to minimize the cost function using a local optimization strategy extending gradient descent, which implies that it only converges to a local minimum. The algorithm is indeterministic, it does not converge to the same solution on each run due to the stochastic nature of initial conditions. Thus, following the general proceeding in literature on NMF (Devarajan, 2008, Langville et al., 2006), we compare the local minima from several different starting points (multiple restarts), using the results of the best local minimum found.

In order to analyze the stability of the algorithm, we applied it to a bipartite toy network with well defined cluster structure and three clusters per partition. We chose a graph with $n_1 = 900$ nodes in partition 1 and $n_2 = 800$ nodes in partition 2. Then, we employed a stability score to compare the clustering results and to quantify their stability. Garge et al. (2005) proposed to use Cramer's v^2 from Equation (1.5) to measure replicability. However, since crisp assignment of nodes to clusters would be required, we instead use the Fuzzy Rand Index (FRI) defined in Equation (1.7).

We calculated the pairwise similarity of all clustering results to the reference fuzzy memberships for both partitions separately. The resulting FRI distribution is shown in Figure 5.4. In more than 70%, we reached a FRI of 1, indicating that our algorithm produces stable results with close fuzzy memberships. However, although the studied input graph had a well defined cluster structure, our algorithm did not always converge to a meaningful decomposition. Sometimes, only one cluster could be determined correctly, which shows that we can not guarantee that the local optimization finds a global minimum of the cost function, as discussed before. This illustrates the critical need for multiple restarts.

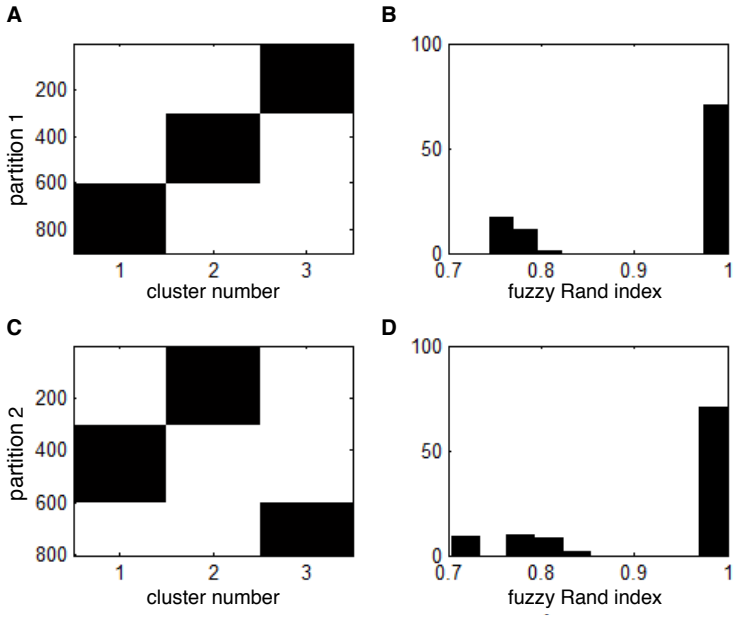


Figure 5.4: In order to analyze algorithm stability, we decomposed a bipartite toy network with a well defined cluster structure and three clusters per partition. (A) and (C) show good decompositions of partition 1 and 2 into the true cluster structure with $m_i = 3$ clusters in each partition. (B) and (D) give the distribution of the fuzzy Rand index (FRI) in the two partitions over 100 runs, where we extracted three clusters per partition. We reach a FRI of 1 in more than 70%, indicating a reasonably good reproducibility.

5.3.3 The cluster structure depending on m

Using a bipartite graph with six well defined, yet not hard clusters, we studied the dependency of the clusters on the cluster number m in Figure 5.5. The graph is illustrated in Figure 5.8(A). In this example we found that extracting less clusters than really present in the data, our algorithm identified clusters composed by the union of true ones. If we extracted more clusters than the correct number is, it split up true clusters into strongly overlapping sub-clusters. Hence, even when not extracting the exactly correct number of clusters, the decomposition obtained appears to be meaningful and allows for a deeper interpretation of the extracted clusters.

Histograms of the obtained degrees of membership showed a characteristic behavior. In graphs without cluster structure, compare Figure 5.6, large degrees of membership were completely missing. If the graph contained a well defined cluster structure, the histograms had an U-like shape: we found a large number of high values almost up to one, see Figure 5.5. Hence, we propose to employ such histograms in order to give an indication whether a graph is modularly organized or not.

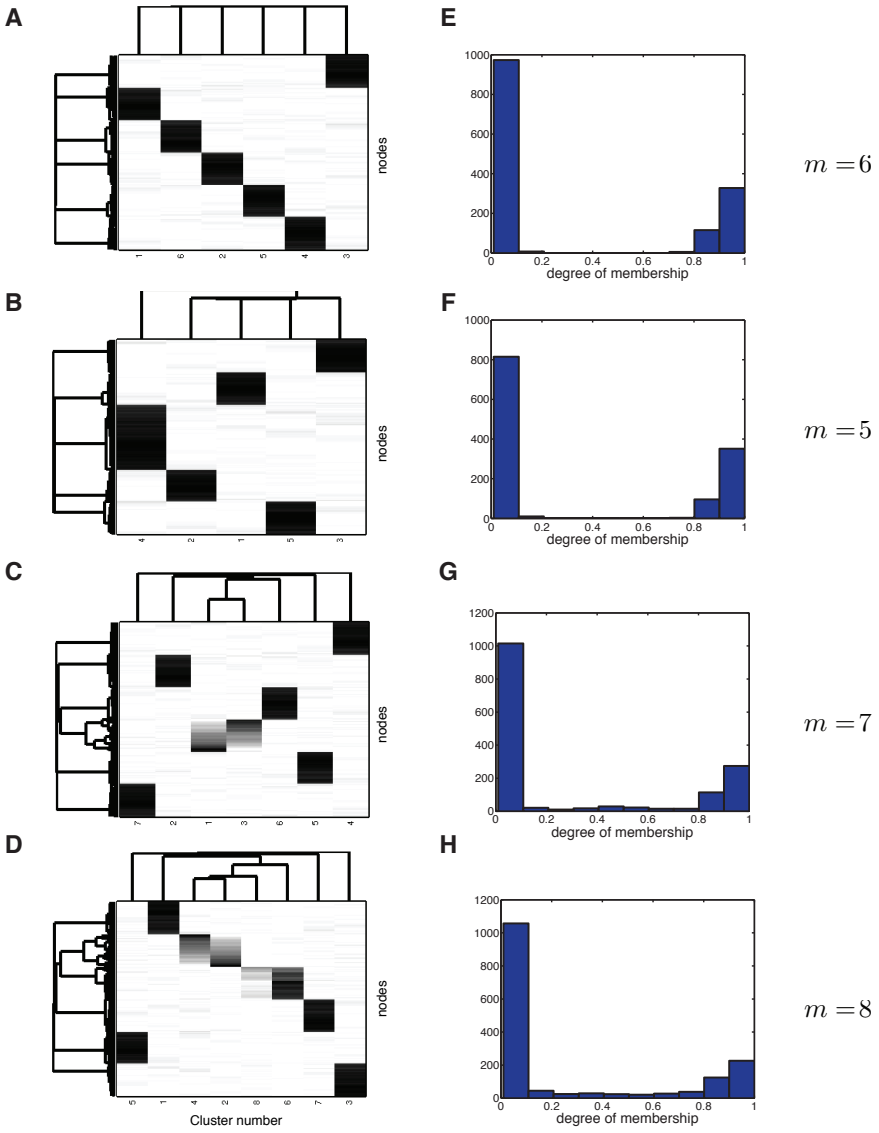


Figure 5.5: (A-D) give hierarchical clusterings of the degree of membership matrices obtained for the true $m = 6$ and also for $m = 5$ and $m = 7, 8$. Each plot shows the best of 25 runs. The algorithm behaves well: When we extracted only five clusters, it detected four of the six clusters, the fifth cluster being the union of the last two true clusters. Extracting more than six clusters, it split up true clusters into two strongly overlapping sub-clusters. In (E-H) we give histograms of the degrees of membership obtained in these situations (for better recognizability, we counted only entries ≥ 0.01). These histograms have a typical U-like shape, with a peak at small entries and a second peak at large degrees of membership around one indicating the well defined cluster structure of the studied graph.

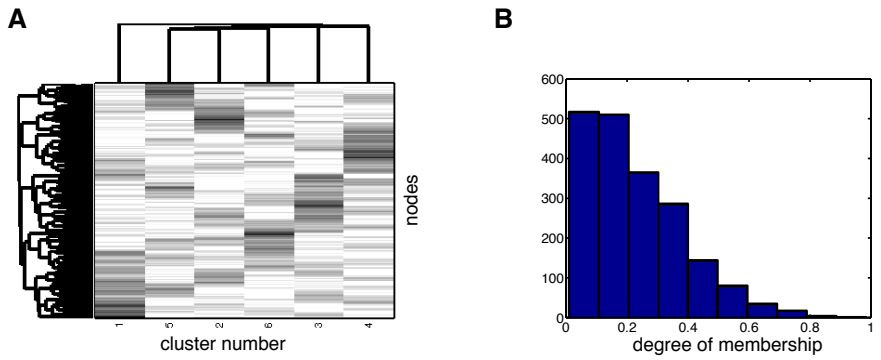


Figure 5.6: (A) gives a hierarchical clustering of the best of 25 decompositions of the graph from Figure 5.8(C), which has no cluster structure. Although the algorithm identified clusters visible in this illustration, they are not well defined and blurred out. Consequently, in the histogram of the degrees of membership (B) we see a large number of small values, the peak at large degrees of membership is missing.

5.4 Decomposition of a gene-disease-protein complex graph

We exemplifed the analysis of biological networks in collaboration with the group of Volker Stümpflen (BIS, IBIS, Helmholtz Zentrum München): The algorithm was applied to a tripartite disease-gene-protein complex graph as illustrated in Figure 5.7. In this graph, a disorder and a gene are connected if mutations in that gene are implicated in that disorder. A complex and a gene are linked if the gene is coding for a protein part of the complex. We constructed this graph by integrating the human gene-disease network from Goh et al. (2007) and protein complexes from the CORUM database. We used the CORUM core set as of July 2009 (Ruepp et al., 2008). Both data sets are manually curated, hence we expect their quality to be quite high. The resulting graph consists of 5672 nodes and 7795 edges with all genetic disorders, all known disease genes and human protein complexes. We extracted the largest connected component where $|V| = 3737$ and $|E| = 6219$. It consists of 854 complexes in V_c, 590 diseases in V_d and 2293 genes in V_g.

5.4.1 Choice of parameters

An important feature of biological networks is their hierarchical organization: Higher-level structure is composed of multiple instances of various lower-level structures (Clauset et al., 2008). Good examples, where systems are repeatedly connected to form other systems are illustrated by Barabási and Oltvai (2004) or Ahn et al. (2010). Hierarchical organization implies that small groups of nodes organize in a hierarchical manner to increasingly large groups on many different scales (Barabási and Oltvai, 2004, Ravasz and Barabási, 2003). In order to account for this topological

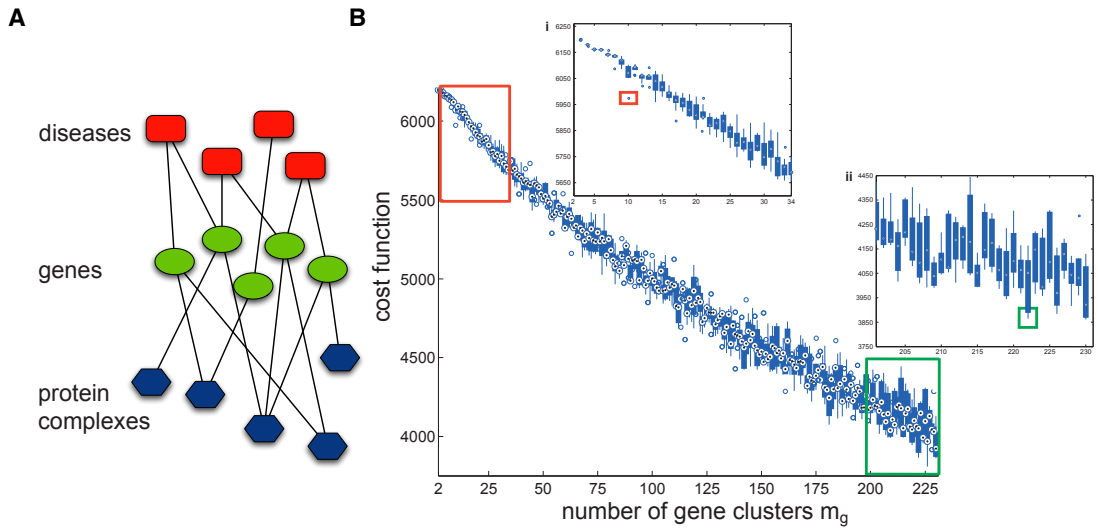


Figure 5.7: We combine the gene-disease network from Goh et al. (2007) with human protein complexes from the CORUM database (Ruepp et al., 2008). This resulted in a layered tripartite graph, which is schematically drawn in (A). We performed a 10-fold approximation of this graph to estimate appropriate numbers of clusters. The boxplot curve (B) shows how the cost function $f(H, C)$ from Equation (5.1) depends on the number of gene clusters m_g. The true minima of the cost function are decreasing with m_g, and this is also visible in the approximated minima identified by our proposed algorithm. Hence, it allows detecting structure on various resolution levels. The details show the cost function course for large-scale clustering (i) and a decomposition on small scale (ii), respectively. For our detailed analyses, we used the decompositions showing steep drops in the cost function marked by the red and green boxes.

characteristic one has to be able to extract relevant information on an appropriate, pre-defined resolution level. We addressed this issue by analyzing the very global structure and a detailed local level of the disease-gene-protein complex network. In the following, we first present the results of a decomposition into large clusters which demonstrates that our method is generally applicable to biological data. Then, we discuss smaller clusters that allowed for a precise interpretation of single elements.

As discussed before, due to its random initialization our algorithm is inherently indeterministic. Different clustering results have of course a significant impact on the interpretation of the biological meaning of the results. We already showed that our algorithm is quite stable on graphs with well defined cluster structure. Hence, we verify that the disease-gene-protein complex network has indeed a defined cluster structure. In order to avoid analyzing a local minimum, we always discuss performance over 10 runs.

Dealing with a theoretically monotonous cost function, it is hard to determine the optimal numbers of clusters for each node type in which the graph has to be

partitioned. Even in the case of unipartite k-means or principal component analysis (compare Chapter 1) there is no direct and computationally simple solution. Appropriate values are also not apparent from prior knowledge about our data set. We therefore chose desired approximate resolutions m_g for the gene partition. For large-scale clustering, we approximated the number of clusters to be found by limiting the maximal number of gene clusters to $m_g = \lfloor\sqrt{|n_g|/2}\rfloor$, as suggested by Mardia et al. (1979). The number of complex clusters m_c for V_c and disease clusters m_d for V_d were then scaled according to their partitions' sizes:

$$m_i = \lceil m_g\sqrt{n_i/n_g}\rceil,$$

where $i \in \{c, d\}$. We used this heuristics, since a brute-force sampling of the three-dimensional parameter space is computationally out of reach. Then, we looked for plateaus and steep drops in the cost function within a certain range around this value m_g and chose a local optimum of the algorithmically found decompositions. In Figure 5.8 we performed simulations illustrating that the profile of the cost function may indeed indicate for a proper number of clusters in graphs with known cluster structure.

5.4.2 Clusters on a large scale

First, we focused on the identification of large clusters. Figure 5.7 shows the cost function values after algorithm convergence for each parameter setting. In the following discussion, we used $(m_g, m_c, m_d) = (10, 5, 6)$ as it showed the first steep drop in the cost function. Moreover, here we observed a significant local minimum of the cost function values of the algorithmically determined decompositions.

From the illustration of the decomposition in Figure 5.10 we see that the resulting clusters vary strongly in size. For all partitions, the majority of elements was assigned to a single cluster with degree of membership above 0.9. The corresponding histograms are given in Figure 5.9. As discussed before, such large degrees of membership are rarely found in graphs lacking any cluster structure. Thus we have evidence for the presence of a well defined cluster structure at the desired resolution level. However, there also exists a considerable amount of elements simultaneously assigned to several clusters, e.g. in complex clusters 3 and 5, gene clusters 1 and 3 or disease clusters 3 and 4. This shows the need for a fuzzy approach, especially in the case of the disease partition.

5.4.2.1 Cluster evaluation

To determine whether the resulting clusters are biologically reasonable, we again applied GO enrichment analysis to the clusters of the gene partition. To this end, the

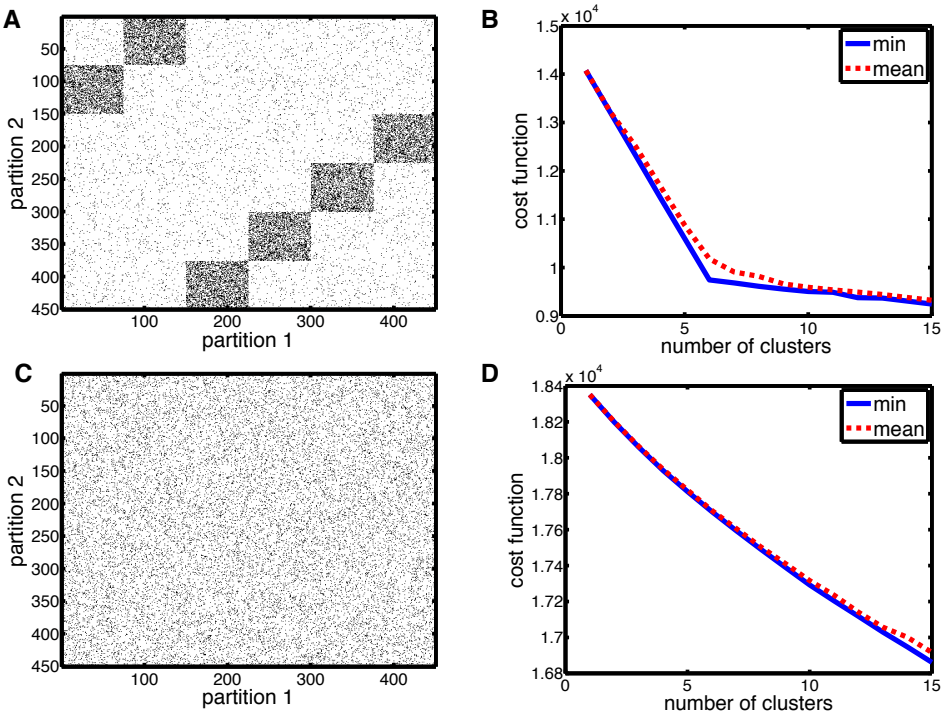


Figure 5.8: To analyze the profile of the cost function, we generated two bipartite example graphs. Their adjacency matrices are illustrated in (A) and (C) as heatmaps (black codes for ones, white for zeros). The first graph has six obvious (yet not hard) clusters in each partition, each of them connected to only one cluster of the other partition. It contains 75 nodes per cluster and two nodes of different color stemming from linked clusters are connected with a probability of 0.4. Additionally, we introduced random connections between the other nodes with a probability of 0.05. The second graph also contains 450 nodes per partition, but has no cluster structure (all pairs of nodes connected with probability 0.12).
(B) and (D) show the profile of the cost function after algorithm convergence (average and minimum value over 100 runs) when extracting between two and 15 clusters. While in the second, cluster-free example there is no structure in the profile of the cost function, the well defined cluster structure of the first example has a sharp break. For $m = 1 \dots 6$, i.e. until all present six clusters are detected, we observe steep drops in the cost function, followed by a flat plateau with little refinement for $m > 6$.

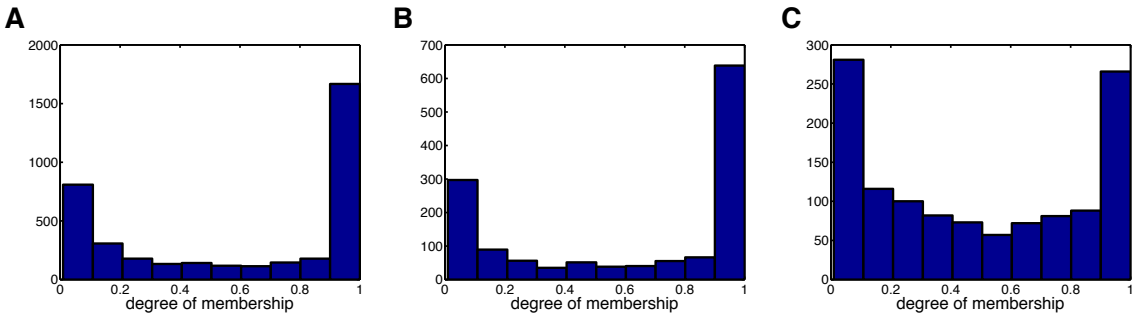


Figure 5.9: The degrees of membership for the large-scale clustering of the gene-disease-protein complex graph: (A) genes, (B) protein complexes, (C) diseases. The U-like shape gives evidence for the existence of a well defined cluster structure. For better recognizability, we counted only entries ≥ 0.01.

genes used in the analysis (degree of membership above 0.2) were tagged with their respective GO categories and analyzed within each cluster for overrepresentation of certain categories versus the "background" level of the population (in this case, all genes in the tripartite graph). We used Ontologizer (Bauer et al., 2008) with the setting "Parent-Child-Intersection" restricting the analysis to the *biological process* category. For multiple testing correction we employed Bonferroni correction. To assign GO-Terms to gene sets, a p-value cutoff of 0.05 was used.

We found, for instance, that in the genes within the two overlapping clusters 1 and 3 the GO-Terms *cell cycle* and *cellular response to stimulus/stress* are significantly enriched. Genes in cluster 4 can be related to e.g. *death, cell proliferation* and *developmental processes*, whereas cluster 6 represents *translation. Gene expression*-associated GO-Terms, such as *RNA processing* and *splicing*, were detected in cluster 7. This shows that our method was able to identify biologically meaningful functionally enriched clusters.

5.4.2.2 Backbone evaluation

The interconnectivity of the in total 21 clusters is sparse (see Figure 5.10). The skeleton for the global cluster structure for both underlying bipartite graphs (gene-complex and gene-disease) demonstrates that locally overlapping clusters also tend to interconnect with the same clusters of the other partition; for instance, disease clusters 3 and 4 are both connected with gene cluster 9. To evaluate the extracted backbone graph, in the following we tested the hypothesis that interconnected clusters of different partitions are also functionally correlated. Assuming that the resulting interconnected gene and complex or disease clusters are functionally related, one expects to see a similar profile for functional annotation and backbone intercon-

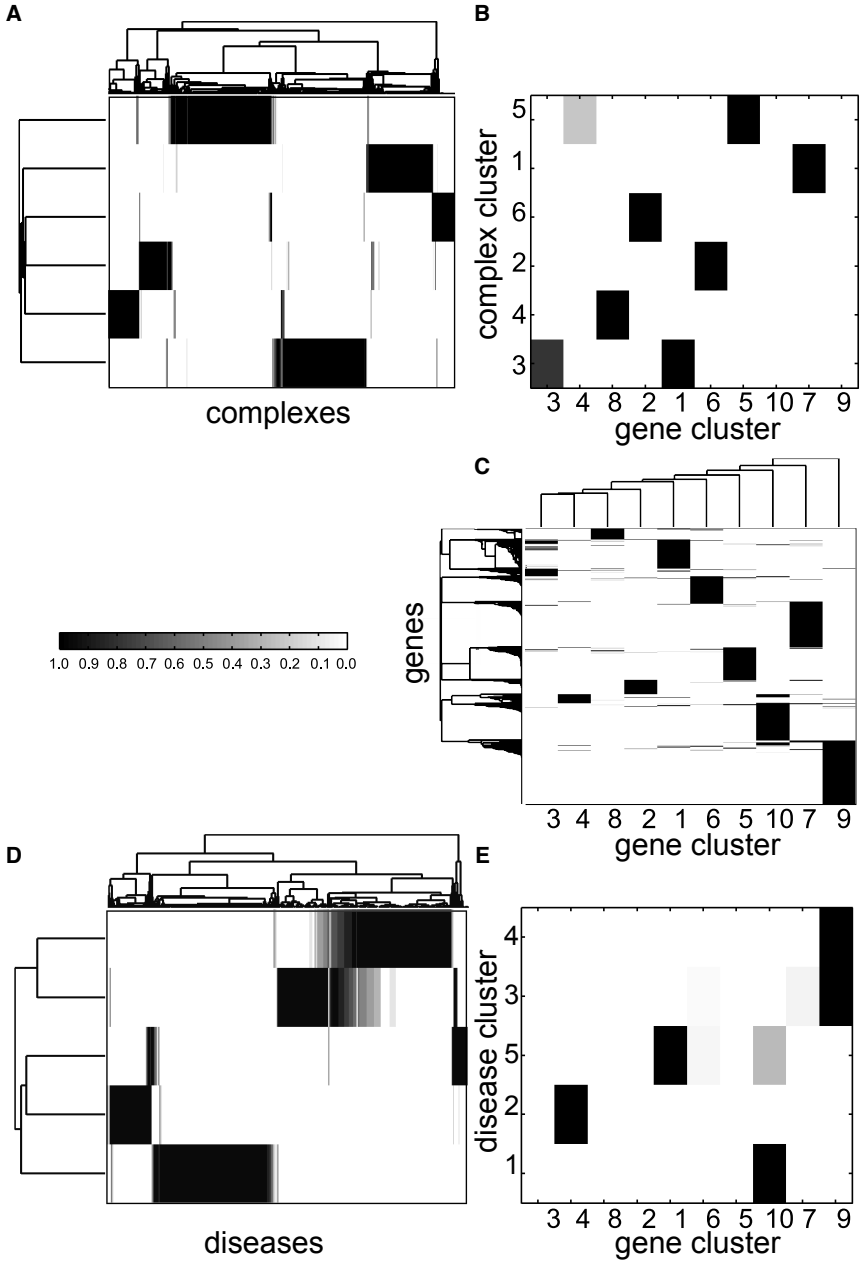


Figure 5.10: The large-scale decomposition of the gene-disease-protein complex network. We performed hierarchical clusterings of the nodes' degrees of membership of the (A) complex, (C) gene and (D) disease partition. These show that the majority of elements was assigned to single clusters. However, a considerable amount of cluster overlaps exists, e.g. for the disease clusters 3 and 4. The backbones for gene-complex (B) and for gene-disease (E) are sparsely connected, but show that locally overlapping clusters also tend to interconnect with the same clusters of the other partition; e.g. disease cluster 3 and 4 are both connected to gene cluster 9 with large weights.

nectivity of each cluster.

Evaluation strategy For backbone evaluation, we used functional annotations from FunCat (Ruepp et al., 2004), which are available for all genes and protein complexes. We chose to use FunCat, since Gene Ontology associations for genes could be mapped to their according FunCat categories, but not vice versa. A subset of 13 main categories was used, subcategory annotations were mapped to corresponding main category terms. Disorder classifications for genes and diseases were taken from (Goh et al., 2007), where the classification classes *grey* and *multiple* were combined for pleiotropic genes. We calculated Pearson's correlation coefficients between cluster FunCat/disorder annotations by weighting a cluster element's classification by its degree of membership to the particular cluster. As the difference score between normalized backbone interconnectivity and annotation correlation we define the Frobenius norm of their difference.

The null models for the evaluation of the backbone were generated by applying a weighted bipartite randomization procedure to each partition-cluster subgraph $\mathbf{C}^{(i)}$. To this end, we generalized the degree-preserving rewiring of complex networks first introduced by Maslov and Sneppen (2002) that was already used in the preceding Chapter. In the weighted case, one has to decide between preserving either the number of neighbors of all nodes, or the total weight of their adjacent edges during the rewiring procedure. We chose to maintain the first quantity: In every randomization step we randomly picked two edges and exchanged their endpoints of the partition type, thereby keeping the weights attached to the edges. With this we also conserved the weighted degree of the partition nodes which reflects the right-stochasticity of the fuzzy clusterings. The degree of randomization can be monitored by a loss of degree-correlations between first and second neighbors. In practice, correlations vanish after about one randomization step per edge. So, for our analyses we used five times this number as we suggested in Wong et al. (2008). We calculated p-values from a sampling over 10^5 runs.

Evaluation results The evaluation results are illustrated in Figure 5.11. Here, we find, for instance, that complex cluster 3 and the interconnected gene clusters 1 and 3 show a high binary FunCat correlation. The difference score between backbone interconnectivity and annotation correlation is 2.48, resulting in a p-value $< 10^{-5}$. To compare the results of the fuzzy clustering approach with the results for the disjoint clustering method of Long et al. (2006), we applied the algorithm with the same parameter settings and identical annotation and randomization procedure to the obtained clusters. For hard clustering we achieved a larger difference score of

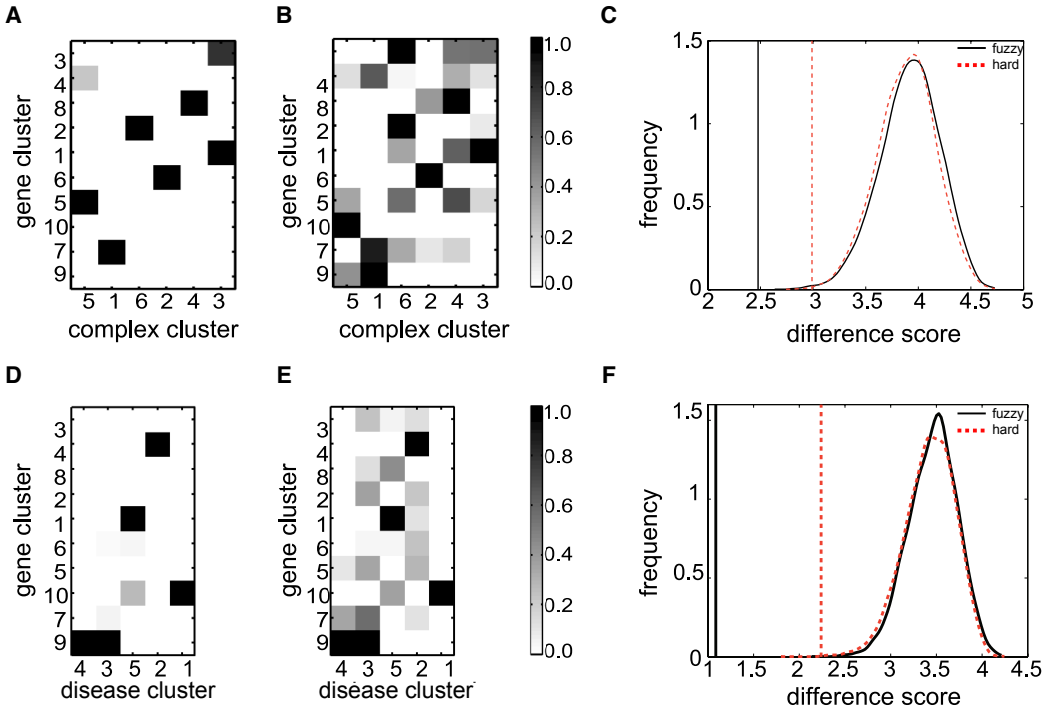


Figure 5.11: To evaluate the large-scale clustering we additionally included functional annotations. (A) and (B) compare the gene-complex backbone with functional correlations of the extracted clusters according to FunCat annotation. Similarly, (D) and (E) show the gene-disease backbone and the clusters' disorder class correlations. Interconnected clusters also seem to correlate in their annotations. To test this hypothesis rigorously, we calculated difference scores that quantify the correlation of the backbones and their annotations, respectively. Vertical lines in (C) and (F) correspond to the difference scores for the fuzzy (black) and the hard (red) clustering. Comparing these values to the difference scores for 10^5 randomized cluster assignments we obtain significant p-values $< 10^{-5}$. The correlation between annotations of connected clusters of the backbone is higher when applying the fuzzy approach.

2.99 which corresponds to a significant p-value of 0.0015.

To ascertain that our method was able to detect biological feasible clusters in all partitions, we also determined disorder class profiles for each gene and disease cluster. Again, we observed high similarity between backbone interconnectivity and disorder correlation having a difference score of 1.09 (p-value $< 10^{-5}$). For instance, gene cluster 1 and 10 and the interconnected disease clusters 1 and 5 show a high disorder correlation (see Figure 5.11).

The results for the hard clustering approach showed a larger difference score of 2.24 which corresponds to a still significant p-value of $2 \cdot 10^{-4}$. We see that the annotation correlations between connected clusters of the backbone graph is higher when applying the fuzzy approach.

5.4.3 Clusters on a small scale

We showed that our method is able to both detect and interconnect biologically meaningful clusters. However, due to their size of about 279 genes on average the single clusters are hard to interpret. The detection of smaller clusters representing biological units enables a precise biological interpretation. To detect smaller clusters, we set the maximum number of gene clusters to $m_g = \lceil \frac{|V_g|}{10} \rceil$. This number seems biologically useful, as most functional complexes contain 10 to 30 protein components (Mete et al., 2008, Voevodski et al., 2009). Furthermore, genes are involved in up to 10 disorders (Goh et al., 2007). In the following, we describe results for $(m_g, m_c, m_d) = (222, 135, 112)$, where we found the lowest value of the cost function, compare Figure 5.7(B). This setting accounts for an average cluster size of 10 genes in the gene partition.

In order to make use of the cluster overlaps, we looked for genes assigned to more than one cluster with a degree of membership of above 0.2. We considered this threshold as significant as it is 50-fold higher than assigning each gene uniformly to all 222 gene clusters with equal degree of membership of 0.0045.

As a showcase we chose *MECP2*, a protein that functions as a key factor in epigenetic transcriptional regulation. It is known to be involved in neurodevelopmental and psychiatric disorders such as *Autism, Mental retardation* and *Angelman syndrome* (Campos et al., 2009, Goh et al., 2007, Samaco et al., 2005), and was assigned to three distinct gene clusters with degrees of membership of 0.42, 0.31 and 0.24, respectively. These clusters mainly cover neurological (23%), psychiatric (81%) and pleiotropic (7%) genes having a degree of membership larger than 0.2. This is illustrated in Figure 5.12, where we visualized the backbone interconnectivity and the fuzzy clustering of the nodes in the neighborhood of *MECP2*.

We then analyzed the nine disease clusters interconnected with the three gene clusters in the backbone network. In total, 45 disorders representing mainly psychiatric (66%) and neurological (20%) disorders were assigned to eight disease clusters with a degree of membership of above 0.2; 6 out of 9 psychiatric disorders available in the network analyzed were present in three disease clusters.

Another large fraction of these disease clusters are disorders classified as *multiple*. Most of them (*Shprintzen-Goldberg syndrome* or *Aarskog Scott syndrome*) show also neurological diseases such as mental retardation (Lebel et al., 2002, Shprintzen and Goldberg, 1982). We also identified the *ophthalmological* disorder *Blepharospasm*, an adult-onset focal dystonia that causes involuntary blinking and eyelid spasms (Fiorio et al., 2008) for that a known polymorphism in the dopamine receptor *DRD5* is associated with (Misbahuddin et al., 2002). This is a subform of *Dystonia* and classified as a *neurological* disorder (ICD-10 G24.5) by the WHO (Isaac et al., 1994).

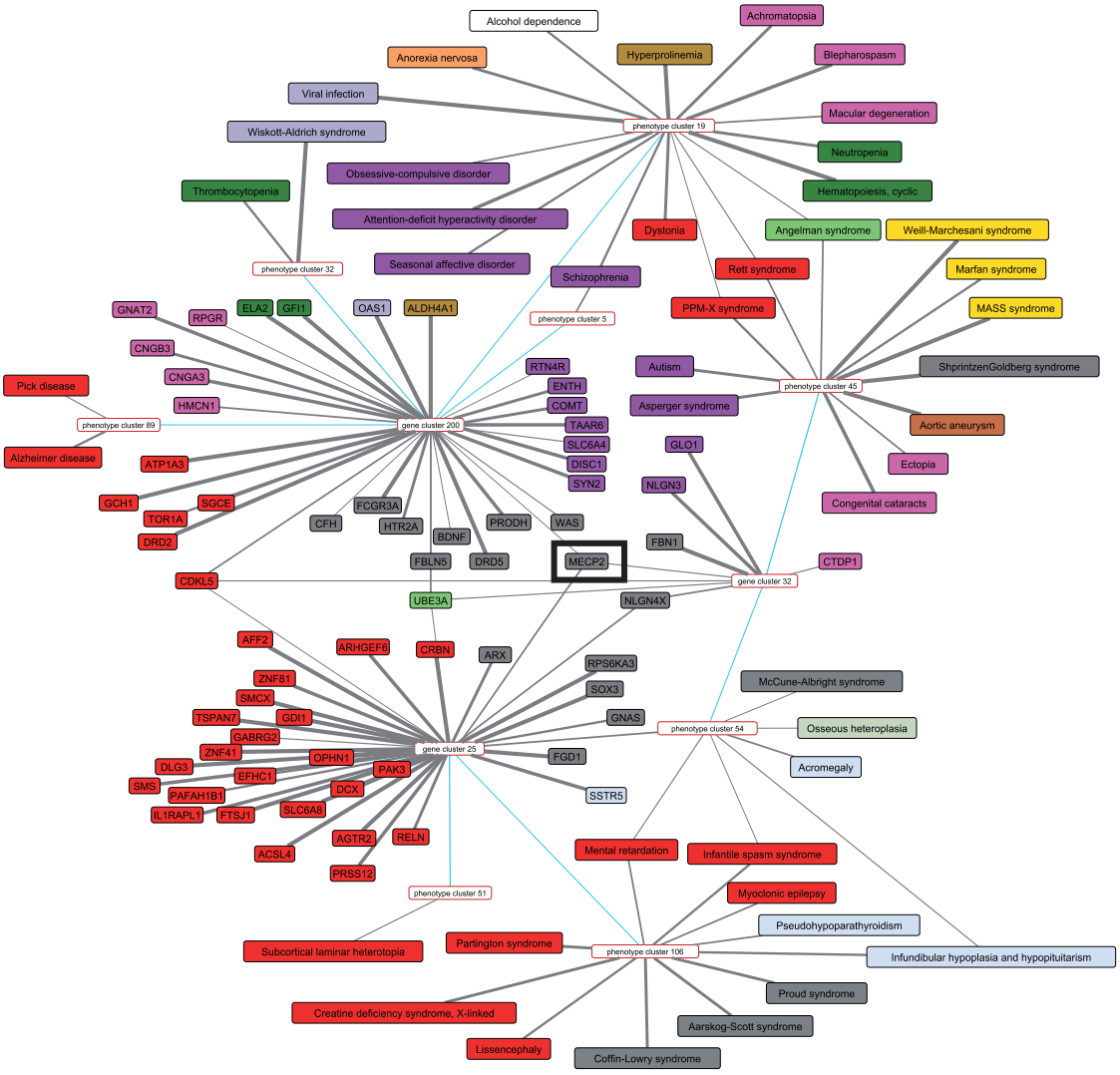


Figure 5.12: We illustrate the results – the backbone network and the nodes' degrees of membership to clusters, thresholded by a degree of membership of 0.2 – of the small-scale clustering in the neighborhood of *MECP2* using the fuzzy approach. Nodes are colored according to their disorder class annotations, for a legend see Figure 5.13. Blue edges indicate backbone interconnectivity, grey edges cluster assignment. Edge thickness indicates the degree of membership. *MECP2* is connected to three gene clusters mainly covering neurological (red) and psychiatric (purple) genes. The seven interconnected disease clusters also represent mainly psychiatric and neurological disorders. Also unclassified disorders are present such as e.g. *Alcohol dependence* (white), which is classified as a mental and behavioral disorder. In a broader sense, however, it can be considered as psychiatric disorder.

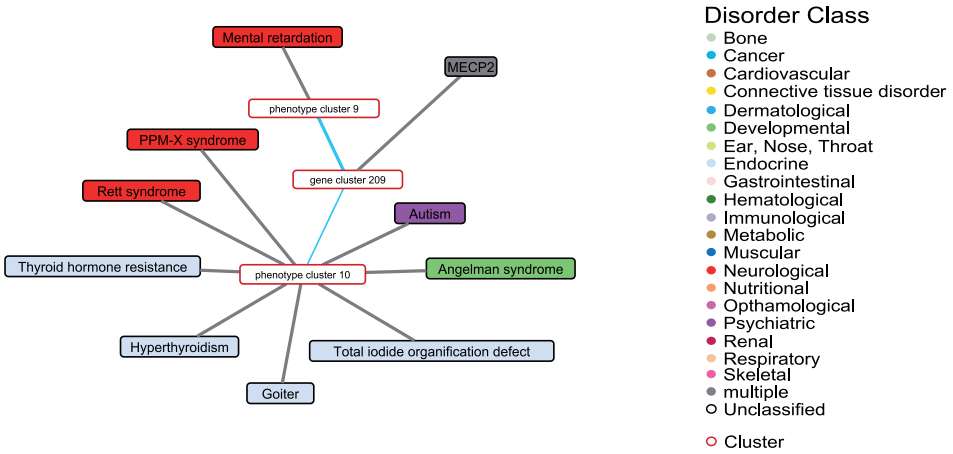


Figure 5.13: The results of the small-scale clustering in the neighborhood of *MECP2* using hard clustering. *MECP2* is assigned to gene cluster 209 which is connected to two disease clusters only. Although all associated disorders are identified correctly, in contrast to the fuzzy clustering no further information can be obtained from the decomposition.

Furthermore, we found *Anorexia nervosa* to be present in the analyzed clusters. It is annotated as a *nutritional* disorder by Goh et al. (2007), however it represents a life-threatening complex psychiatric disorder (Sylvester and Forman, 2008). Another so far unclassified disease, *Alcohol dependence*, was assigned to the interconnected cluster. It is classified as a mental and behavioral disorder (ICD-10 F10.2) and in a broader sense can be considered as psychiatric disorder.

In contrast, applying the hard clustering algorithm, *MECP2* was assigned to a single gene cluster which is connected to two disease clusters, see Figure 5.13. Although all associated disorders were identified correctly, no further information could be obtained from the clusters. However, Samaco et al. (2005) report an epigenetic overlap in autism-spectrum neurodevelopmental disorders as *MECP2* affects the regulation of *UBE3A* expression. These relations became immediately apparent in the cluster result of our fuzzy approach: Both genes were mutually assigned to gene cluster 25 that identifies the phenotypic and genotypic overlaps, whereas direct links to known connected genes are missing in the hard clustering.

5.5 Conclusions and outlook

The widespread application of high-throughput methods such as microarrays or next generation sequencing has considerably increased the amount of experimental data

and the information available in biomedical literature that is accessible to text-mining approaches. These data can usually be represented in terms of networks which over the last years have emerged as an invaluable tool for describing and analyzing complex systems. However, one has to take into account that network information is commonly available for various types of nodes. Especially integrative biological networks are k-partite (Goh et al., 2007, Yildirim et al., 2007).

Another important feature of biological networks is their hierarchical organization, implying that small groups of nodes organize in a hierarchical manner to increasingly larger groups on many different scales (Barabási and Oltvai, 2004, Clauset et al., 2008, Ravasz and Barabási, 2003). This necessitates the analysis of these objects on multiple resolution levels. Furthermore, many proteins or genes are pleiotropic, and often associated with many functions. Hence, clustering algorithms that assign elements into several functional modules are essential (Gasch and Eisen, 2002, Gulbahce and Lehmann, 2008, Palla et al., 2005).

This Chapter presented a novel, computationally efficient and scalable graph clustering algorithm that is capable to deal with all these described issues. Furthermore, it does not require any a priori knowledge about the data set. Results on a tripartite network, constructed by integrating the human disease network and protein complexes, demonstrated that we could identify and interconnect biologically meaningful clusters on different scales. Overlapping modules gave a more comprehensive picture of e.g. gene-disease connections than looking at disjoint clusters alone.

Summarizing, the proposed fuzzy clustering algorithm is suitable to compress and approximate the underlying topology of heterogeneous biological networks, which facilitates the understanding of such networks on multiple scales. In the future, this algorithm can be generalized to allow links within a partition, which will enable the decomposition of uni-partite, but also arbitrary colored graphs. This will allow to include the crucial missing elements of protein-protein interactions and transcription factor regulation into the studied biological networks. Currently, we explore whether the algorithm may also help to detect synonyms in large networks extracted by text-mining strategies. However, the method is readily applicable to many further problems from outside bioinformatics, as for instance in collaboration networks or when connecting customer-product relations with additional information sources.

6 Latent causes in biological systems: a proof of principle

Blind source separation techniques have been in the focus of rather intense research during the past decade. With the nowadays available robust algorithms in particular for the linear case – the basic ones were discussed in Section 1.1.2 – more and more people turn towards model generalizations and applications of BSS. One area of application for matrix factorization techniques and machine learning in general has been bioinformatics (Tarca et al., 2007). As we saw in Chapter 4, this field commonly deals with the analysis of large-scale high-throughput data sets from genomics and metabolomics. With the basic methods being robustly established, a trend in this field is to deal with smaller-scale fine-grained models closely integrating information from experiments (Alon, 2006). This systems-biology ansatz is increasingly bringing forth concise explanations for biological phenomena. This Chapter will derive some possible application areas and extensions of BSS in systems biology.

A key ingredient for the system modeling is in this context the detailed description of the system dynamics; we will consider mass action and Hill kinetics, but also a non-mechanistic neural-network approach. We then address the question of how to model unknown sources (latent variables) that can be inferred from the observations. Here, we think of previously unknown transcription factors or small molecules like microRNAs coupling to a gene regulatory network, or crosstalk of different metabolic pathways. We give proof of principles that in some situations such latent causes can be estimated even within a linear mixing model, however in more general settings this task leads to extensions of the standard BSS models.

In the following, we will denote measured time-courses by x and latent causes by h, dropping their explicit time dependence for simplicity of notation.

6.1 Mass action kinetics

The modeling of a macroscopic chemical system such as a metabolic or gene regulatory network is commonly simplified by neglecting the discrete nature of the participating reactants and their reactions. Therefore one introduces continuous concentrations as well as continuous reaction rates linked by a system of *ordinary*

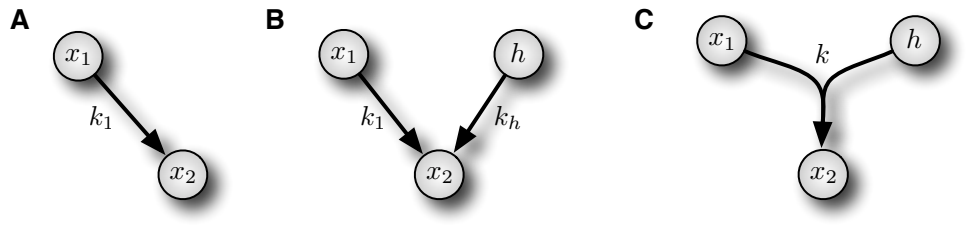


Figure 6.1: Reactions of order one and two: (A) shows a reaction, where x_1 reacts to x_2 with rate k_1, see Equation (6.1). In (B) we add a latent cause h, which also contributes to the x_2 concentration. (C) depicts a second-order reaction, where x_1 and h together react to x_2.

differential equations (ODE), the so-called *rate equations*. These rate laws are governed by the law of mass action, which says that the instantaneous rate of a reaction is proportional to the concentration of each reactant, raised to the power of its stoichiometry. The largest degree occurring in this equation is called the *order of the reaction*, which rarely exceeds two (Alon, 2006). Figure 6.1 shows the elementary reactions of order one and two, which we will take as starting point for the following discussion.

6.1.1 First-order mass action kinetics

For the rate equations of the direct conversion $x_1 \xrightarrow{k_1} x_2$ in Figure 6.1, we find

$$\begin{aligned} \dot{x_1} &= -\tau_1 x_1 - k_1 x_1 \quad \text{and} \\ \dot{x_2} &= -\tau_2 x_2 + k_1 x_1 \,. \end{aligned} \tag{6.1}$$

Here, the reaction runs with rate k_1. We additionally introduce decay terms quantifying the loss of reactants due to degradation with time constants $\tau_{1/2}$. If we now allow one latent cause h that produces x_2 in a first-order reaction (Figure 6.1(B)), we have to change the rate law for x_2 to

$$\dot{x_2} = -\tau_2 x_2 + k_1 x_1 + k_h h \,.$$

In this situation — provided that we know both the decay rates and x_1 — we can directly calculate the time-course of $k_h h$ via

$$k_h h = \dot{x_2} + \tau_2 x_2 - k_1 x_1 \,.$$

Obviously, we cannot determine the scale of the latent cause and its reaction rate k_h observing only x_1 and x_2.

However, if x_1 and h are assumed to be uncorrelated or independent (e.g. because they stem from different biological processes), we can even estimate k_1 from the observed time-courses: we may simply minimize the absolute correlation between

x_1 and $k_h h$, which has a unique solution in this case. Simulations with various parameter sets and shapes of latent causes showed that k_1 could be estimated up to an absolute error that depended on the size of the (in practice non-vanishing) correlation between x_1 and h. For instance, with the latent causes in Figure 6.2, randomly sampled $k_1 \in (0.1, 1)$ could be estimated with an absolute error of 0.005, averaged over 100 runs with random decay rates in $(0.1, 1)$.

In a general reaction network of this kind with N species x_i, let $x_j \xrightarrow{k_{ij}} x_i$ denote the reaction with rates k_{ij} vanishing if no reaction occurs. We may write the rate law for any of the reactants in a system with n latent causes as

$$\dot{x}_i + \tau_i x_i - \sum_{l=1}^{N} (k_{il} x_l - k_{li} x_i) = \sum_{j=1}^{n} a_{ij} h_j \,. \tag{6.2}$$

Denoting the left hand side of this equation by y_i and using matrix notation, we arrive at the common linear blind-source separation problem

$$\mathbf{y} = \mathbf{A}\mathbf{h} \,. \tag{6.3}$$

This problem can be solved with the various techniques from Section 1.1.2, if we assume that the latent causes fulfill certain conditions like decorrelation, statistical independence or non-negativity. Moreover, these properties can be used to estimate reaction rates analogously to the upper example. If at least one reactant x_u is known not to be affected by the latent ones, we may estimate the rates $\mathbf{k}$ occurring in its rate law as $\hat{\mathbf{k}} = \operatorname{argmin}_{\mathbf{k}} |\operatorname{corr}(y_u, x_u)|$. This estimate has — possibly many — indeterminacies depending on the network topology.

6.1.2 Example: a feed-forward loop

A frequently occurring motif in metabolic networks is the coherent feed-forward loop (Zhu and Qin, 2005), as shown in Figure 6.2. We study the first-order coupling of two statistically independent latent causes $h_{1/2}$ to this system, which for example may correspond to a separation of overlapping metabolic pathways. We write the rate equations for the depicted problem as linear mixing model

$$\begin{aligned} \dot{x}_2 + \tau_2 x_2 + k_1 x_1 - k_2 x_2 &= \sum a_{1j} h_j \\ \dot{x}_3 + \tau_3 x_3 + k_2 x_2 + k_3 x_1 &= \sum a_{2j} h_j \,. \end{aligned} \tag{6.4}$$

Hence, if the time-courses of the reactants are measured and the reaction rates and time constants are known, we can estimate $\dot{x}_i$ and reconstruct $h_{1/2}$ by assuming approximate statistical independence. In our simulations we performed ICA using the FastICA algorithm from Section 1.1.2.3. With this, in 100 simulations (rate

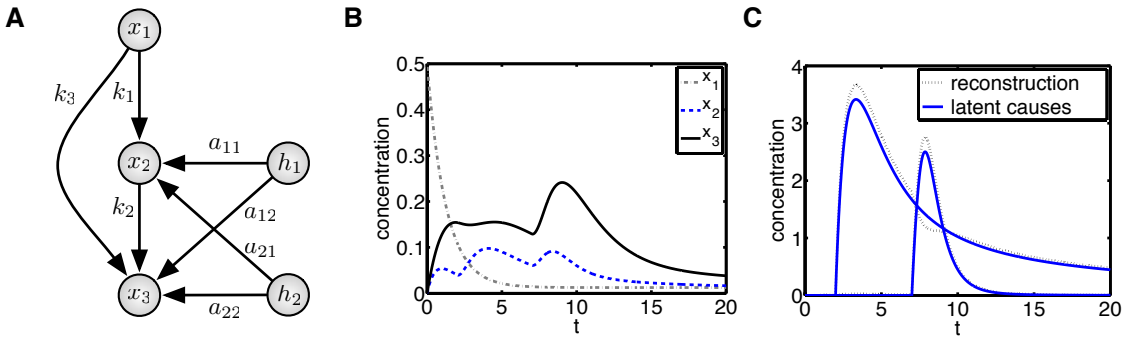


Figure 6.2: (A) A coherent feed-forward loop with two latent causes. (B) Simulated time-courses of the measured reactants for $k_1 = 0.3$, $k_2 = 0.7$, $k_3 = 0.4$ and degradation rates $\tau_1 = 0.1$, $\tau_2 = 0.6$, and $\tau_3 = 0.5$. (C) The two simulated latent causes and their reconstructions using FastICA. We used the solutions of $\dot{h}_1 = (2(t-1))^{-1} - h_1$ for $t > 2$ and $2\dot{h}_2 = (1+(t-7)^3)^{-1} - 3h_2$ for $t > 7$ as latent causes. As mixing matrix we chose $\mathbf{A} = (0.6, 0.4; 0, 0.6)$, where we found SNR of 22 and 36 dB.

parameters given in Figure 6.2) with random positive mixing coefficients, the latent causes could be reconstructed with a mean signal-to-noise ratio of 25 ± 9 dB.

6.1.3 Second-order mass action kinetics

The easiest and analytically solvable example for second-order mass action kinetics with a latent cause h is a reaction $x_1 + h \xrightarrow{k} x_2$, as shown in Figure 6.1(C). The corresponding ODE system contains only degradation terms and the product term representing the reaction:

$$\begin{aligned} \dot{x}_1 &= -\tau_1 x_1 - k x_1 h \,, \\ \dot{x}_2 &= -\tau_2 x_2 + k x_1 h \,. \end{aligned} \tag{6.5}$$

Hence, if we measure the time-courses of x_1 and x_2 and also know their decay rates we can determine

$$kh = \left(\dot{x}_2 + \tau_2 x_2\right) / x_1 \,,$$

given that $x_1 \neq 0$. If additionally the reaction rate k is known, we can extract the latent cause. Otherwise this constant and with it the scale of h remains an indeterminacy of this problem, even if e.g. we assume independence of the two reactants.

Now consider a cascade $x_1 \longrightarrow x_2 \longrightarrow x_3$ of second-order reactions. Here we could try to estimate the time-courses of two unobserved reaction partners $h_{1/2}$ that may take part in both reactions — e.g. two enzymes that have similar functions but

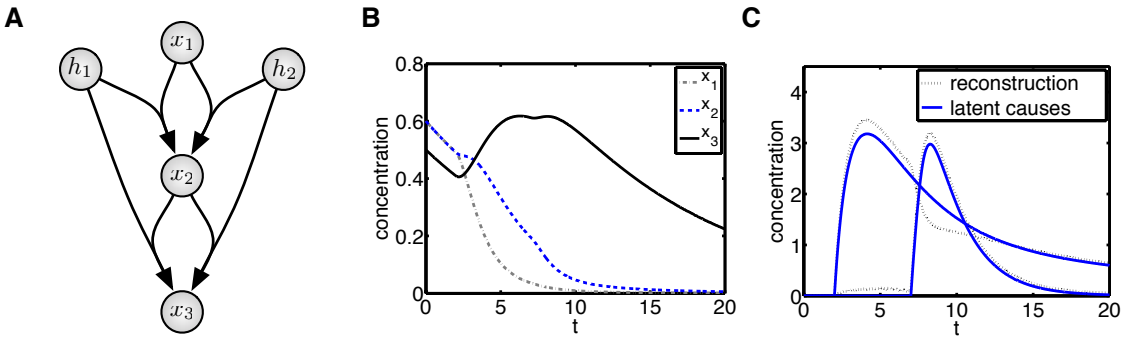


Figure 6.3: Second-order mass action kinetics: a three node cascade with 2 latent causes (A) and a simulated time course (B). All time constants were set to 0.01; we chose a mixing matrix $\mathbf{A} = (0.8, 0; 0.2, 0.5)$ and the latent causes from Figure 6.2. With FastICA we could reconstruct them with a SNR of 14 dB and 27 dB. The hidden influences and their reconstructions are plotted in (C).

are regulated by different processes. The rate equations for the first and the third reactant again lead to a linear mixing model

$$(\dot{x}_l + \tau_l x_l)/g_l = \sum_{m=1}^{2} a_{lk} h_k \quad \text{for } l = 1, 3.$$

As before, this can be easily solved by ICA. Figure 6.3 shows a simulated example.

However, these procedures only work in the discussed simple cases. If, for instance, there exists a direct second-order reaction $x_1 \longrightarrow x_3$ in the three node cascade or if we have a larger cascade with more than two influences, we arrive at a mixing model with time-dependent mixture coefficients. For a general network of second-order reactions $r : x_{r_a} + h_j \longrightarrow x_{r_b}$ with n latent causes we find rate equations

$$\begin{aligned}\dot{x}_i = &- \tau_i x_i - \sum_{r_a=i} \sum_{j=1}^{n} a_{r_a j} x_i h_j + \sum_{r_b=i} \sum_{j=1}^{n} a_{j r_a} x_{r_a} h_j \\ &+ \sum (\text{reactions of } x_i \text{ not affected by latent causes}).\end{aligned}$$

We therefore will have to solve the BSS problem

$$y_i = \sum_{j=1}^{n} \left(- \sum_{r_a=i} a_{r_a j} g_i + \sum_{r_b=i} a_{j r_a} g_{r_a} \right) h_j \,, \tag{6.6}$$

where we know the time-courses of all g. Hence, the estimation of latent causes in such systems leads to a novel class of linear mixing models with time-dependent coefficients.

Many reactions, especially enzymatic ones, are composed from processes of complex formation and dissociation like $x_1 + x_2 \longleftrightarrow C \longrightarrow x_3 + x_2$. This can be modeled by a detailed second-order mass action system. If however the concentration of x_2 (the enzyme) is much lower than the concentration of x_1 (the substrate), the dynamics can be approximated by *Michaelis-Menten kinetics* (cf. Alon (2006)). However, this type of reaction is formally a special case of Hill kinetics, which will be discussed in the following.

6.2 Gene regulatory networks

Molecular interactions on a genetic level are known to show a switch-like behavior. Motivated by the analysis of a promotor binding model (see e.g. (Alon, 2006)), they are usually described by activating and inhibiting *Hill functions*

$$H^+_{n,\theta}(x) = x^n(x^n + \theta^n)^{-1} \quad \text{and} \quad H^-_{n,\theta}(x) = (x^n + \theta^n)^{-1}\,. \tag{6.7}$$

As we have already seen in Section 3.2.4, where we only have taken into account activations, the *Hill coefficient* n is a measure for the co-operativity of the interaction. The *threshold parameter* θ corresponds to the concentration at half maximum activation.

6.2.1 A negative feedback loop modeled with Hill kinetics

As showcase for the issues we face in estimating latent causes in gene regulation, imagine the mutual inhibition of two genes shown in Figure 6.4, a bistable motif that is found in many developmental processes:

$$\begin{aligned} \dot{x}_1 &= -\tau_1 x_1 + H^-_{n_2,\theta_2}(x_2)\ , \\ \dot{x}_2 &= -\tau_2 x_2 + H^-_{n_1,\theta_1}(x_1)\ . \end{aligned} \tag{6.8}$$

Again we can measure the time-courses of both genes and know all parameters in the rate equations (6.8). If we now allow two latent causes h_1 and h_2, we first have to specify the logical operations that couple them to the system. The translation of the potentially complex 'molecular computations' taking place in a gene's promotor region into differential equations is still a field of active research, cf. Wittmann et al. (2009). However, purely additive as well as purely multiplicative coupling of all incoming regulations are widely-used approaches. This corresponds to a combination of the inputs by Boolean OR and AND logic, respectively. Both logics can only be transformed to the common linear mixing model, if any latent cause couples to measured genes with the same Hill function.

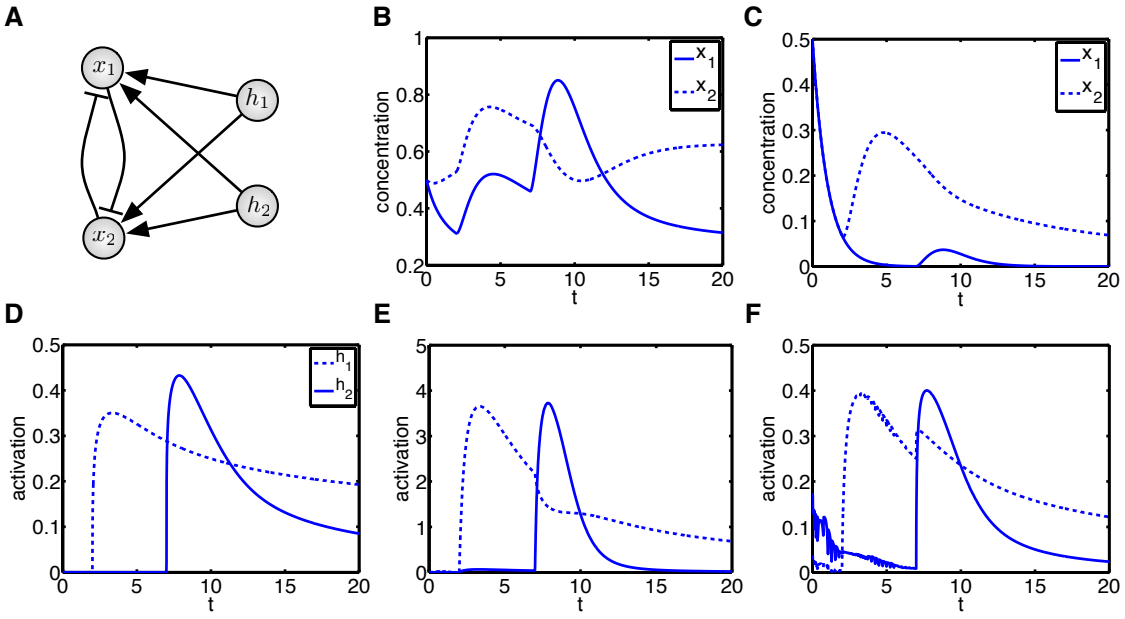


Figure 6.4: (A) Mutual inhibition of two species x_1 and x_2 with two latent causes h_1 and h_2. We added h_1 and h_2 as defined in Figure 6.2 as activators with mixing matrix $(1, 1; 1, 0)$ to the system. This led to completely different time-courses of g_1 and g_2, depending on whether we chose additive (B) or multiplicative (C) logic, despite using the same parameters (we took all n and τ equal to one, k_1 to k_4 were 0.4, 0.2, 0.4, 0.3) and initial conditions. Under the assumption of equal coupling, the Hill-transformed latent causes (D) could be reconstructed fairly well by FastICA: for OR logic (E) we got SNR of 18 dB and 34 dB, for AND logic ((F), sign corrected before exponentiating) SNR of 11 dB in both cases.

In this simplified situation we can proceed analogously to the last section and extract at least the Hill-transformed latent causes, as demonstrated in Figure 6.4. In the case of multiplicative coupling, we have rate equations

$$\dot{x}_1 = -\tau_1 x_1 + H_{n_2,\theta_2}(x_2) \prod_{j=1}^{2} H^{a_{1j}}_{n_{hj},\theta_{hj}}(h_j) \tag{6.9}$$

and symmetrically for g_2. Here, after bringing the degradation term to the left hand side, we can take the logarithm and subtract the regulation of the measured gene, arriving at

$$\log(\dot{x}_1 + \tau_1 x_1) - \log(H_{n_2,\theta_2}(x_2)) = \sum_{j=1}^{2} a_{1j} \log\left(H_{n_{hj},\theta_{hj}}(h_j)\right) . \tag{6.10}$$

Hence, in this situation we can determine the $H_{n_{hj},\theta_{hj}}(h_j)$ up to an exponent.

6.2.2 Gene regulatory networks with Hill kinetics

In general regulatory networks, a participating gene as well as a latent cause can act both as inhibitor $I^-(x)$ and activator $I^+(x)$ to a gene x. Additionally, the Hill parameters may vary in every interaction. For additive coupling, this situation leads to ODEs of the form

$$\dot{x}_i = -\tau_i x_i + \sum_{\sigma=\pm} \left(\sum_{x_j \in I^\sigma(x_i)} H^\sigma_{n_{ji},\theta_{ji}}(x_j) + \sum_{h_l \in I^\sigma(x_i)} a_{il} H^\sigma_{n_{li},\theta_{li}}(h_l) \right) . \tag{6.11}$$

If AND logic is used, we have to write products and exponents instead of the weighted sums of Hill functions, which after logarithmic transformation is converted to weighted sums again. With this, when rearranging the rate equations two new mixing models arise:

$$y_i = \sum_{j \in I^-(g_i)} \frac{a_{ij}}{\theta_{ij}^{n_j} + h_j^{n_j}} + \sum_{j \in I^+(g_i)} \frac{a_{ij} h_j^{n_j}}{\theta_{ij}^{n_j} + h_j^{n_j}} \qquad \text{(for OR logic)} \tag{6.12}$$

$$y_i = \sum_j a_{ij} \log\left(\theta_{ij}^{n_j} + h_j^{n_j}\right) - \sum_{j \in I^+(g_i)} a_{ij} \log\left(h_j^{n_j}\right) \qquad \text{(for AND logic)} \tag{6.13}$$

The use of different functions for activation and inhibition is a consequence of the non-negativity of Hill functions and their combinations. These models cannot be solved by linear BSS.

6.2.3 Continuous-time recurrent neural networks

In a recent approach to gene expression data analysis, Busch et al. (2008) use generalized continuous-time recurrent neural networks (CTRNN) as abstract dynamical models of regulatory systems. This leads to ODEs of the form

$$\dot{x}_i(t) = \tau_i \left(-x_i(t) + \sum_l \mathbf{W}_{li} \sigma\left(x_l(t - \Delta_l) - \theta_l\right) + I_i(t) \right) . \tag{6.14}$$

Here τ_i denotes the degradation rate, I_i an external input and θ_l are thresholds. Interactions are incorporated via the activation function

$$\sigma(x) = (1 + \mathrm{e}^{-ax})^{-1} ,$$

and additively connected by a real weight matrix $\mathbf{W}$. The delay constants Δ_l account for the time delay due to gene induction, transcription and translation. Of course, this approach has the advantage of a single function for both inhibition and

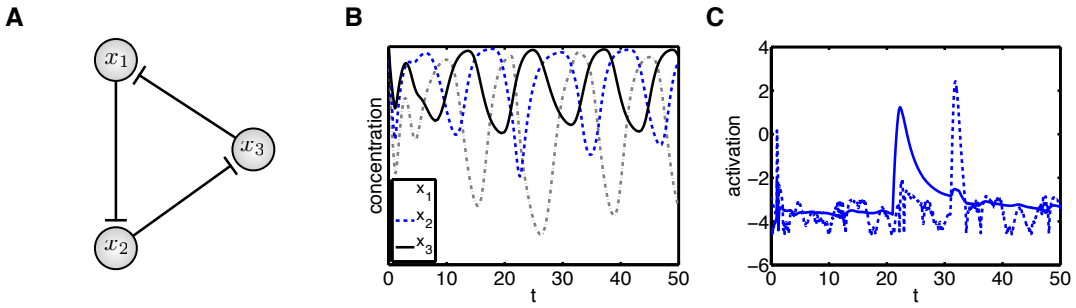


Figure 6.5: The repressilator (A), a three-node motif giving stable oscillations and (B) a simulated time-course with our two latent causes (with a time shift of 20, then oscillations without the h_l were stable), coupling to g_1 and g_2 with mixing matrix $\mathbf{W} = (20, -1; 10, 0)$. We chose $a = 1$, the same for all degradation rates, all time delays were 5, and the non-vanishing interaction weights are $20, 15, 10$. (C) shows reconstruction of the transformed latent causes with FastICA.

activation, but on the other hand we loose the biological model and direct interpretability. However, for the estimation of latent causes h_j in such an ODE system we obtain the mixing model

$$y_i = \sum_j \mathbf{W}_{ji} \sigma \left(h_j \left(t - \Delta_j \right) - \theta_j \right) . \tag{6.15}$$

Here the time delays will remain an indeterminacy. In the case of equal coupling, this again reduces to a problem solvable using ICA. In Figure 6.5 we discuss an example for this, where two latent causes couple to the so-called repressilator motif (Elowitz and Leibler, 2000). If, which is more realistic, we assume interaction-specific delay times Δ_{ij}, we can use convolutive ICA (Hyvärinen et al., 2001). However, linear BSS is unable to estimate additional parameters such as the k_j and exponents in σ.

6.3 Conclusions and outlook

With the availability of more and more quantitative data the estimation of latent causes in biological systems is an upcoming challenge and crucial for the interpretation of many experiments. For simple processes in mass action kinetics and first-order reaction networks we showed that this task leads to a linear BSS problem. As a proof of principle we reconstructed latent causes from artificial data using ICA.

In the more delicate case of second-order reactions and regulatory interactions, where we have to deal with sigmoidal input functions that may be connected by different logical operations, new mixing models arose. The analysis of these models

and the estimation of the occurring parameters in non-linear situations necessitates a treatment within a Bayesian framework, similar to e.g. (Duarte et al., 2009, Vyshemirsky and Girolami, 2008, Wilkinson, 2006).

In practice, we will be confronted with networks involving reactions and interaction of more than one of the discussed types, leading to a variety of hybrid models and corresponding likelihood functions. Moreover we will only observe a fraction of the known network elements, so there exists an added layer of model inference. These involved learning problems – and more generally, latent causes in arbitrary dynamical systems – may be solved by extensions of BSS and Bayesian methods.

From the network perspective, this Chapter is the first attempt to the estimation of missing nodes and their connections in complex networks, a task that can not be solved without access to some dynamical properties of the system. In contrast, the estimation of missing links can in principle be tackled with topological properties alone, as Clauset et al. (2008) recently demonstrated in an impressive fashion.

7 Conclusions and summary of main contributions

The lasts years have brought enormous progress in most scientific disciplines, ranging from modern experimental technologies and exploding computational resources to novel theoretical and algorithmic tools. This provides us novel possibilities to gain a deeper understanding of complex systems, some of which confront us with the most fascinating challenges in all of science. Especially examples from the fast developing biological sciences have been the driving forces in unraveling the underlying principles governing highly interconnected systems from cells to animal populations or brain formation. However, we are still at the beginning. Likewise, major problems currently facing mankind are social and economic in nature. Will we be able to decipher the common fundamental laws?

Many classical challenges seem to be accessible in the near future, partially within the framework of complex networks. During the time I spent in research – spanning more than four years now – I could witness network science maturing to a new branch of statistical physics, with own conferences and sections in the prestigious journals of the field. We have shown in our literature review in Chapter 1, but also with our own research e.g. on economics in Chapter 2, that this language indeed has proven to be the appropriate formulation in many examples.

The goal of this thesis was to bring together methods for describing complex systems, mainly this language of complex network science, and machine learning techniques. These automated approaches play a crucial role whenever dealing with the occurring large-scale data sets and ensure the necessary objectivity in their analysis. The results presented in the previous six Chapters showed that these two fields can considerably profit from each other also beyond the well-studied shared interest in community detection. The motivation for each of the individual projects we followed in the course of this thesis however arose from concrete problems in different fields of science: we covered empirical trade, mathematical biology, bioinformatics, and finally molecular biology. With this, the developed techniques are not only valuable from a theoretical point of view, but their subsequent application also led to results that are of impact to a broad range of disciplines.

As already explained in the introductory Chapter 1, each Chapter of this thesis

was intended to be as self-contained as possible. Therefore, we would like to refer the reader to the respective Chapters for detailed conclusions and suggestions for future work. Here we want to briefly formulate what we consider the main scientific contributions of this thesis:

- Starting from an economic question, Chapter 2 analyzed node centralities in input-output networks, which we built from data collected by the OECD. So far, these networks have not been studied in the complex network community. Due to the large number of countries available and the possibility to observe the evolution over time we expect this data set to be rich source for other researchers both in economics and statistical physics. Input-output networks have special properties that sets them apart from other examples in the field, for instance they contain strong self-loops. To the best of our knowledge, this Chapter derived the first measures of node centrality that are applicable to such networks. Moreover, our hierarchical clustering of countries based on node centralities is a novel approach in empirical economics. It showed how a machine learning technique can be employed to cope with the high complexity of the network data available in this field.

- From the viewpoint of theoretical biology, Chapter 3 formulated a generic model of signal transduction and introduced the concept of effective parameters therein. It provided general solutions that allow to immediately calculate these objets in a given system in the case of Heaviside and linear activation functions. We demonstrated that the effective parameters are valuable tools not only for interpreting the biological properties of the system under study, but also for stabilizing the estimation of parameters when fitting experimental data to quantitative models. From a physicist's point of view, Chapter 3 identified the relationship between interaction topology and system dynamics in those special cases of hierarchical systems.

- The GraDe algorithm in Chapter 4 is the first matrix factorization algorithm that is based on prior knowledge. We proved identifiability in our factorization model, where we posed constraints that are based on the novel concept of graph-delayed correlations. In both, simulations with artificial data and real-world microarray data, we demonstrated the applicability as well as robustness of the proposed approach. Applying GraDe to data from a time-course experiment on *IL-6* stimulated primary hepatocytes resulted in new biological insights. We observed that *IL-6* activates cell cycle progression, while it down-regulates metabolic processes and programmed cell death. Therefore, *IL-6* mediated priming renders hepatocytes more responsive towards cell prolifera-

tion and reduces expenditures for the energy metabolism. The methodological strength of the proposed approach is two-fold: first, it naturally arises from a network approximation of the general ODE model of gene regulation. Second, instead of ignoring the sample dependencies in biological high-throughput data by assuming i.i.d. samples, we explicitly model them.

- To the best of our knowledge, the NMF-type community detection algorithm we developed in Chapter 5 is the first method that allows detection of overlapping communities in k-partite graphs. We analyzed the algorithm's performance and stability, and demonstrated its applicability both in toy data and the real-world example of a gene-disease-protein complex graph. Employing functional annotations we showed that the communities as well as their connecting backbones are biologically reasonable.

- Chapter 6 formulated perspectives for novel developments in the BSS field: The estimation of latent causes in biological systems is an upcoming challenge and crucial for the interpretation of any systems biology experiment. For simple processes in mass action kinetics and first-order reaction networks, but also some examples from gene regulation modeled by Hill kinetics or within a neural network framework we showed that this task leads to a linear BSS problem. By reconstructing latent causes from artificial data using ICA we gave proof of principles that this strategy is indeed practicable. We showed how more complex situations lead to non-linear mixing models which will be analyzed in a follow-up project. From the network scientist's perspective, Chapter 6 is the first, preliminary, attempt to the estimation of missing nodes and their connections in complex networks.

The algorithms and methods developed in this thesis are readily applicable to any other example. However, as always in science, this thesis yielded novel methods and provided first answers to the problems it originated from, but in the end even more new questions arose. In particular, our proof of principles on latent causes in biological systems is a starting point for a much larger project that will be promoted by a European Research Council grant over the next years. Here, the inference of time courses of multiple latent causes as well as their targets in more complex dynamical, or even stochastic systems will be tackled within a fully Bayesian formulation of the non-linear source separation problems. A second working package will be devoted to the additional efficient inclusion of prior knowledge into the used methods. This will help to regularize the upcoming estimation problems. To this end, various extensions of our GraDe algorithm can be considered. Finally, both approaches will be integrated into a combined framework. Bayesian source separation techniques

will also be crucial for extensions of our fuzzy community detection algorithm since they enable the automatic determination of the optimal number of communities to be extracted.

The novel techniques can then be tested within the various systems biology models. We plan, for instance, the application to a differentiation model of embryonic stem cell lineage segregation. Here, possible latent causes include unknown transcription factors and regulation by small molecules like microRNAs, but also off-target effects of drugs. In the end, this may lead to efficient differentiation protocols for cell replacement therapy, a promising candidate for the treatment of a wide range of debilitating diseases like type I diabetes or Parkinson's disease.

Bibliography

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, USA, 2008.

A. Agresti. *Introduction to categorical data analysis*. John Wiley and Sons, New York, 1996.

Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, New Jersey, 1993.

R. Albert, H. Jeong, and A.-L. Barabási. The diameter of the world wide web. *Nature*, 401(6749): 130–131, 1999.

J. H. Albrecht and L. K. Hansen. Cyclin D1 promotes mitogen-independent cell cycle progression in hepatocytes. *Cell growth & differentiation*, 10(6):397–404, 1999.

U. Alon. *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall/CRC, 2006.

L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley. Classes of small-world networks. *Proceedings of the National Academy of Sciences of the United States of America*, 97(21):11149–11152, 2000.

A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman Divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

A.-L. Barabási. *Linked*. Perseus, Cambridge, Massachusetts, 2002.

A.-L. Barabási. *Bursts: The Hidden Pattern Behind Everything We Do*. Dutton Adult, New York, 2010.

A.-L. Barabási and R. Albert. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.

A.-L. Barabási and Z. N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.

A.-L. Barabási, R. Albert, and H. Jeong. Mean-field theory for scale-free random networks. *Physica A*, 272:173–187, 1999.

M. J. Barber. Modularity and community detection in bipartite networks. *Physical Review E*, 76 (6):066102, 2007.

A. Barrat and M. Weigt. On the properties of small-world network models. *European Physics Journal B*, 13:574–560, 2000.

T. Baskaran, F. Blöchl, T. Brück, and F. J. Theis. The Heckscher-Ohlin Model and the Network Structure of International Trade. *International Review of Economics and Finance, in press*, 2010.

H. Bateman. The solution of a system of differential equations occurring in the theory of radioactive transformations. *Proceedings of the Cambridge Philosophical Society*, 15:423–427, 1910.

S. Bauer, S. Grossmann, M. Vingron, and P. N. Robinson. Ontologizer 2.0–a multifunctional tool for go term enrichment analysis and data exploration. *Bioinformatics*, 24(14):1650–1651, 2008.

A. Bavelas. A mathematical model for small group structures. *Human Organisation*, 7:16–30, 1948.

A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on Signal Processing*, 45(2):434–444, 1997.

Y. Benjamini and Y. Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

M. W. Berry, M. Browne, A. N. Langville, P. V. Pauca, and R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52(1):155–173, 2007.

J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

K. Bhattacharya, G. Mukherjee, J. Saramaki, K. Kaski, and S. S. Manna. The international trade network: weighted network analysis and modelling. *Journal of Statistical Mechanics*, 2008(02): P02002, 2008.

C. M. Bishop. *Pattern recognition and machine learning*. Springer New York, 2006.

F. Black. *Business Cycles and Equilibrium*. Basil Blackwell, New York, 1987.

F. Blöchl. Independent graph analysis. Diploma thesis, Universität Regensburg, 2007.

F. Blöchl, A. Rascle, J. Kastner, R. Witzgall, E. W. Lang, and F. J. Theis. Are we to integrate previous information into microarray analyses? Interpretation of a Lmx1b-knockout experiment. In J. M. Górriz, E. W. Lang, and J. Ramírez, editors, *Recent Progress in Biomedical Signal Processing*. Bentham Science Publishers, 2010.

S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4-5):175–308, 2006.

B. Bollobas. *Modern Graph Theory*. Springer, Berlin, 1998.

B. Bollobás. *Random graphs*. Cambridge Studies in Advanced Mathematics, Cambridge, UK, 2001.

H. Bolouri and E. Davidson. Transcriptional regulatory cascades in development: Initial rates, not steady state, determine network kinetics. *Proceedings of the National Academy of Sciences of the United States of America*, 100(16):9371–9376, 2003.

P. Bonacich. Power and centrality: a family of measures. *American Journal of Sociology*, 92: 1170–1182, 1987.

S. P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, 2005.

S. P. Borgatti. A graph-theoretic perspective on centrality. *Social Networks*, 28(4):466–484, 2006.

R. Boscolo, C. Sabatti, J. C. Liao, and V. P. Roychowdhury. A generalized framework for network component analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(4):289–301, 2005.

U. Brandes. A Faster Algorithm for Betweenness Centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

D. Brockmann and F. J. Theis. Money circulation, trackable items, and the emergence of universal human mobility patterns. *IEEE Pervasive Computing*, pages 28–35, 2008.

D. Brockmann, L. Hufnagel, and T. Geisel. The scaling laws of human travel. *Nature*, 439(7075): 462, 2006.

R. Burt. Positions in networks. *Social Forces*, 55:93, 1976.

H. Busch, D. Camacho-Trullio, Z. Rogon, K. Breuhahn, P. Angel, R. Eils, and A. Szabowski. Gene network dynamics controlling keratinocyte migration. *Molecular Systems Biology*, 4(200), 2008.

E. Camon, M. Magrane, D. Barrell, V. Lee, E. Dimmer, J. Maslen, D. Binns, N. Harte, R. Lopez, and R. Apweiler. The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology. *Nucleic acids research*, 32(Database issue):D262–6, 2004.

M. Campos, C. B. Abdalla, A. V. dos Santos, C. P. Pestana, J. M. dos Santos, C. B. Santos-Reboucas, and M. M. G. Pimentel. A MECP2 mutation in a highly conserved aminoacid causing mental retardation in a male. *Brain Development*, 31(2):176–178, 2009.

J. F. Cardoso. Higher-order contrasts for independent component analysis. *Neural Computation*, 11:157–192, 1999.

J. F. Cardoso and A. Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 17(1):161–164, 1995.

A. Cichocki and S. Amari. *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications.* Wiley, New York, 2002.

A. Clauset, C. Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.

P. Comon. Independent component analysis - a new concept? *Signal Processing*, 36:287–314, 1994.

F. Davidescu and S. Jŏrgensen. Structural parameter identifiability analysis for dynamic reaction networks. *Chemical Engineering Science*, 63(19):4754–4762, 2008.

B. Davies. *Integral Transforms and Their Applications.* Springer Verlag, 2002.

H. de Jong, J. L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselmann. Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bulletin of Mathematical Biology*, 66(2):301–340, 2004.

D. del Vecchio, A. J. Ninfa, and E. D. Sontag. Modular cell biology: retroactivity and insulation. *Molecular Systems Biology*, 4(161), 2008.

L. Denis-Vidal, G. Joly-Blanchard, and C. Noiret. System Identifiability (Symbolic Computation) and Parameter Estimation (Numerical Computation). *Numerical Algorithms*, 34(2):283–292, 2003.

K. Devarajan. Nonnegative matrix factorization: an analytical and interpretive tool in computational biology. *PLoS Computational Biology*, 4(7):e1000029, 2008.

I. S. Dhillon and S. Sra. Generalized nonnegative matrix approximations with bregman divergences. In *Proc. NIPS 2005*, pages 283–290, 2006.

J. Diesner, T. L. Frantz, and K. M. Carley. Communication Networks from the Enron Email Corpus "It's Always About the People. Enron is no Different". *Computational and Mathematical Organization Theory*, 11(3):201–228, 2005.

S. N. Dorogovtsev and J. F. F. Mendes. Language as an evolving word web. *Proceedings of The Royal Society of London. Series B, Biological Sciences*, 268(1485):2603–2606, 2001.

S. N. Dorogovtsev and J. F. F. Mendes. *Evolution of Networks*. Oxford University Press, 2003.

J. P. K. Doye and C. P. Massen. Characterizing the network topology of the energy landscapes of atomic clusters. *Journal of Chemical Physics*, 122:084105. 14 p, 2004.

S. Drulhe, G. Ferrari-Trecate, H. de Jong, and A. Viari. Reconstruction of Switching Thresholds in Piecewise-Affine Models of Genetic Regulatory Networks. *Lecture Notes in Computer Science*, 3927:184–199, 2006.

L. T. Duarte, C. Jutten, and S. Moussaoui. Ion-selective electrode array based on a bayesian nonlinear source separation method. In *ICA '09: Proceedings of the 8th International Conference on Independent Component Analysis and Signal Separation*, pages 662–669, Berlin, Heidelberg, 2009. Springer-Verlag.

J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57, 1973.

A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

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 of the United States of America*, 95(25):14863–8, 1998.

M. B. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 4(6767):335–338, 2000.

P. Erdös and A. Rényi. On Random Graphs. I. *Publicationes Mathematicae*, 6:290–297, 1959.

E. Estrada, D. J. Higham, and N. Hatano. Communicability betweenness in complex networks. *Physica A*, 388(5):764–774, 2009.

G. Fagiolo, J. Reyes, and S. Schiavo. World-trade web: Topological properties, dynamics, and evolution. *Physical Review E*, 79(3):036115, 2009.

N. Fausto. Liver regeneration. *Journal of hepatology*, 32(1 Suppl):19–31, 2000.

R. Ferrer-i-Cancho and R. V. Solé. The small world of human language. *Proceedings of The Royal Society of London. Series B, Biological Sciences*, 268(1482):2261–2265, 2001.

R. Ferrer-i-Cancho, C. Janssen, and R. V. Solé. Topology of technology graphs: Small world patterns in electronic circuits. *Physical Review E*, 64(4):046119+, 2001.

C. Févotte and C. Doncarli. Two contributions to blind source separation using time-frequency distributions. *IEEE Signal Processing Letters*, 11(3):386–389, 2004.

C. Févotte and F. J. Theis. Orthonormal approximate joint block-diagonalization. Technical report, GET/Télécom Paris, 2007.

M. Fiorio, M. Tinazzi, A. Scontrini, C. Stanzani, M. Gambarin, A. Fiaschi, G. Moretto, G. Fabbrini, and A. Berardelli. Tactile temporal discrimination in patients with blepharospasm. *Journal of Neurology, Neurosurgery & Psychiatry*, 79(7):796–798, 2008.

S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the United States of America*, 104(1):36, 2007.

R. Franke, M. Müller, N. Wundrack, E.-D. Gilles, S. Klamt, T. Kähne, and M. Naumann. Host-pathogen systems biology: logical modelling of hepatocyte growth factor and Helicobacter pylori induced c-Met signal transduction. *BMC Systems Biology*, 2(4), 2008.

L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:31–41, 1977.

L. C. Freeman. Centrality in social networks: Conceptual clarification ı. *Social Networks*, 1(4): 215–239, 1979.

L. C. Freeman. *The Development of Social Network Analysis.* Empirical Press, Vancouver, 2006.

L. C. Freeman, S. P. Borgatti, and D. R. White. Centrality in valued graphs: A measure of betweenness based on network flow. *Social Networks*, 13(2):141–154, 1991.

T. S. Gardner, D. di Bernardo, D. Lorenz, and J. J. Collins. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, 301(5629):102–5, 2003.

N. R. Garge, G. P. Page, A. P. Sprague, B. S. Gorman, and D. B. Allison. Reproducible clusters from microarray research: whither? *BMC Bioinformatics*, 6 Suppl 2:S10, 2005.

D. Garlaschelli and M. I. Loffredo. Fitness-dependent topological properties of the world trade web. *Physical Review Letters*, 93:188701, 2004.

D. Garlaschelli and M. I. Loffredo. Structure and evolution of the world trade network. *Physica A*, 355(1):138–144, 2005.

D. Garlaschelli, G. Caldarelli, and L. Pietronero. Universal scaling relations in food webs. *Nature*, 423(6936):165–268, 2003.

A. P. Gasch and M. B. Eisen. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome biology*, 3(11):1–22, 2002.

J. Gauldie, C. Richards, D. Harnish, P. Lansdorp, and H. Baumann. Interferon beta 2/B-cell stimulatory factor type 2 shares identity with monocyte-derived hepatocyte-stimulating factor and regulates the major acute phase protein response in liver cells. *Proceedings of the National Academy of Sciences of the United States of America*, 84(20):7251, 1987.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis.* Chapman and Hall, New York, 2004.

R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, and J. Gentry. Bioconductor: open software development for computational biology and bioinformatics. *Genome biology*, 5(10):R80, 2004.

G. Ghoshal, V. Zlatic, G. Caldarelli, and M. E. J. Newman. Random hypergraphs and their applications. *Physical Review E*, 79(6 Pt 2), 2009.

M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826, 2002.

L. Glass and S. A. Kauffman. The logical analysis of continuous, non-linear biochemical control networks. *Journal of Theoretical Biology*, 39(1):103–129, 1973.

K.-I. Goh, M. E. Cusick, D. Valle, B. Childs, M. Vidal, and A.-L. Barabási. The human disease network. *Proceedings of the National Academy of Sciences of the United States of America*, 104 (21):8685–8690, 2007.

G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, 1996.

R. Grassi. Vertex centrality as a measure of information flow in italian corporate board networks. *Physica A*, 389:2455–2464, 2010.

P. Gruber and F. J. Theis. Grassmann clustering. In *Proc. EUSIPCO 2006*, Florence, Italy, 2006.

J.-L. Guillaume and M. Latapy. Bipartite structure of all complex networks. *Information Processing Letter*, 90(5):215–221, 2004.

N. Gulbahce and S. Lehmann. The art of community detection. *Bioessays*, 30(10):934–938, 2008.

H. Haken. *Synergetics - An Introduction. Nonequilibrium Phase Transitions and Self-Organization in Physics, Chemistry and Biology*. Springer Verlag Berlin, 1977.

H. Haken. *Advanced synergetics*. Springer Berlin, 1983.

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

S. Hengl, C. Kreutz, J. Timmer, and T. Maiwald. Data-based identifiability analysis of non-linear dynamical models. *Bioinformatics*, 23(19):2612–2618, 2007.

C. A. Hidalgo, B. Klinger, A.-L. Barabási, and R. Hausmann. The product space conditions the development of nations. *Science*, 317:482–487, 2007.

P. Holme, M. Huss, and H. Jeong. Subnetwork hierarchies of biochemical pathways. *Bioinformatics*, 19(4):532–538, 2003.

P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

E. Hüllermeier and M. Rifqi. A fuzzy variant of the Rand index for comparing clustering structures. *Proceedings of the IFSAEUSFLAT*, pages 1294–1298, 2009.

A. Hyvärinen. Fast and robust fixedpoint algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

G. Iori, G. De Masi, O. V. Precup, G. Gabbi, and G. Caldarelli. A network analysis of the italian overnight money market. *Journal of Economic Dynamics and Control*, 32(1):259–278, 2008.

M. Isaac, A. Janca, and N. Sartorius. *ICD-10 Symptom Glossary for Mental Disorders.* World Health Organization, Division of Mental Health, Geneva, 1994.

A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411:41–42, 2001.

M. Johannes, J. Brase, H. Fröhlich, S. Gade, M. Gehrmann, M. Fälth, H. Sültmann, and T. Beißbarth. Integration of pathway knowledge into a reweighted recursive feature elimination approach for risk stratification of cancer patients. *Bioinformatics*, 26(17):2136, 2010.

M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, and Y. Yamanishi. KEGG for linking genomes to life and the environment. *Nucleic acids research*, 36(Database Issue):480–484, 2008.

G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: application in vlsi domain. In *Proc. DAC '97*, pages 526–529. ACM Press, 1997.

L. Katz. A new index derived from sociometric data analysis. *Psychometrika*, 18:39–43, 1953.

L. Kaufman and P. J. Rousseew. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Series in Probaility and Statistics. John Wiley & Sons, 2005.

B. Kaufmann. Fitting a sum of exponentials to numerical data. *arXiv:physics/0305019v1*, 2003.

B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, 49(1):291–307, 1970.

M. K. Kerr and G. A. Churchill. Experimental design for gene expression microarrays. *Biostatistics*, 2(2):183–201, 2001.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

S. Klamt and A. von Kamp. Computing paths and cycles in biological interaction graphs. *BMC Bioinformatics*, 10(1):181, 2009.

S. Klamt, J. Saez-Rodriguez, J. Lindquist, L. Simeoni, and E. Gilles. A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC Bioinformatics*, 7(1):56, 2006.

S. Klamt, U.-U. Haus, and F. J. Theis. Hypergraphs and cellular networks. *PLoS Computational Biology*, 5(5):e1000385, 2009.

M. Krull, S. Pistor, N. Voss, A. Kel, I. Reuter, D. Kronenberg, H. Michael, K. Schwarzer, A. Potapov, C. Choi, O. Kel-Margoulis, and E. Wingender. TRANSPATH: an information resource for storing and visualizing signaling pathways and their pathological aberrations. *Nucleic acids research*, 34(Database issue):D546–51, 2006.

A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5):056117, 2009.

A. N. Langville, C. D. Meyer, and R. Albright. Initializations for the nonnegative matrix factorization. Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA USA, 2006.

M. Latapy, C. Magnien, and N. D. Vecchio. Basic notions for the analysis of large two-mode networks. *Social networks*, 30(1):31–48, 2008.

V. Latora and M. Marchiori. Economic small-world behavior in weighted networks. *European Physics Journal B*, 32(2):249–263, 2003.

V. Latora and M. Marchiori. A measure of centrality based on network efficiency. *New Journal of Physics*, 9(6):188, 2007.

H. J. Leavitt. Some effects of certain communication patterns on group performance. *Journal of Abnormal and Social Psychology*, 46:38–50, 1951.

R. R. Lebel, M. May, S. Pouls, H. A. Lubs, R. E. Stevenson, and C. E. Schwartz. Non-syndromic x-linked mental retardation associated with a missense mutation (p312l) in the fgd1 gene. *Clinical Genetics*, 61(2):139–145, 2002.

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 40:788–791, 1999.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2001.

S. Legewie, H. Herzel, H. V. Westerhoff, and N. Bluthgen. Recurrent design patterns in the feedback regulation of the mammalian signalling network. *Molecular Systems Biology*, 4, 2008.

W. Leontief. *Input-Output Economics (2nd ed.)*. Oxford University Press, New York, 1986.

X. Li, Y.-Y. Jin, and G.-R. Cheng. On the topology of the world exchange arrangements web. *Physica A*, 343:573–582, 2004.

J. C. Liao, R. Boscolo, Y.-L. Yang, L. M. Tran, C. Sabatti, and V. P. Roychowdhury. Network component analysis: reconstruction of regulatory signals in biological systems. *Proceedings of the National Academy of Sciences of the United States of America*, 100(26):15522–7, 2003.

W. Liebermeister. Linear modes of gene expression determined by independent component analysis. *Bioinformatics*, 18(1):51–60, 2002.

S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28 (2):129–137, 1982.

B. Long, X. Wu, Z. Zhang, and P. S. Yu. Unsupervised learning on k-partite graphs. In *Proc. SIGKDD 2006*, pages 317–326, 2006.

J. B. Long and C. I. Plosser. Real business cycles. *Journal of Political Economy*, 91:39–69, 1983.

L. Lovász. Random walks on graphs: a survey. *Combinatorics*, 2(80):1–46, 1993.

D. R. L. Lutter, P. Ugocsai, M. Grandl, E. Orso, F. J. Theis, E. W. Lang, and G. Schmitz. Analyzing m-csf dependent monocyte/macrophage differentiation: Expression modes and meta-modes derived from an independent component analysis. *BMC Bioinformatics*, 9(1):100, 2008.

J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

R. Malmgren, D. Stouffer, A. Campanharo, and L. Amaral. On universality in human correspondence activity. *Science*, 325:1696–1700, 2009.

K. V. Mardia, J. M. Bibby, and J. T. Kent. *Multivariate analysis*. Academic Press, 1979.

A. R. Mashaghi, A. Ramezanpour, and V. Karimipour. Investigation of a protein complex network. *The European Physical Journal B*, 41:113–121, 2004.

S. Maslov and K. Sneppen. Specificity and stability in topology of protein networks. *Science*, 296 (5569):910–913, 2002.

E. J. McCluskey and T. C. Bartee. *A survey of switching circuit theory*. McGraw-Hill, 1962.

M. Mete, F. Tang, X. Xu, and N. Yuruk. A structural approach for finding functional modules from large biological networks. *BMC Bioinformatics*, 9 Suppl 9:S19, 2008.

S. Milgram. The small world problem. *Psychology Today*, 2:60, 1967.

R. E. Miller and P. D. Blair. *Input-output analysis : foundations and extensions*. Cambridge University Press, Cambridge, UK, 2nd ed. edition, 2009.

R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

A. Misbahuddin, M. R. Placzek, K. R. Chaudhuri, N. W. Wood, K. P. Bhatia, and T. T. Warner. A polymorphism in the dopamine receptor drd5 is associated with blepharospasm. *Neurology*, 58(1):124–126, 2002.

F. S. Mishkin. *The economics of money, banking, and financial markets*. Addison-Wesley, 2007.

L. Molgedey and H. G. Schuster. Separation of a mixture of independent signals using time-delayed correlations. *Physical Review Letters*, 72(23):3634–3637, 1994.

R. Montanez, M. A. Medina, R. V. Solé, and C. Rodríguez-Caso. When metabolism meets topology: Reconciling metabolite and reaction networks. *Bioessays*, 32(3):246–256, 2010.

M. Müller-Linow, C. C. Hilgetag, and M.-T. Hütt. Organization of excitable dynamics in hierarchical biological networks. *PLoS Computational Biology*, 4(9):e1000190, 2008.

C. R. Myers. Software systems as complex networks: Structure, function, and evolvability of software collaboration graphs. *Physical Review E*, 68(4):046116, 2003.

I. Nachman, A. Regev, and N. Friedman. Inferring quantitative models of regulatory networks from expression data. *Bioinformatics*, 20(Suppl 1):i248, 2004.

R. Neher, M. Mitkovski, F. Kirchhoff, E. Neher, F. Theis, and A. Zeug. Blind source separation techniques for the decomposition of multiply labeled fluorescence images. *Biophysical journal*, 96(9):3791–3800, 2009.

M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America*, 98(2):404–409, 2001.

M. E. J. Newman. Assortative mixing in networks. *Physical Review Letters*, 89(20):208701, 2002.

M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6):066133, 2004.

M. E. J. Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 27(1):39–54, 2005.

M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, New York, NY, USA, 2010.

M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113+, 2004.

H. B. Nielsen. Separable nonlinear least squares. Technical report, Technical University of Denmark, 2000a.

H. B. Nielsen. Multi-exponential fitting of low-field 1h nmr data. Technical report, Technical University of Denmark, 2000b.

J. D. Noh and H. Rieger. Random walks on complex networks. *Physical Review Letters*, 92(11): 118701, 2004.

H. Öktem. A survey on piecewise-linear models of regulatory dynamical systems. *Nonlinear Analysis*, 63(3):336–349, 2005.

D. P. O'Leary. Fitting Exponentials: An Interest in Rates. *Computing in Science & Engineering*, 6(3):66–69, 2004.

P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.

G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

R. Pastor-Satorras, A. Vázquez, and A. Vespignani. Dynamical and correlation properties of the internet. *Physical Review Letters*, 87(25):258701, 2001.

K. Pearson. Principal components analysis. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 6(2):559, 1901.

H. T. Pedersen, R. Bro, and S. B. Engelsen. Towards Rapid and Unique Curve Resolution of Low-Field NMR Relaxation Data: Trilinear SLICING versus Two-Dimensional Curve Fitting. *Journal of Magnetic Resonance*, 157(1):141–155, 2002.

J. B. Pereira-Leal, A. J. Enright, and C. A. Ouzounis. Detection of functional modules from protein interaction networks. *Proteins*, 54(1):49–57, 2004.

E. Plahte, T. Mestl, and S. W. Omholt. A methodological basis for description and analysis of systems with complex switch-like interactions. *Journal of Mathematical Biology*, 36(4):321–348, 1998.

P. Pons and M. Latapy. Computing communities in large networks using random walks. *Computer and Information Sciences-ISCIS 2005*, pages 284–293, 2005.

M. Porter, J. Onnela, and P. Mucha. Communities in networks. *Notices of the American Mathematical Society*, 56(9):1082–1097, 2009.

R. Prony. Essai expérimental et analytique sur les lois de la dilatabilité et sur celles de la force expansive de la vapeur de l'eau et de la vapeur de l'alkool, a différente températures. *Journal de l'Ecole polytechnique*, 1:24–76, 1795.

W. V. Quine. The problem of simplifying truth functions. *American Mathematical Monthly*, 59 (8):521–531, 1952.

F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2658, 2004.

F. Rao and A. Caflisch. The protein folding network. *Journal of Molecular Biology*, 342:299, 2004.

E. Ravasz and A.-L. Barabási. Hierarchical organization in complex networks. *Physical Review E*, 67(2 Pt 2):026112, 2003.

E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabási. Hierarchical Organization of Modularity in Metabolic Networks. *Science*, 297(5586):1551–1555, 2002.

J. Reichardt and S. Bornholdt. Detecting Fuzzy Community Structures in Complex Networks with a Potts Model. *Physical Review Letters*, 93(21):218701, 2004.

J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74(1):016110, 2006.

M. Ringnér. What is principal component analysis? *Nature biotechnology*, 26(3):303–304, 2008.

M. Rosvall and C. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences of the United States of America*, 105 (4):1118, 2008.

A. Ruepp, A. Zollner, D. Maier, K. Albermann, J. Hani, M. Mokrejs, I. Tetko, U. Güldener, G. Mannhaupt, M. Münsterkötter, and H. Mewes. The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic acids research*, 32(18): 5539–5545, 2004.

A. Ruepp, B. Brauner, I. Dunger-Kaltenbach, G. Frishman, C. Montrone, M. Stransky, B. Waegele, T. Schmidt, O. N. Doudieu, V. Stümpflen, and H. Mewes. Corum: the comprehensive resource of mammalian protein complexes. *Nucleic acids research*, 36(Database issue):D646–D650, 2008.

G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966.

J. Saez-Rodriguez, L. Simeoni, J. A. Lindquist, R. Hemenway, U. Bommhardt, B. Arndt, U.-U. Haus, R. Weismantel, E. D. Gilles, S. Klamt, and B. Schraven. A logical model provides insights into t cell receptor signaling. *PLoS Computational Biology*, 3(8):e163, 2007.

R. C. Samaco, A. Hogart, and J. M. LaSalle. Epigenetic overlap in autism-spectrum neurodevelopmental disorders: Mecp2 deficiency causes reduced expression of ube3a and gabrb3. *Human Molecular Genetics*, 14(4):483–492, 2005.

A. Scala, L. A. N. Amaral, and M. Barthélémy. Small-world networks and the conformation space of a short lattice polymer chain. *Europhysics Letters*, 55(4):594, 2001.

R. Schachtner, D. R. L. Lutter, P. Knollmüller, A. M. Tomé, F. J. Theis, G. Schmitz, M. Stetter, P. G. Vilda, and E. W. Lang. Knowledge-based gene expression classification via matrix factorization. *Bioinformatics*, 24(15):1688–1697, 2008.

F. Schweitzer, G. Fagiolo, D. Sornette, F. Vega-Redondo, A. Vespignani, and D. R. White. Economic networks: the new challenges. *Science*, 325(5939):422–425, 2009.

M. A. Serrano and M. Boguñá. Topology of the world trade web. *Physical Review E*, 68(1):015101, 2003.

L. V. Sharova, A. A. Sharov, T. Nedorezov, Y. Piao, N. Shaik, and M. S. H. Ko. Database for mRNA Half-Life of 19 977 Genes Obtained by DNA Microarray Analysis of Pluripotent and Differentiating Mouse Embryonic Stem Cells. *DNA Research*, 16(1):45–58, 2008.

S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, 31(1):64–68, 2002.

R. J. Shprintzen and R. B. Goldberg. A recurrent pattern syndrome of craniosynostosis associated with arachnodactyly and abdominal hernias. *Journal of Craniofacial Genetics and Developmental Biology*, 2(1):65–74, 1982.

G. Smyth. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical applications in genetics and molecular biology*, 3(1):1027, 2004.

G. K. Smyth, M. Ritchie, N. Thorne, and J. Wettenhall. Limma: linear models for microarray data. In *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, pages 397–420. Springer, 2005.

E. H. Snoussi. Qualitative dynamics of piecewise-linear differential equations: A discrete mapping approach. *Dynamics and Stability of Systems*, 4(3-4):189–207, 1989.

E. H. Snoussi and R. Thomas. Logical identification of all steady states: the concept of feedback loop characteristic states. *Bulletin of Mathematical Biology*, 55(5):973–991, 1993.

D. Soete. *Neutron Activation Analysis*. John Wiley & Sons, 1972.

D.-M. Song, Z.-Q. Jiang, and W.-X. Zhou. Statistical properties of world investment networks. *Physica A*, 388(12):2450–2460, 2009.

K. L. Streetx, T. Luedde, M. Manns, and C. Trautwein. Interleukin 6 and liver regeneration. *Gut*, 47(2):309–312, 2000.

A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–50, 2005.

C. J. Sylvester and S. F. Forman. Clinical practice guidelines for treating restrictive eating disorder patients during medical hospitalization. *Current Opinion in Pediatrics*, 20(4):390–397, 2008.

A. L. Tarca, V. J. Carey, X. Chen, R. Romero, and S. Draghici. Machine learning and its applications to biology. *PLoS computational biology*, 3(6):e116, 2007.

S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church. Systematic determination of genetic network architecture. *Nature genetics*, 22(3):281–5, 1999.

T. ten Raa. *The Economics of Input-Output Analysis*. Cambridge Books, Cambridge University Press, Cambridge, UK, 2006.

A. E. Teschendorff, M. Journée, P. A. Absil, R. Sepulchre, and C. Caldas. Elucidating the altered transcriptional programs in breast cancer using independent component analysis. *PLoS computational biology*, 3(8):e161, 2007.

M. Thattai and A. van Oudenaarden. Attenuation of Noise in Ultrasensitive Signaling Cascades. *Biophysical Journal*, 82(6):2943–2950, 2002.

F. J. Theis. A new concept for separability problems in blind source separation. *Neural Computation*, 16(9):1827–1850, 2004.

F. J. Theis and A. Meyer-Bäse. *Biomedical Signal Analysis: Contemporary Methods and Applications*. The MIT Press, 2010.

F. J. Theis, A. Meyer-Bäse, and E. W. Lang. Second-order blind source separation based on multi-dimensional autocovariances. In *Proc. ICA 2004*, volume 3195 of *LNCS*, pages 726–733, Granada, Spain, 2004. Springer.

F. J. Theis, K. Stadlthanner, and T. Tanaka. First results on uniqueness of sparse non-negative matrix factorization. In *Proc. EUSIPCO 2005*, Antalya, Turkey, 2005.

L. Tong, R.-W. Liu, V. Soon, and Y.-F. Huang. Indeterminacy and identifiability of blind identification. *IEEE Transactions on Circuits and Systems*, 38:499–509, 1991.

F. Vega-Redondo. *Complex Social Networks*. Cambridge University Press, Cambridge, UK, 2007.

K. Voevodski, S.-H. Teng, and Y. Xia. Finding local communities in protein networks. *BMC Bioinformatics*, 10(1):297, 2009.

U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

V. Vyshemirsky and M. A. Girolami. Bayesian ranking of biochemical system models. *Bioinformatics*, 24(6):833–839, 2008.

P. Wang, M. C. González, C. A. Hidalgo, and A.-L. Barabasi. Understanding the spreading patterns of mobile phone viruses. *Science*, 324(5930):1071–1076, 2009.

D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393: 440–442, 1998.

W. Weidlich. *Sociodynamics: A Systemic Approach to Mathematical Modelling in the Social Sciences*. Taylor & Francis, London, 2000.

R. S. Weiss and E. Jacobson. A method for the analysis of the structure of complex organizations. *American Sociological Review*, 20:661–668, 1955.

J. K. Westwick, C. Weitzel, A. Minden, M. Karin, and D. A. Brenner. Tumor necrosis factor alpha stimulates AP-1 activity through prolonged activation of the c-Jun kinase. *The Journal of Biological Chemistry*, 269(42):26396–26401, 1994.

D. R. White and S. P. Borgatti. Betweenness centrality measures for directed graphs. *Social Networks*, 16(4):335–346, 1994.

J. G. White, E. Southgate, J. N. Thomson, and S. Brenner. The structure of the nervous system of the nematode caenorhabditis elegans. *Philosophical Transactions of the Royal Society of London. Series B*, 314(1165):1–340, 1986.

D. J. Wilkinson. *Stochastic Modelling for Systems Biology (Mathematical and Computational Biology)*. Chapman & Hall/CRC, 2006.

D. J. Wilkinson. Bayesian methods in bioinformatics and computational systems biology. *Briefings in Bioinformatics*, 8(2):109–116, 2007.

W. Windig and B. Antalek. Direct exponential curve resolution algorithm (decra): A novel application of the generalized rank annihilation method for a single spectral mixture data set with exponentially decaying contribution profiles. *Chemometrics and Intelligent Laboratory Systems*, 37(2):241–254, 1997.

D. M. Wittmann, J. Krumsiek, J. Saez-Rodriguez, D. A. Lauffenburger, S. Klamt, and F. J. Theis. Transforming Boolean Models to Continuous Models: Methodology and Application to T-Cell Receptor Signaling. *BMC Systems Biology*, 3(98), 2009.

P. Wong, S. Althammer, A. Hildebrand, A. Kirschner, P. Pagel, B. Geissler, P. Smialowski, F. Blöchl, M. Oesterheld, T. Schmidt, N. Strack, F. J. Theis, A. Ruepp, and D. Frishman. An evolutionary and structural characterization of mammalian protein complex organization. *BMC Genomics*, 9(1):629, 2008.

H. Yen, Q. Xu, D. Chou, Z. Zhao, and S. Elledge. Global Protein Stability Profiling in Mammalian Cells. *Science*, 322(5903):918–923, 2008.

A. Yeredor. Non-orthogonal joint diagonalization in the leastsquares sense with application in blind source separation. *IEEE Transactions on Signal Processing*, 50(7):1545–1553, 2002.

M. A. Yildirim, K.-I. Goh, M. E. Cusick, A.-L. Barabási, and M. Vidal. Drug-target network. *Nature Biotechnology*, 25(10):1119–1126, 2007.

D. Zhou, J. Huang, and B. Schoelkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.

D. Zhu and Z. S. Qin. Structural comparison of metabolic networks in selected single cell organisms. *BMC Bioinformatics*, 6(1), 2005.

A. Ziehe and K.-R. Mueller. TDSEP – an efficient algorithm for blind separation using time structure. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *Proc. of ICANN'98*, pages 675–680, Skövde, Sweden, 1998. Springer Verlag, Berlin.

A. Ziehe, P. Laskov, K.-R. Mueller, and G. Nolte. A linear least-squares algorithm for joint diagonalization. In *Proc. of ICA 2003*, pages 469–474, Nara, Japan, 2003.