Statistical methods for the inference of interaction networks

Sebastian Alexander Max Dümcke

München 2014

Dissertation zur Erlangung des Doktorgrades
der Fakultät für Chemie und Pharmazie
der Ludwig-Maximilians-Universität München

Statistical methods for the inference of interaction networks

Sebastian Alexander Max Dümcke

aus

Heidelberg, Deutschland

2014

Erklärung:

Diese Dissertation wurde im Sinne von §7 der Promotionsordnung vom 28. November 2011 von Herrn Prof. Dr. Patrick Cramer betreut.

Eidesstattliche Versicherung:

Diese Dissertation wurde eigenständig und ohne unerlaubte Hilfe erarbeitet.

München, am 31. März 2014

Sebastian Alexander Max Dümcke

Dissertation eingereicht am 31. März 2014

1. Gutachter: Prof. Dr. Patrick Cramer
2. Gutachter: Prof. Dr. Achim Tresch

Mündliche Prüfung am 28. Juli 2014

To science

You push at the boundary for a few years
Until one day, the boundary gives way
And, that dent you've made is called a Ph.D.

Matt Might *The Illustrated Guide to a Ph.D.*
http://matt.might.net/articles/phd-school-in-pictures/

Summary

Each cell is a complex network of interacting components. It is crucial to understand biology as a system of strong interactions within submodules and loose interactions between submodules. It is toward that goal that computational methods for network inference lead. Ideally these methods help to infer submodules of the global network that is the cell from various biological data.

This thesis presents three statistical methods for the inference of interaction networks. One method uses linear regression followed by hierarchical clustering on gene expression data to predict condition specific transcription factor interactions. This method can infer the dynamic network of interacting transcription factors across conditions where transcription factors can have different interacting partners under different conditions.

The second method can infer regulatory networks. It employs dynamic Boolean networks with unknown time delays on protein abundance data to model interactions between regulators and targets of regulation, allowing for feedback between the two. This method recovers the interaction network responsible for murine embryonic stem cell differentiation,

The third method is a novel test of independence. It is based on the exact distribution of ith nearest neighbours (also a part of this thesis) to derive a test statistic that is especially powerful in the detection of circular dependencies. This method can be used in the PC algorithm for Bayesian network inference. The latter are used to model diverse biological networks.

Preface

Respected Corrector and Dear Reader,

it is with a great sense of accomplishment that I present to you my dissertation. I have taken great care with both the content and the presentation of this document. The first chapter 'Transcription Factor combinatorics' contains work that was done while employed at the GeneCenter of the Ludwig-Maximilians-Universität in Munich. The second and third chapters 'Signal Network reconstruction' and 'Independence testing' contain work that was realized while employed at the Universität zu Köln and working at the Max-Planck Institute for Plant Breeding Research, both in Cologne. This change of employers is a direct consequence of my supervisors move to Cologne after being appointed Jeff Schell professor, a joint chair at the University of Cologne and the Max-Planck Institute for Plant Breeding. This gave me the opportunity to experience two different scientific work environments and two very different german cities.

When working in computational biology it is not important where you work from. More important are the people you work with. In this matter I have been blessed with some brilliant collaborators. I am very grateful to the laboratory of Patrick Cramer, especially to Martin Seizl for designing validation experiments for me and to Nicole Pirkl and Stefanie Etzold for prompt and accurate realization of these experiments. I thank Dietmar Martin for helping me putting the results in context and for his expertise working on the yeast model organism. I also thank Ulrich Mansmann for supporting me with his statistical knowledge and for pertinent remarks on independence testing.

During my thesis I had the opportunity to work with data from many different organisms, including yeast, drosophila and human data and obtained with many different experimental techniques such as microarray hybridization, ChIP-chip, and ChIP-seq. This variety has been a challenge but it helped me hone my skills and expand my toolbox of statistical methods.

The dissertation is organized as follows: It starts with an introduction into the field of interaction networks, giving the reader background information and a context into which to place the current work. Then each chapter begins with an introductory paragraph linking it to the context established in the 'Introduction' and is subsequently organized typically as a research paper containing an introduction, a description of the materials and methods used followed by the results

and ending with a discussion. Further information for each chapter is bundled in the appendix. It should serve to finding answers to specific methodological questions and asides referenced in the main text. You can probably skip this section in the study of the text and it is not formatted for easy reading. The dissertation as a whole is concluded by a chapter conveniently named 'Conclusion' that will be a summary of the achievements and contributions of this work to the field and context established in the Introduction.

I invite the motivated reader to read through this dissertation cover-to-cover to gain an in-depth understanding of the interplay between key mechanisms of genetic regulation. Another reader might prefer to directly skip to the chapter of most interest to him or only to the result section of each chapter. Others will be satisfied in their curiosity with the 'Introduction' and 'Conclusion' chapters alone. I hope to have structured the dissertation in a way to accommodate the different types of readers. Last but not least I wish you all an insightful and pleasurable reading!

Kind Regards,

Sebastian Dümcke

Acknowledgments

During the three and a half years it took to research this thesis, I had the support of many people that all deserve my thanks.

First and foremost, I want to thank my mentor Achim Tresch for his constant support on– and off work and for everything that I learned under his supervision.

Secondly I thank my colleagues in Munich and Cologne with whom it was always a pleasure to share the office and meet up for an after work beer. Thank you Theresa Niederberger, Björn Schwalb, Phillipp Torkler, Benedikt Zacher, Diana Uskat, Carina Demel, Maria Hauser, Eckhart Guthörlein, Jörn Marialke, Armin Meier, Vincent Wolowski, Johannes Söding, Anja Kiesel, Mark Heron, Kemal Akman, Florentine Scharf and Anke Nissen for the Munich part and Henrik Failmezger, Arijit Das, Vipul Patel, Eva Willing, Florence Jacob, Felix Frey, Frederike Horn, Anja Bus, Markus Berns, Lukas Müller, Friederike Brüssow, Nora Peine, Karin Komsick-Buchman, Burkhart Becker and Benjamin Jaegle in Cologne.

They have been mentioned in the Preface already but all the people I collaborated with deserve my special thanks. These include the members of the Cramer lab: Patrick Cramer, Martin Seizl, Nicole Pirkl, Stefanie Etzold and Phillip Eser as well as Ulrich Mansmann from the IBE for his help on the last part of the thesis.

I thank the members of my thesis advisory committee (TAC) Sven Rahman, Ulrike Gaul, Julien Gagneur as well as all members of my thesis defense committee in order Patrick Cramer, Achim Tresch, Ulrich Mansmann, Dietmar Martin, Klaus Förstemann and Daniel Wilson.

I am grateful for the support of my girlfriend, Nasim, whose love and kindness manage to lift all the weight off my shoulders. I thank my sister and my parents for believing in me. I would also like to acknowledge my various flatmates, Lisa, Wibke, Chris, Janina, Michel, Lenka and Natalia for contributing to a safe and restful living environment that allowed me to relax after a hard day at work. I am thankful to Stefan for his hospitality on my trips to munich and for his loyal friendship.

To anybody interested in keeping in touch I can be reached via email at `duemcke@sam-d.com` and online at my personal website sam-d.com.

Contents

List of Figures

List of Tables

List of Algorithms

Introduction

This thesis explores diverse aspects of biological interaction network inference. From the detection of specific types of interactions (condition specific transcription factor interactions, see chapter 1) to the inference of complete small scale interaction networks (using Boolean networks, see chapter 2) via new methods for independence testing which are used specifically in some inference algorithms (e. g. the PC algorithm [1]) described in chapter 3.

Biological interactions have initially been characterized individually using specific low throughput experimental techniques (e. g. co-immunoprecipitation or any knockdown technique). This approach has been replaced by high throughput techniques (e. g. yeast two-hybrid) that yield whole interaction networks. Using such techniques, nearly complete protein interaction networks have been proposed for many model organisms (e. g. for yeast [2], fly [3] and human [4]). New bioinformatic tools have been created to learn from this wealth of data from which we can learn more about the system as a whole than with individual interaction experiments.

In parallel it was tried to reconstruct such large interaction networks *in silico* without all the financial efforts involved in large scale experiments. So far computational interaction network inference is still not feasible at global scale. Bioinformatics has developed a vast toolbox of inference techniques for man different types of networks (e..g. Bayesian networks [5, 6] and Boolean networks [7]).

This thesis is structured as follows: In the first chapter about transcription factor (TF) combinatorics (chapter 1, *Transcription Factor combinatorics*) I investigate how to detect interactions between TFs under the assumption that TF combinatorics are specific to growth conditions. The second chapter deals with inferring interaction network that include unknown time delays (chapter 2, *Signal Network reconstruction*). For this specific task I provide the exact closed form likelihood calculations. The third chapter is about independence testing (chapter 3, *Independence testing*) where amongst other results I show the exact likelihood distribution of the ith nearest neighbour of a point on a two dimensional torus. This distribution leads to the development of two novel tests of independence. Finally the thesis contains an appendix with numerous interesting bits of information that would otherwise have disrupted the flow of argumentation in the main text.

1 Transcription Factor combinatorics

The necessity of combinatorial transcription factor (TF) interaction rapidly becomes apparent when one considers that the number of TFs is small but the number of different and often adverse conditions an organism must adapt to is vast. Therefore combinations of factors exist that interact with different partners under different conditions. This increases the number of conditions that any organism can adapt to.

In this chapter I present One Hand Clapping, a method for the detection of condition-specific interactions between transcription factors from genome-wide gene activity measurements. Using this new method on many TFs and conditions shows a dense interaction network (refer to Figure 1.8 for an example).

1.1 Introduction

Homeostasis, the ability to respond to a plethora of environmental challenges, is vital to the cell. This adaptation is achieved by an orchestrated regulation of gene expression. It was discovered that some TFs act as master regulators in many different conditions, and that the specificity of the regulatory response is obtained through dispatching the signal from the master regulators to downstream TFs [8]. It is quite clear that direct TF interactions, both physical and genetic, are the prevalent mechanisms of this dispatching [9, 10, 11]. A method for the detection of functionally relevant, condition-specific transcription factor interactions (TFIs) would therefore greatly contribute to our understanding of gene regulation.

A necessary first step towards the detection of TFIs is the quantification of individual TF activity. It is difficult to deduce the activity of a TF by its expression alone (only a small fraction of transcription factors show expression levels that correlate with those of their target genes [12]), as there are many alternative mechanisms to activate TFs. A complementary approach is the quantification of TF-DNA binding with ChIP assays [13]. Computational approaches rely on a known TF-target interaction graph [13, 14]. A linear model that describes gene expression as the product of a position-specific activity matrix derived from binding data, and the unknown TF activities is presented in [15]. The experimental detection of TFIs is based on techniques like co-immunoprecipitation and protein binding arrays [13, 16], which are costly and time-consuming. A statistical

framework to deduce TF cooperativity from overrepresentation of common TF motifs at the promoter region of target genes is presented in [17, 18]. However, these approaches do not make direct use of gene expression profiles, nor are their predictions condition-specific. The most promising approaches integrate multiple sources of information, e.g. expression data with binding sites from chromatin immunoprecipitation. The idea is that if two TFs act cooperatively then there should exist a sufficiently large target gene set to which both TFs bind, and the expression profiles of these target genes should be similar across a series of experiments [19]. This concept is used to rigorously assess cooperativity among TFs in the yeast cell cycle [20]. BarJoseph *et al.* [21] construct regulatory gene modules by requiring co-regulation and the co-occurrence of binding sites for a pair of interacting TFs. Beer *et al.* [22] cluster gene expression profiles in a preliminary step, and apply a Bayesian classifier to predict TF modules, i.e. groups of TFs that act together in regulating a set of targets. Advanced statistical models for the integration of binding data and expression data are used in [23]. Single TFs and TF sets are modeled as hidden variables in a sparse regression model. In this way, the authors can assign a significance value for the combinatorial activity of each TF set. Wang *et al.* [24] view the problem of TFI identification as a learning task and use Bayesian networks for the integration of multiple sources of evidence to predict cooperatively binding TFs.

While there are only few studies that focus on TFIs, genetic interactions in general have been investigated extensively. Classically, the biological concept of genetic interaction (e.g. epistasis) between two components relies on the simultaneous perturbation of two components that yields an effect which is different from what one would expect from the perturbation of the individual components. This was applied at large scale in synthetic lethality/growth defect screens like [25, 26, 27], to name a few of them. Typically, as many genes as possible are screened for interaction in an automated way by measuring the fitness of single and double gene deletions. Both fitness measures (growth and lethality) are one dimensional. It is still under debate how the deviation of the double deletion fitness from the fitness of the single deletions can be appropriately measured and tested in a rigid mathematical framework [28, 29]. While this direct interaction measure proved to be rather fragile, the comparison of interaction profiles (the vector of all interaction scores of one gene with all others) yielded surprisingly robust and good results [29]. Furthermore, it became evident that the experimental effort can be reduced considerably if not all pairwise combinations of the genes of interest (~ 5.4 million combinations tested in [27]) are screened, and that even more information can be gained from measurements under different conditions. This insight is reflected in the work of Bandyopadhyay *et al.* [30] which identified genes interacting with DNA damage specific partners, screening a comparably low number of $80,000$ double mutants.

In the present work, I extend the concept of genetic interaction to high dimensional phenotypes (e.g. genome-wide mRNA measurements, RNA-seq) as these become increasingly available. I formulate a mathematical concept of TFI which relies on the assumption that the common targets of interacting TFs should behave significantly different than the genes targeted by only one TF alone. So far, each pairwise genetic interaction had to be tested in an individual experiment, requiring a huge number of combinatorial perturbations. This method instead needs only one global intervention to the system (the fact that led to the name One Hand Clapping) in the form of an environmental stimulus, and a high dimensional gene activity readout in order to score all pairwise TFIs. As in the case of synthetic genetic arrays, I compare the obtained interaction profiles between TFs to obtain reliable and stable predictions. A first proof of concept of this method was given in [31], where the authors applied One Hand Clapping (OHC) to transcriptional activity data obtained under osmotic stress. Here I establish a solid methodological basis and provide a proof of its universal applicability. After benchmarking the performance of OHC, I construct a compendium of high confidence, condition-specific TFIs based on a large gene expression screen [32]. Finally, I validate two of the novel TFI predictions under osmotic stress, one of them in silico, the other one in vivo. OHC is available

as an open source, user-friendly R package (resource `OneHandClapping_1.5.tar.gz` and online `http://cran.r-project.org/web/packages/OneHandClapping/index.html`). The current best practice in the study of gene regulation, consisting of quantification of differential expression and gene set enrichment analysis, can now be extended by the screening for combinatorial TF activity.

1.2 Materials and Methods

1.2.1 TF Interaction model

Let there be gene activity measurements e_g for all genes $g \in G$. G is the set of all genes of the organism. In this case, the values e_g will be the log folds of the activity in a perturbation experiment versus a wild-type control. Suppose we knew all TF-target relations (for a discussion how to obtain such a TF-target annotation see the next subsection). For each TF T, we then had a binary indicator function $I(g \in T)$ taking on value 1 if gene g belongs to the target set of T, and 0 otherwise. The main idea of this method is to divide the set of all genes into four subsets (Figure 1.1): Those genes that are targeted by none of the two TFs, those that are targeted by only one of the TFs, and those that are targeted by both TFs. Apart from a possible baseline shift β_0 in gene activity, TF T_j alone is assumed to have an effect β_j on its targets ($j = 1, 2$). Disregarding the baseline shift, the common targets of T_1 and T_2 are expected to

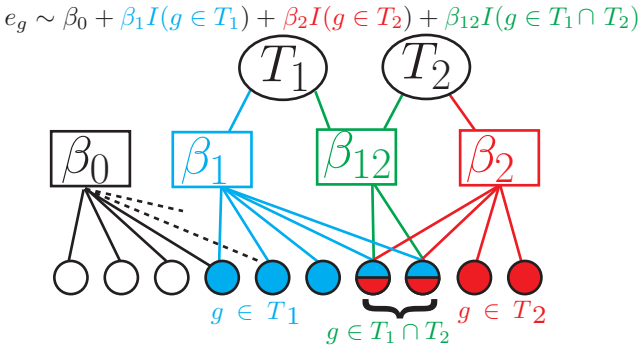


Figure 1.1: Schematic description of the linear regression model: For two TFs T_1 and T_2, expression of all genes that are targets of T_1 is described by coefficient β_1 (cyan), expression of genes that are targets of T_2 is described by coefficient β_2(red) and expression of genes that are targets of both TFs is described by coefficient β_{12} (green). β_0 (white) is the coefficient for the baseline activity. It is connected to all genes including those that are targets of neither T_1nor T_2 (white circles). Remaining connections are symbolized by dotted lines. Circles at the left symbolize genes and are colored according to the TF that targets them. The whole formula of the logistic linear regression is shown at the top, with the relevant parts highlighted at the bottom.

show a change in activity that amounts to $\beta_1+\beta_2$, if the two TFs do not interact. The deviation from this expectation is quantified by the interaction term β_{12}, which presents the most interest. Formally, this can be cast as a second order linear regression of e_g versus the covariates $I(g \in T_1)$ and $I(g \in T_2)$,

$$e_g \sim \beta_0 + \beta_1 I(g \in T_1) + \beta_2 I(g \in T_2) + \beta_{12} I(g \in T_1) I(g \in T_2),$$

with $g \in G$. The regression is performed for each TF pair separately, since including more TFs and their interaction terms would lead to overfitting. This cannot be alleviated by using regularization methods like ridge regression or lasso regression (data not shown). Running the regression in an all-against-all fashion for a set of TFs T results in a symmetric $|T| \times |T|$ interaction matrix M containing all interaction terms β_{12}. I noticed that the interaction terms alone are not strong predictors of interaction (data not shown). The possible explanations for this are threefold: The definition of the target sets T_1 and T_2 is imperfect, the expression measurements are prone to unsystematic variation, or the model of TF activity might be too simplistic in some cases.

1.2.2 TF annotation

One cornerstone for finding TFIs by looking at commonly regulated target genes is the availability of a sufficiently accurate TF-target gene mapping. Such a mapping is rarely available, especially for different growth conditions. This is a limitation of the method that will hopefully be alleviated with the advent of ChIP-seq data of TFs in many organisms, as they are being generated by the ENCODE and modENCODE consortia [33, 34, 35].

For *Saccharomyces cerevisiae* there are fortunately several high-quality TF-target mappings available. TF-target relations mined from a manually curated literature repository can be found in the YEASTRACT database [36] which is used in this work. I filter this annotation removing TFs with less than 10 annotated target genes. This leaves 165 TFs with a median of 167 annotated genes per TF. Please refer to the Appendix (chapter A, *TF-target graphs for different organisms and their characteristics*) for a discussion of alternative TF-target graphs and some characteristic numbers on them.

Figure 1.2A shows a box plot of expression folds (total fraction) of the TFs from YEASTRACT. Prominent differentially expressed TFs are explicitly shown (XBP1, MAG1, SIP4, CIN5, NRG1, CUP2, TEC1, ASH1 and BAS1). Most of these outliers are not directly involved in the salt stress or general stress response pathways, confirming that TF activity is not regulated at the transcriptional level.

When looking at the coefficients β_1, β_2 and β_{12} from the regression model of all TF pairs in the YEASTRACT database there is no apparent structure (Fig-

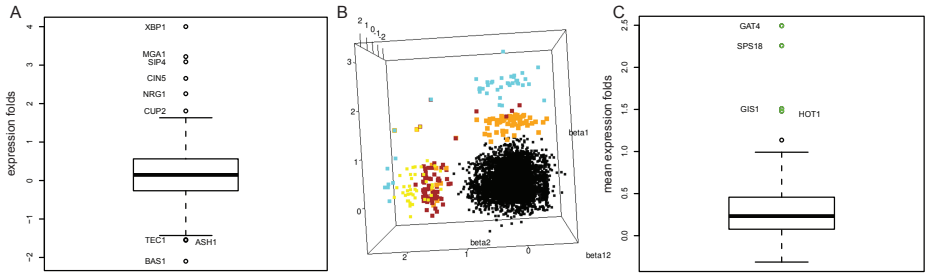


Figure 1.2: **A**: mRNA expression folds (data set D1) of genes coding for all TF from YEASTRACT. Strong differentially expressed genes are not necessarily involved in the osmotic stress response pathway suggesting that TF activity is regulated post-transcriptionally. **B**: 3D plot of coefficients β_1, β_2 and β_{12} from the interaction model. β_1 is in the x-direction, increasing to the left, β_2 is in the y-direction increasing upwards and β_{12} in the z-direction. The coefficients of interactions involving Gis1p, Gat4p, Hot1p and Sps18p are highlighted in orange, brown, cyan and yellow respectively. There is no apparent correlation between the single effects and the interaction effect. **C**: mean expression folds in data set D1 of the target sets of each TF in YEASTRACT. The target sets of Gis1p, Gat4p, Hot1p and Sps18p (highlighted in green) show a strong ($> \log_2(1.5)$ fold) differential expression.

ure 1.2B). Closer investigation reveals extreme values that are due to pairwise interactions between a small set of four TFs (Hot1p, Sps18p, Gis1p, Gat4p see Figure 1.2C). Indeed these TFs have target genes which are strongly differentially expressed, thus giving rise to a high β_{12} coefficient to every TF having a considerable overlap with one of these four TFs. The mean expression of all target genes is above that of all other TFs (Figure 1.2C). A Gene Ontology analysis revealed that they are stress responder genes involved in response to various stimuli and to heat shock (Table 1.1) I removed these four outlier TFs from the TF-target graph, leaving us with a final annotation containing 161 TFs.

1.2.3 TFI prediction

To arrive at robust TFI predictions I use a "guilt-by-association" principle that has been commonly applied in genetic interaction screens [27]. Instead of comparing single interaction values, I compare the interaction profiles of each TF (the rows of the interaction matrix M) by means of their correlation. More specifically, we use $1 -$ Pearson correlation as a distance measure. I apply hierarchical average linkage clustering to the rows of M using this distance measure. The two descendants of the terminal branches of this dendrogramm define the TFI predictions (Algorithm 1.1). The reasoning behind this is that one expects many TFs to have at least one interaction partner in a given condition, and the most likely partner is the one with the most similar interaction profile. Alternatively, I tried to predict TFIs based on p-values derived from a null distribution of the correlation distances. Such null distributions can be either derived from Pearson's Product Moment Coefficient [37] or, more conservatively, from resampling procedures (shuffling target genes). Still the simple clustering procedure works best in terms of area under the curve (AUC) (data not shown; for a definition of AUC see the Results).

1.2.4 Gene activity data

In this thesis I use several data sets as input to the OHC method. First I use mRNA expression data from a time course experiment exposing a wild-type yeast strain to osmotic stress by adding 0.8 M NaCl (see [31] for more details). The paper provides standard total mRNA expression data after 36 minutes of osmotic stress (data set D1), as well as the corresponding measurements of "newly synthesized" mRNA (data set D2), which are roughly proportional to the mRNA synthesis rates at the time of measurement. Throughout this thesis I always mean log expression folds (log quotient of expression under the experimental condition against expression in the control experiment) when referring to expression data. To test the reliability of the method, I included an unrelated gene

GOBPID	Pvalue	OddsRatio	ExpCount	Count	Size	Term
GO:0009266	1.4E-34	9.1E+00	1.3E+01	7.1E+01	2.1E+02	response to temperature stimulus
GO:0009628	2.8E-30	6.2E+00	2.1E+01	8.3E+01	3.3E+02	response to abiotic stimulus
GO:0009408	6.1E-30	8.3E+00	1.3E+01	6.4E+01	2.0E+02	response to heat
GO:0034605	1.1E-29	8.8E+00	1.2E+01	6.1E+01	1.8E+02	cellular response to heat
GO:0007039	1.0E-19	8.4E+00	7.5E+00	4.0E+01	1.2E+02	vacuolar protein catabolic process
GO:0006950	1.9E-19	3.2E+00	5.2E+01	1.2E+02	8.2E+02	response to stress
GO:0050896	2.4E-16	2.7E+00	7.2E+01	1.4E+02	1.1E+03	response to stimulus
GO:0044275	4.0E-12	7.0E+00	5.4E+00	2.6E+01	8.4E+01	cellular carbohydrate catabolic process
GO:0016052	4.7E-12	6.6E+00	5.8E+00	2.7E+01	9.1E+01	carbohydrate catabolic process
GO:0006066	2.3E-11	3.6E+00	1.7E+01	4.7E+01	2.6E+02	alcohol metabolic process
GO:0006091	2.3E-11	3.7E+00	1.6E+01	4.6E+01	2.5E+02	generation of precursor metabolites and energy
GO:0044282	5.8E-11	4.7E+00	9.3E+00	3.3E+01	1.4E+02	small molecule catabolic process
GO:0055114	1.7E-10	3.0E+00	2.2E+01	5.5E+01	3.5E+02	oxidation reduction
GO:0033554	3.3E-10	2.5E+00	3.7E+01	7.5E+01	5.7E+02	cellular response to stress
GO:0005975	4.6E-10	3.0E+00	2.2E+01	5.4E+01	3.5E+02	carbohydrate metabolic process

Table 1.1: Gene Ontology term enrichment analysis of the targets of four outlier TFs Hot1p, Sps18p, Gis1p and Gat4p against all genes in yeast. The 15 most significant GO terms are shown. The results indicate that the target genes of Hot1p, Sps18p, Gis1p and Gat4p are all stress responder genes.

input : root of dendrogram obtained through hierarchical clustering
output: list of leaf pairs

dfs (n)*:*
  if *length*(`leafs(`n`)`) ≤ 2 **then**
    return `leafs(`n`)`
  end
  else
    return [**dfs** (`leftchild(`n`)`),**dfs** (`rightchild(`n`)`)]
  end
endw

`leafs(`n`)`: function returning all leafs under node n
`leftchild(`n`)`: function returning left children node of node n
`rightchild(`n`)`: function returning right children node of node n

Algorithm 1.1: Depth-first search to get TF pairs from a clustering dendrogram.

expression data set generated by Mitchell *et al.* [38] obtained from *S. cerevisiae* under osmotic stress. The total mRNA expression level (corresponding to the total fraction of Miller *et al.*) 30 minutes after addition of 0.8 M NaCl was measured (data set D3). The gene expression data sets from Miller *et al.* and Mitchell *et al.* should be highly comparable, since the same yeast strain, the same microarray platform and a similar protocol were used. Microarray data were downloaded as raw files from GEO [39] (accession number: GSE15936) for Mitchell *et al.* data and from ArrayExpress (accession number: E-MTAB-439) for Miller *et al.* Normalization was performed using gcrma [40] (as implemented in R/Bioconductor [41]) without quantile normalization, since I expect global effects of the perturbation on mRNA expression. As a completely different way of assessing gene activity, Miller *et al.* [31] also provide RNA Polymerase II (Pol II) occupancies from ChIP-chip experiments 24 minutes after addition of salt. I use their Pol II mean occupancy on each gene (between transcription start site and polyadenylation site) as another proxy for gene activity (data set D4).

1.2.5 Yeast strains and growth assays

The *S. cerevisiae* deletion strains *hog1*Δ*, arr1*Δ*, gcn4*Δ, as well as the wild-type strain BY4741 were obtained from Open Biosystems (Huntsville, USA). The double deletion strain *arr1*Δ*/gcn4*Δ was generated by integrating a ClonNat cassette in the ARR1 locus of the *gcn4*Δ strain. Correct gene disruptions were verified by PCR. Spot dilutions were done to assess fitness and growth under osmotic stress. Equal amounts of freshly grown yeast cells in YPD were re-

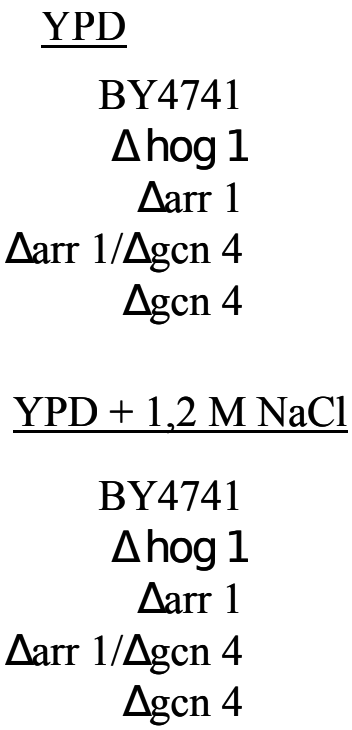


Figure 1.3: Cultivated growth plates after 4 days of incubation. Strains were spotted on YPD plates with 1.2 M sodium chloride at 30°C. Starting with cells grown to an OD_{600} of 0.1 and a 1:10 dilution series of length 5 was spotted. From left to rigth, dilutions of 1:1, 1:10, 1:100, 1:1000, $1{:}10^5$

suspended in water, 10-fold dilutions were spotted on YPD plates and YPD plates with 1.2 M NaCl. Plates were incubated for 4 days at 30°C. Results are found in Figure 1.3.

1.2.6 Gene expression microarrays

Overnight cultures were diluted in fresh synthetic complete medium with 2% glucose to $OD_{600\,nm} = 0.1$ (120 ml cultures, 160 rpm shaking incubator, 30°C). In the early log phase ($OD_{600\,nm} = 0.8$) 20 ml of the culture were harvested by centrifugation (no salt stress sample). Afterwards, NaCl was added to the remaining culture to a final concentration of 0.8 M. 30 min after addition, 20 ml of culture were harvested (salt stress sample). Total RNA was prepared after cell lysis using a FastPrep-24 instrument (Millipore) and subsequent pu-

rification using the RiboPure-Yeast Kit (Ambion) following the manufacturer's instructions. All following steps were conducted according to the Affymetrix GeneChip 3'IVT Express Kit protocol. Briefly, one-cycle cDNA synthesis was performed with 300 ng of total RNA. In vitro reverse transcription labeling was carried out for 16 h. The fragmented samples were hybridized for 16 h on *Yeast Genome 2.0* expression arrays (Affymetrix), washed and stained using a Fluidics 450 station and scanned on an Affymetrix GeneArray scanner 3000 7G. Micorarray data have been deposited to the ArrayExpress database (`http://www.ebi.ac.uk/microarray`) under accession number E-MEXP-3566

1.3 Results

1.3.1 OHC accurately predicts pairwise TF interactions

I first applied OHC to mRNA expression data from the total mRNA fraction of Miller *et al.* (data set D1) using the filtered YEASTRACT database as TF-target annotation (see Methods). The resulting interaction matrix is shown as a heatmap (Figure 1.4). The rows of the matrix were clustered and TFI predictions were made as described in the Methods section. I predict 59 mutually disjoint TF interaction pairs, while for 43 single TFs no interaction partners were predicted. Validation of the predictions was done through the BioGRID database ([42], version 3.1.71). It contains physical and genetic interactions for many yeast proteins that were derived from high and low throughput experiments in the literature. The subgraph of BioGRID corresponding to interactions between TFs, as well as their degree distribution, is shown in Figure 1.5. From the 59 predicted TFIs, I validate 13 of them as listed in BioGRID (a positive predictive value of 22%). Validated TFI predictions had a significantly lower correlation distance than unvalidated TFIs (Wilcoxon's test, p-value 0.004). This shows that interacting TF pairs are more closely related (considering the interaction measure and distance function) than unvalidated predictions. This is further investigated through a ROC plot (Figure 1.7A). The area under the curve (AUC, 76%) shows a strong deviation from random predictions (diagonal) and shows that the profile correlation measure can serve as a proxy for predicting interactions. To better assess the performance of the clustering and prediction algorithm, I verified the over-representation of validated prediction using Fisher's test (p-value $< 10^{-5}$, odds ratio: 5.291, with a 95% confidence interval $[2.67; 10]$). When testing only genetic or physical interactions from BioGRID (p-values $< 10^{-5}$ and 0.003, respectively) I find a bias towards prediction of genetic interactions, as defined by BioGRID.

I tested the consistency of predictions on incomplete TF-target annotations by removing an increasing percentage of TFs from the annotation. I measured the

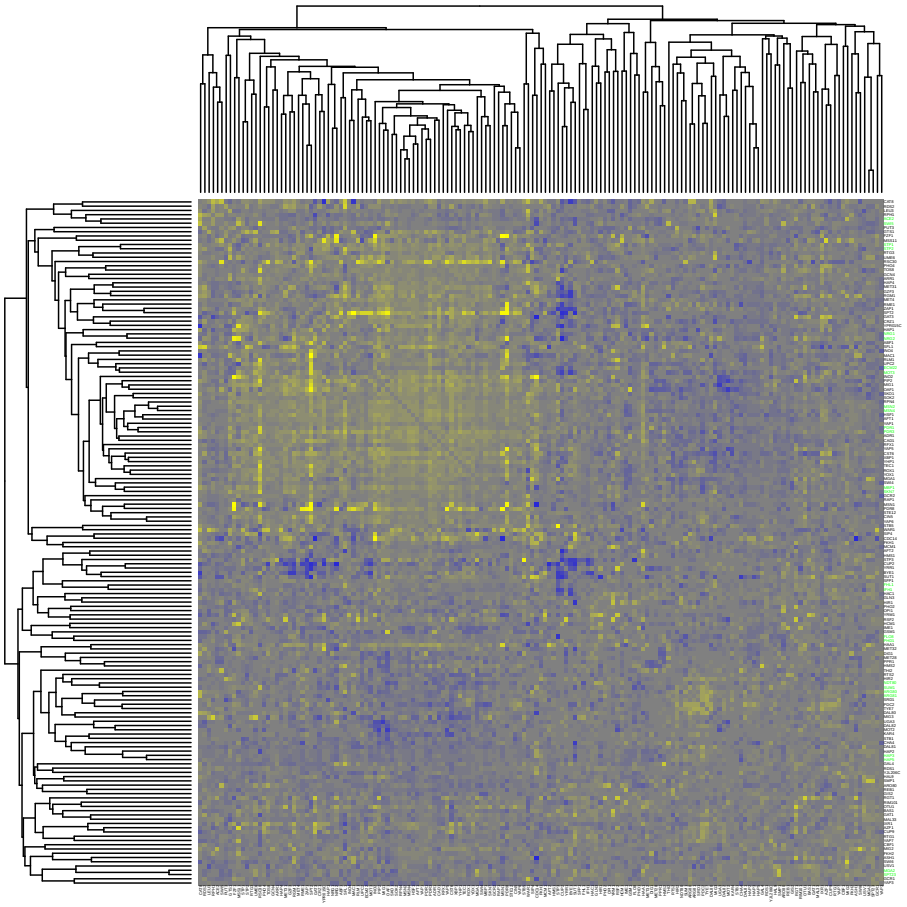


Figure 1.4: Heatmap showing resulting interaction score matrix of running One-HandClapping on dataset D1 (mRNA folds of total fraction after 30 minutes osmotic stress). Pairs validated by BioGRID are highlighted in green. Negative interaction scores are colored blue, positive interaction scores yellow and interaction scores around 0 are colored grey.

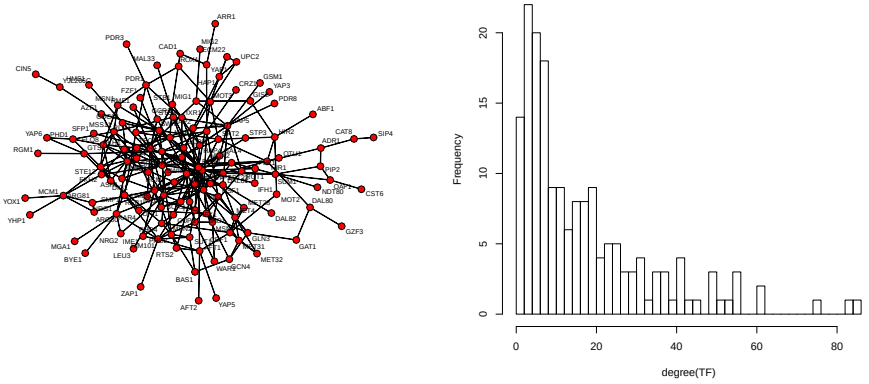


Figure 1.5: Graph showing all interactions found in BioGRID between TFs from YEASTRACT. Right plot shows the degree distribution of the graph. There are many TFs with few edges and few with many edges.

agreement of predictions on the smaller TF annotation with predictions made on the original annotation (Figure 1.6). Additionally I measured the performance as the number of validated pairs according to BioGRID. Expectedly the drop in agreement is stronger than the drop in performance, because removing one TF from a pair will regroup the remaining TF with another with high probability, thus changing the predictions. Simultaneously performance decreases more slowly, showing that the regrouping of TFs leads to new validated pairs. After removal of 20% of TFs performance merely drops from 22% to 18%.

1.3.2 OHC is stable on a wide range of gene activity data

To test the stability of the method I applied it to the mRNA expression data of the labeled fraction from the same osmotic stress experiment used previously (termed data set D2, see Methods). Both data sets are similar (Spearman's $\rho = 0.85$, Figure 1.7C) and we expect similar results. On this data set I predict 60 pairwise interactions, 11 validated by the BioGRID database (18% prediction accuracy; predicted pairs: Nrg1p-Nrg2p, Fhl1p-Ifh1p, Stp1p-Stp2p, Msn2p-Msn4p, Mbp1p-Swi4p, Ecm22p-Upc2p, Cbf1p-Met28p Ndt80p-Sum1p, Arg80p-Arg81p, Hap3p-Hap5p and Mga2p-Spt23p). The validated interactions highly agree between both data sets, 8 pairs being validated by both runs (Figure 1.7B). The interactions Ace2p-Swi5p, Ecm22p-Mot3p, Pdr1p-Pdr3p, Mbp1p-Skn7p and Flo8p-Phd1p found in the first data set are lost in the second, the interactions Mbp1p-Swi4p, Ecm22p-Upc2p and Cbf1p-Met28p in the second are

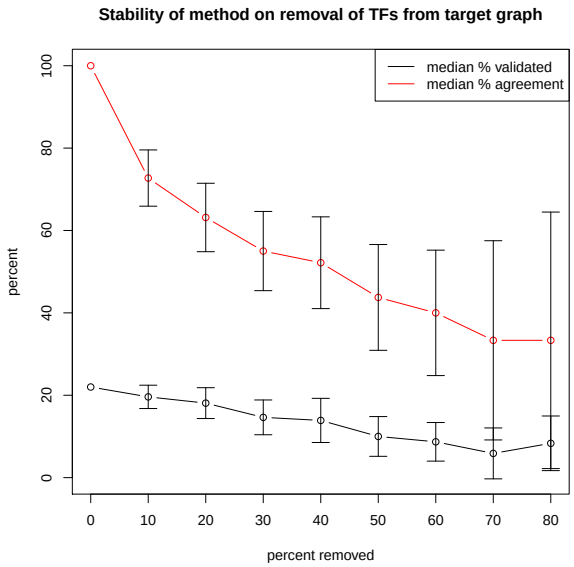


Figure 1.6: Figure showing the consistency of the predictions. I tested the stabililty of OHC's predictions against incomplete annotations. Starting with the initial set T of transcription factors, I used OHC to predict an original set P of TFIs. Then I removed a certain fractions of randomly selected TFs to obtain a reduced TF set T'. I applied OHC again to the reduced interaction matrix to obtain a new prediction P'. The consistency (i.e agreement between predictions) of OHC was assessed by calculating: $\frac{|P \cap P'|}{|P \cap (T' \times T)|}$ This is the fraction of original TFIs that were also found in the reduced TF screen divided by all original TFIs that could potentially be found in the reduced TF annotation (red line). Moreover I calculated the fraction of validated pairs in each reduced TF set (black line) using BioGRID (as described in the Methods section). The procedure was repeated 100 times for each scenario (i.e. for the removal of 10%, 20% ... 80% of all TFs). The points in the plot are the medians over 100 runs and the error bars show the standard deviations.

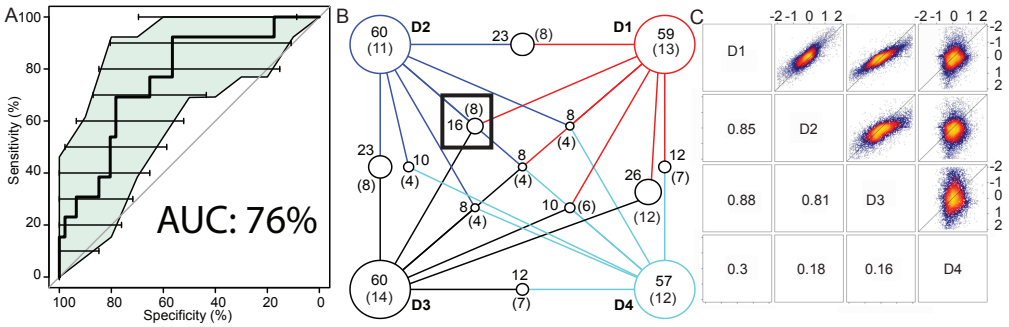


Figure 1.7: Validation of OHC predictions **A**: ROC curve using TF interactions in BioGRID as benchmark, colored area and horizontal lines are confidence intervals of sensitivity and specificity, respectively. **B**: Overlap of predicted and validated pairs between all data sets. D1: total mRNA fraction D2: labeled mRNA fraction D3: mRNA data from Mitchell *et al.* D4: Pol II ChIP-chip occupancy measurements; predictions across data sets agree well, data set D4 having the most distinct predictions. Each intersect is shown as a circle, with radius proportional to the intersect size. The black box shows the intersect used as novel predictions. The numbers in parentheses indicate the subset of interactions that are validated by BioGRID. **C**: pairwise comparison of expression or occupancy values for all genes. Numbers in lower part indicate Spearman's correlation between data sets. D1 and D3 have the highest correlation as they are both total mRNA expression measurements. D2 has a very good but lower correlation with D1 and D3. This is due to subtle differences when measuring newly synthesized mRNA. D4 has a very weak positive correlation as Pol2 occupancy as an indicator of gene activity is very different from mRNA expression.

not present in the first data set. Comparison of all predicted interactions (Figure 1.7B) features an overlap of 23 pairwise interactions (38%).

Reproducibility was tested by running the method on another osmotic stress data set from [38] (mRNA expression measurement 30 minutes after addition of NaCl) termed D3 (Spearman's $\rho = 0.88$, Figure 1.7C). The method predicts 60 pairwise interactions and 14 validated interactions (23%). The overlap with the previous two data sets is 26 and 23 pairs for data sets D1 and D2 respectively. Validated interactions agree strongly; they overlap at 12 and 8 validated interactions for D1 and D2 respectively (Figure 1.7B). It is interesting to notice that the data sets D3/D1 agree more closely than D3/D2 and D1/D2. This might be due to the fact that D1 and D3 measure the total mRNA at the extraction timepoint and thus include mRNAs transcribed before the onset of stress and not yet degraded, contrary to D2 which corresponds to the labeled mRNA fraction and thus contains only mRNAs transcribed after the onset of stress. Indeed D1/D3 have a higher correlation than D1/D2 and D3/D2 (Fig 1.7C).

To show that the method also works on proxies of gene activity other than mRNA expression measurements, I used the Pol II ChIP-chip data from [31] (termed D4). On this data set the method predicts 57 interactions, 12 of which can be validated (21% accuracy). Its performance is thus comparable to the performance on mRNA expression data. The predictions vary strongly as there are only 12, 10 and 12 predicted interactions shared with the data sets D1, D2 and D3 respectively (Figure 1.7B). This is due to a low correlation between the data sets D1 to D3 varying between 0.16 and 0.3 (Spearman's rank correlation, see Figure 1.7C). Despite the low correlation, a core of 8 interactions is shared between all data sets (including 4 novel predictions) and shows that the method is robust enough to adapt to various measures of gene activity.

1.3.3 OHC finds *cis* and *trans* TF interactions

I distinguish between two main types of combinatorial TF interactions: *cis* regulatory interactions and *trans* regulatory interactions [43]. *Cis* interactions are mediated by a specific TF binding site configuration at the *cis* regulatory region of a gene, possibly resulting in cooperative or competitive binding of TFs. Competitive binding occurs when two TFs share a common or overlapping binding motif. Cooperative binding of TFs occurs if two TFs are required to bind simultaneously to be functional, or if the binding of the second TF is enhanced by the binding of the first TF, which is the case e.g. for nucleosome-mediated cooperativity [44].

Trans interactions are defined as direct protein-protein interactions of both TFs prior to DNA binding, either by forming a protein complex or by complex formation with other co-factors involved in Pol II recruitment and transcription

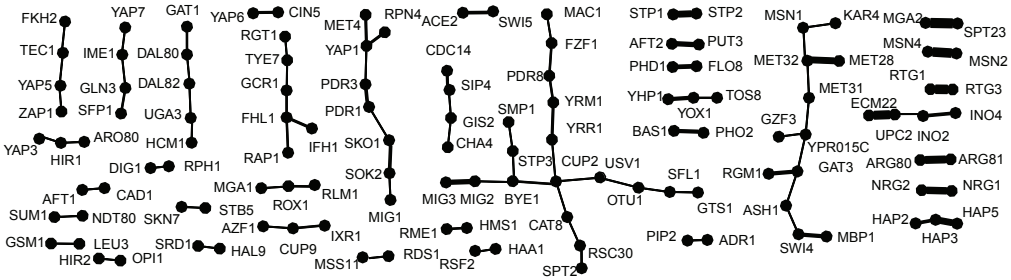


Figure 1.8: Components of the network of confident TF interactions predicted in more than half of the experiments from each condition. Edge width is proportional to the number of conditions in which the edge was found and the components are sorted in increasing edge width from left to right. TF pairs on the right, that are found in many conditions are either protein complexes or homologous or highly similar proteins. TF pairs on the left are highly condition specific and only interact in a single condition.

initiation.

TF pairs predicted by the method on data set D1 and validated by BioGRID include the following types of interaction: Ace2p-Swi5p [45] and Sum1p-Ndt80p [46] undergo competitive *cis* regulatory interactions, the former having identical binding sites, the latter having overlapping binding sites. Mot3p-Ecm22p [47], Mbp1p-Skn7p [48], Arg80p-Arg81p [49], Hap3p-Hap5p [50] and Pdr1p-Pdr3p [51] are all examples of *trans* regulatory protein interactions forming prior to DNA binding. The pair Ifh1p-Fhl1p represents a special type of *trans* interaction. Fhl1p is by default bound to the promoter of ribosomal protein genes without influencing transcription. The phosphorylation of Ifh1p enables the binding and activation of Fhl1p [52].

Three interactions (Msn2p-Msn4p [53], Mga2p-Spt23p [54] and Stp1-Stp2p [55]) could not be categorized unambiguously. They consist of homologous or functionally redundant proteins, implying that both *cis* and *trans* interactions could serve as regulatory mechanism. I call these interactions *ambiguous*.

1.3.4 OHC provides a compendium of condition-specific TF interactions

Absolutely no changes to the model are required when applying the method to large data sets containing gene activity measurements under diverse conditions. Consequently, I ran the method on mRNA expression data from 173 experiments

(data compiled by Gash et al. [32]) which is grouped into 16 conditions with at least five experiments. Clustering the experiments according to the correlation of the expression profiles across conditions recovers the grouping into 16 conditions defined above. Similarly, clustering the predictions made by OHC on each experiment according to the number of common TF-interactions between experiments recovers the condition classes as well (Figure 1.9). This demonstrates that predictions by OHC are truly condition specific and reproducible.

I compiled a compendium of confident condition-specific TF interactions. For each condition, I selected the OHC interactions that are found in more than half of the experiments for that condition. This compendium is provided as Resource (`table.confident.txt`). The graph representation of this compendium (Figure 1.8) is sparsely connected with many isolated pairs. The number of conditions for a pair of transcription factors is encoded by edge width, indicating the specificity of the interaction. Due to false negatives and the limited variety of environmental conditions in [32], this network is far from being complete, and too sparse to be conclusive about its topological properties, such as edge degree distribution and connectivity. Yet it highlights an important organisational feature of signalling pathways, namely a functional hierarchy, where information is flowing from general to specific regulators: Some TF pairs interact in more than one condition. Most of them are either protein complexes (e.g. Hap2p-Hap3p-Hap5p), form heterodimer (e.g. Arg80-Arg81p, Ino2p-Ino4p) or are highly similar or homolog transcription factors (e.g. Nrg1p-Nrg2p, Msn2p-Msn4p, Mga2-Spt23p and Upc2p-Ecm22p). This is the reason why the aforementioned interactions are detected in multiple conditions; simply because the activation of one interaction partner leads to complex formation. The other TFs which interact with different partners, need both to be active under the same condition for an interaction to be predicted. This is the case for the interaction between Skn7p and Stb5p which is exclusively predicted by OHC under diamide treatment, which seems plausible as both have a role in the oxidative stress response [56, 57]. Skn7p is the more general transcription factor while Stb5 is diamide specific. Indeed STB5 null mutants have a decreased resistance to diamide [58]. Another interesting finding is the TF Tye1p which, as this model postulates, regulates glycolysis together with Gcr1p under H_2O_2 exposure and together with Rgt1p when cells are exposed to dithiothreitol (DTT). Condition-specific transcriptional control is achieved by activating Tye1p under several oxidative stress inducing agents and specifically pairing it with TFs only active under one such agent. OHC helps in discovering this type of combinatorial gene regulation.

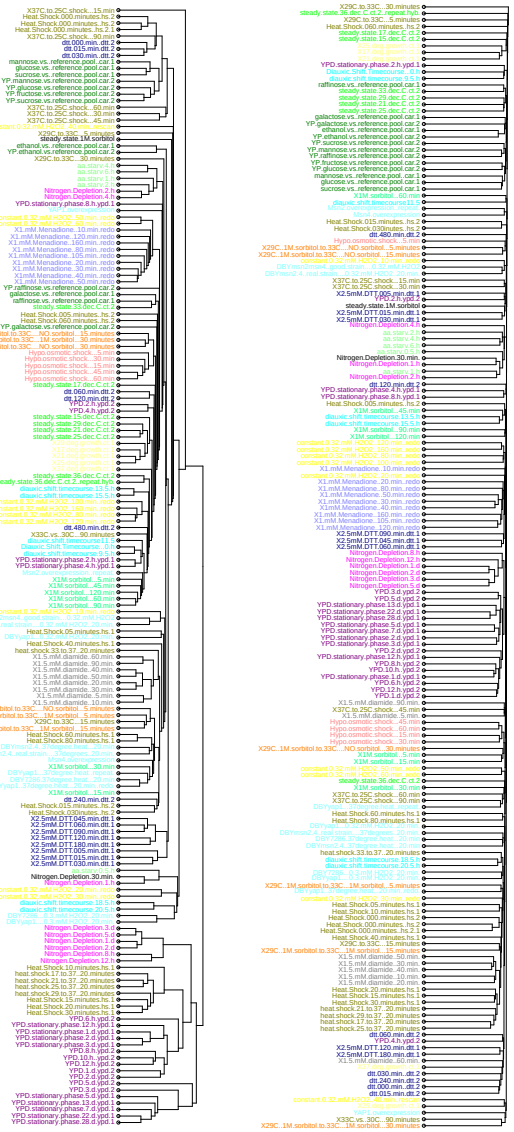


Figure 1.9: Hierarchical average clustering of Gash *et al.* expression using euclidean distance (left) and hierarchical average clustering of intersect size between predicted pairs in each condition from Gash *et al.* (right). The experiments from the dataset are grouped into 16 conditions indicated by different colors. Both clustering procedures group the experiments according to the condition they belong to. This validates that predictions made by OHC are condition specific

1.3.5 Novel predictions of TF interactions can be validated experimentally

Novel predictions are defined as consensus predictions between data sets D1, D2 and D3 (indicated by a black box in Figure 1.7B). I left data set D4 out because of the low correlation with the rest of the data. This gives eight novel predictions namely the pairs: Cin5p-Yap6p, Gcn4p-Arr1p, Zap1p-Spt2p, Sko1p-Sok2p, Hsf1p-Aft1p, Sip4p-Cdc14p, Cup2p-Yrr1p and Rim101p-Otu1p.

Cin5p and Yap6p bind competitively

I realized both Cin5p and Yap6p have very similar binding motifs (Figure 1.10A) according to the YeTFaSCo database [59], choosing the motifs with high expert confidence. They are derived from ChIP-chip data by Harbison *et al.* [13] and MacIsaac *et al.* [14] for CIN5 and YAP6 respectively

I searched for both motifs using these position-specific weight matrices (PWMs) and the MEME suite [60] (FIMO version 4.7.0 using default parameters for pvalue and qvalue thresholds) on intergenic regions defined by [13]. Testing for co-occurence of both motifs on all intergenic regions is highly significant (p-value $< 10^{-5}$). I found 135 intergenic regions where both motifs have one or several matches. In this set I find 149 competitive matches, where the distance between both motif occurrences is 0 and 36 cases having 5 or more nucleotides between motif occurrences. Motif search also shows, that the TFs can bind alone as for some intergenic regions only a match for a single TF falls below the p-value threshold. As there are protein-binding microarray (PBM [16]) derived motifs for each TF I deduce that both proteins can bind DNA on their own. The motif similarity from ChIP-chip data is thus not due to a protein complex between Cin5p and Yap6p and I conclude that both TFs bind competitively to the promoter of their common target genes.

The other novel predictions do not show such a clear evidence for an interaction based on their motifs, so I decided to perform experimental validation for one additional pair. I chose the pair Gcn4p-Arr1p as both interaction partners have the largest target sets among all predicted pairs (as defined by YEASTRACT, 1260 and 743 target genes for Gcn4p and Arr1p, respectively).

GCN4/ARR1 show a synthetic rescue phenotype

To validate the interaction between GCN4 and ARR1 a classical genetic interaction screen was performed (Figure 1.10B and Figure 1.3). It assayed the growth of a wild-type strain as well as single and double deletion strains in rich medium (YPD) and under osmotic stress (YPD + 1.2 M NaCl). The single deletions had no effect in rich media due to the condition specificity of the prediction.

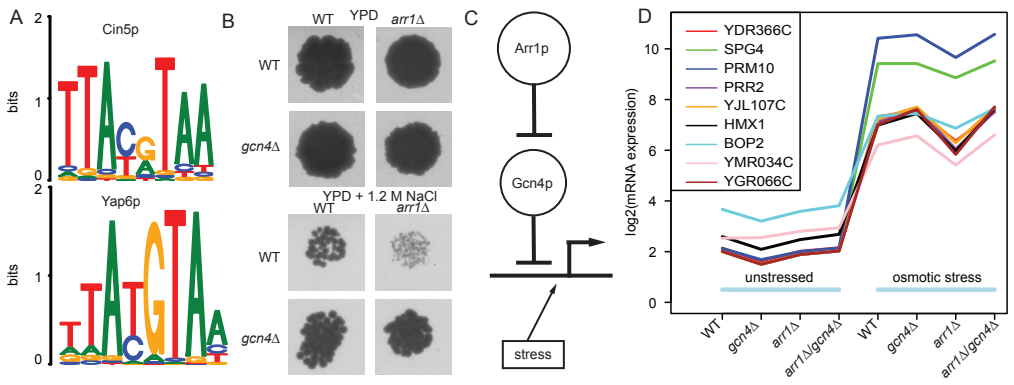


Figure 1.10: **A**: Motifs of CIN5 and YAP6 from YeTFaSCo database. Both motifs are very similar, which is confirmed by motif search, as a test for co-occurrence of both motifs is significant (see text). **B**: Growth assay on YPD plates and YPD plates containing 1.2 M NaCl after incubating for 4 days at 30°C. Only cell growth at a dilution of 1:100 is shown and the relevant parts have been extracted from Figure 1.3. Growth phenotype is not affected in YPD medium. Wild type cells have decreased phenotype under osmotic stress and arr1Δ mutants show strong decrease in growth phenotype, while gcn4Δ mutants and double mutants do not. This shows the synthetic rescue of the effect of the knockout of ARR1 in the double mutant. **C**: Hypothetical model explaining the observations from the growth assay experiments (**B**). This model is focused on the genes that respond positively to osmotic stress (symbolized by the line with arrowhead). A double inhibition chain of Arr1p ⊣ Gcn4p leads to the observed phenotypes under osmotic stress: In wild-type cells the inhibitory effect of Gcn4p is prevented by Arr1p and the cells grow normally. The same observation is made when knocking out GCN4 as the logic of regulation does not change. The knockout of ARR1 relieves the inhibition on Gcn4p, which in turn downregulates the target genes. I speculate that this is causing problems with osmo-adaptation, leading to a reduced cell growth. The double mutant rescues that phenotype as the genes are only driven by the osmotic stress (as in wild-type). **D**: Log-expression values of candidate genes responsible for synthetic rescue across all arrays. All candidates are affected by the knockout of ARR1. Four candidate genes are uncharacterized ORFs, two are proteins of unknown function and the rest has a variety of different roles in different pathways. The genes have not yet been linked to osmotic stress.

While wild-type and *gcn4*Δ grew normal under osmotic stress, *arr1*Δ showed a strong decrease in cell growth. The growth defect is rescued in the double deletion strain *gcn4*Δ*/arr1*Δ. This indicates an interaction between both proteins, though the experimental design cannot distinguish between a *cis* or *trans* interaction.

My current working hypothesis on the mechanism of the interaction is shown in Figure 1.10C. I expect most genes commonly regulated by both TFs to be salt stress responders (because of the condition-specificity of OHC). It is known from previous experiments [31] that Gcn4p acts as a repressor under osmotic stress. By positioning Arr1p upstream of and inhibiting Gcn4p, this model explains the observations from the growth assay experiments. The removal of Arr1p from the system probably leads to genes important for osmo-adaptation to be repressed by Gcn4p, reducing cell growth rate. The removal of Gcn4p has no noticeable effect in this model. The double knockout reestablishes conditions close to wild-type, where genes are only regulated by the osmotic stress response.

Mutant cycle analysis was performed (see [61]) to elucidate the mechanism of this interaction. Briefly, transcriptional profiling was done for single and double deletion strains, before and after exposure to osmotic stress conditions (0.8 M NaCl, see Figure 1.11 for a comparison of all profiles). For each gene, its expression under osmotic stress was explained by a linear model accounting for an effect of the GCN4 deletion, an effect of the ARR1 deletion, and their interaction effect. I selected the genes whose interaction effect was positive and larger than $\log_2 1.5$ (45 genes). Then, I filtered this group for genes showing a decrease in expression in the *arr1*Δ arrays and an expression similar to wild-type in the double mutant (leaving 37 genes). The genes should be salt stress responders and thus should show a two-fold increase of their wild-type expression under osmotic stress relative to wild-type expression in synthetic complete medium. This criterion reduced the candidate set to nine genes (Figure 1.10D). The filtering criteria were chosen in accordance to the expected model (Figure 1.10C). When shuffling the arrays and applying the same criteria I find at most two genes, showing that the result is not random. Four of the nine candidate genes are uncharacterized ORFs (YDR366C, YJL107C, YMR034C and YGR066C), Bop2p and Spg4p are proteins of unknown function. The other candidates are involved in a variety of biological processes such as heme degradation, pheromone induced signalling, survival at high temperature or as a membrane protein (SGD [62]). This suggests a novel function of these genes as a part of the osmotic stress response pathway, albeit their roles are unclear and a Blastn/Blastp homology search did not help reveal their function.

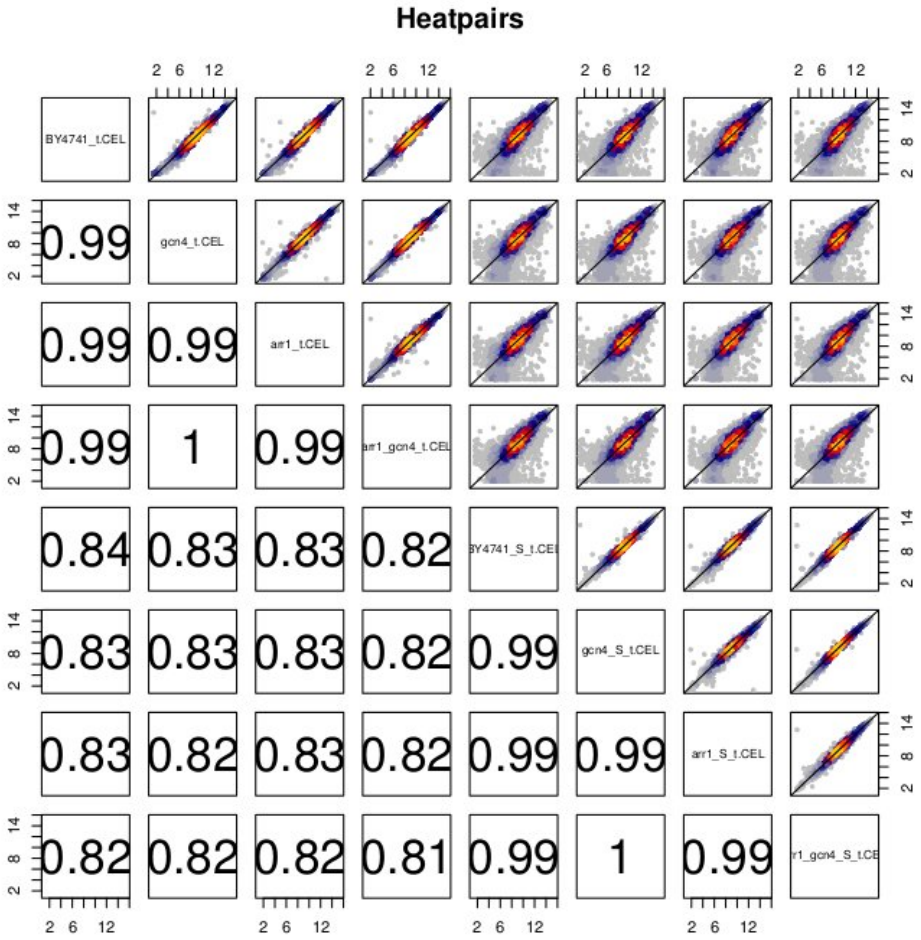


Figure 1.11: Pairwise comparison of mRNA log expression from the mutant cycle analysis microarrays. I compared wild-type strains, GCN4 and ARR1 knockouts as well as double knockouts in normal growth conditions and osmotic stress conditions. Most of the variance between arrays stems from the comparison of normal growth condition and osmotic stress (upper right quadrant).

1.4 Discussion

One Hand Clapping has been established as a method to predict condition-specific TF interactions; its implementation, which does not require any parameter adjustments by the user, is provided as a software package for R. It takes advantage of the increasingly reliable and comprehensive resources on gene-specific transcriptional regulators. OHC is data-inexpensive; *two* genome-wide gene activity measurements (under normal and stress conditions) are already sufficient. With this sparse input I derive a robust interaction measure that is stable on many different types of gene activity data. Despite its modest sensitivity, its predictions are relevant due to their high specificity.

Applied on osmotic stress data and TF-target relations from YEASTRACT, OHC predicts 59 interactions. 23 of the interactions can be validated by BioGRID (22%). While gene activity data is available for many different conditions in all organisms, it may be difficult to find a mapping of TFs to a set of target genes suitable for OHC in other organisms. For the yeast *S. cerevisiae* there are fortunately several options available, the most important being the YEASTRACT database [36] and the data set provided by MacIsaac *et al.* [14]. When I run the method using the latter, I predict 38 interactions, only 6 of which can be validated by BioGRID (16% prediction accuracy). While the annotation from MacIsaac *et al.*, based on ChIP-chip data, is of high quality, it does not suit my purpose, as it contains assignments made under standard experimental conditions. YEASTRACT contains many TF-target gene assignments under different stress conditions and knockout strains.

It is important to note that the predictions made by OHC are entirely different from predictions based on target genes set alone. Indeed, a straightforward Fisher test for target gene overlap does not find the same TF interactions as OHC (data not shown). In particular, the method can and does predict interactions between TFs that have no overlap in target genes and thus no interaction score. This is possible because I predict interactions based on profile similarity which takes into account the interaction scores with all other TFs. I found three TF interactions without target gene overlap: Kar4p-Stb1p, Rds1p-YJL206C and Cbf1p-Mig2p.

In silico validation of the method is based on all interactions between TFs submitted to BioGRID. As this repository is not exhaustive, the performance measurements in this chapter represent conservative estimates. Moreover, entries in BioGRID are biased towards interactions present under normal growth conditions and frequently studied stress conditions, as these account for the large part of the studies that contributed to BioGRID. I selected one novel candidate pair (Gcn4p-Arr1p) for in vivo validation by growth assays under osmotic stress. The growth defect in the *arr1*Δ strain showed a synthetic rescue phenotype in

the *arr1*Δ*/gcn4*Δ double deletion strain. Subsequent gene expression analysis revealed nine candidate genes potentially involved in the synthetic rescue, not previously connected to osmotic stress.

Application to a large data set comprising 16 conditions showed that different pairs are detected in each condition. I compiled a compendium of confident condition-specific interactions, where each pair has to be predicted in at least half of the experiments for each condition (stability). This provides a resource for studying functionally relevant condition-specific TF interactions. Since different interactions are predicted in different conditions I confirm that TF combinatorics drive adaptation to environmental challenges.

This method can be extended in several ways: First, the linear model from which the interaction score is derived can be replaced by a more elaborate physical model of TF activation, as has been attempted by [63, 64]. Currently these models fall short of describing TF competition adequately [65, 66]. Nonetheless I speculate that the inclusion of chromatin structure, in particular nucleosome positioning, in the interaction score will improve the method. Second, OHC can be generalized to other organisms, as reliable TF-target annotations will become available. Finally, the screening principle introduced here lends itself to generalization: The only property of TFs that enters the model is that each TF splits the genes into two disjoint sets (targets vs. non-targets), i.e., each TF defines a binary property on the set of genes. It is therefore straightforward to perform a condition-specific interaction screen on any collection of binary properties, such as pathway membership (e.g. KEGG [67]) or functional annotation (e.g. GO [68]).

2 Signal Network reconstruction

At the center of molecular biology are the processes surrounding transcription and translation. These well studied processes are accompanied by regulatory mechanisms. When modelling theses processes one has to account for the temporal dimension. There is for instance a delay between the transcription of messenger RNA and the existence in the cell of an active translated protein that can take part in regulating downstream mechanisms. Most of the time this delay is not known or measurable and has to be treated as missing data in any interaction network inference method (though some sensible interval can be safely assumed for any delay). To solve this task I propose dynamic Boolean networks and report the exact likelihood for the case of unknown time delays.

2.1 Introduction

The inference of signaling networks from biological data is of fundamental importance for a systemic understanding of regulatory processes. The statistical methods that have been developed for that purpose can be grouped according to the type of data which they expect as input. Many approaches use gene expression data. Some methods are based solely on static observations of the unperturbed system; they exploit the fact that fluctuations of interacting components are dependent [69, 70]. The use of perturbation data greatly improves network reconstruction [71, 72, 73]. In order to resolve the order of events in a signaling cascade, time-resolved measurements after perturbation yield further improvements [6, 74]. Boolean networks are an appropriate tool for dealing with this type of data [7, 75, 76, 77]. The most difficult problem lies in accounting for the mostly unknown time delays with which the signal is propagated through the network [78].

In this work, I propose Boolean networks with probabilistic time delays as a novel statistical network inference method. There have been attempts to calculate the likelihood of a Boolean network in special cases by using MCMC sampling [79] and for dynamic nested effects models [80, 81]. Exact results were so far obtained only under strong restrictions on the logic functions involved, like in the context of conjunctive Bayesian networks [82, 83]. By analytically marginalizing over the unknown delay times, I derive the main result, an exact and efficient recursive likelihood formula for a very broad class of Boolean

networks with exponentially-distributed time delays that may include feedback loops. I evaluate this method in various simulation scenarios for its ability to recover the unknown topology. The method is then applied to a murine stem cell knockdown data set by Ivanova et al. [84], which consists of a set of whole genome gene expression time series after the knockout of six genes (Essrb, Sox2, Nanog, Tcl1, Oct4 and Tbx3) that are considered key regulators in the maintenance and differentiation of mouse embryonic stem cells. My analysis reveals more feedback loops than previously detected.

The algorithm is implemented in the statistical language R. Code and documentation are available as resource `codeboons.zip` accompanying this thesis.

2.2 Methods

I aim to model central aspects of dynamic signaling networks, namely combinatorial regulation, and time delayed responses in gene activity. All signaling components are considered either active or inactive, i.e., they are represented as binary variables. The activity of each component is modeled as a Boolean function of its parent variables in the network. Signaling in biological networks occurs with time delays, which are suitably modeled by the Boolean networks introduced below.

2.2.1 Boolean Networks with unknown time delays and interventions

Let $\mathcal{G} = \{1, ..., N\}$ be a set of N signaling components that dynamically interact with each other via transcriptional regulation, and let $\mathbb{F} = \{0, 1\}$ be a Boolean field. This model represents intracellular gene regulation by a directed graph given by an adjacency matrix $\Gamma \in \mathbb{F}^{\mathcal{G}\times\mathcal{G}}$. It is understood that $\Gamma_{ij} = 1$ whenever i is a parent, i.e., a regulator of j. At each time point t, a gene $j \in \mathcal{G}$ is characterized by two Boolean variables $A_j(t)$ and $B_j(t)$. The induction state variable $A_j(t)$ tells us whether gene j is either transcribed at its basic rate or whether it exhibits altered transcription ($A_j(t) = 0$ or 1, respectively). The activity state variable $B_j(t)$ reports whether the signaling molecule j is in its basic functional state or whether its function is altered at time point t ($B_j(t) = 0$ or 1, respectively, see Figure 2.1). It helps to think of the induction states as genes and their expression, and the activity states as the corresponding gene products (proteins) and their activity as transcription factors. Although protein activity can be measured in some instances, it is generally hard to obtain time-resolved data. Therefore, I will infer the activity variables from the expression of their known target genes. The induction state $A_j(t)$ of j at time t is determined

instantaneously by the activity states $B_i(t)$ of its parents $i \in \mathrm{pa}\,(j) \subseteq \mathcal{G}$ via a Boolean function $f_j : \mathbb{F}^{pa(j)} \to \mathbb{F}$,

$$A_j(t) = f_j(B_i(t);\, i \in \mathrm{pa}\,(j))\ ,\quad j \in \mathcal{G},\, t \in [0,\infty) \tag{2.1}$$

If $\mathrm{pa}(j) = \emptyset$, f_j is a constant. The family $\{f_j \mid j \in \mathcal{G}\}$ of Boolean functions is denoted by $\mathcal{F}$. The changes in the activity state of gene j are transmitted to changes in the corresponding activity state with a constant time delay $d_j \in [0,\infty)$,

$$B_j(t) = \begin{cases} A_j(t-d_j) & \text{for } t \geq d_j \\ A_j(0) & \text{else} \end{cases}\ ,\quad j \in \mathcal{G},\, t \in [0,\infty) \tag{2.2}$$

Let $\Delta = \{d_j \mid j \in \mathcal{G}\}$. The graph Γ, together with $\mathcal{F}$ and Δ define the dynamics of all binary variables in the model.

In order to completely specify the Boolean network, one needs to initialize the values of $A_j(t)$ at $t = 0$. Through an intervention experiment, some induction states are actively set to 1, $A_j(0) = 1$ (e.g., by a gene knockdown), while the rest of the variables are initialized by 0. At the same time, all feedback to an actively perturbed induction state variable A_j is blocked, which is reflected by the removal of all incoming edges to A_j.

In practical situations the delay times Δ are rarely known. I account for this fact by considering the delay times as unknowns for which one specifies their prior distribution. The prior is a product of independent exponential distributions, one for each individual delay time,

$$\pi(\Delta;\Lambda) = \prod_{j=1}^{N} \pi_j(d_j;\,\lambda_j)\ ,\quad \pi_j(d_j;\,\lambda_j) = \begin{cases} \lambda_j \exp(-\lambda_j d_j) & \text{if } d_j \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

Here, $\Lambda = (\lambda_j)$ is a tuple of appropriately chosen positive hyper-parameters, and a complete parametrization of the model is given by the tuple $\mathcal{M} = (\Gamma, \mathcal{F}, \mathcal{L}, \Lambda)$.

2.2.2 The likelihood function

Let $\mathbf{B} = \{B_j(\tau_k); j \in \mathcal{G},\, k = 0, ..., K\}$ the observations of the binary state variables B_j at $K+1$ time points $0 = \tau_0 < \tau_1 < \tau_2 < ... < \tau_K$. Given a parametrization $\mathcal{M}$ of the model and some initial activation pattern, one seeks to calculate the probability of observing $\mathbf{B}$, by integration over the unknown delay times,

$$P(\mathbf{B} \mid \Gamma, \mathcal{F}, \Lambda) = \int_{\Delta} P(\mathbf{B} \mid \Gamma, \mathcal{F}, \Delta) \cdot \pi(\Delta;\Lambda) \tag{2.4}$$

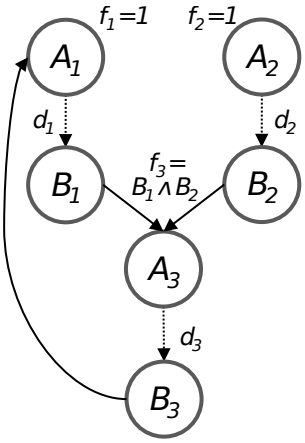


Figure 2.1: Schematic of the model for a fixed timepoint t: A_i and B_i are the induction and activity states, respectively of each regulator $\{1, 2, 3\}$. The delays in signaling between an alteration of the gene state and an ensuing alteration of the activity state, are given by $\Delta = (d_i)$. Given all parent-child relationships of the network, $\mathcal{F} = \{f_1, f_2, f_3\}$ is the family of Boolean functions. Functions for nodes with less than two parents (A_1 and A_2) are constant.

The major technical achievement of this work is the closed-form solution of the integral in Equation (2.4) for arbitrary Boolean networks (possibly including cycles) that satisfy a rather general admissibility condition (see section 2.2.3). Note that the class of Boolean networks that can be inferred includes all acyclic networks, and all networks that allow each node to switch only once, yet it is substantially larger. As the derivation of this result requires tedious calculations and elaborate notation, I give the algorithm for the likelihood calculation separately in Algorithms 2.1 and 2.2 and put the details of the mathematical derivation of the closed form solution of $P(\mathbf{B} \mid \mathcal{M})$ into its own section (section 2.2.4) . I also prepared a table of all symbols used throughout this chapter in Appendix A, *List of symbols used in dynamic Boolean network learning.* Some quantities arising during the calculation become extremely small which bears the risk of underflow errors. Therefore all necessary computations were performed in log space instead of using standard floating point arithmetic (see Appendix A, *Calculations in Log space* for details). Having scored a Boolean network, we search the space of all admissible signaling graphs by Markov Chain Monte Carlo as outlined in Husmeier [85] and Appendix A, *Details on MCMC.*

Note that this framework easily allows the modeling of a series of intervention experiments. Each intervention will produce its own sequence of state observa-

tions $\mathbf{B}$, and each sequence will be evaluated separately by actively initializing the expression states of perturbed variables with 1 and blocking all feedback to this state by the removal of all incoming edges.

input: maximal scoring state sequence: $\mathbf{B}_{max}$
hyper parameter for the distribution of the delay times: Λ
local probability functions: $P(D|\mathbf{B}, \mathcal{L})$
1 Find $\mathcal{N}(\mathbf{B}_{max})$, all state sequences at Hamming distance 1 of $\mathbf{B}_{max}$
2 Run an MCMC chain over all admissible Boolean networks $(\Gamma, \mathcal{F})$. Acceptance or rejection of proposed models $(\Gamma, \mathcal{F})$ is based on their likelihood L
3 **foreach** *proposed Boolean network* **do**
4 **foreach** $\mathbf{B} \in \mathcal{N}(\mathbf{B}_{max})$ **do**
// calculate $S = P(\mathbf{B} \mid \Gamma, \mathcal{F}, \Lambda)$
5 find the set $\mathcal{K}$ of all compatible κ
6 $S_{\mathbf{B}} = \sum_{\kappa \in \mathcal{K}} \texttt{score}(\mathbf{B}, \kappa, \Lambda)$
7 **end**
// calculate likelihood L of $(\Gamma, \mathcal{F})$
8 $L = \sum_{\mathbf{B} \in \mathcal{N}(\mathbf{B}_{max})} S_{\mathbf{B}} \cdot P(D \mid \mathbf{B}, \mathcal{L})$
9 **end**

Algorithm 2.1: Calculation of the likelihood and search through the space of admissible Boolean networks. The scoring function (line 6) is detailed in Algorithm 2.2

2.2.3 Admissibility check for a Boolean Network $(\Gamma, \mathcal{F})$

I state a condition on the logical structure $(\Gamma, \mathcal{F})$ of the Boolean Network under which one is able to solve the problem of likelihood computation (Equation 4 of the main text). Starting with an initial configuration of $A_j(0) = B_j(0)$, $j \in \mathcal{G}$, we update the Boolean network asynchronously, i. e. one node per time. Let A_j^{old}, B_j^{old}, $j \in \mathcal{G}$, denote the configuration before an update step, and A_j, B_j, $j \in \mathcal{G}$ denote the configuration after the update step. The required condition is that for all configurations that can be reached from any initial configuration using asynchronous updates, one has

$$A_j \neq A_j^{old} \Rightarrow B_j^{old} = A_j^{old} \tag{2.5}$$

All networks satisfying (2.5) are called admissible. This condition can be checked by enumerating all possible trajectories that can result from the starting states configuration. Although this might in theory be of complexity $2^{|\mathcal{G}|}$, this

Function `score`$(\mathbf{B}, \kappa, \alpha = \Lambda)$:
input: state sequence $\mathbf{B}$,
switch time κ,
parameter of the integral α
For each k find the interval $[\tau_{i_k}, \tau_{i_{k+1}}]$ where the switch in the state sequence $\mathbf{B}$ happens
calculate:

$$F(j,\beta,\alpha;\kappa) = \int_{t_1=\tau_{i_1}}^{\tau_{i_1+1}} ... \int_{t_k=\tau_{i_k}}^{\tau_{i_k+1}} \left[exp(\beta t_j) \prod_{i=1}^{k} \pi_i(t_i - t_{\kappa(i)}; \alpha_i) \right] dt_k dt_{k-1} \dots dt_1$$

This is done using the following recursion formula:

$$F(k,\beta,\alpha;\kappa) \; = \; \begin{cases} 1 \\ \hat{c}(k,\beta;\alpha) \cdot F(k, 0, \hat{\alpha}(k,\beta;\alpha);\kappa) \\ F(0,0,(\alpha_1,...,\alpha_{k-1});\kappa) - \exp(-\alpha_k \tau_{i_k+1}) \cdot F(\kappa(k), \alpha_k, (\alpha_1,...,\alpha_{k-1});\kappa) \\ [\exp(-\alpha_k \tau_{i_k}) - \exp(-\alpha_k \tau_{i_k+1})] \cdot F(\kappa(k), \alpha_k, (\alpha_1,...,\alpha_{k-1});\kappa) \end{cases}$$

$$\begin{cases} \text{if } \alpha = \emptyset \\ \text{if } \alpha \neq \emptyset, \beta > 0 \\ \text{if } \alpha \neq \emptyset, \beta = 0, t_{\kappa(k)} \geq \tau_{i_k} \\ \text{if } \alpha \neq \emptyset, \beta = 0, t_{\kappa(k)} < \tau_{i_k} \end{cases}$$

Here, $\hat{\alpha}(j,\beta;\alpha)$ and $\hat{c}(j,\beta;\alpha)$ are constants defined as

$$\hat{\alpha}(j,\beta;\alpha) = (\hat{\alpha}_i(j,\beta;\alpha))_{i=1,...,k}, \; \hat{\alpha}_i(j,\beta;\alpha) = \begin{cases} \alpha_i & \text{if } i \notin \{j, \kappa(j), \kappa^2(j), ...\} \\ \alpha_i - \beta & \text{if } i \in \{j, \kappa(j), \kappa^2(j), ...\} \end{cases}$$

$$\hat{c}(j,\beta;\alpha) = \prod_{s \in \{j,\kappa(j),\kappa^2(j),...\}} \frac{\alpha_s}{\alpha_s - \beta}$$

Algorithm 2.2: Scoring a single state sequence $\mathbf{B}$, given κ and α. The recursion will split into two separate cases whenever B_k and its predecessor $B_{\kappa(k)}$ switched values within the same observation interval. If this happens too often, network reconstruction will be impossible anyway. In practice, a sufficient temporal resolution will imply that scaling of the algorithm is roughly linear in the number of state switches

is irrelevant for networks of moderate size ($|\mathcal{G}| < 10$). Additionally, the set of all trajectories that emerge from a given starting configuration is typically much smaller than $2^{|\mathcal{G}|}$. Condition (2.5) can be stated equivalently and more intuitively: if for some $j \in \mathcal{G}$, $t_1 < t_3$, we have $A_j(t_1) \neq A_j(t_3)$ then $B_j(t_2) = A_j(t_1)$ for some t_2, $t_1 < t_2 < t_3$

Remark that this is a rather weak restriction. In many practical cases, condition (2.5) is automatically fulfilled and does not need to be checked at all. E.g., if all functions $f \in \mathcal{F}$ are monotonic in the sense that $x = (x_1, ..., x_n) \leq y = (y_1, ..., y_n)$ (component-wise inequality, letting $0 \leq 1$) implies $f(x) \leq f(y)$, then condition (2.5) always holds. Another important case in which condition (2.5) is automatically met is if the topology Γ of the Boolean network does not include cycles.

2.2.4 Closed form solution of $P(\mathbf{B} \mid \mathcal{M})$

Each observation $\mathbf{B}$ can equivalently be represented by a list of 'switching events', i.e. a list of tuples $(j_s, \tau_s, B_{j_s}(\tau_s))$, $s = 1, ..., S$ denoting that variable B_{j_s} switched its value form $B_{j_s}(\tau_s)$ to $\sim B_{j_s}(\tau_s)$ in the interval between the observation time points τ_s and τ_{s+1}. Let $\sigma \in \mathrm{Sym}(S)$, denote the true order in which these switching events occurred. Although σ is unknown, it is strongly constrained by the intervals $[\tau_s, \tau_{s+1}]$ in which the switching events occurred. The reason is that after sorting the events list according to σ, $\tau_s \leq \tau_{s+1}$ must hold for all s, i.e., the event $s+1$ occur ed in the same or in a later observation interval than event s. Such an order σ of events is called 'admissible'. For the moment, fix one admissible σ and rearrange the switching events list according to σ. By the admissibility condition (2.5), each switching event $(j_s, \tau_s, B_{j_s}(\tau_s))$ is preceded by a switch of $A_{j_s}(t)$. Since $A_{j_s}(t) = f_{j_s}(B_i(t); i \in \mathrm{pa}\,(j_s))$, there is necessarily a unique most recent switching event $\kappa(s) \in pa(j_s)$ among the input nodes of f_{j_s} which led to the change in $A_{j_s}(t)$. Note that $\kappa(s) < s$.

I split the calculation of $P(\mathbf{B} \mid \Gamma, \mathcal{F}, \Lambda)$ into disjoint areas. For all permutations $\sigma \in \mathrm{Sym}(S)$, define the pairwise disjoint sets $R_\sigma = \{t = (t_1, ..., t_S) \mid 0 < t_{\sigma 1} < t_{\sigma 2} < ... < t_{\sigma N}\}$. The positive orthant of $\mathbb{R}^N$ is (up to some set of Lebesgue measure zero) the disjoint union of R_σ, $\sigma \in \mathrm{Sym}(S)$. Denote by $T = (T_1, ..., T_S)$ the change points at which the switching event s occurred. Thus,

$$P(\mathbf{B} \mid \Gamma, \mathcal{F}, \Lambda) \;=\; \sum_{\sigma \in \mathrm{Sym}(N)} P(\mathbf{B}, T \in R_\sigma \mid \Gamma, \mathcal{F}, \Lambda) \tag{2.6}$$

I will now provide a closed form solution for each summand in Equation (2.6).

By Bayes' rule,

$$P(\mathbf{B}, T \in R_\sigma \mid \Gamma, \mathcal{F}, \Lambda) = \int_\Delta P(\mathbf{B}, T \in R_\sigma \mid \Gamma, \mathcal{F}, \Delta) \cdot \pi(\Delta; \Lambda) \tag{2.7}$$

If A_j does not have any parents, $\kappa(j)$ is set to zero. for all $j = 1, ..., N$. The change points T depend deterministically on the delay times $\Delta = (d_1, ..., d_S)$, namely

$$T_s = d_s + T_{\kappa(s)} \ , \ s = 1, ..., S$$

It is convenient to re-parametrize the integral in (2.7) in terms of the change point coordinates. To that end, define the transformation

$$\varphi : \mathbb{R}^S \to \mathbb{R}^S \ , \quad \varphi(d) = \varphi(d_1, ..., d_S) = (t_1 = d_1 + T_{\kappa(1)}, ..., t_S = d_S + T_{\kappa(S)})$$

(in the above definition, I set $t_0 = 0$). Note that φ is well defined, because the definition of t_j recurs only on an already defined t_k, $k < j$. Moreover, φ is a bijective linear transformation with determinant 1, because it can be represented as $\varphi(d) = Ad$, where A is a lower triangle matrix with unit diagonal. Using integral transformation by φ, one can therefore re-parametrize the integral in (2.7),

$$\begin{aligned} P(\mathbf{B}, T \in R_\sigma \mid \Gamma, \mathcal{F}) &= \int_d P(\mathbf{B}, T \in R_\sigma \mid \Gamma, \mathcal{F}, d) \cdot \pi(d; \Lambda) && (2.8) \\ &= \int_t P(\mathbf{B}, T \in R_\sigma \mid \Gamma, \mathcal{F}, t_s - t_{\kappa(s)}; s = 1, ..., S) \cdot \\ &\quad \cdot \prod_{s=1}^{S} \pi_s(t_s - t_{\kappa(s)}; \lambda_{j_s}) && (2.9) \end{aligned}$$

Importantly, here I assume that the delay times from switching events of the nodes $B_{j_s}(t)$ all have independent delay times sampled from the identical prior $\pi_j(d_j; \lambda_j)$ whenever $j_s = j$. Given the protein states $\mathbf{B}$ at the observation time points, the interval within which a variable B_j changed its state is known. This means that T_s is confined to the unique interval $I_s = [\tau_{j_s}, \tau_{j_s+1}]$, $j_s \in \{0, ..., K\}$, for which $B_{j_s}(\tau_{j_s}) = 0$ and $B_{j_s}(\tau_{j_s+1}) = 1$ (here let $\tau_0 = 0$ and $\tau_{K+1} = \infty$). The vector $T = (T_1, ..., T_S)$ can therefore only assume values in the Cartesian product $I = I_1 \times ... \times I_S$. Thus, realize that $P(\mathbf{B}, T \in R_\sigma \mid \Gamma, \mathcal{F}, t_s - t_{\kappa(s)}; s = 1, ..., S)$ simply is an indicator function $\delta(t \in I \cap R_\sigma)$, and the integral (2.9) can be written as

$$\int_{t \in I \cap R_\sigma} \prod_{s=1}^{S} \pi_{j_s}(t_s - t_{\kappa(s)}; \lambda_{j_s}) = \int_{t_1 = \tau_{i_1}}^{\tau_{i_1+1}} ... \int_{t_S = \tau_{i_S}}^{\tau_{i_S+1}} \prod_{s=1}^{S} \pi_{j_s}(t_s - t_{\kappa(s)}; \lambda_{j_s}) \tag{2.10}$$

The last expression in (2.10) can be calculated recursively. Evaluating the innermost integral yields terms of the form $\exp(\alpha_i t_j)$ which contain the integrand and cannot be removed. Thus for any $\alpha = (\alpha_1, ..., \alpha_k)$, β, $j, k \in \{1, ..., N\}$, one will have to evaluate expressions of the form

$$B(j, \beta, \alpha; \kappa) = \int_{t_1=\tau_{i_1}}^{\tau_{i_1+1}} ... \int_{t_k=\tau_{i_k}}^{\tau_{i_k+1}} \left[exp(\beta t_j) \prod_{i=1}^{k} \pi_i(t_i - t_{\kappa(i)}; \alpha_i) \right] dt_k dt_{k-1} \dots dt_1 \tag{2.11}$$

A preparatory Lemma is needed which will help us rearrange these terms in the course of the calculations.

Lemma 1. Let $\alpha = (\alpha_1, ..., \alpha_k)$, $j \in \{1, ..., N\}$. Provided that

$$\beta \notin \{j, \kappa(j), \kappa^2(j), ...\}$$

we have

$$\prod_{j=1}^{k} \pi_j(t_j - t_{\kappa(j)}; \alpha_j) \cdot exp(\beta t_j) = \hat{c}(j, \beta; \alpha) \prod_{j=1}^{k} \pi_j(t_j - t_{\kappa(j)}; \hat{\alpha}_j(j, \beta; \alpha)) \tag{2.12}$$

for

$$\hat{c}(j, \beta, \alpha; \kappa) = \prod_{s \in \{j, \kappa(j), \kappa^2(j), ...\}} \frac{\alpha_s}{\alpha_s - \beta} \tag{2.13}$$

and

$$\begin{aligned} \hat{\alpha}(j, \beta, \alpha; \kappa) &= (\hat{\alpha}_i(j, \beta, \alpha; \kappa); i = 1, ..., k) \\ &= \hat{\alpha}_i(j, \beta, \alpha; \kappa) = \begin{cases} \alpha_i & \text{if } i \notin \{j, \kappa(j), \kappa^2(j), ...\} \\ \alpha_i - \beta & \text{if } i \in \{j, \kappa(j), \kappa^2(j), ...\} \end{cases} \end{aligned}$$

with the convention that the empty product equals 1.

Proof.

It is sufficient to show

$$\exp(\beta t_j) \prod_{i=1}^{k} \pi_i(t_i - t_{\kappa(i)}; \alpha_i) = \hat{c}(\beta, j; \alpha) \cdot \prod_{i=1}^{k} \pi_i(t_i - t_{\kappa(i)}; \hat{\alpha}_i(\beta, j; \alpha)) \tag{2.14}$$

Both sides of Equation (2.14) are zero if $t_i < t_{\kappa(i)}$ for some $i = 1, ..., k$. One

may therefore assume without loss that $t_i \geq t_{\kappa(i)}$ for all $i = 1, ..., k$, and

$$\begin{aligned}
\exp(\beta t_j) \cdot \pi_j(t_j - t_{\kappa(j)}; \alpha_j) &= \exp(\beta t_j) \cdot \alpha_j \exp(-\alpha_j(t_j - t_{\kappa(j)})) \\
&= \alpha_j \exp(-\alpha_j(t_j - t_{\kappa(j)}) + \beta(t_j - t_{\kappa(j)}) + \beta t_{\kappa(j)}) \\
&= \frac{\alpha_j}{\alpha_j - \beta} \cdot (\alpha_j - \beta) \exp(-(\alpha_j - \beta)(t_j - t_{\kappa(j)})) \cdot \\
&\quad \cdot \exp(\beta t_{\kappa(j)}) \\
&= \frac{\alpha_j}{\alpha_j - \beta} \cdot \pi_j(t_j - t_{\kappa(j)}; \alpha_j - \beta) \cdot \exp(\beta t_{\kappa(j)})
\end{aligned} \tag{2.15}$$

The result follows by a simple induction on k in Eq. 2.14.

Lemma 2.

The following equation holds:

$$\begin{aligned}
\int_{t_k=\tau_{i_k}}^{\tau_{i_k+1}} \pi_k(t_k - t_{\kappa(k)}; \alpha_k) &\overset{(*)}{=} \int_{t_k=\max(t_{\kappa(k)}, \tau_{i_k})}^{\tau_{i_k+1}} \alpha_k \exp(-\alpha_k(t_k - t_{\kappa(k)})) \\
&= \begin{cases} 1 - \exp(-\alpha_k \tau_{i_k+1}) \cdot \exp(\alpha_k t_{\kappa(k)}) \\ [\exp(-\alpha_k \tau_{i_k}) - \exp(-\alpha_k \tau_{i_k+1})] \cdot \exp(\alpha_k t_{\kappa(k)}) \end{cases} \\
&\quad \begin{cases} \text{if } t_{\kappa(k)} \geq \tau_{i_k} \\ \text{if } t_{\kappa(k)} < \tau_{i_k} \end{cases}
\end{aligned} \tag{2.16}$$

Remember the convention $\kappa(k) = 0$ if node A_k has no parent, and $t_0 = 0$.

Proof.

First, note that in step$(*)$, $t_{\kappa(k)} < t_k \leq \tau_{i_k+1}$, and the lower integration bound $\max(\tau_{i_k}, t_{\kappa(k)})$ is indeed not greater than the upper integration bound.

For $t_{\kappa(k)} \geq \tau_{i_k}$:

$$\begin{aligned}
\int_{t_k=max(t_{\kappa(k)}, \tau_{i_k})}^{\tau_{i_k+1}} \alpha_k \cdot \exp(-\alpha_k(t_k - t_{\kappa(k)})) &= \int_{t_k=t_{\kappa(k)}}^{\tau_{i_k+1}} \alpha_k \exp(-\alpha_k(t_k - t_{\kappa(k)})) \\
&= [-\exp(-\alpha_k(t_k - t_{\kappa(k)}))]_{t_{\kappa(k)}}^{\tau_{i_k+1}} \\
&= -\exp(-\alpha_k(\tau_{i_k+1} - t_{\kappa(k)})) + \\
&\quad + \exp(-\alpha_k(t_{\kappa(k)} - t_{\kappa(k)})) \\
&= 1 - \exp(-\alpha_k \tau_{i_k+1}) \cdot \exp(\alpha_k t_{\kappa(k)})
\end{aligned}$$

For $t_{\kappa(k)} < \tau_{i_k}$:

$$\begin{aligned}\int_{t_k=max(t_{\kappa(k)},\tau_{i_k})}^{\tau_{i_k+1}} \alpha_k \cdot \exp(-\alpha_k(t_k - t_{\kappa(k)})) &= \int_{t_k=\tau_{i_k}}^{\tau_{i_k+1}} \alpha_k \cdot \exp(-\alpha_k(t_k - t_{\kappa(k)})) \\ &= [-\exp(-\alpha_k(t_k - t_{\kappa(k)}))]_{\tau_{i_k}}^{\tau_{i_k+1}} \\ &= \exp(-\alpha_k(\tau_{i_k} - t_{\kappa(k)})) - \\ &\quad - \exp(-\alpha_k(\tau_{i_k+1} - t_{\kappa(k)})) \\ &= [\exp(-\alpha_k\tau_{i_k}) - \exp(-\alpha_k\tau_{i_k+1})] \cdot \\ &\quad \cdot \exp(-\alpha_k t_{\kappa(k)})\end{aligned}$$

Theorem 3. Let $\alpha = (\alpha_1, ..., \alpha_k)$. Set $B(0, \beta, \emptyset; \kappa) = 1$. The following recursions hold:

$$B(k, 0, \alpha; \kappa) = \begin{cases} B(0, 0, (\alpha_1, ..., \alpha_{k-1}); \kappa) - \exp(-\alpha_k\tau_{i_k+1}) \cdot \\ \quad \cdot B(\kappa(k), \alpha_k, (\alpha_1, ..., \alpha_{k-1}); \kappa) & \text{if } t_{\kappa(k)} \geq \tau_{i_k} \\ [\exp(-\alpha_k\tau_{i_k}) - \exp(-\alpha_k\tau_{i_k+1})] \cdot \\ \quad \cdot B(\kappa(k), \alpha_k, (\alpha_1, ..., \alpha_{k-1}); \kappa) & \text{if } t_{\kappa(k)} < \tau_{i_k} \end{cases} \tag{2.17}$$

$$B(k, \beta, \alpha; \kappa) = \hat{c}(k, \beta, \alpha; \kappa) \cdot B(k, 0, \hat{\alpha}(k, \beta, \alpha; \kappa); \kappa)$$

with $\hat{c}$ and $\hat{\alpha}$ as defined in Lemma 1.

Proof.

Using Lemma 2 (L2),

$$
\begin{aligned}
B(k,0,\alpha;\kappa) &= \int_{t_1=\tau_{i_1}}^{\tau_{i_1+1}} ... \int_{t_k=\tau_{i_k}}^{\tau_{i_k+1}} \left[\prod_{j=1}^{k} \pi_j(t_j - t_{\kappa(j)};\alpha_j) \right] \\
&= \int_{t_1=\tau_{i_1}}^{\tau_{i_1+1}} ... \int_{t_k=\tau_{i_{k-1}}}^{\tau_{i_{k-1}+1}} \prod_{j=1}^{k-1} \pi_j(t_j - t_{\kappa(j)};\alpha_j) \quad \left[\int_{t_k=\tau_{i_k}}^{\tau_{i_k+1}} \pi_k(t_k - t_{\kappa(k)};\alpha_k) \right] \\
&\overset{\text{L2}}{=} \int_{t_1=\tau_{i_1}}^{\tau_{i_1+1}} ... \int_{t_k=\tau_{i_{k-1}}}^{\tau_{i_{k-1}+1}} \prod_{j=1}^{k-1} \pi_j(t_j - t_{\kappa(j)};\alpha_j) \\
&\quad \cdot \begin{cases} 1 - \exp(-\alpha_k \tau_{i_k+1}) \cdot \exp(\alpha_k t_{\kappa(k)}) & \text{if } t_{\kappa(k)} \geq \tau_{i_k} \\ [\exp(-\alpha_k \tau_{i_k}) - \exp(-\alpha_k \tau_{i_k+1})] \cdot \exp(\alpha_k t_{\kappa(k)}) & \text{if } t_{\kappa(k)} < \tau_{i_k} \end{cases} \\
&= \begin{cases} B(0,0,(\alpha_1,...,\alpha_{k-1});\kappa) - \exp(-\alpha_k \tau_{i_k+1}) \cdot \\ \quad \cdot B(\kappa(k),\alpha_k,(\alpha_1,...,\alpha_{k-1});\kappa) & \text{if } t_{\kappa(k)} \geq \tau_{i_k} \\ [\exp(-\alpha_k \tau_{i_k}) - \exp(-\alpha_k \tau_{i_k+1})] \cdot \\ \quad \cdot B(\kappa(k),\alpha_k,(\alpha_1,...,\alpha_{k-1});\kappa) & \text{if } t_{\kappa(k)} < \tau_{i_k} \end{cases}
\end{aligned}
$$

$$
\begin{aligned}
B(k,\beta,\alpha;\kappa) &= \int_{t_1=\tau_{i_1}}^{\tau_{i_1+1}} ... \int_{t_k=\tau_{i_k}}^{\tau_{i_k+1}} \left[exp(\beta t_k) \prod_{j=1}^{k} \pi_j(t_j - t_{\kappa(j)};\alpha_j) \right] \\
&= \int_{t_1=\tau_{i_1}}^{\tau_{i_1+1}} ... \int_{t_k=\tau_{i_k}}^{\tau_{i_k+1}} \left[\hat{c}(k,\beta,\alpha;\kappa) \prod_{j=1}^{k} \pi_j(t_j - t_{\kappa(j)};\hat{\alpha}_i(k,\beta,\alpha;\kappa)) \right] \\
&\overset{\text{L1}}{=} \hat{c}(k,\beta,\alpha;\kappa) \cdot B(k,0,\hat{\alpha}(k,\beta,\alpha;\kappa);\kappa)
\end{aligned}
$$

2.3 Likelihood in the case of hidden activity states

In this application, the activity states can not be observed directly and are therefore treated as hidden variables that emit data D. The likelihood function in this case is obtained by formally integrating over all possible state sequences $\mathbf{B}$,

$$
P(D \mid \Gamma, \mathcal{L}, \mathcal{F}, \Delta) = \sum_{\mathbf{B} \in \mathbb{F}^{N \times K}} P(D \mid \mathbf{B}, \mathcal{L}) \cdot P(\mathbf{B} \mid \Gamma, \mathcal{F}, \Delta) \qquad (2.18)
$$

The computation of the last factor $P(\mathbf{B} \mid \Gamma, \mathcal{F}, \Delta)$ has been described above. In addition to that, there is another computational challenge. The summation over all states $\mathbf{B} \in \mathbb{F}^{N \times K}$ is not feasible, because the state space is by far too large.

One can view Equation (2.18) as a weighted sum of the terms $P(\mathbf{B} \mid \Gamma, \mathcal{F}, \Delta)$, weighted $P(D \mid \mathbf{B}, \mathcal{L})$, respectively, then restrict the summation to only those terms whose weight is sufficiently high. How to find the highest weights without the enumeration of all states? Exploit the fact that in this model, the hidden state variables B_j change their value from 0 to 1 at most once in their time course. This means that for each B_j one can summarize its time course by denoting the time at which the state change occurs by the random variable T_j called change point. The contribution of the hidden state B_j to $P(D \mid \mathbf{B}, \mathcal{L})$ is

$$\prod_{k=1}^{K} P(D_j(\tau_k) \mid B_j(\tau_k)) = \prod_{k=1}^{K} P(D_j(\tau_k) \mid B_j = \delta(\tau_k > T_j)) \tag{2.19}$$

where $\delta(\tau_k > T_j) = \begin{cases} 1 & \text{if } \tau_k > T_j \\ 0 & \text{else} \end{cases}$ is the indicator function. Thus, there are at most $K+1$ different time courses for B_j ((0,0,...,0), (0,0,...,0,1), ... (0,1,1,...,1), (1,1,...,1)). Enumerating these, one finds the time course for B_j that maximizes the term in (2.19). Doing so for all $j \in \mathcal{G}$, one can find the best scoring state $\mathbf{B}_{max}$. I realized that each deviation from $\mathbf{B}_{max}$ reduces the score in (2.19) substantially. This suggests that all high scoring states can be found in the direct vicinity (as measured by Hamming distance) of $\mathbf{B}_{max}$. I realized that in practice, to speed up calculations, it is sufficient to use $\mathbf{B}_{max}$ only (Figure 2.2).

2.4 Results

2.4.1 Performance on synthetic data

Having in mind the application to stem cell differentiation data with six genes (see Section 2.4.2), I manually chose five representative topologies with six nodes for the simulation studies with an OR as sole Boolean function. The delay times d_g for each gene g were sampled uniformly from the interval $[1, 30]$ minutes. The measurements were generated after $t = \{0, 15, 30, 45, 60\}$ minutes. For each topology, I then calculated the binary activity patterns $B_g(t)$ for each single gene knockout g. The local probability distributions $\mathcal{L} = \{P(D_j \mid B_j);\ j \in \mathcal{G}\}$ are taken as

$$P(D_j \mid B_j) \sim \mathcal{N}(D_j; \mu = B_j, \sigma^2)$$

for $\sigma \in \{0.006, 0.12, 0.24, 0.36, 0.46, 0.58\}$. I assume that vague prior knowledge about the delay times is available by choosing the hyper-parameter λ_g of the exponential delay time prior such that their expected value equals the respective true delay time d_g.

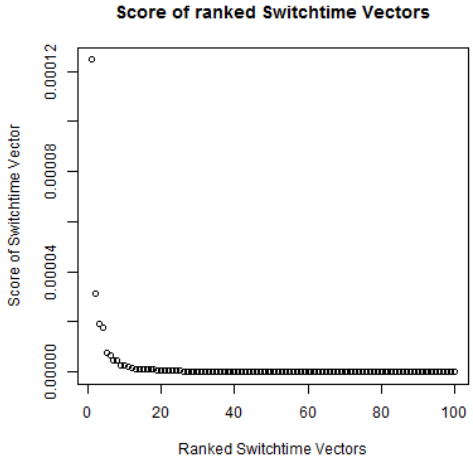


Figure 2.2: Time course vectors $B_j = (\delta(\tau_k > T_j; j,k \in \{1,...,K\})$ ranked by score. The scores decrease rapidly with increasing Hamming distance to the best scoring state $\mathbf{B}_{max}$. In my calculation I will use $\mathbf{B}_{max}$ as only contribution to the sum in Equation 2.19

For each topology I started 100 MCMC runs with 2000 steps (see Appendix chapter A, *Details on MCMC* for details on the MCMC method). The likelihood requires the summation over all possible state sequences $\mathbf{B} \in \mathbb{F}^{N \times K}$. This makes calculations infeasible even for medium sized networks. I address this problem by restricting the model space search to state sequences that are in the immediate vicinity of the best scoring state sequence B_{max}. To find B_{max}, we exploit the fact that in this model, the hidden state variables B_j change their value from 0 to 1 at most once in their time course (due to the monotonicity of the chosen Boolean function, OR). This means that for each B_j onw can summarize its time course by denoting the time at which the state change occurs by the random variable T_j called change point. The contribution of the hidden state B_j to $P(D \mid \mathbf{B}, \mathcal{L})$ is

$$\prod_{k=1}^{K} P(D_j(\tau_k) \mid B_j(\tau_k)) = \prod_{k=1}^{K} P(D_j(\tau_k) \mid B_j = \delta(\tau_k > T_j)) \tag{2.20}$$

where $\delta(\tau_k > T_j) = \begin{cases} 1 & \text{if } \tau_k > T_j \\ 0 & \text{else} \end{cases}$ is the indicator function. Thus, there are at most $K+1$ different time courses for B_j ((0,0,...,0), (0,0,...,0,1), ... (0,1,1,...,1), (1,1,...,1)). Enumerating these, I find the time course for B_j that maximizes the

term in (2.20). Doing so for all $j \in \mathcal{G}$, I find the best scoring state $\mathbf{B}_{max}$. Figure 2.3 shows the results for all five topologies. The model shows a good overall performance for low and moderate noise levels. It performs best on tree topologies (Figure 2.3E), which are often encountered in biological pathways. Another frequent pathway motif is the feed-forward loop, as modeled in Figure 2.3B. The addition of feedback to the linear topology in Figure 2.3A decreases performance, but it still remains at a reasonable level. Figure 2.3F shows the results on a biological network from literature (of the stem cell differentiation pathway from [79]). Specificity and sensitivity are comparable to the simpler topologies A-E.

2.4.2 Application to stem cell differentiation data

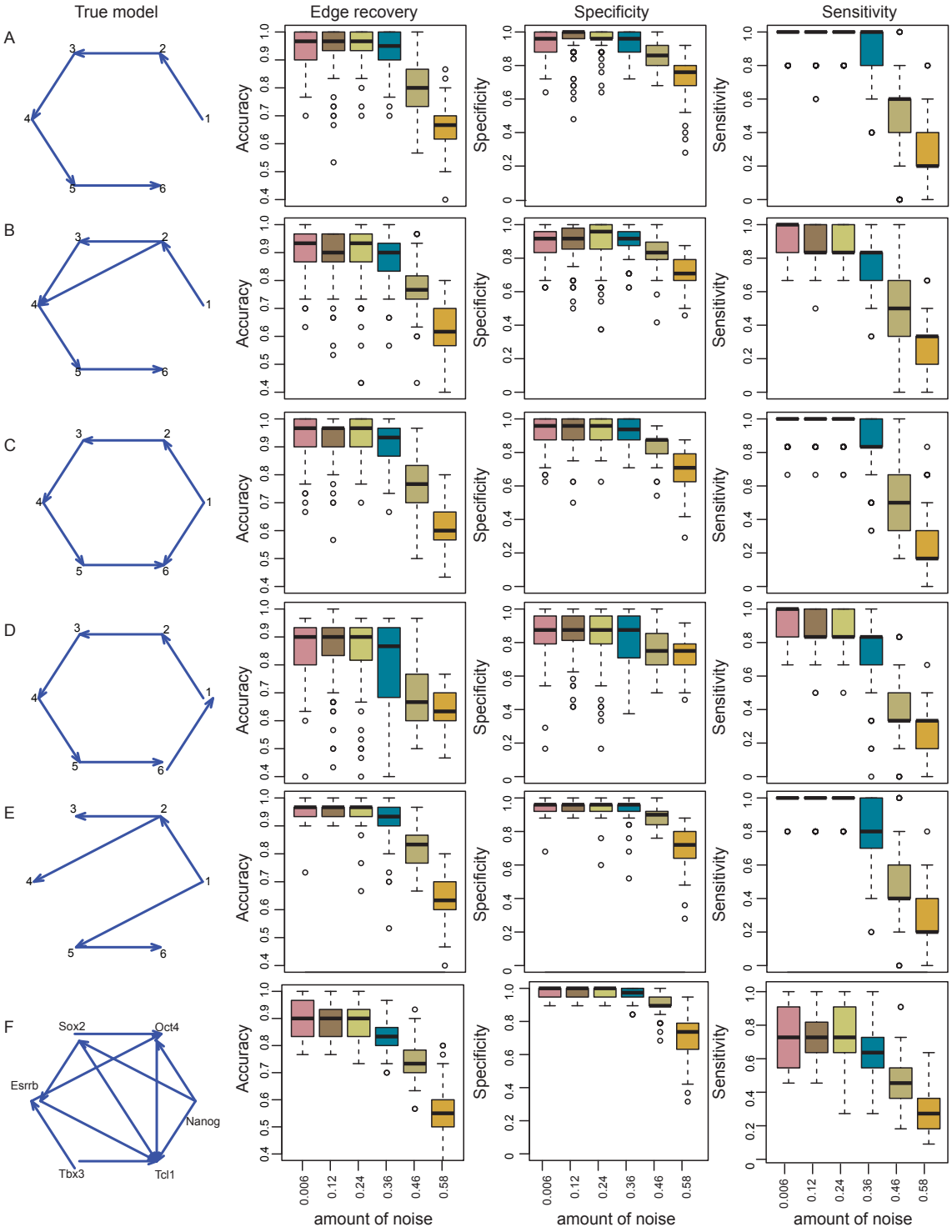
This model calls for time series measurements of protein activities after single gene knockouts. Data of that kind is still sparse. I circumvented this problem and increase the applicability of the method by treating the binary activity state variables as hidden variables. The data consists of time series of measurements $D = (D_j(\tau_k))$ of the activity states $B_j(\tau_k)$, $j \in \mathcal{G}$, at a finite number of $K+1$ time points $0 = \tau_0 < \tau_1 < \tau_2 < ... < \tau_K$. The data $D_j(\tau_k)$ can be thought of as a noisy, possibly replicate, quantification of the hidden activity states $\mathbf{B} = \{B_j(\tau_k); j \in \mathcal{G}, k = 0, ..., K\}$. I relate measurements to their underlying activity state through time-independent local probability distributions $\mathcal{L} = \{P(D_j \mid B_j); j \in \mathcal{G}\}$. Given the hidden induction states $\mathbf{B}$ and the local probabilities $\mathcal{L}$, the probability of observing D is

$$P(D \mid \mathbf{B}, \mathcal{L}) = \prod_{j=1}^{N} \prod_{k=1}^{K} P(D_j(\tau_k) \mid B_j(\tau_k)) \tag{2.21}$$

Note that Equation (2.21) assumes independence of observations. The likelihood then becomes

$$P(D \mid \Gamma, \mathcal{L}, \mathcal{F}, \Delta) \quad = \quad \sum_{\mathbf{B} \in \mathbb{F}^{N \times K}} P(D \mid \mathbf{B}, \mathcal{L}) \cdot P(\mathbf{B} \mid \Gamma, \mathcal{F}, \Delta) \tag{2.22}$$

Thus I can apply the method to the dataset of Ivanova et al. [84] who used short hairpin RNA (shRNA) loss of function techniques to down regulate genes whose expression patterns suggest self-renewal regulatory functions in mouse embryonic stem cells. Genome-wide gene expression time series measurements after $t = 0, 1, ..., 7$ days were obtained after knockdown of each of the following genes: Nanog, Oct4, Sox2, Tbx3, Esrrb and Tcl1. These genes are known to play a major role in stem cell differentiation and are therefore called "major genes". Anchang et al. [79] built a model with this knock-down data using dynamic nested effect models.


True model
Edge recovery
Specificity
Sensitivity
A
B
C
D
E
F
Sox2
Oct4
Esrrb
Nanog
Tbx3
Tcl1
Accuracy
Specificity
Sensitivity
amount of noise
0.006
0.12
0.24
0.36
0.46
0.58

The major genes represent the nodes in this network. The variables $A_j(t)$ and $B_j(t)$ correspond to their gene and protein activities respectively. Since the activity states $B_j(t)$ are not directly measured by [84], I use the expression activity of gene groups under the regulatory control of the major genes (the E-Genes in the nested effect model of [79]) as a proxy for their protein activity. To get the local probabilities $P(D_j|B_j)$ needed in the case of assuming hidden $B_j(t)$ we use data from 122 genes given as discretized time series representing admissible patterns (see the Supplemental Materials of [79] for details). In accordance with this definition, genes in their basic state were assigned the value 0, and assumed the value 1 upon activation. I kept the grouping of the 122 genes into six groups of genes depending on Nanog, Oct4, Sox2, Tbx3, Esrrb or Tcl1 discovered from the E-Genes graph from Anchang et al. Since the data contain

---

Figure 2.3 *(facing page)*: Results of the simulations study on five topologies (first column). The second column shows the performance as percentage of correctly predicted edges (presence and absence) for different noise levels σ added to the binary activity patterns as a box plot over all 100 runs of the MCMC. The third and fourth column show the distribution of sensitivity and specificity of network reconstruction across all runs. **A**: linear graph. This topology can be predicted with high accuracy up to noise level 0.36. **B**: linear graph with feed-forward loop. This topology is also correctly predicted up to noise level 0.36 although losing 0.1 performance points compared to the linear graph without shortcut. **C**: linear graph with forward-jump to the last node. The model can better predict this case than the intra-node forward-jump in B. **D**: full cycle. This difficult topology can be predicted by the model with accuracy over 80% up to noise level 0.36. Performance then rapidly degrades. This topology has a high variance in sensitivity/specificity values between the different runs even for low levels of added noise. **E**: tree structure. The model is well adapted to this topology and shows a high performance until noise level 0.36. Its performance is comparable to the linear topology (A). **F**: network of stem cell differentiation as reconstructed by Anchang *et al.*

time series representing the undifferentiated cell culture (the negative control), and the cell culture undergoing normal differentiation (the positive control), I filtered for genes whose expression differed more than two fold at the last time point between the two control experiments. Then, I assigned to each gene at each time point a probability to belong to the basal or the active state, according to whether its expression resembled more the negative or positive control (a likelihood ratio was calculated under the assumption of Gaussian distributions). Using the gene groups defined earlier I calculated a likelihood ratio for each major gene to be active vs. inactive as the product of the corresponding likelihood ratios of the assigned genes (this was done separately for each time point and each knockout). The likelihood ratios are then converted into a probability of being active (at a certain time point, in a certain knockdown experiment), which corresponds to the input required for this model.

In this application I only use the Boolean function AND, leading to monotonic activity states $\mathbf{B}$. As described in Section 2.4.1 I chose the state sequence $\mathbf{B}_{max}$ that maximizes $P(D \mid \mathbf{B}, \mathcal{L})$.

Using the same MCMC procedure as in the simulation setting, the stationary chain comprised 155 unique models. We used model averaging and calculated the weighted frequencies of each edge. Each model was weighted by its number of occurrences in the Markov chain, resulting in a probabilistic adjacency matrix (Figure 2.4A). Tcl1 has the lowest connectivity, while Nanog has the highest. To compare the results of this model with the model from Anchang *et al.* (Figure 2.4C), I converted the probabilistic adjacency matrix into a graph by drawing all edges with a probability above 0.5 (Figure 2.4B). The most striking difference of Figure 2.4B compared to Figure 2.4C is the presence of cycles. In particular, the major genes Oct4, Sox2, Nanog and Esrrb form a maximal clique of the graph. The two graphs essentially agree on the position of Tcl1, which in both cases is targeted by Tbx3 and Esrrb. Also, Tbx3 is located mostly upstream of the Oct4, Sox2, Nanog, Esrrb clique in both graphs. Still, it is puzzling why my method finds a highly interconnected, feedback-loop rich structure, whereas Anchang *et al.* find a sparser solution. Note that the method in Anchang *et al.*, assumes an acyclic graph structure, and hence by definition cannot find cycles. As the simulation studies have shown that the model can accurately predict circular structures in regulatory graphs, the feedback in this network might be higher, and the signaling hierarchy less pronounced than previously thought. This is confirmed by a different approach to mouse embryonic stem cell network reconstruction [86] that also discovers a large amount of interplay between the key regulators of stem cell differentiation. Zhou et al. have also reconstructed a mouse embryonic stem cell network based on transcription factor binding sites, protein interactions and literature annotation. They show bidirectional interactions of Oct4 with Nanog and Sox2 coinciding with my findings (Figure 2.4D).

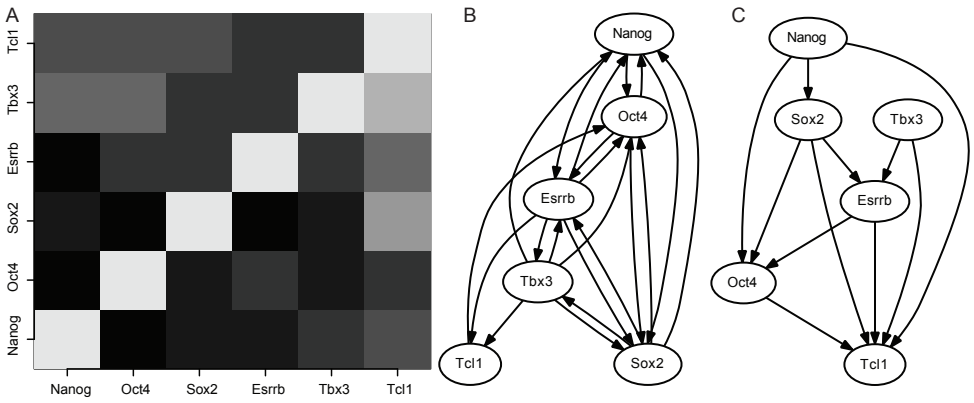


Figure 2.4: **A**: Adjacency matrix of the result of the network inference on the biological dataset. Each entry corresponds to the observed frequency and is colored accordingly with lighter colors representing lower frequencies **B**: network obtained from A by setting a threshold of 0.5 on the edge probability **C**: Network inferred by Anchang *et al.* [79] **D**: Extract from the network published in Zhou *et al.* [86]. The authors did not include Tbx3 and Tcl1 in their findings. Dashed edges in B and C represent edges that are not present in my model (A). All other edges from B and C are also found in model A.

2.5 Discussion

In this work I developed an algorithm that permits to analyze gene knockdown time series experiments which have high dimensional readouts (such as gene expression). In order to elucidate the interplay of the major regulators, all of them need to be perturbed and measured individually. On the side of methods development, I have solved the problem of calculating the likelihood function for data generated from a Boolean network with probabilistic, exponentially distributed time delays (Algorithm 2). The likelihood function can be used for network reconstruction, as has been demonstrated in simulation studies. Having a closed form solution for the likelihood has several further applications that I did not mention so far. It is possible to sample the joint distribution $P(\mathbf{B} \mid \Gamma, \mathcal{F}, \Lambda)$ rather efficiently, because many observations $\mathbf{B}$ can be excluded a priori knowing Γ and $\mathcal{F}$. This allows for accounting for some hidden variables B_k among the observed $\mathbf{B}$ by integrating them out. Furthermore it is possible to calculate the expectation of a certain B_j to be on or off in a given time interval. As an application, I have devised a method to apply it to data in which the values of the Boolean network can only be observed indirectly (Algorithm 1). I analyzed murine stem cell differentiation data of Ivanova et al. [84] for the purpose of signaling network reconstruction. Comparison to a previous reconstruction attempt in Anchang et al. [79] revealed a much richer feedback structure than expected. The method suggests regulatory feedback loops that lead to a better understanding of the dynamic interplay of some master regulators in murine embryonic stem cell development. I expect our method to find numerous applications, as protein abundance data becomes increasingly available [87].

3 Independence testing

Independence testing is useful in many diverse applications such as economics, sports betting, physics, biology and many more. It is also closely related to network inference. A well-known inference algorithm for Bayesian networks, the PC algorithm [1] is based on independence testing to infer network structure. This algorithm will search for a subset of nodes S such that two nodes X and Y are conditionally independent (conditioned on S). One writes $I(X, Y \mid S)$ in that case. Starting from a complete graph, for each connected nodes X and Y for which there exists a subset of nodes S such that $I(X, Y \mid S)$ one removes the edge between X and Y. After this procedure is done for all nodes of the node set, the remaining edges in the graph are oriented according to a fixed set of rules.

If the set of independence relationships (defined by the network) is faithful (see [1] for a definition of this term) to a graph and one has a perfect way of determining whether $I(X, Y \mid S)$, then the algorithm guarantees to infer a graph equivalent (meaning it represents the same set of independence relationships) to the graph that generated the data. This means that the result of the inference is strongly dependent on the test used for determining (conditional) independence.

In this chapter I present different novel tests of independence based on the exact distribution of the ith nearest neighbour of any point on a two dimensional torus.

3.1 Introduction

Dependence measures and tests for independence have recently attracted a lot of attention, because they are the cornerstone of algorithms for network inference in probabilistic graphical models. Pearson's product moment correlation coefficient is still by far the most widely used statistic in areas such as economy, biology and the social sciences. Yet Pearson's correlation is largely constrained to detecting linear relationships. Spearman [88] and Kendall [89] extended Pearson's work to monotonic dependencies. In 1948, Hoeffding [90] proposed a non parametric test for independence that is suited for many different functional relationships. Székely et al. introduced the distance correlation (dcor) as a generalization of Pearson's correlation.

Other approaches build on mutual information (MI). MI characterizes independence in the sense that the MI of a joint distribution of two variables is zero if and only if these variables are independent. However, MI is difficult to estimate from finite samples. Kraskov et al. [91] proposed an accurate MI estimator derived from nearest neighbor distances. Reshef et al. [92] presented the maximal information coefficient (MIC), a measure of dependence for two-variable relationships which was heavily advertised [93] but lacks any statistical motivation.

dcor and Kraskov's estimator use the pair-wise distances of the points in a sample as a sufficient statistic. In this work I provide an exact formula for the ith nearest neighbor distance distribution of rank-transformed data ($i = 1, 2, ...$). Based on that, I propose two novel tests for independence. An implementation of these tests, together with a general benchmark framework for independence testing, are freely available as a CRAN software package (`http://cran.r-project.org/web/packages/knnIndep`) and as resource `knnIndep_1.0.tar.gz` accompanying this thesis. In this thesis I have benchmarked Pearson's correlation, Hoeffding's D, dcor, Kraskov's estimator for MI, MIC and my two tests. I conclude that no particular method is generally superior to all other methods. However, dcor and Hoeffding's D are the most powerful tests for many different types of dependence. Circular dependencies, e.g., are best recognized by my tests. This type of dependence is fairly common, e.g., if two dependent periodic processes are monitored. An example from biology is the expression of a transcription factor and one of its target genes during the cell cycle [94].

3.2 Exact distribution of the ith nearest neighbour distances

Consider a set of $N \geq 4$ points that are distributed 'randomly' on a surface. In what follows, I derive the distribution (conditional distribution) of the $(i+1)$th nearest neighbor of a point (given the distance to its ith neighbor). I assume the points drawn from the following model: Let $X = (x_j)_{j=1,...,N}$ and $Y = (y_j)_{j=1,...,N}$ be permutations of the numbers $0, ..., N-1$ that are drawn uniformly from the set of all permutations of $\{0, ..., N-1\}$. The points $z_j = (x_j, y_j)$, $j = 1, ..., N$, lie on a torus of size N which is endowed with the maximum distance as a metric. I. e., the distance between two points is given by

$$dist(z_1, z_2) = \max(\min(|x_1 - x_2|, N - |x_1 - x_2|), \min(|y_1 - y_2|, N - |y_1 - y_2|))$$

Fix a reference point z_1. Let d_i, $i = 1, ..., N-1$ denote the distance of the ith nearest neighbor of z_1 to z_1 and D_i the random variable associated with it. Since this distance measure is translation invariant, let without loss

$z_1 = (x_1, y_1) = (0, 0)$. My target is the calculation of the conditional probability $P(D_{i+1} \mid D_i)$ and the marginal $P(D_i)$. The main work will be the calculation of the probability $P(D_{i+1} \geq c,\, D_i \leq a)$ for given values a and c. Once this is done, $P(D_i)$ and $P(D_{i+1} \mid D_i)$ can be derived by the following calculations.

Set $d_0 = 0$ for convenience, and let $a \leq c$, $a \leq \lfloor \frac{N}{2} \rfloor$, $i \in \{0, ..., N-2\}$. Note that

$$P(D_{i+1} \geq c,\, D_i = a) = P(D_{i+1} \geq c,\, D_i \leq a) - P(D_{i+1} \geq c,\, D_i \leq a-1) \quad (3.1)$$

First calculate the marginal distribution

$$\begin{aligned} P(D_i = a) &= P(D_{i+1} \geq a,\, D_i = a) \\ &= P(D_{i+1} \geq a+1,\, D_i = a) + P(D_{i+1} = a,\, D_i = a) \\ &= P(D_{i+1} \geq a+1,\, D_i \leq a) - P(D_{i+1} \geq a+1,\, D_i \leq a-1) + \\ &\quad + P(D_{i+1} = a,\, D_i = a) \end{aligned} \quad (3.2)$$

The three terms in the expression will be explicitly derived further on (Equations 3.5 and 3.2).

Consequently,

$$\begin{aligned} P(D_{i+1} \leq c \mid D_i = a) &= 1 - P(D_{i+1} \geq c+1 \mid D_i = a) \\ &= 1 - \tfrac{P(D_{i+1} \geq c+1,\, D_i = a)}{P(D_i = a)} \end{aligned} \quad (3.3)$$

which then leads to

$$P(D_{i+1} = c \mid D_i = a) = P(D_{i+1} \leq c \mid D_i = a) - P(D_{i+1} \leq c-1 \mid D_i = a) \quad (3.4)$$

All these formulas of course only hold for those choices of a and i for which the probability $P(D_{i+1} \geq a,\, D_i = a)$ is non-zero.

I determine $P(D_{i+1} \geq c,\, D_i \leq a)$ by counting the number of admissible point configurations and dividing through $(N-1)!$, the number of all possible point configurations with $z_1 = (0, 0)$ fixed. When counting configurations, I repeatedly exploit the fact that each horizontal and each vertical grid line contains exactly one point from the sample. In case of $c > a$, I split the torus into 3 regions (Figure 3.1). Region I is a square of side length $2a + 1$. It contains z_1 and i additional points at arbitrary positions. The number of possibilities to draw an i-tuple from $2a$ positions (recall that one position is already taken by z_1) without replacement is $\frac{(2a)!}{(2a-i)!}$. Thus, there are $\left(\frac{(2a)!}{(2a-i)!}\right)^2$ i-tuples describing an admissible configuration in region I. However, each configuration is counted $i!$

times, since the order of the points does not matter. Hence, the number of unique configurations in region I equals $\frac{1}{i!}\left(\frac{(2a)!}{(2a-i)!}\right)^2 = \left(\begin{array}{c} 2a \\ i \end{array}\right)^2 i!$. For the second region there are $N-2c+1$ possible y-coordinates and $2c-1-(i+1) = 2c-i-2$ columns to be filled with sample points (note that the columns $-c$ and c belong to region III and that $i+1$ columns are already taken by points in region I). This yields $\frac{(N-2c+1)!}{(N-4c-i+3)!}$ unique configurations for region II. There are $N-2c+1$ points remaining which can be placed freely in the remaining $N-2c$ columns/rows, yielding $(N-2c+1)!$ possibilities. Together we obtain:

$$\begin{aligned} P(D_{i+1} \geq c,\, D_i \leq a) &= \frac{1}{(N-1)!} \cdot \underbrace{\left(\begin{array}{c} 2a \\ i \end{array}\right)^2 i!}_{\text{region I}} \cdot \underbrace{\frac{(N-2c+1)!}{(N-4c+i+3)!}}_{\text{region II}} \cdot \\ &\quad \cdot \underbrace{(N-2c+1)!}_{\text{region III}} \quad N \geq 4 \end{aligned} \tag{3.5}$$

In the case of $c = a$ there is one more complication, because there is a region R of points exactly at distance c, containing at least the ith and $(i+1)$th neighbor of z_1, where the region I overlaps with regions IIa and IIb (Figure 3.1). Let $r \in \{2,3,4\}$ be the number of points in region R and i_0 the number of points strictly inside the square of distance c. I derive a general formula for all admissible configuration in the case of $c = a$, $f(r, i_0, N, c)$. Denote by $k(r, i_0, c)$ the number of admissible point configurations in region R .

On each side of the square region R one can place

$$\begin{aligned} \epsilon &= 2c+1-i_0-\underbrace{1}_{z_1}-\underbrace{2}_{\text{corner points}} \\ &= 2c-2-i_o \end{aligned}$$

points without the two corner points. For each possible number of points r, in region R, $r = \{2,3,4\}$, the number of possible configurations is counted (see Figure 3.2 for $r = 2$). The approach is analogous for $r = 3$ and $r = 4$.

Table 3.2 lists all possible admissible combinations of points in region R. Counting the admissible configurations strictly inside regions I, IIa, IIb and III happens similar to the above cases (Equation 3.5). This leads to the following general formula for all admissible configurations:

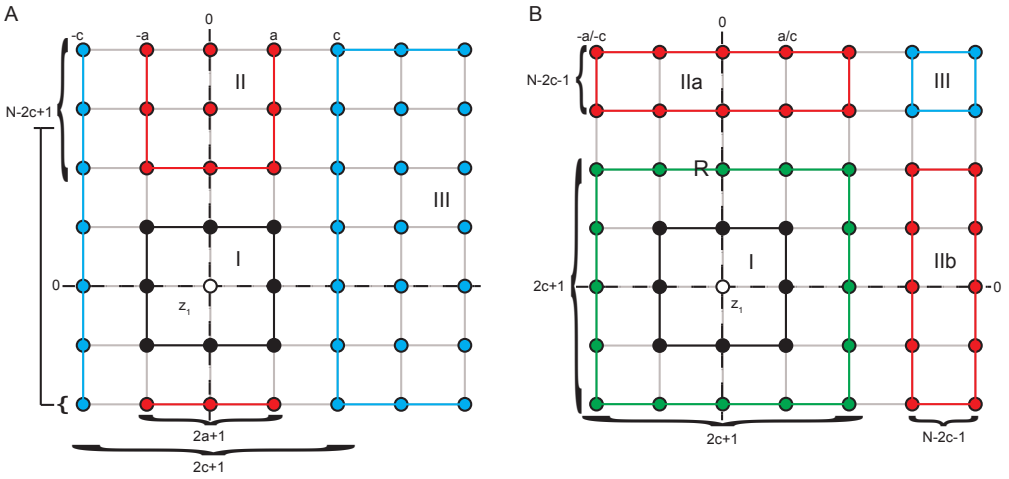


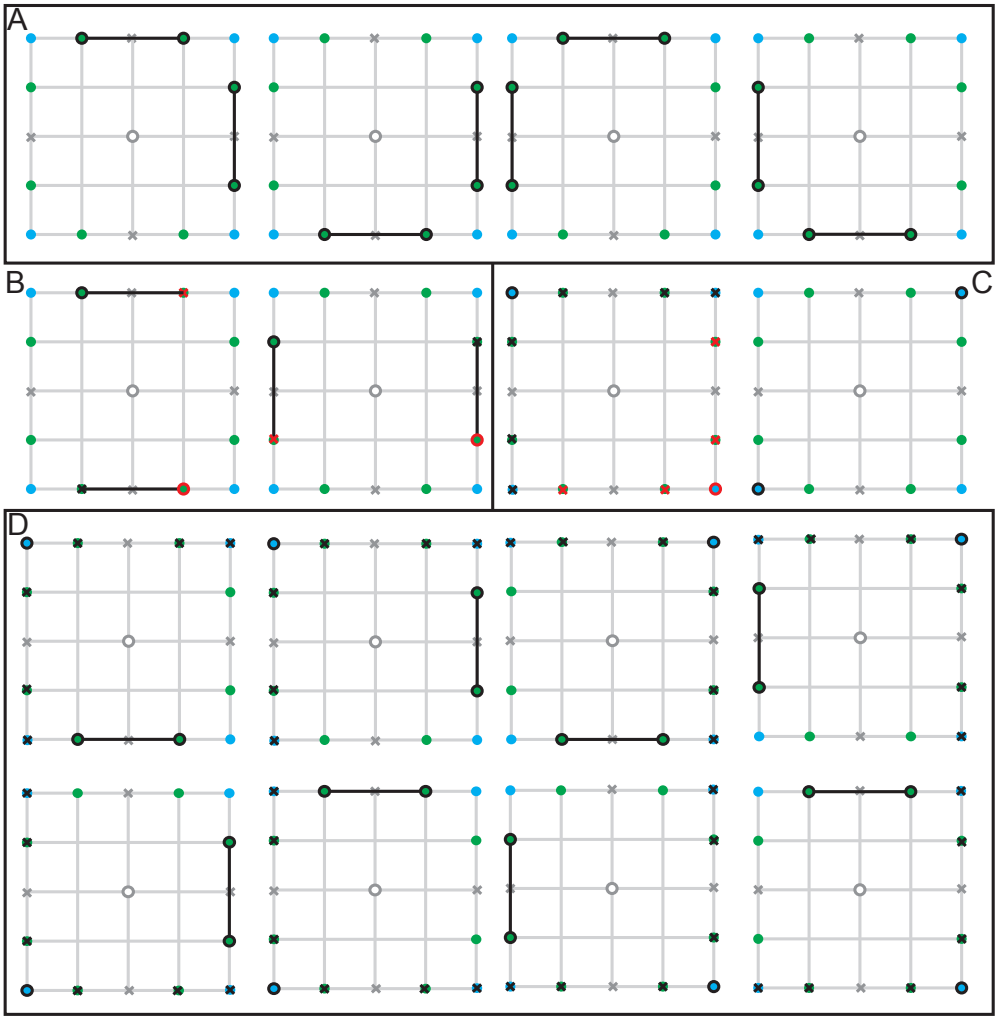
Figure 3.1: Diagrams explaining Equations 3.5 and 3.6 for $N = 7$, $a = 1$ and $c = 2$ (panel A) and $a = c = 2$ (panel B) with the reference point z_1 at coordinates $(0, 0)$. **A**: Let us define 3 regions I, II and III (black, red and blue points respectively). Region I has the least number of constraints and the number of admissible configurations is the number of possibilities to draw i points from $2a$ positions without replacement nor ordering: $\binom{2a}{i}^2 i!$. The number of admissible configurations for region II is given by the number of rows $n_r = N - 2c + 1$ available and the number of columns which remain to be filled $n_c = 2c - i - 2$ according to $\frac{n_r!}{(n_r - n_c)!}$. Region III has the remaining $N-2c+1$ points freely distributed, yielding $(N-2c+1)!$ admissible configurations. **B**: In the case $a = c$ we add an additional region R of r points exactly at distance c (green points). There can be $r = 2$, 3 or 4 such points. Region I has size $(2(c-1))^2$ and $\binom{2c-2}{i_0}^2 i_0!$ admissible configurations with i_0 the number of points strictly inside the square of distance c. Region IIa and IIb are symmetric and handled analogous to region II in panel A with $n_r = N-2c-1$ and $n_c = 2c-i_0-r$. Region III has $(N+i_0+r-4c-1)!$ admissible configurations analogous to panel A.

$$f(r,i_0,N,c)i \;=\; \underbrace{\left(\begin{array}{c}2c-2\\ i_0\end{array}\right)^2 i_0!}_{\text{region I}} \cdot \underbrace{k(r,i_0,c)}_{\text{region R}} \cdot \underbrace{\left[\frac{(N-2c-1)!}{(N-4c+i_0+r-1)!}\right]^2}_{\text{region IIa+IIb}} \cdot \underbrace{(N+i_0+r-4c-1)!}_{\text{region III}} \tag{3.6}$$

The sum over all possible tuple (r, i_0) in Table 3.2 gives the probability $P(d_{i+1} = c,\ d_i = c)$ in the general case:

$$P(\mathrm{R}) = \begin{cases} 0 & \text{if } i_0 > N - r \\ \sum_{(r,i_0)} \frac{1}{(N-1)!} f(r,i_0,N,c) & \text{else} \end{cases} \tag{3.7}$$

Figure 3.2 *(facing page)*: Counting the number of possible configuration for placing 2 points on region R. Let $\epsilon = 2c - i_0 - 2$, the number of free points on an edge of region R (without the corner points shown in blue and the possibilities excluded by z_1(at the center in grey), shown as grey crosses). Here $i_0 = 0$ for convenience, thus $\epsilon = 2$ points (shown in green) **A**: One can choose any points from two adjacent edges. That means ϵ^2 possibilities to place two points and there are 4 such configurations so in total, $4\epsilon^2$ possible configurations. **B**: When placing points on opposing edges, the placement of the first point forbids one possibility on the opposing edge (black cross for the black point and red cross for the red point). Thus there are $2\epsilon(\epsilon-1)$ possible configurations. **C**: Placing one point on the corner forbids both edges connected by this corner (all points marked with a cross). The points thus have to be on opposite corners. There are only 2 possible configurations in this case **D**: Placing the first point on a corner leaves one of the two opposing edges to place points on. There are 8ϵ possibilities. In total for all cases there are $2\epsilon(\epsilon-1) + 4\epsilon^2 + 8\epsilon + 2 = 6\epsilon^2 + 6\epsilon + 2$ possibilities (see Table 3.2 for the number of possibilities for $r = 3$ and $r = 4$).



r	i_0	$k(r, i_0, c)$; let $\epsilon = 2c - 2 - i_0$	condition
2	$i-1$	$2\epsilon(\epsilon-1) + 4\epsilon^2 + 8\epsilon + 2 = 6\epsilon^2 + 6\epsilon + 2$	
3	$i-1$,$i-2$	$4\epsilon^2(\epsilon-1) + 4\epsilon^2 = 4\epsilon^3$	if $i_0 < N - r$
4	$i-1$,$i-2$,$i-3$	$\epsilon^2(\epsilon-1)^2$	

Table 3.1: Table of possible configurations of $r = 2, 3, 4$ points lying exactly on the border region R depending on the parameter i_0. Figure 3.2 shows how the configurations are counted in te case of $r = 2$

The above calculations only hold if region R is a genuine square, for large values of c R degenerates to a pair of lines (one horizontal and one vertical line). These cases are covered in the extended formula

$$P(D_{i+1}=c,\, D_i=c) = \begin{cases} i=1 & \begin{cases} c=1: & P(d_3 \geq 2,\, d_2 \leq 1) \\ c>1: & P(\mathrm{R}), \text{ Equation (3.7)} \end{cases} \\ 1<i<N-2 & \begin{cases} 1<c\leq \lfloor \frac{N}{2} \rfloor: & P(\mathrm{R}), \text{ Equation (3.7)} \\ \text{else}: & 0 \end{cases} \\ i=N-2 & \begin{cases} c=\frac{N}{2},\, N \text{ even}: & \begin{pmatrix} N-2 \\ i-1 \end{pmatrix}^2 (i-1)! \\ \text{else}: & P(\mathrm{R}), \text{ Equation (3.7)} \end{cases} \\ i=N-1 & \begin{cases} c=\lfloor \frac{N}{2} \rfloor: & 1 \\ \text{else:} & 0 \end{cases} \end{cases}$$

Since the above formulas involve tedious calculations, I validated the formulas for $N=7$ and $N=8$ by counting the occurrence of each possible configuration among all $N!$ configurations. Additionally, I checked the validity of the formula for larger N ($N=20$) by taking 10^6 random configurations and comparing the empirical frequency $h(d_i)$ with $P(d_i)$ (see Appendix A, *Validation of the formula*).

Figure 3.3 shows the distribution of $P(d_{i+1} \mid d_i)$ and $P(d_i)$. The conditional distribution is shown for $i=50$. The marginal distribution is highly peaked with a low variance that decreases with increasing i (and reaches 0 for $i=N$).

The formulas have been implemented in the statistical language R [95] with emphasis on a numerically stable implementation as we deal with small numbers. The implementation is vectorized for speed. Still there is a computational penalty through the many factorials and logarithms that have to be calculated. For a sample of size 320, calculating all $P(d_{i+1} \mid d_i)$ takes 4.1 seconds on a single workstation (single thread, Intel(R) Core(TM) i5-2500 CPU @ 3.30GHz). Runtime for larger samples is shown in Figure 3.4 and indicates a practical limit on the sample size of $N<3000$ (which takes up to 3 minutes) and a complexity of $O(N^2)$.

For practical reasons, I assumed that the points lie on a torus (distances on the torus are translation-invariant and therefore the formulas for $P(d_{i+1} \mid d_i)$ and $P(d_i)$ hold for all points in the sample). This will bias results when applied to points on a plane, as points on the border will have different nearest neighbors when projected on the torus. The bias is less pronounced for close neighbors (i small), thus I limit the statistics to $i_{max}=N/2$. I do not expect to lose statistical power, since the information content of $P(d_i)$ for large i approaches zero (see Figure 3.3).

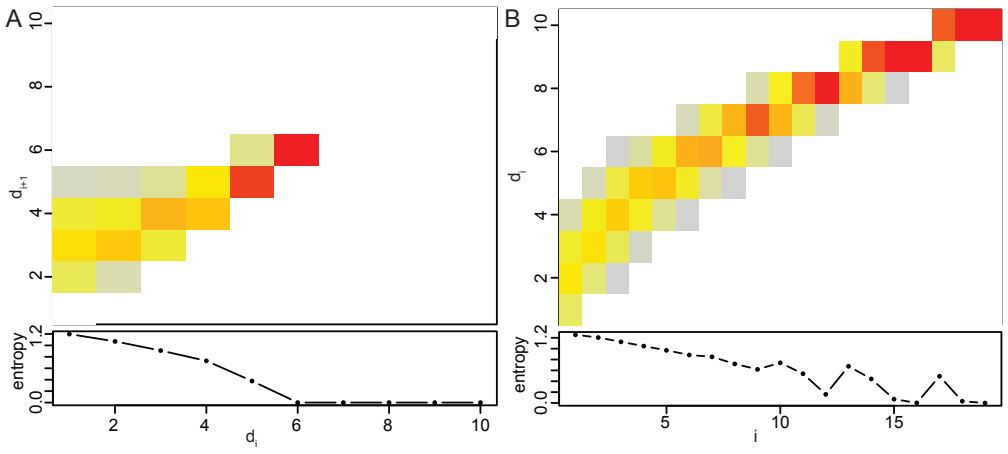


Figure 3.3: **A**: Conditional distribution $p_c = P(D_{i+1} = d_{i+1} \mid D_i = d_i)$ for $i = 2$, $N = 21$ (top) and the entropy $-\sum_{d_{i+1}=1}^{\lfloor \frac{N}{2} \rfloor} p_c \log p_c$(bottom). The probability p_c of observing large (d_{i+1}, d_i) is zero for distances larger than $(6, 6)$ when $i = 2$. The lower triangle is empty because $d_{i+1} \geq d_i$ and the entropy is constantly decreasing for increasing values of d_i because the possible (d_{i+1}, d_i) decrease towards $(6, 6)$. **B**: Marginal distribution $P(d_i)$ for $N = 21$ (top) and entropy $-\sum_{d_i=1}^{\lfloor \frac{N}{2} \rfloor} P(d_i) \log P(d_i)$ (bottom). With increasing i, the distribution becomes narrower and the entropy tends towards 0, as the number of possible distances to the ith nearest neighbour decrease. The non-monotonic behavior of the entropy for large values of i is due to downstream constraints imposed by the maximal distance $\frac{N}{2}$. For testing independence, we advise using all $P(D_{i+1} \mid D_i)$ until the value of i where the entropy starts increasing again ($i = 9$ in this example).

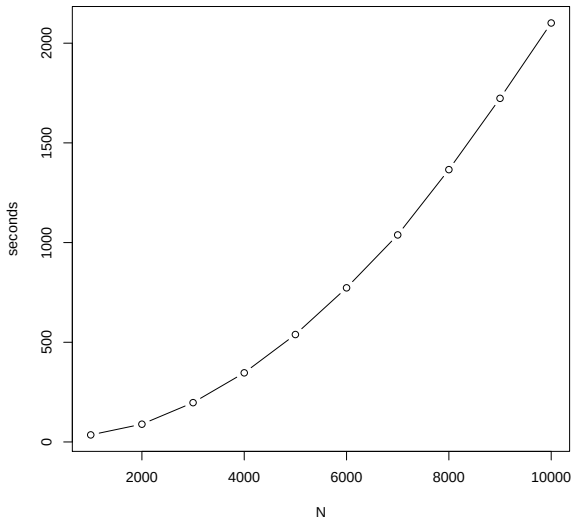


Figure 3.4: Runtime according to sample size N of calculating $P(D_i \mid D_{i-1})$ for all points and all nearest neighbors

3.3 Tests based on the ith nearest neighbour distribution

It has been shown that the distance of the ith nearest neighbour of some point z can be used to estimate the local (log) density at z [91]. My idea is to use the full sequence of nearest neighbour distances for assessing local density. For a sample point z, let $(D_0 = d_0^z = 0, D_1 = d_1^z, D_2 = d_2^z, \ldots, D_{N-1} = d_{N-1}^z)$ the sequence of neighbour distances. If z lies in a dense region, we expect this sequence to increase slower than in a region with lower density.

3.3.1 Distributional tests

Note that $(D_0 = d_0^z = 0, D_1 = d_1^z, D_2 = d_2^z, \ldots, D_{N-1} = d_{N-1}^z)$ is a Markov chain, i.e.,

$$P(d_0^z, d_1^z, d_2^z, \ldots, d_{N-1}^z) = \prod_{i=0}^{N-2} P(d_{i+1}^z \mid d_i^z)$$

That way, taking z as the center point, the distances d_{i+1}^z, given the previous distance d_i^z , are pairwise independent for all i. On the other hand this is not true

for the distances $d_{i+1}^{z_1}$ and $d_{i+1}^{z_2}$ (not even if one conditions on $d_i^{z_1}$ and $d_i^{z_2}$ respectively). This follows from the triangle inequality in metric spaces, $dist(z_1, x) \leq dist(z_2, x) + dist(z_1, z_2)$, which implies that $d_{i+1}^{z_1} \leq d_{i+1}^{z_2} + dist(z_1, z_2))$.

Let the random variable C_i be defined by the process of drawing Z uniformly from $1, ..., N$ and then drawing C_i according to the distribution $P(D_i \mid D_{i-1} = d_{i-1}^z)$. Let f_i denote the probability function of C_i, it is given by

$$\begin{aligned} f_i(c) &= P(C_i = c) \\ &= \sum_{z=1}^{N} P(C_i = c \mid Z = z) \cdot P(Z = z) \\ &= \sum_{z=1}^{N} P(D_i = c \mid D_{i-1} = d_{i-1}^z) \cdot P(Z = z) \\ &= \frac{1}{N} \sum_{z=1}^{N} P(D_i = c \mid D_{i-1} = d_{i-1}^z) \end{aligned}$$

Consider the observed values d_i^z, $z = 1, ..., N$, as (not necessarily independent) realizations of D_i. Their empirical frequency e_i is

$$e_i(c) = \frac{1}{N} \sum_{z=1}^{N} \mathbf{I}[d_i^z = c]$$

where $\mathbf{I}[.]$ denotes the indicator function with values in $\{0, 1\}$. Pearson's χ^2 test [96] can be used to test for the fit of f_i to e_i:

$$X_i = \sum_{c=1}^{\lfloor \frac{N}{2} \rfloor} \frac{(e_i(c) - f_i(c))^2}{f_i(c)} \sim \chi^2_{\phi_i - 1}$$

X_i is a χ^2-distributed test statistic with $\phi_i - 1$ degrees of freedom where ϕ_i is the number distances c with $f_i(c)$ strictly positive. The final test statistic is:

$$\sum_{i}^{N-1} X_i \sim \chi^2_{\sum_i^{N-1}(\phi_i - 1)}$$

Alternatively one can compare the empirical and theoretical cumulative distribution functions $F_i(c)$ and $E_i(c)$ defined as follows:

$$F_i(c) = P(C_i \leq c) = \frac{1}{N} \sum_{z=1}^{N} P(D_i \leq c \mid D_{i-1} = d_{i-1}^z)$$

$$E_i(c) = \frac{1}{N}\sum_{z=1}^{N} \mathbf{I}[d_i^z \leq c]$$

E_i can be compared to F_i by an Anderson-Darling [97] or a Cramér-von Mises test, which proved inferior to Pearson's χ^2 test (see Figure 3.5)

3.3.2 Test for location

I have the idea to compare the distribution of the ith neighbour distances observed in a sample with a suitable null distribution by means of their location. The most robust measures of location are mean or median, however in my studies of samples taken from joint distributions with low mutual information, I realized that many points do not show exceptional nearest neighbour distances. The difference to a sample drawn from independent X and Y distributions was made up by few points that had extreme nearest neighbour distances. This lead me to use extreme values as a test for location. The pvalue of a two-sided test based on $P(D_i^z \mid d_{i-1}^z)$ is $p_i^z = 2\min(v_i^z, 1 - v_i^z)$, with $v_i^z = P(D_i^z \leq d_i^z \mid d_{i-1}^z)$. I summarize, for all ith neighbours, the 2-sided pvalues by their minimum

$$V_i = \min(p_i^z;\ z = 1, ..., N)$$

Our test statistic V is obtained by aggregating the V_i values:

$$V = -2\sum_{i=1}^{N-1} \ln V_i$$

.

3.4 Construction of a benchmark set

Benchmarking was done on distributions (X, Y) given by $X \sim U[0,1]$, and $Y \sim f(X) + \mathcal{N}(0, \sigma^2)$. Here, $U[0,1]$ denotes a uniform distribution on the interval $[0,1]$, and $\mathcal{N}(0, \sigma^2)$ denotes a Gaussian distribution with mean 0 and variance σ^2. The function f was chosen as one of the following: linear, quadratic, cubic, sine with period 4π, circular, $f(x) = x^{1/4}$ and a step function (see Figure 3.6). This choice was inspired by a comment by Simon & Tibshirani (`http://statweb.stanford.edu/~tibs/reshef/script.R`, [98]) to the publication of the method MIC by Reshef et al. [92]. The noise parameter σ^2 determines the degree of dependence between X and Y, i.e., the mutual information $MI(X, Y;\ f, \sigma^2)$. The latter was estimated using an approximation $q_{XY}(X, Y)$

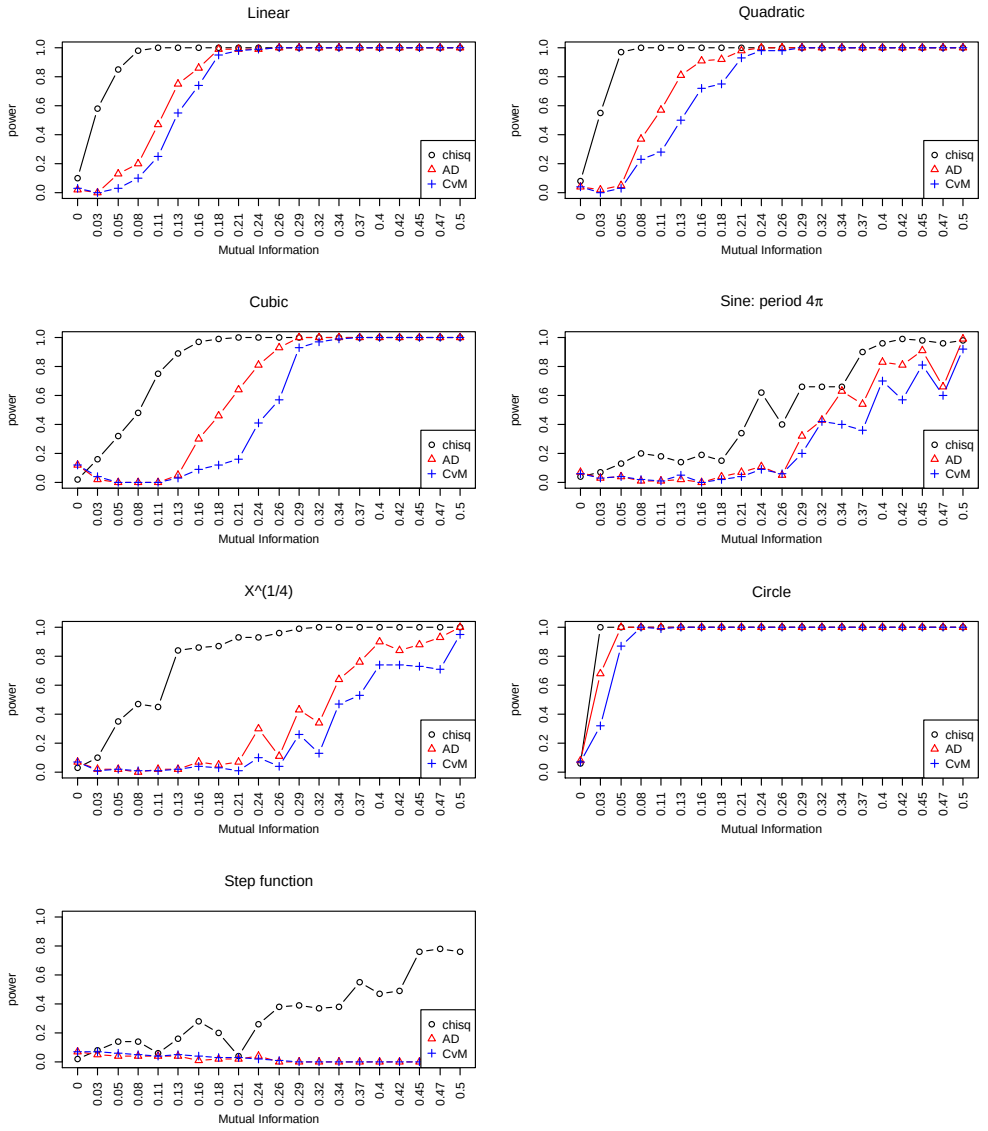


Figure 3.5: Benchmark comparing Pearson's χ^2 test as novel distributional test based on the theoretical and empirical probability functions (black curve labeled chisq) against using an Anderson-Darling (red curve, labeled AD) or a Cramér-von Mises test (blue curve, labeled CvM) on the cumulative distribution functions.

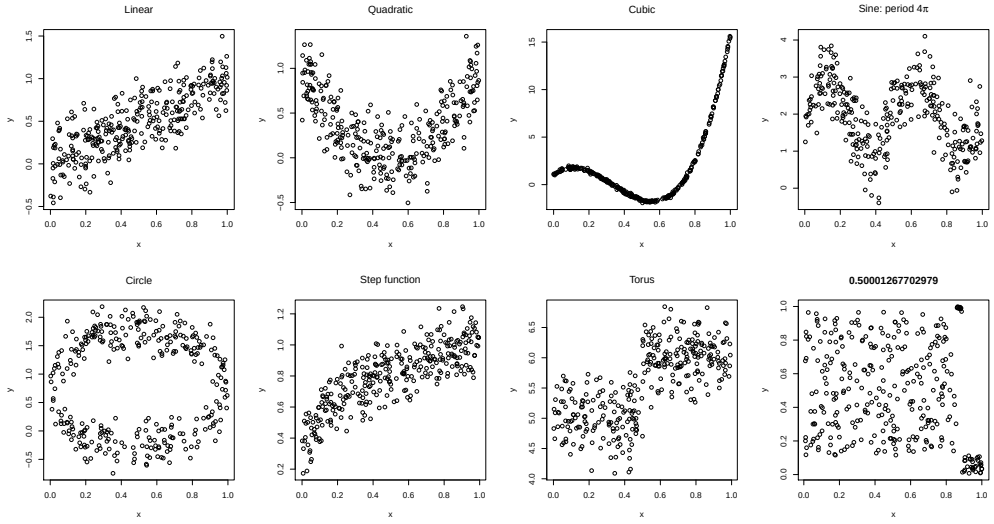


Figure 3.6: All considered functional dependencies at MI= 0.5. Top row from left to right: linear, quadratic, cubic, sine with period 4π, bottom row: $x^{1/4}$, circle, step function and the dependence called "patchwork copula".

to the density $p(X, Y)$ for which the mutual information can easily be calculated. Make q_{XY} a piecewise-constant density on a sufficiently fine quadratic grid $\{(\epsilon x, \epsilon y,) \mid x, y \in \mathbb{Z}\}$ with $q_{XY}(x, y) = p(\epsilon \left\lfloor \frac{x}{\epsilon} + 0.5 \right\rfloor, \epsilon \left\lfloor \frac{y}{\epsilon} + 0.5 \right\rfloor)$. In this case, $\epsilon = 0.01$ yielded sufficient precision. It is elementary to calculate the mutual information of q by

$$MI = \epsilon^2 \cdot \sum_{x,y \in \mathbb{Z}} q_{XY}(\epsilon x, \epsilon y) \cdot \log \frac{q_{XY}(\epsilon x, \epsilon y)}{q_X(\epsilon x) q_Y(\epsilon y)}$$

Here, q_X and q_Y denote the marginal densities with respect to x and y.

To make the results comparable for different f, I fixed an MI value M and chose $\sigma^2_{f,M}$ such that $MI(X, Y;\ f, \sigma^2_{f,M}) = M$. This was done for 20 MI values, M ranging from 0.01 to 0.5. The noise levels $\sigma^2_{f,M}$ are listed in Table 3.2. Samples from all dependencies f with $M = 0.5$ is shown in Figure 3.6.

So far performance evaluation of measure of dependence was only done on functional dependencies. Here I introduce "patchwork copulas" as a new non-functional dependence of x and y. Fix a grid size B, say $B = 10$. My density q will be a piece-wise constant function defined on a rectangular 2D grid on the unit square (with uneven grid line spacing) such that its marginal distributions are uniform (i.e., I will define a copula). The parameters of my distribution

MI	lin	para	quadratic	sin1	sin2	circ	x14	step
0.001	6.39	6.59	15.88	15.64	15.64	6.39	3.60	11.05
0.027	1.21	1.25	2.71	2.97	2.97	1.21	0.69	2.10
0.054	0.86	0.88	1.79	2.10	2.10	0.86	0.48	1.48
0.08	0.69	0.71	1.37	1.70	1.70	0.69	0.39	1.20
0.106	0.59	0.61	1.11	1.45	1.45	0.59	0.33	1.02
0.132	0.52	0.54	0.93	1.28	1.28	0.52	0.29	0.91
0.159	0.47	0.49	0.79	1.15	1.16	0.47	0.26	0.81
0.185	0.43	0.44	0.68	1.05	1.05	0.43	0.24	0.74
0.211	0.40	0.41	0.59	0.97	0.97	0.40	0.22	0.69
0.237	0.37	0.38	0.52	0.90	0.90	0.37	0.21	0.64
0.264	0.35	0.35	0.45	0.84	0.84	0.35	0.19	0.59
0.29	0.32	0.33	0.39	0.79	0.79	0.32	0.18	0.56
0.316	0.31	0.31	0.34	0.75	0.75	0.31	0.17	0.52
0.342	0.29	0.30	0.30	0.71	0.71	0.29	0.16	0.49
0.369	0.27	0.28	0.25	0.67	0.67	0.27	0.15	0.47
0.395	0.26	0.26	0.22	0.64	0.64	0.26	0.14	0.44
0.421	0.25	0.25	0.18	0.61	0.60	0.25	0.14	0.42
0.447	0.24	0.24	0.15	0.58	0.58	0.24	0.13	0.40
0.474	0.23	0.23	0.12	0.55	0.55	0.23	0.12	0.37
0.5	0.22	0.22	0.09	0.53	0.53	0.22	0.12	0.35

Table 3.2: All noise levels generated from the target mutual information values between 0.001 and 0.5 shown in the first column. Each column starting from the second represents a functional dependency in the order: linear, quadratic, cubic, sine with period 4π, sine with period 16π, circular, $x^{1/4}$ and step function and gives the 20 different noise levels used in the benchmark for each case.

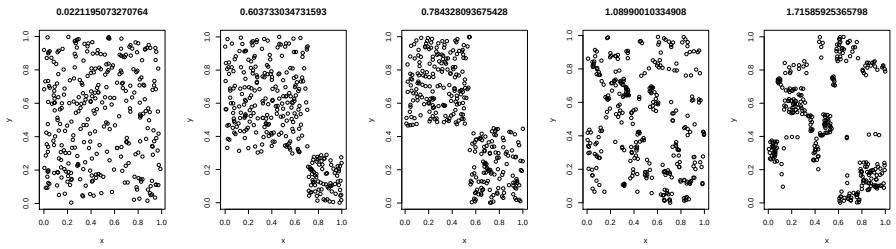


Figure 3.7: Scatterplots of the patchwork copula dependence for 320 points and 20×20 grid (parameters for Beta distribution: $(0.01, 1)$) for different MI values This data is uniform in x and y but the joint distribution has a non-functional dependence. The mutual information is indicated above each plot.

are the values p_{ij}, $i, j = 1, ..., B$, with $\sum_{i,j=1}^{B} p_{ij} = 1$. Let $p_{i*} = \sum_{j=1}^{B} p_{ij}$ and $p_{*j} = \sum_{i=1}^{B} p_{ij}$. Let (I, J) be a random variable which selects the grid rectangle (i, j) with probability p_{ij}, i.e., $P((I, J) = (i, j)) = p_{ij}$, $i, j = 1, ..., B$. My distribution (X, Y) is then defined by $X \sim \sum_{i=1}^{I-1} p_{i*} + U_I$, $U_I \sim U[0, p_I]$, and $Y \sim \sum_{j=1}^{J-1} p_{*j} + V_J$, $V_J \sim U[0, p_J]$. The density in the grid rectangle (i, j) can be computed as $q_{ij} = \frac{p_{ij}}{p_{i*} p_{*j}}$. It is elementary to verify that the marginals of q are uniform and that the mutual information of (X, Y) is

$$MI(X, Y;\ (p_{ij})) = \sum_{i,j=1}^{B} p_{ij} \log \left(\frac{p_{ij}}{p_{i*} p_{*j}} \right)$$

To generate samples with a desired MI value, choose suitable values for α and β. Then draw i.i.d. samples $p_{ij} \sim \text{Beta}(\alpha, \beta)$, $i, j = 1, ..., B$, and rescale the p_{ij} by dividing them by their sum. This process is repeated with different α, β until $MI(X, Y;\ (p_{ij}))$ is close enough to the desired MI value.The resulting dependence resembles a patchwork quilt of dense and spread out point clouds (see Figure 3.7).

Typically the points are considered embedded in Euclidean spaces [91], however the distance function can easily be adapted to model the geometry of a torus. I benchmarked some methods on both geometries (Euclidean plane and torus) and found that all methods were sensitive to changes of geometry.

I made the benchmark framework publicly available under a GPL 3.0 license. It is implemented in R [95] and contains code for generating the dependence structures as well as plotting the results. It is provided as Resource `knnIndep_1.0.tar.gz` accompanying this thesis as well as on CRAN`http://cran.r-project.org/web/packages/knnIndep`.

3.5 Comparison of methods

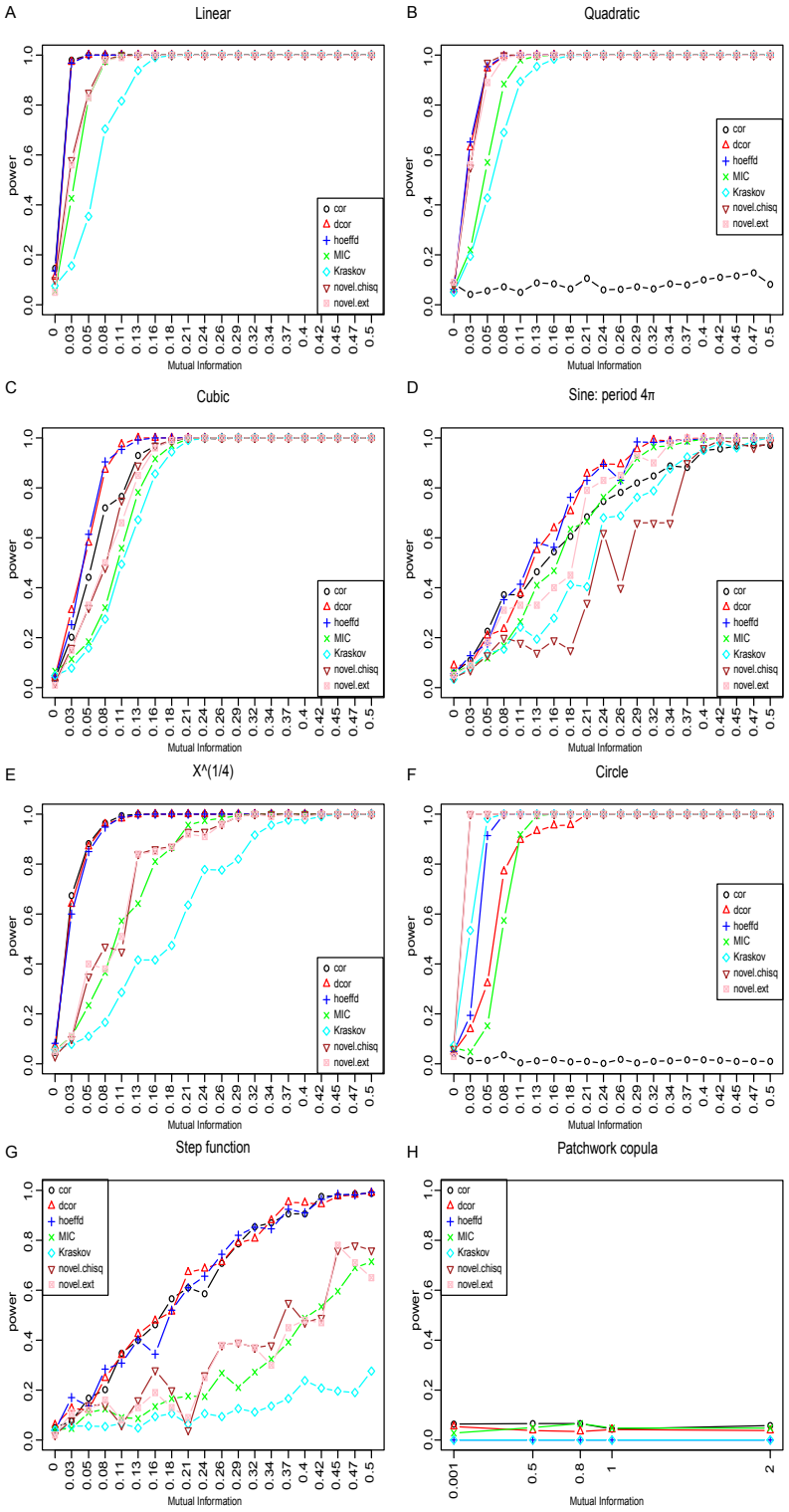
I compared both my tests (based on χ^2 and extreme paths, see section 3.3) to Pearson's product moment correlation coefficient, distance correlation (dcor, [99]) , Hoeffding's D [90], Kraskov's estimator for mutual information [91] and MIC [92]. For each type of dependence and each given value of MI, I generated a test set of 500 samples each consisting of 320 points from the respective dependence type. Test statistics were calculated for each sample. Additionally I generated a reference set of 500 samples with x and y values drawn independently which is used to calculate the cutoff value corresponding to a significance level of 5%. The power of each method was estimated as the fraction of samples that were called significant according to the cutoff. Results are shown in Figure 3.8. Additionally I generated receiver operating curves (ROC) for each type of dependence and MI value as Appendix A, *ROC curves for each method.*

The method of Hoeffding and dcor perform well throughout all types of dependence considered except for the circular dependence. My methods have a performance that places them after dcor and Hoeffding's method and before MIC. In the case of the circular dependence, my methods perform best, achieving maximum power at mutual information of 0.03. I suspect that is due to the fact that a circle geometrically resembles two crossing lines when projected onto a torus. To test this hypothesis we projected all types of dependence onto the torus and reran the whole benchmark (see Figure 3.9) . I observe that the cubic, sine and step functions are not detected by any method, even at the same MI.

The scaling of the plots in Figure 3.8 to the MI of the underlying joint distribution, enables the direct visual comparison of different dependence types. On the one hand this reveals that some types of dependence seem to be more difficult to detect for all methods (step function and sine curve). On the other hand each method performs best on different types of dependence.

3.6 Discussion

I have derived an exact formula for the distibution of the distances of the ith nearest neighbour of a given point. This distribution assumes rank transformed bivariate data from two independent variables. While this result is of independent interest, I used it to construct two non-parametric tests of independence for bivariate data. Similar to Kraskov's estimator, this test statistic is purely based on nearest neighbour distances. In contrast to Kraskov's estimator which requires an arbitrarily fixed i, I simultaneously take into account the whole sequence of ith nearest neighbours ($i = 1, 2, ...$). This improves on Kraskov's estimator, if used as a score for independence testing. My tests use rank trans-


A
Linear
B
Quadratic
C
Cubic
D
Sine: period 4π
E
X^(1/4)
F
Circle
G
Step function
H
Patchwork copula
power
Mutual Information
cor
dcor
hoeffd
MIC
Kraskov
novel.chisq
novel.ext

formed data, because this is a prerequisite for applying the exact nearest neighbour distributions derived in Section 3.2. The rank transformation is often used as a primary step to estimating mutual information, therefore I consider it an uncritical step in the procedure. My tests perform almost as well as the best competitors dcor and Hoeffding's D and they perform better than the recently proposed MIC statistic. I believe that the power of this method could be further improved in the Euclidean plane if my ith neighbour statistic would be adapted to account for boundary effects in the Euclidean plane. Although the methods try to account for the dependence of the variables D_i^z, $z = 1, ..., N$, I necessarily lose power because their exact dependence structure is not known. Alternatively I propose to take all distances d_i^z for a point z and apply a sequential testing approach for calling points that are located in dense regions. The number of these points could serve as a test statistic. The rationale is that under the null hypothesis of independence there should be fewer points z considered significant in the sequential test than for dependent samples.

Next I reviewed competing methods and presented a benchmark framework for performance testing on different types of dependence structures and topologies (Euclidean and toroidal). The benchmark framework and our novel tests for independence are publicly available as an R [95] package on CRAN (`http://cran.r-project.org/web/packages/knnIndep`). By scaling each type of dependence to a common set of mutual information values I allow comparison between all dependence types. Remarkably, when benchmarked on patchwork copulas, all methods fail. This is particularly intriguing for MIC as by design it should detect the grid structure of the data. In the case of the circular dependence, my methods perform best, while the method of Hoeffding and dcor perform well throughout all types of dependence considered. This in turn shows, that all tests I investigated are biased towards the detection of certain types of dependence structures.

Figure 3.8 *(facing page)*: Benchmark of all methods. cor denotes Pearson's product moment correlation coefficient, dcor distance covariance, hoeffd Hoeffding's D, MIC denotes MIC, novelTest.chisq is my test based on Pearson's χ^2 test and novelTest.ext is my test based on extreme paths. Each plot shows the power (on the y-axis) against the MI (x-axis). I examine 8 different types of dependence: linear, quadratic, cubic, sine with period 4π, $x^{1/4}$, circle, step function and the dependence called "patchwork copula" (**A-H**)

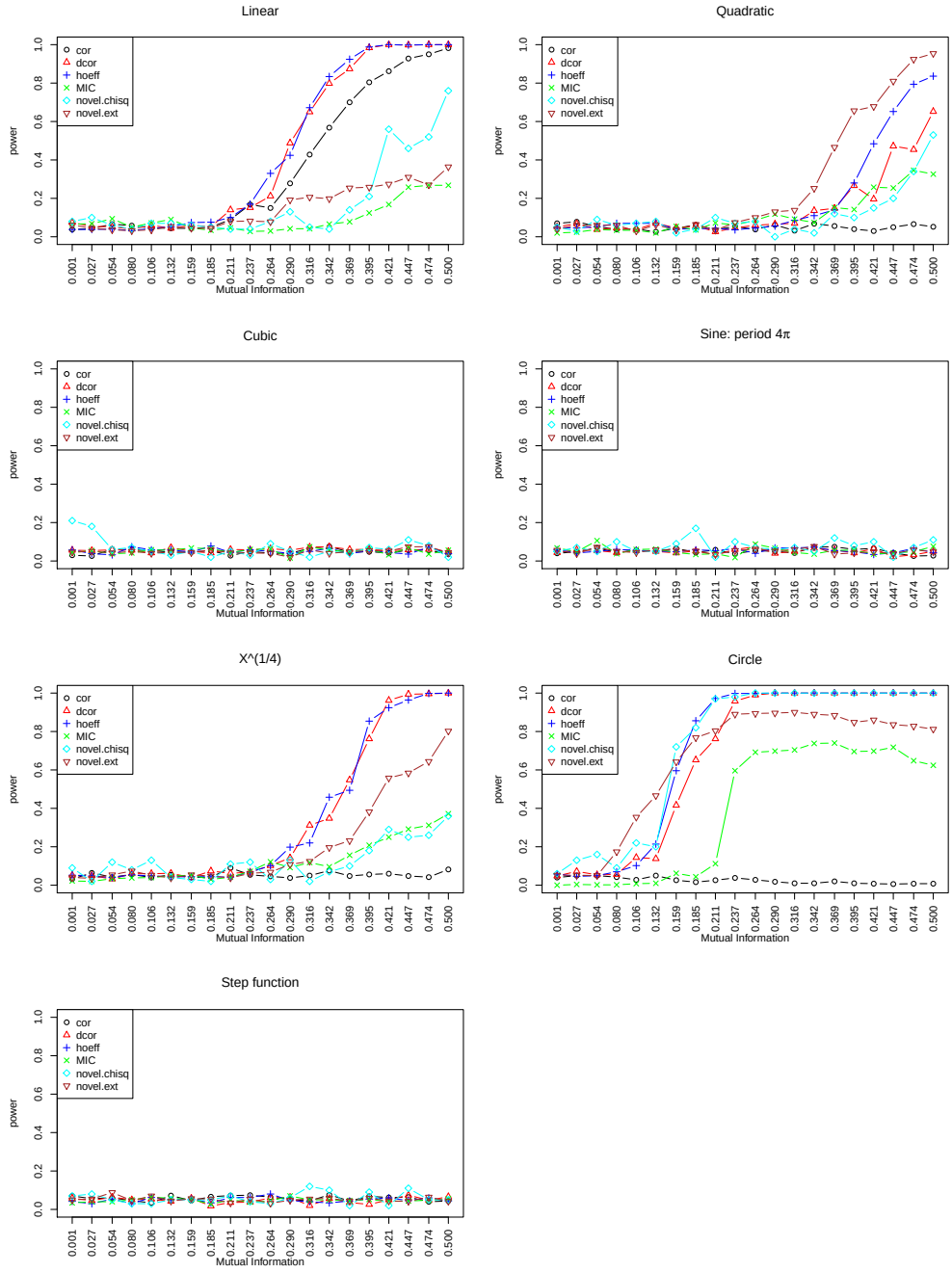

Linear
Quadratic
Cubic
Sine: period 4π
X^(1/4)
Circle
Step function
power
Mutual Information
cor
dcor
hoeff
MIC
novel.chisq
novel.ext
0.0
0.2
0.4
0.6
0.8
1.0
0.001
0.027
0.054
0.080
0.106
0.132
0.159
0.185
0.211
0.237
0.264
0.290
0.316
0.342
0.369
0.395
0.421
0.447
0.474
0.500

Figure 3.9 *(facing page)*: Benchmark of all methods on all functional dependence structures projected onto the torus. Cubic, sine and step function dependence are harder to detect on the torus even though the projection does not change to mutual information of the dependence. For the novel test only 100 samples were used to measure power, which explains the higher variance of these curves. My novel test for location beats all other method on data with a quadratic functional dependence.

Conclusion

In this thesis I examined three different statistical methods for the inference of interaction networks. In the first chapter I have shown how to detect condition specific transcription factor interactions and gave the network of transcription factor interactions across 16 conditions from the yeast gene expression compendium of Gash *et al.* [32].

In the second chapter I have given a closed form solution for the likelihood of dynamic Boolean networks with unknown time delays. This allows for the inference of such networks via a search and score approach using Markov chain Monte Carlo methods. I applied this method to derive the interaction network between the main TFs involved in the murine embryonic stem cells differentiation pathway.

Instead of the search and score approach to network inference there exists constraint based inference methods. A well known constraint based inference algorithm for Bayesian networks is the PC algorithm [1]. Instead of enumerating all possible model from the model space it starts with the complete graph and removes edges between nodes based on the result of a test of independence between two variables (conditioned on a third). I devised such a test in the third chapter and also show the exact distribution of the ith nearest neighbour of any point in a sample. This test is also applicable outside of network inference and can be used as a replacement for Pearson's product moment correlation coefficient when the expected functional relationship between two variables is not linear.

As noted in the Discussion section ending each chapter I see some potential for improvement in all three methods. The method for transcription factor interactions will benefit from better experimental techniques for the detection of transcription factor binding sites or simply through new method of aggregating all new datasets generated by (mod)ENCODE projects. I also suspect that the study of organisms under non optimal conditions will increasingly become the center of investigations.

The inference of dynamic Boolean networks will greatly profit from advances in proteomics that will allow the direct measurement of protein activity. So far in my applications of the model I circumvented the problem of missing protein activity measurements by using e. g. that target genes of a transcription factor as proxy for the activity of that transcription factor. That would become unnec-

essary if one can directly measure the activity of protein and remove a source of errors form the model.

As I have seen in my benchmarks there exists very strong contestants in the field of independence testing. Yet all methods are biased, meaning they perform better in some types of dependencies even though all types of dependence were calibrated to the same mutual information values. I believe the next breakthrough will com from better estimators of entropy (and thus of mutual information) which will be adapted for independence testing.

A Additional Resources

Additional Data for Chapter 1: Transcription Factor combinatorics

TF-target graphs for different organisms and their characteristics

At the heart of the OHC method lies the availability of a TF-target graph of good quality. This data should assign each TF a set of regulated genes. Ideally we would have such a graph for each condition and cell type, as the assignment of TFs and target genes changes over these variables. However there are currently no resources to construct such a graph at that level of detail. Therefore it is helpful to have a graph that encompasses as much variation as possible. We thus prefer data that has more edges than necessary and prefer to devise robust methods to handle false positives for each condition.

There are several possibilities to obtain a TF-target graph. Either compile validated TF-gene binding event from literature, mostly issued from low-throughput experiment, or use ChIP-chip or ChIP-seq techniques to either derive binding motifs for the TFs or assigning the binding peaks to genes directly. The analysis of ChIP-* data is still difficult due to the high amount of noise and because the binding events cannot be attributed to a specific strand. These methods are often combined with phylogenetic data to look at the conservation of binding sites through different species. Conserved binding events should represent functional binding of the TF. Binding motifs can be transformed into a TF-target graph by scanning gene promoters for these motifs and assigning matching promoters as regulated genes to the TF.

I will present of few TF-target graphs for S. cerevisiaethat I have used throughout my work. For each graph I present the sources and a few key statistics.

As S. cerevisiae is an extremely well studied organism, there is a wealth of resources available for TFs and their regulators. Table A.1 present TF-target resources I found most helpful.

Using these resources lead to the TF-target graphs detailed in the following.

JASPAR The JASPAR TF-target graph is determined by scanning the region 500 base pairs (bp) upstream of the open-reading frame for all motifs available for S. cerevisiae using PSCAN [102] with a score threshold of 0.9. This results

Type	Name (citation)	#TF
Experimental	Harbison *et al.* ([13])	203
	MacIsaac *et al.* ([14])	118
Motif based	JASPAR ([100])	177
	YeTFaSCo ([59])	256
	ScerTF ([101])	196
Literature	YEASTRACT ([36])	185

Table A.1: Useful resources for TF-target annotations and TF binding motifs. Indicated number of TFs are approximate values and subject to change (as of July 2012)

in a graph with 176 TFs having on average 1447 annotated target genes (see Figure A.1. This TF-target graph can be found as `jasparList.RData` in the accompanying Resources.

YEASTRACT TF-target relations mined from a manually curated literature repository can be found in the YEASTRACT database [36]. This resource has the advantage to contain annotations from many different experiments under different conditions and is thus the favored TF-target graph for S. cerevisiae. It is used, after filtering in Chapter 1. In its original form it contains 183 TFs each with, on average 263 genes (see Figure A.2). This TF-target graph can be found as `yeastract.RData` in the accompanying Resources.

ChIP-chip data from Harbison *et al.* The experimental data set from Harbison *et al.* is substantially smaller than the rest. TF binding sites have been inferred under different conditions and validated by conservation across *Saccharomyces* species. I took all TF-target relations with a p-value below 0.001. Keeping the same conservative criteria than the original study, [14] reanalyzed the Harbison data set and identified some more regulatory interactions. This leads to an annotation for 118 TF with an average 69 genes annotated (see Figure A.3). This TF-target graph can be found as `fraenkelList.RData` in the accompanying Resources.

ScerTF The entire database of optimal matrices selected for all S. cerevisiae-transcription factors is available for download from the ScerTF homepage `http://stormo.wustl.edu/ScerTF/` . Each matrix included in this collection was chosen based on its discriminative ability to correctly identify bound and unbound

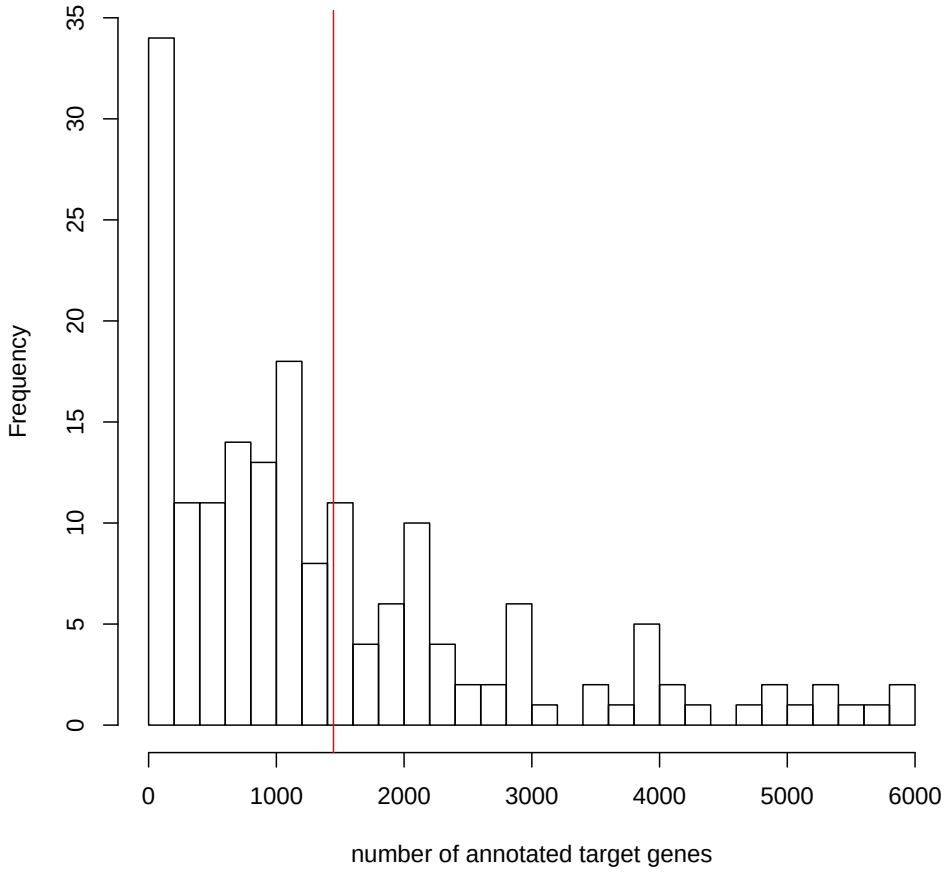


Figure A.1: Histogram of annotated target genes for all TF motifs found in the JASPAR database. Regulated genes where associated to each motif through a motif search using PSCAN on a promoter region defined as 500 bp upstream of the ORF. The red line shows the mean size.

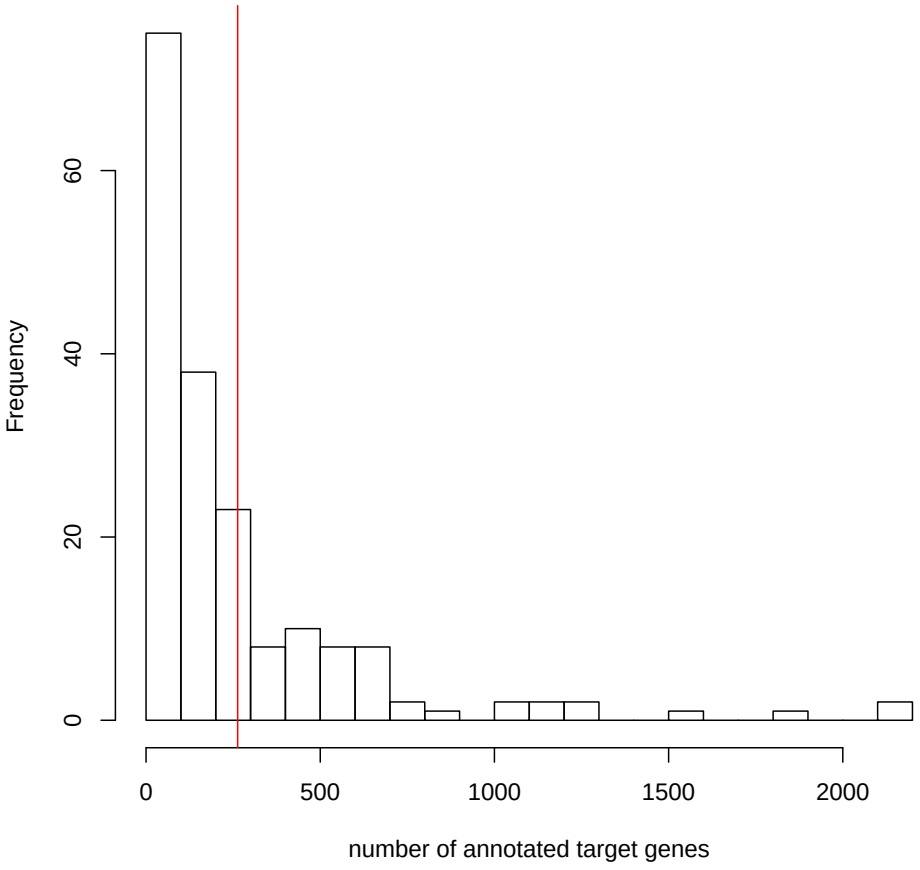


Figure A.2: Histogram of the number of annotated target genes to each TF of the YEASTRACT database. The red line shows the mean size.

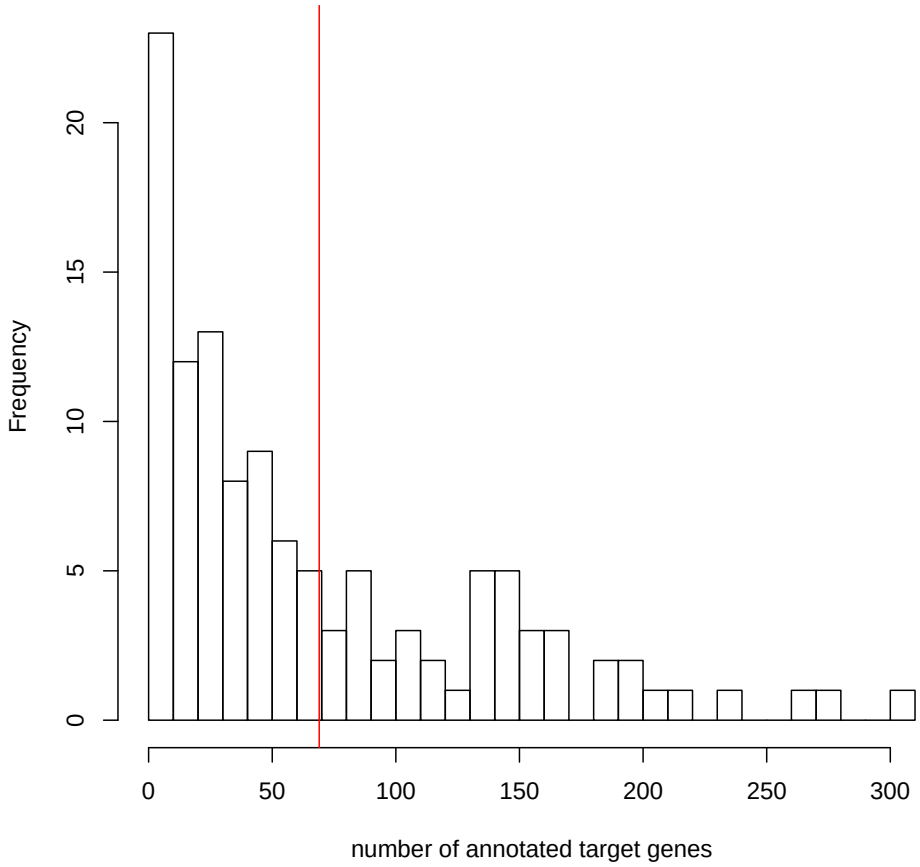


Figure A.3: Histogram of the number of annotated target genes to each TF at conservation threshold 0.001 in the work of MacIsaacs *et al.*. The red line shows the mean size.

probes in the Harbison ChIP-chip dataset as well as its information content and agreement with corroborating evidence in the literature. The recommended matrices were converted to the MEME format using the tool `jaspar2meme` from the MEME suite (version 4.8.1) on the count matrices provided in the `PCM` folder of the download from the ScerTF homepage. The conversion was done using the default arguments.

To find regulated target genes for each motif, I used the FIMO web server (version 4.8.1, [103]). to search for motif occurrences in the region 1000 base pairs upstream and 200 base pairs downstream of the start codon (database provided by the FIMO web interface as "Saccharomyces cerevisiae (upstream) (nucleotide only)"). The pvalue threshold was set at 10^{-4} and I used the motif file described above. Resulting matches below threshold were mapped to the nearest yeast ORF using data supplied by SGD [62] (accessed on July 26th 2012).

These are 169 matrices all unique, each regulating 775 genes on average. The minimum number of regulated genes is 198. The plot shown in Figure A.4 has bell curve shape which shows one problem of motif search: it returns many results that are probably false positive results.

This TF-target graph can be found as `scertf.RData` in the accompanying Resources.

YeTFaSco This dataset consists of 244 expert curated matrices. Duplicates (matrices for the same TF) were merged. The final 198 matrices given as IUPAC strings are converted using `iupac2meme` from the MEME suite. The target genes search was done as for ScerTF dataset (previous paragraph) using the FIMO webserver. In this annotation each TF has on average 735 target genes and all motifs have at least 203 genes (see Figure A.5). This TF-target graph can be found as `yetfasco.RData` in the accompanying Resources.

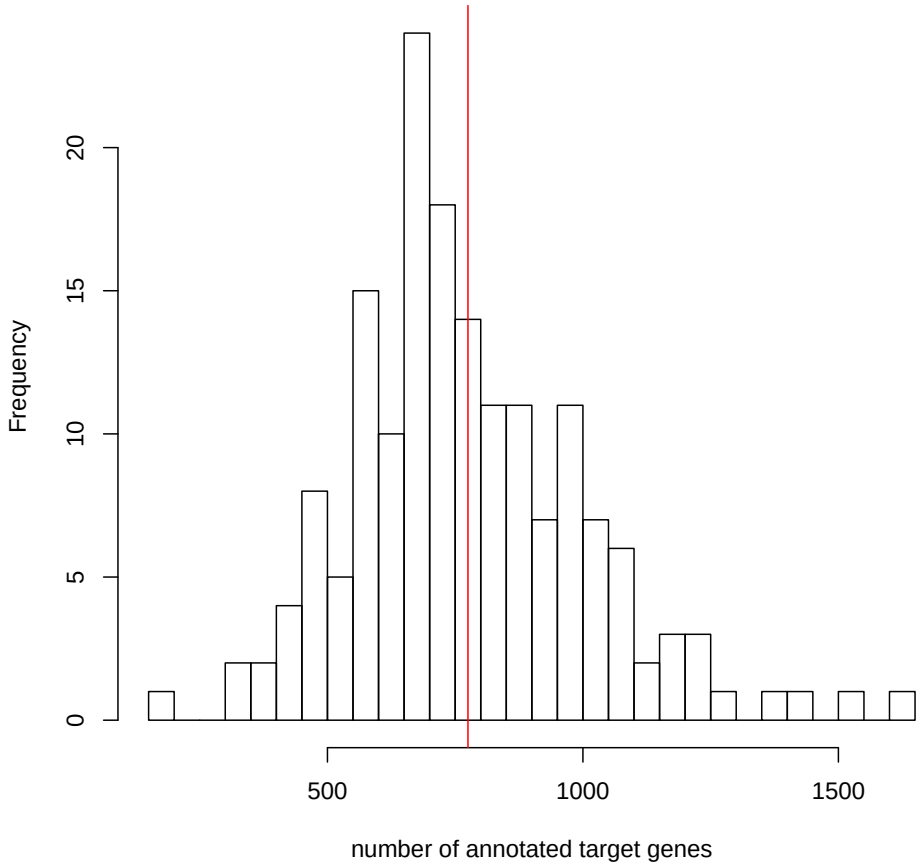


Figure A.4: Histogram of the number of annotated target genes found by FIMO for each TF motif from the ScerTF database of recommended motifs. The red line shows the mean size.

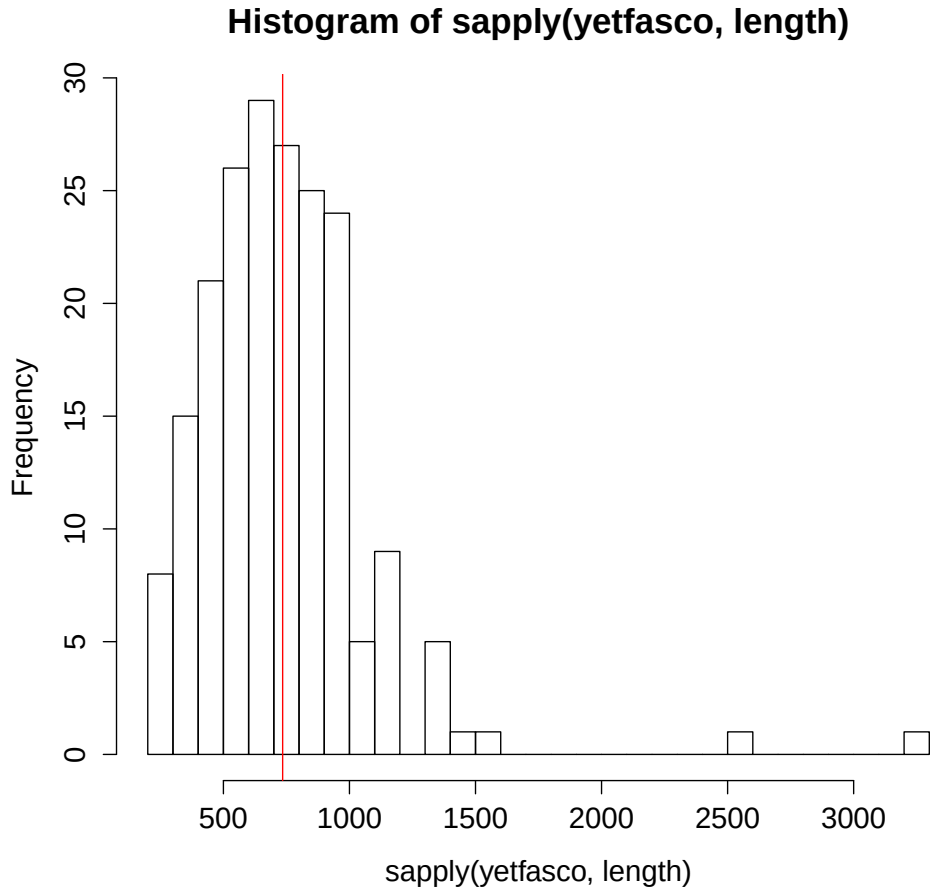


Figure A.5: Histogram of the number of annotated target genes found by FIMO for each TF motif from the YeTFaSco database of recommended motifs. The red line shows the mean size.

Additional Information for Chapter 2: Signal Network reconstruction

Calculations in Log space

When calculating the likelihoods/probabilities, I face the problem of dealing with very small numbers. Overflow/underflow errors are almost inevitable for larger networks. Therefore, I need to perform my calculations in log space, which I will define shortly. More specifically, I introduce a new algebra $(R, \oplus, \odot)$ which is isomorphic to $(\mathbb{R}, +, \cdot)$, and in which the quantities I actually deal with are essentially the logarithms of the corresponding (absolute) values in $\mathbb{R}$.

Definition. Let $R = (\mathbb{R} \times \{\pm 1\}) \cup \{-\infty\}$. Define the maps $\log^* : \mathbb{R} \to R$ and $\exp^* : R \to \mathbb{R}$ by

$$\log^*(a) = \begin{cases} (\log|a|\,,\, sign(a)) & \text{if } a \neq 0 \\ -\infty & \text{if } a = 0 \end{cases}$$

and

$$\exp^* A = \begin{cases} A_2 \exp(A_1) & \text{if } A = (A_1, A_2) \\ 0 & \text{if } A = -\infty \end{cases}$$

It is easy to see that $\log^*$ and $\exp^*$ are inverse to each other and hence bijections between $\mathbb{R}$ and R. I make $\log^*$ an isomorphism of algebras by defining an additive and a multiplicative structure on R via

$$\begin{aligned} A \oplus B &= \log^*(\exp^*(A) + \exp^*(B)) \quad \text{for } A, B \in R \\ A \odot B &= \log^*(\exp^*(A) \cdot \exp^*(B)) \quad \text{for } A, B \in R \end{aligned} \tag{A.1}$$

Note that this definition is equivalent to requiring

$$\begin{aligned} \log^*(a) \oplus \log^*(b) &= \log^*(a + b) \quad \text{for } a, b \in \mathbb{R} \\ \log^*(a) \odot \log^*(b) &= \log^*(a \cdot b) \quad \text{for } a, b \in \mathbb{R} \end{aligned}$$

If one would apply this definition in order to effectively calculate $A \oplus B$ or $A \odot B$, nothing would be gained, since we recur on the original addition/multiplication task. In the following, I will develop alternative ways to evaluate these expressions, avoiding overflow/underflow problems. First note that if A or B, or both, equal $-\infty$, then $A \odot B = -\infty$. If $A = -\infty$, then $A \oplus B = B$; analogously if $B = -\infty$, then $A \oplus B = A$. Excluding these trivial cases, let $A = (A_1, A_2) = (\log|a|, sign(a)) = \log^*(a)$, and $B = (B_1, B_2) = \log^*(b)$, for

some $a, b \in \mathbb{R} \setminus \{0\}$. Observe that

$$\begin{aligned} A \odot B &= \log^*(a \cdot b) = (\log|ab|, sign(ab)) = (\log|a| + \log|b|, sign(a)sign(b)) \\ &= (A_1 + B_1, A_2 \cdot B_2) \end{aligned}$$

Multiplication in R is therefore very easy. The addition in R requires more care. If $A_1 = B_1$, then

$$A \oplus B = \begin{cases} -\infty & \text{if } sign(A_2) \neq sign(B_2) \\ (A_1 + \log 2, A_2) & \text{if } sign(A_2) = sign(B_2) \end{cases}$$

Hence without loss, one may assume $A_1 > B_1$, i.e., $|a| > |b|$. It follows that $sign(a+b) = sign(a) = A_2$, and

$$|a+b| = A_2(a+b) = A_2a + A_2b = |a| + A_2B_2|b| = |a| \cdot (1 + A_2B_2|b|/|a|)$$

the last term, $1 + A_2B_2|b|/|a|$, is necessarily positive, thus

$$\begin{aligned} A \oplus B &= \log^*(a+b) \\ &= (\log|a+b|\,,\, sign(a+b)) \\ &= (\log|a| + \log(1 + A_2B_2|b|/|a|)\,,\, A_2) \\ &= (A_1 + \log(1 + A_2B_2\exp(\log|b| - \log|a|))\,,\, A_2) \\ &= (A_1 + \log(1 + A_2B_2\exp(B_1 - A_1))\,,\, A_2) \qquad \text{(A.2)} \end{aligned}$$

Since $A_1 > B_1$ implies $\exp(B_1 - A_1) < 1$, the calculation of (A.2) is numerically stable, except for $A_1 \approx B_1$ (equal up to many significant digits), $A_2 = -B_2$. This however is unavoidable, because the addition using standard floating point arithmetic in $\mathbb{R}$ is numerically unstable in this situation as well.

One is now able to rephrase Theorem 2.17) in the algebra $(R, \oplus, \odot)$, which leads to a numerically stable rule for the calculation of $B(k, \beta, \alpha; \kappa)$. Let $b(k, 0; \alpha) = \log^* B(k, 0; \alpha)$. The recursion in Theorem 3 becomes

$$b(k,0,\alpha;\kappa) = \begin{cases} b(0,0,(\alpha_1,...,\alpha_{k-1});\kappa) \oplus (-\alpha_k\tau_{i_k+1}, -1)\odot \\ \quad \odot\,(b(\kappa(k),\alpha_k,(\alpha_1,...,\alpha_{k-1});\kappa),\,1) & \text{if} \quad t_{\kappa(k)} \geq \tau_{i_k} \\ [(-\alpha_k\tau_{i_k}, 1) \oplus (-\alpha_k\tau_{i_k+1}, -1)] \odot \\ \quad \odot\,(b(\kappa(k),\alpha_k,(\alpha_1,...,\alpha_{k-1});\kappa),\,1) & \text{if} \quad t_{\kappa(k)} < \tau_{i_k} \end{cases}$$

$$b(k,\beta,\alpha;\kappa) = \log^*(\hat{c}(k,\beta,\alpha;\kappa)) \odot b(k,0,\hat{\alpha}(k,\beta,\alpha;\kappa);\kappa)$$

with $\hat{c}(k, \beta, \alpha; \kappa)$ and $\alpha(k, \hat{\beta}, \alpha; \kappa)$ defined as in Lemma 1.

List of symbols used in dynamic Boolean network learning

Symbol	Description
$\mathcal{G} = \{1, \ldots, N\}$	a set of N signalling components that dynamically interact with each other via transcriptional regulation
$\mathbb{F} = \{0, 1\}$	a Boolean field
$\Gamma \in \mathbb{F}^{\mathcal{G}\times\mathcal{G}}$	adjacency matrix describing a directed graph
$A_j(t)$	induction state variable of gene j (at timepoint t)
$B_j(t)$	induction state variable of signaling molecule j (at timepoint t), usually a protein
$\mathrm{pa}\,(j)$	set of parent nodes of component j
$\mathcal{F} = \{f_j \mid j \in \mathcal{G}\}$	family of Boolean functions
t	time
$\Delta = \{d_j \mid j \in \mathcal{G}\}$	time delays of all components
$\pi(\Delta; \Lambda)$	prior distribution over the delays
$\Lambda = (\lambda_j)$	tuple of appropriately chosen positive hyper-parameters
$\mathbf{B} = \{B_j(\tau_k); j \in \mathcal{G},\, k = 0, ..., K\}$	observations of the variables B_j at $K+1$ time points $0 = \tau_0 < \tau_1 < \tau_2 < ... < \tau_K$
$\mathcal{M}$	a specific, complete parametrisation of the model
T_j	change point (modeled as a random variable)
$\mathbf{B}_{max}$	state sequence with the maximum score
$\kappa(j)$	index of the last parent protein $B_{\kappa(j)}$ of A_j which needs to become active in order to activate A_j
$T_{\kappa(j)}$	smallest time point for which $f_j(\mathrm{pa}\,(A_j)(T_{\kappa(j)})) = 1$
α, β	parameters of $F(j, \beta, \alpha; \kappa)$, the recursive integral
$(R, \oplus, \odot)$	logspace algebra defining the equivalent of sum ($\oplus$) and product ($\odot$)

Details on MCMC

The Markov chain is initialized with a random graph Γ. In each step the algoritm proposes a graph Γ' drawn uniformly from the neighborhood $N(\Gamma)$ of the current graph. The neighborhood is defined as those graphs that differ by exactly one edge. Acceptance or rejection of Γ' is done according to the Metropolis-Hastings ratio $\alpha = \min(\frac{P(D|\Gamma',\mathcal{L},\mathcal{F},\Lambda)\cdot|N(\Gamma)|}{P(D|\Gamma,\mathcal{L},\mathcal{F},\Lambda)\cdot|N(\Gamma')|}, 1)$. I run the chain for 2000 steps discarding the first 100 as burn-in period.

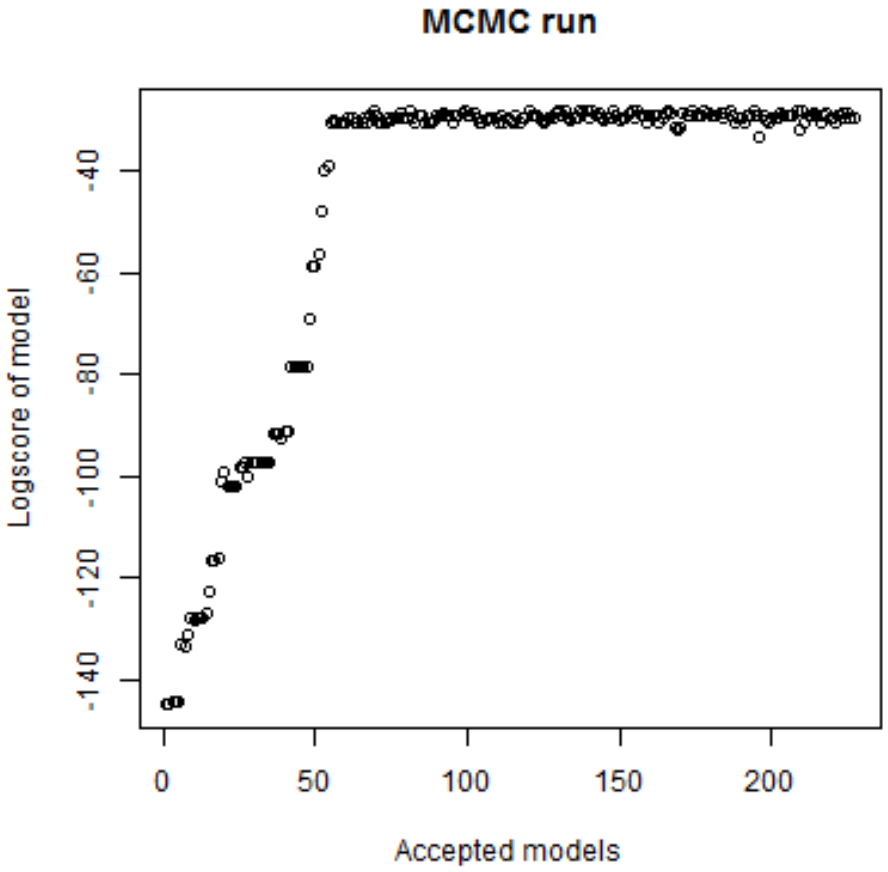


Figure A.6: Log-likelihood scores of accepted models throughout an exemplary MCMC run. On average 150 models are accepted in a run with 2000 MCMC steps. After $\sim$ 60 acceptance steps the likelihood reaches stationary level.

	1	2	3	1	2	3
1	0.6000	0.4000	0.0000	0.6000	0.4000	0.0000
2	0.0667	0.9333	0.0000	0.0667	0.9333	0.0000
3	0.0000	0.6000	0.4000	0.0000	0.6000	0.4000
4	0.0000	0.0667	0.9333	0.0000	0.0667	0.9333
5	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000
6	0.0000	0.0000	1.0000	0.0000	0.0000	1.0000

Table A.2: Exact probabilities (right) and $P(D_i = a)$ (left) for $N = 7$. The rows show the values of i and the columns the values of a.

	1	2	3	4	1	2	3	4
1	0.5238	0.4762	0.0000	0.0000	0.5238	0.4762	0.0000	0.0000
2	0.0476	0.8381	0.1143	0.0000	0.0476	0.8381	0.1143	0.0000
3	0.0000	0.3714	0.6286	0.0000	0.0000	0.3714	0.6286	0.0000
4	0.0000	0.0286	0.9714	0.0000	0.0000	0.0286	0.9714	0.0000
5	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	1.0000	0.0000
6	0.0000	0.0000	0.1429	0.8571	0.0000	0.0000	0.1429	0.8571
7	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	1.0000

Table A.3: Exact probabilities (right) and $P(D_i = a)$ (left) for $N = 8$. The rows show the values of i and the columns the values of a.

Additional Data for Chapter 3: Independence testing

Validation of the formula

To verify the correctness of the formulas, I calculated the exact distribution of $P(D_i = a)$ for $N = 7$ and $N = 8$ by generating all $N!$ possible point configurations and counting the relative frequency of each possible value for D_i. Then I compared the exact values to the values for $P(D_i = a)$ calculated from Formula 3.2 and found that they agree (Tables A.2 and A.3).

Additionally, I checked the validity of the formula for larger N ($N = 20$) by taking 10^6 random configurations and comparing the empirical frequency $h(d_i)$ with $P(d_i)$. The Mean relative difference between $h(d_i)$ and $P(d_i)$ is 0.0001979655. The absolute differences in each cell are shown in Table A.4.

ROC curves for each method

	a									
	1	2	3	4	5	6	7	8	9	10
1	3.33E-5	9.24E-5	6.81E-5	6.67E-5	9.16E-6	0.00	0.00	0.00	0.00	0.00
2	1.63E-5	4.47E-5	2.82E-5	5.07E-6	4.49E-6	4.49E-7	0.00	0.00	0.00	0.00
3	0.00	1.01E-5	8.73E-5	1.55E-4	6.44E-5	7.29E-6	0.00	0.00	0.00	0.00
4	0.00	4.45E-6	2.51E-6	2.37E-4	1.61E-4	6.87E-5	0.00	0.00	0.00	0.00
5	0.00	0.00	2.39E-5	3.59E-5	1.96E-4	2.56E-4	0.00	0.00	0.00	0.00
6	0.00	0.00	9.57E-7	1.82E-5	5.78E-5	7.17E-6	6.78E-5	0.00	0.00	0.00
7	0.00	0.00	0.00	1.01E-5	9.44E-6	1.22E-4	1.23E-4	0.00	0.00	0.00
8	0.00	0.00	0.00	6.19E-7	2.83E-5	9.83E-5	1.27E-4	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	3.12E-6	7.83E-5	8.14E-5	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	1.75E-7	5.47E-5	1.65E-4	2.20E-4	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	5.49E-6	1.11E-4	1.16E-4	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	1.65E-6	1.67E-5	1.51E-5	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	1.10E-5	1.10E-5	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	8.51E-7	1.20E-4	1.21E-4	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.45E-5	2.45E-5	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.24E-5	1.24E-5	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.22E-15	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.72E-6	3.72E-6
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table A.4: Differences between the theoretical values of $P(d_i)$ and the empirical frequencies $h(d_i)$ from 10^6 samples. I sampled data with $N = 20$, a is in the columns of the table and runs form 1 to $\lfloor \frac{N}{2} \rfloor$, i is in the rows in goes from 1 to $N - 1$.

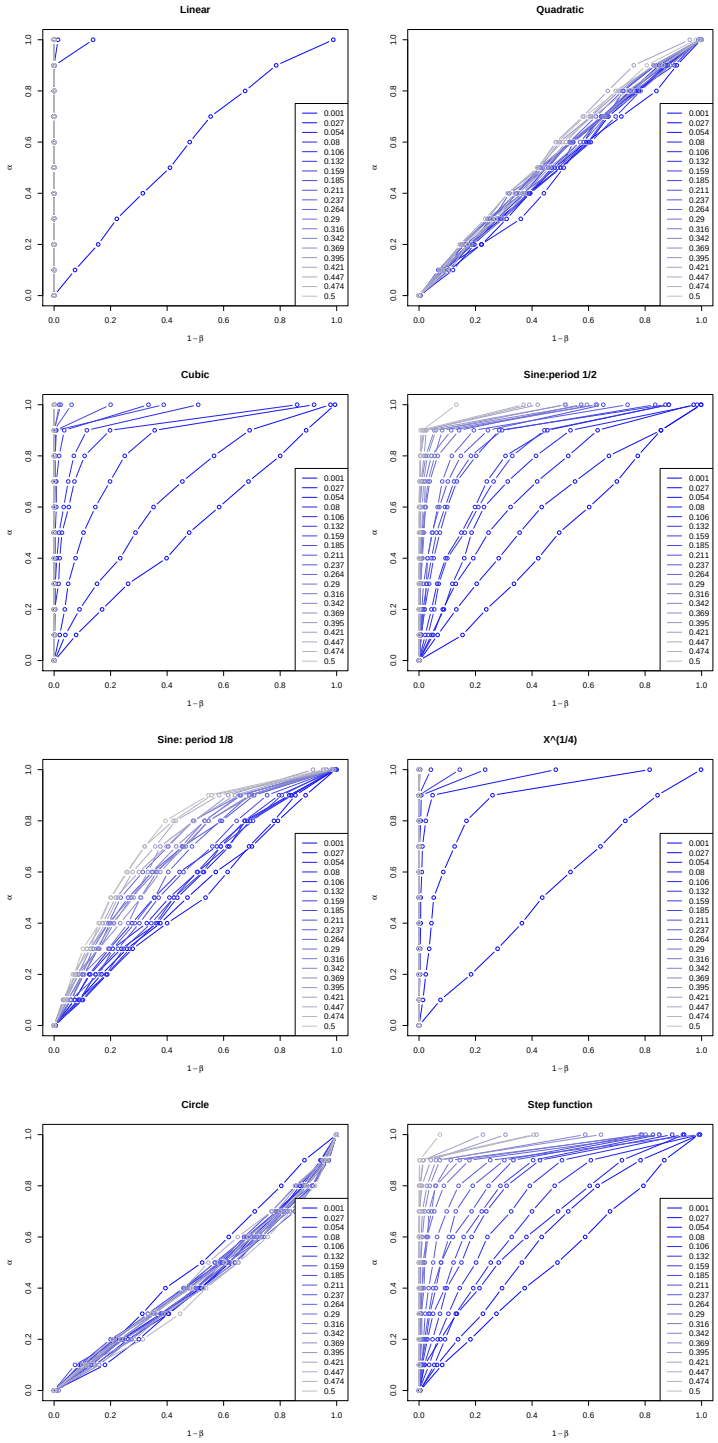


Figure A.7: Pearson's product moment correlation coefficient: ROC curves for all 20 noise levels per functional dependency.

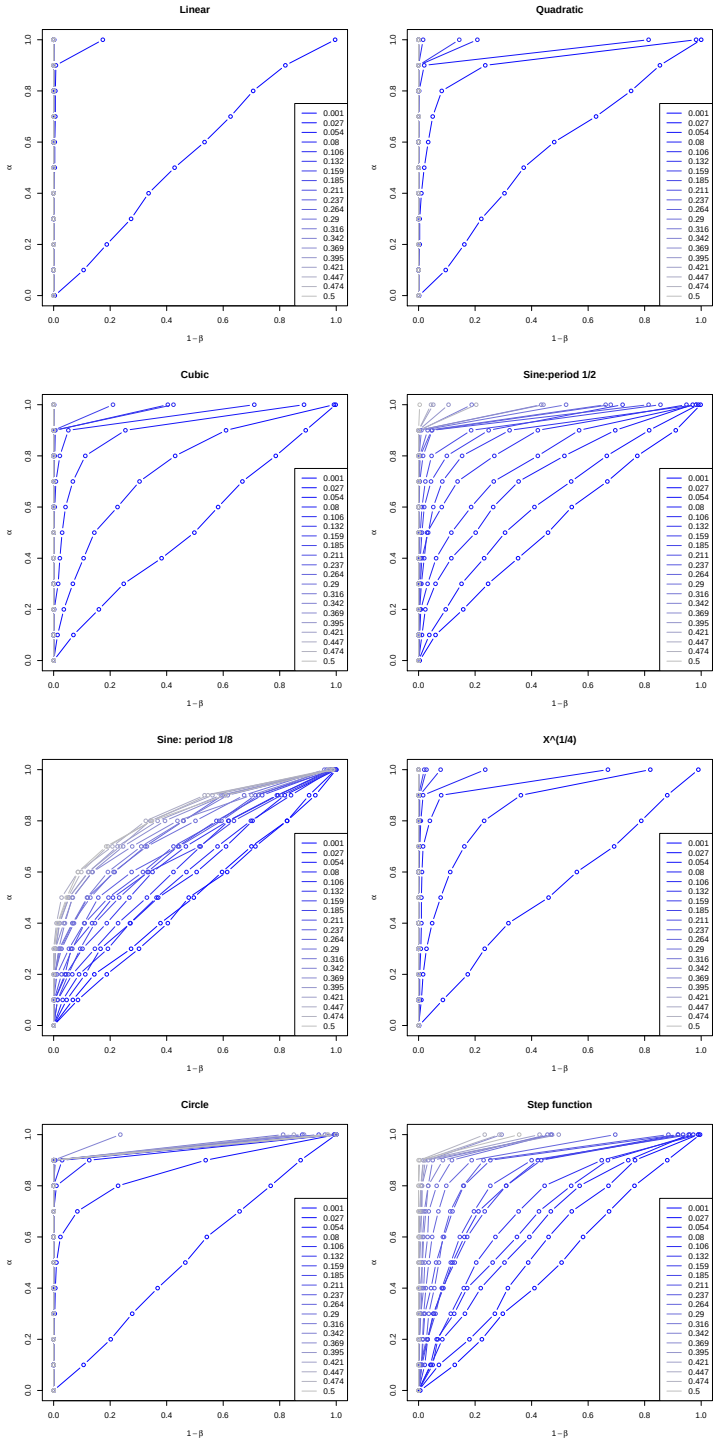


Figure A.8: dcor: ROC curves for all 20 noise levels per functional dependency.

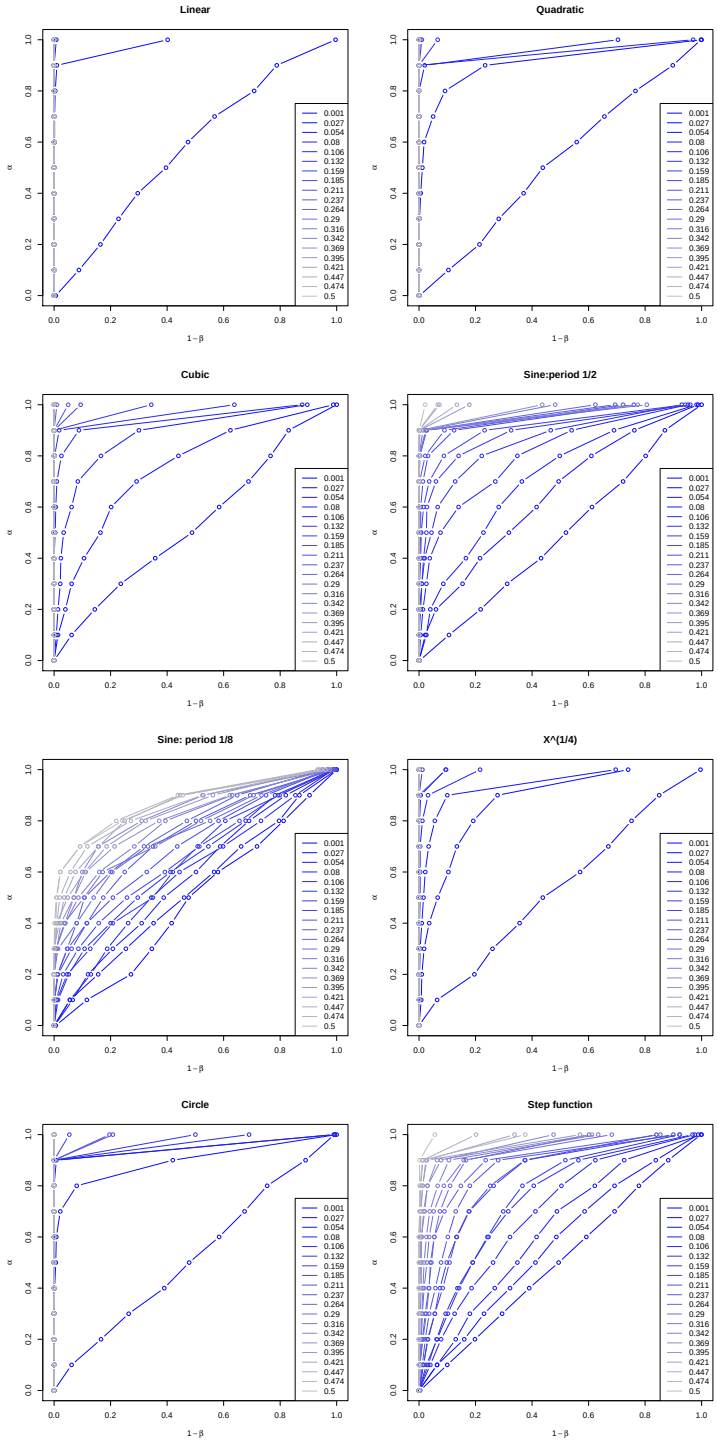


Figure A.9: Hoeffding's method: ROC curves for all 20 noise levels per functional dependency.

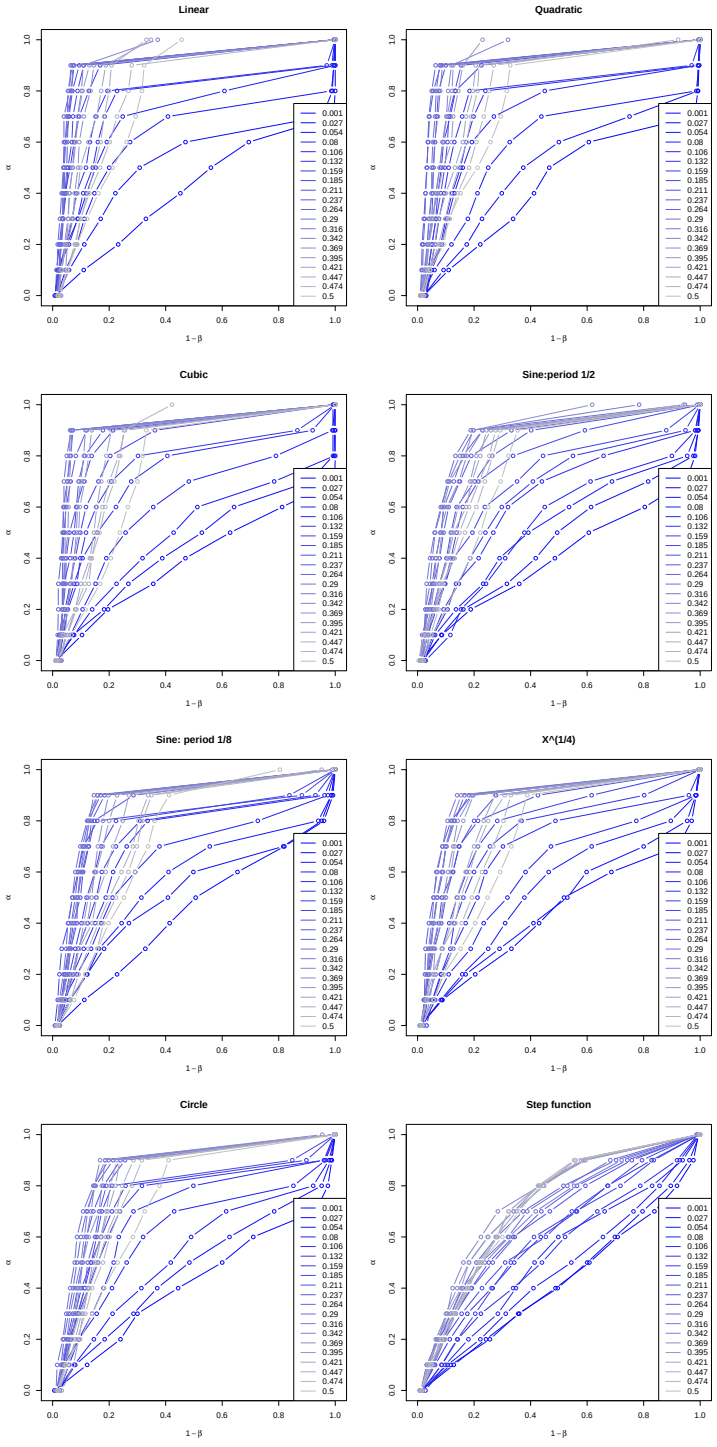


Figure A.10: MIC: ROC curves for all 20 noise levels per functional dependency.

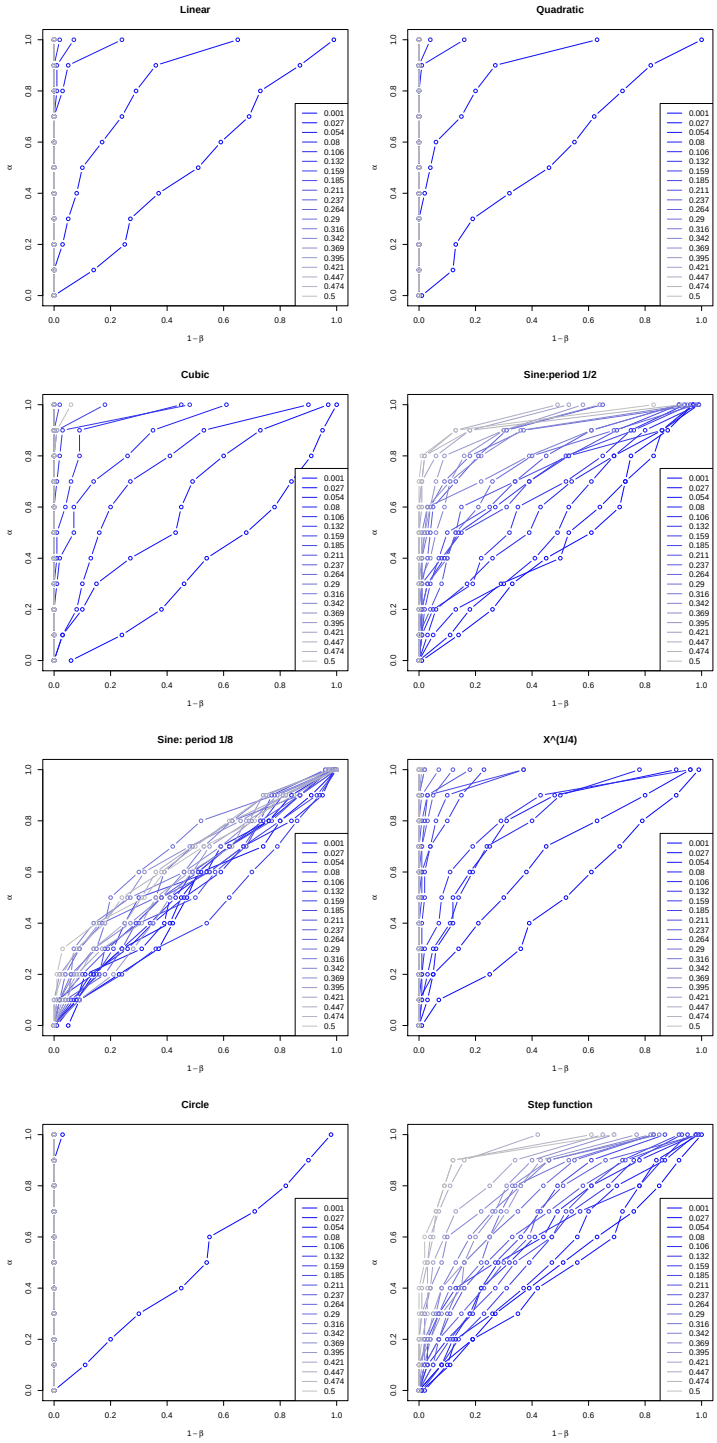


Figure A.11: Novel distributional test: ROC curves for all 20 noise levels per functional dependency.

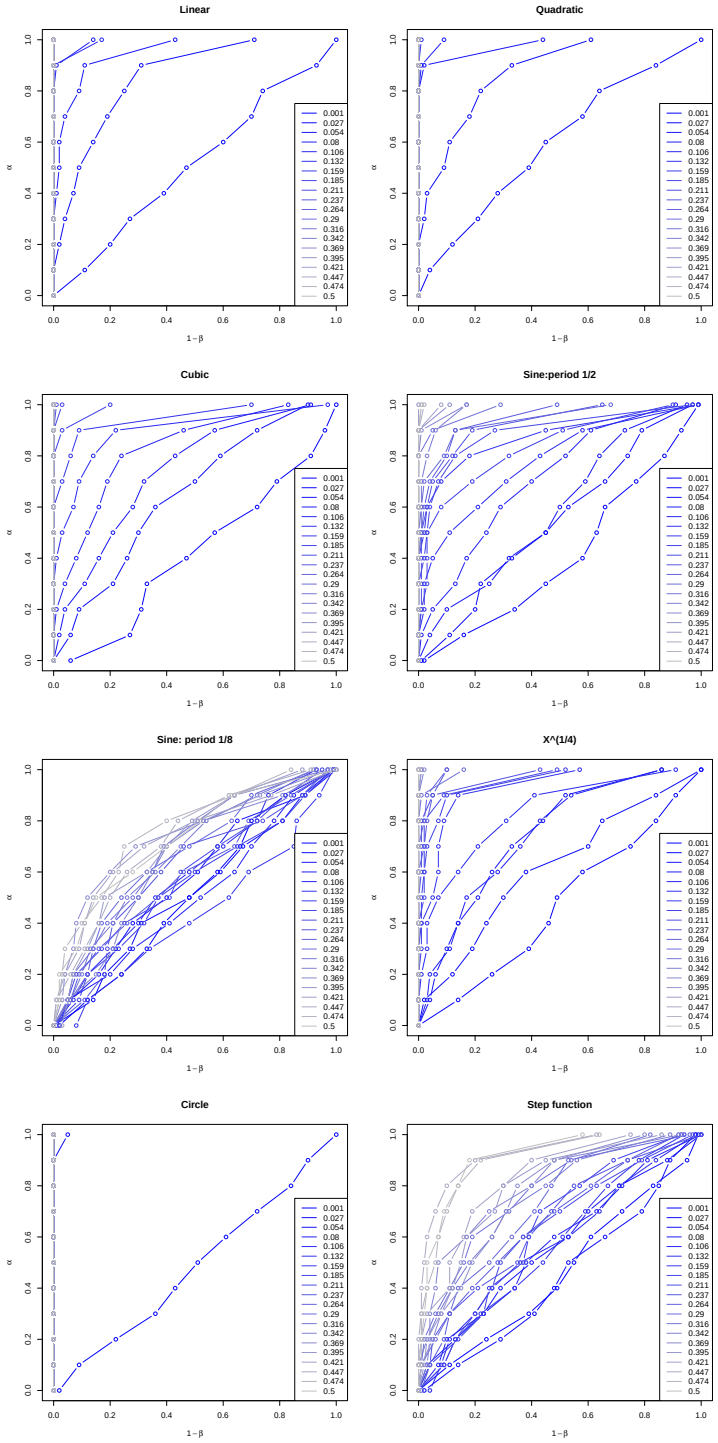


Figure A.12: Novel test for location: ROC curves for all 20 noise levels per functional dependency.

Bibliography

[1] Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*, volume 81. MIT press, 2000.

[2] Takashi Ito, Tomoko Chiba, Ritsuko Ozawa, Mikio Yoshida, Masahira Hattori, and Yoshiyuki Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences*, 98(8):4569–4574, 2001.

[3] Loic Giot, Joel S Bader, C Brouwer, Amitabha Chaudhuri, Bing Kuang, Y Li, YL Hao, CE Ooi, Brian Godwin, E Vitols, et al. A protein interaction map of drosophila melanogaster. *Science*, 302(5651):1727–1736, 2003.

[4] Ulrich Stelzl, Uwe Worm, Maciej Lalowski, Christian Haenig, Felix H Brembeck, Heike Goehler, Martin Stroedicke, Martina Zenkner, Anke Schoenherr, Susanne Koeppen, et al. A human protein-protein interaction network: a resource for annotating the proteome. *Cell*, 122(6):957–968, 2005.

[5] Jing Yu, V Anne Smith, Paul P Wang, Alexander J Hartemink, and Erich D Jarvis. Advances to bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, 20(18):3594–3603, 2004.

[6] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

[7] Ilya Shmulevich, Edward R Dougherty, Seungchan Kim, and Wei Zhang. Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, 2002.

[8] Stephen J. Tapscott. The circuitry of a master switch: Myod and the regulation of skeletal muscle gene transcription. *Development*, 132(12):2685–2695, June 2005.

[9] Tong I. Lee, Nicola J. Rinaldi, François Robert, Duncan T. Odom, Ziv Bar-Joseph, Georg K. Gerber, Nancy M. Hannett, Christopher T. Harbison,

Craig M. Thompson, Itamar Simon, Julia Zeitlinger, Ezra G. Jennings, Heather L. Murray, D. Benjamin Gordon, Bing Ren, John J. Wyrick, Jean-Bosco Tagne, Thomas L. Volkert, Ernest Fraenkel, David K. Gifford, and Richard A. Young. Transcriptional Regulatory Networks in Saccharomyces cerevisiae. *Science*, 298(5594):799–804, October 2002.

[10] Debopriya Das, Nilanjana Banerjee, and Michael Q. Zhang. Interacting models of cooperative gene regulation. *Proceedings of the National Academy of Sciences*, 101(46):16234–16239, November 2004.

[11] Robert P. Zinzen, Charles Girardot, Julien Gagneur, Martina Braun, and Eileen E. M. Furlong. Combinatorial binding predicts spatio-temporal cis-regulatory activity. *Nature*, 462(7269):65–70, November 2009.

[12] Markus J. Herrgård, Markus W. Covert, and Bernhard. Reconciling Gene Expression Data With Known Genome-Scale Regulatory Network Structures. *Genome Research*, 13(11):2423–2434, November 2003.

[13] C. T. Harbison, D. B. Gordon, T. I. Lee, N. J. Rinaldi, K. D. Macisaac, T. W. Danford, N. M. Hannett, J. B. Tagne, D. B. Reynolds, J. Yoo, E. G. Jennings, J. Zeitlinger, D. K. Pokholok, M. Kellis, P. A. Rolfe, K. T. Takusagawa, E. S. Lander, D. K. Gifford, E. Fraenkel, and R. A. Young. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431:99–104, Sep 2004.

[14] Kenzie MacIsaac, Ting Wang, D. Benjamin Gordon, David Gifford, Gary Stormo, and Ernest Fraenkel. An improved map of conserved regulatory sites for saccharomyces cerevisiae. *BMC Bioinformatics*, 7(1):113+, March 2006.

[15] Eunjee Lee and Harmen J. Bussemaker. Identifying the genetic determinants of transcription factor activity. *Molecular Systems Biology*, 6(1):412–412, September 2010.

[16] Sonali Mukherjee, Michael F. Berger, Ghil Jona, Xun S. Wang, Dale Muzzey, Michael Snyder, Richard A. Young, and Martha L. Bulyk. Rapid analysis of the DNA-binding specificities of transcription factors with DNA microarrays. *Nature genetics*, 36(12):1331–1339, December 2004.

[17] Andreas Beyer, Christopher Workman, Jens Hollunder, Dörte Radke, Ulrich Möller, Thomas Wilhelm, and Trey Ideker. Integrated assessment and prediction of transcription factor binding. *PLoS computational biology*, 2(6):e70+, June 2006.

[18] Jens Hollunder, Maik Friedel, Andreas Beyer, Christopher T. Workman, and Thomas Wilhelm. DASS: efficient discovery and p-value calculation of substructures in unordered data. *Bioinformatics*, 23(1):77–83, January 2007.

[19] Yitzhak Pilpel, Priya Sudarsanam, and George M. Church. Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature Genetics*, 29(2):153–159, September 2001.

[20] Nilanjana Banerjee and Michael Q. Zhang. Identifying cooperativity among transcription factors controlling the cell cycle in yeast. *Nucl. Acids Res.*, 31(23):7024–7031, December 2003.

[21] Ziv Bar-Joseph, Georg K. Gerber, Tong I. Lee, Nicola J. Rinaldi, Jane Y. Yoo, Francois Robert, D. Benjamin Gordon, Ernest Fraenkel, Tommi S. Jaakkola, Richard A. Young, and David K. Gifford. Computational discovery of gene modules and regulatory networks. *Nature Biotechnology*, 21(11):1337–1342, October 2003.

[22] Michael A. Beer and Saeed Tavazoie. Predicting gene expression from sequence. *Cell*, 117(2):185 – 198, 2004.

[23] Chiara Sabatti and Gareth M. James. Bayesian sparse hidden components analysis for transcription regulation networks. *Bioinformatics*, 22(6):739–746, March 2006.

[24] Yong Wang, Xiang-Sun Zhang, and Yu Xia. Predicting eukaryotic transcriptional cooperativity by bayesian network integration of genome-wide data. *Nucl. Acids Res.*, 37(18):5943–5958, October 2009.

[25] Amy H. Tong, Marie Evangelista, Ainslie B. Parsons, Hong Xu, Gary D. Bader, Nicholas Pagé, Mark Robinson, Sasan Raghibizadeh, Christopher W. V. Hogue, Howard Bussey, Brenda Andrews, Mike Tyers, and Charles Boone. Systematic Genetic Analysis with Ordered Arrays of Yeast Deletion Mutants. *Science*, 294(5550):2364–2368, December 2001.

[26] Maya Schuldiner, Sean R. Collins, Natalie J. Thompson, Vladimir Denic, Arunashree Bhamidipati, Thanuja Punna, Jan Ihmels, Brenda Andrews, Charles Boone, Jack F. Greenblatt, Jonathan S. Weissman, and Nevan J. Krogan. Exploration of the Function and Organization of the Yeast Early Secretory Pathway through an Epistatic Miniarray Profile. *Cell*, 123(3):507–519, November 2005.

[27] Michael Costanzo, Anastasia Baryshnikova, Jeremy Bellay, Yungil Kim, Eric D. Spear, Carolyn S. Sevier, Huiming Ding, Judice L. Koh, Kiana Toufighi, Sara Mostafavi, Jeany Prinz, Robert P. St Onge, Benjamin VanderSluis, Taras Makhnevych, Franco J. Vizeacoumar, Solmaz Alizadeh, Sondra Bahr, Renee L. Brost, Yiqun Chen, Murat Cokol, Raamesh Deshpande, Zhijian Li, Zhen-Yuan Y. Lin, Wendy Liang, Michaela Marback, Jadine Paw, Bryan-Joseph J. San Luis, Ermira Shuteriqi, Amy Hin Yan H. Tong, Nydia van Dyk, Iain M. Wallace, Joseph A. Whitney, Matthew T. Weirauch, Guoqing Zhong, Hongwei Zhu, Walid A. Houry, Michael Brudno, Sasan Ragibizadeh, Balázs Papp, Csaba Pál, Frederick P. Roth, Guri Giaever, Corey Nislow, Olga G. Troyanskaya, Howard Bussey, Gary D. Bader, Anne-Claude C. Gingras, Quaid D. Morris, Philip M. Kim, Chris A. Kaiser, Chad L. Myers, Brenda J. Andrews, and Charles Boone. The genetic landscape of a cell. *Science (New York, N.Y.)*, 327(5964):425–431, January 2010.

[28] Ramamurthy Mani, St, John L. Hartman, Guri Giaever, and Frederick P. Roth. Defining genetic interaction. *Proceedings of the National Academy of Sciences*, 105(9):3461–3466, March 2008.

[29] Xuewen Pan, Ping Ye, Daniel S. Yuan, Xiaoling Wang, Joel S. Bader, and Jef D. Boeke. A dna integrity network in the yeast saccharomyces cerevisiae. *Cell*, 124(5):1069–1081, March 2006.

[30] Sourav Bandyopadhyay, Monika Mehta, Dwight Kuo, Min-Kyung Sung, Ryan Chuang, Eric J. Jaehnig, Bernd Bodenmiller, Katherine Licon, Wilbert Copeland, Michael Shales, Dorothea Fiedler, Janusz Dutkowski, Aude Guénolé, Haico van Attikum, Kevan M. Shokat, Richard D. Kolodner, Won-Ki Huh, Ruedi Aebersold, Michael-Christopher Keogh, Nevan J. Krogan, and Trey Ideker. Rewiring of Genetic Networks in Response to DNA Damage. *Science*, 330(6009):1385–1389, December 2010.

[31] Christian Miller, Björn Schwalb, Kerstin Maier, Daniel Schulz, Sebastian Dümcke, Benedikt Zacher, Andreas Mayer, Jasmin Sydow, Lisa Marcinowski, Lars Dolken, Dietmar E. Martin, Achim Tresch, and Patrick Cramer. Dynamic transcriptome analysis measures rates of mRNA synthesis and decay in yeast. *Molecular Systems Biology*, 7:458–458, January 2011.

[32] Audrey P. Gasch, Paul T. Spellman, Camilla M. Kao, Orna Carmel-Harel, Michael B. Eisen, Gisela Storz, David Botstein, and Patrick O. Brown. Genomic Expression Programs in the Response of Yeast Cells to Environmen-

tal Changes. *Molecular Biology of the Cell*, 11(12):4241–4257, December 2000.

[33] The ENCODE Project Consortium. The ENCODE (ENCyclopedia Of DNA Elements) Project. *Science*, 306(5696):636–640, October 2004.

[34] Nicolas Negre, Christopher D. Brown, Lijia Ma, Christopher A. Bristow, Steven W. Miller, Ulrich Wagner, Pouya Kheradpour, Matthew L. Eaton, Paul Loriaux, Rachel Sealfon, Zirong Li, Haruhiko Ishii, Rebecca F. Spokony, Jia Chen, Lindsay Hwang, Chao Cheng, Richard P. Auburn, Melissa B. Davis, Marc Domanus, Parantu K. Shah, Carolyn A. Morrison, Jennifer Zieba, Sarah Suchy, Lionel Senderowicz, Alec Victorsen, Nicholas A. Bild, A. Jason Grundstad, David Hanley, David M. MacAlpine, Mattias Mannervik, Koen Venken, Hugo Bellen, Robert White, Mark Gerstein, Steven Russell, Robert L. Grossman, Bing Ren, James W. Posakony, Manolis Kellis, and Kevin P. White. A cis-regulatory map of the Drosophila genome. *Nature*, 471(7339):527–531, March 2011.

[35] Mark B. Gerstein, Zhi J. Lu, Eric L. Van Nostrand, Chao Cheng, Bradley I. Arshinoff, Tao Liu, Kevin Y. Yip, Rebecca Robilotto, Andreas Rechtsteiner, Kohta Ikegami, Pedro Alves, Aurelien Chateigner, Marc Perry, Mitzi Morris, Raymond K. Auerbach, Xin Feng, Jing Leng, Anne Vielle, Wei Niu, Kahn Rhrissorrakrai, Ashish Agarwal, Roger P. Alexander, Galt Barber, Cathleen M. Brdlik, Jennifer Brennan, Jeremy J. Brouillet, Adrian Carr, Ming-Sin Cheung, Hiram Clawson, Sergio Contrino, Luke O. Dannenberg, Abby F. Dernburg, Arshad Desai, Lindsay Dick, Andréa C. Dosé, Jiang Du, Thea Egelhofer, Sevinc Ercan, Ghia Euskirchen, Brent Ewing, Elise A. Feingold, Reto Gassmann, Peter J. Good, Phil Green, Francois Gullier, Michelle Gutwein, Mark S. Guyer, Lukas Habegger, Ting Han, Jorja G. Henikoff, Stefan R. Henz, Angie Hinrichs, Heather Holster, Tony Hyman, A. Leo Iniguez, Judith Janette, Morten Jensen, Masaomi Kato, W. James Kent, Ellen Kephart, Vishal Khivansara, Ekta Khurana, John K. Kim, Paulina Kolasinska-Zwierz, Eric C. Lai, Isabel Latorre, Amber Leahey, Suzanna Lewis, Paul Lloyd, Lucas Lochovsky, Rebecca F. Lowdon, Yaniv Lubling, Rachel Lyne, Michael MacCoss, Sebastian D. Mackowiak, Marco Mangone, Sheldon McKay, Desirea Mecenas, Gennifer Merrihew, David M. Miller, Andrew Muroyama, John I. Murray, Siew-Loon Ooi, Hoang Pham, Taryn Phippen, Elicia A. Preston, Nikolaus Rajewsky, Gunnar Rätsch, Heidi Rosenbaum, Joel Rozowsky, Kim Rutherford, Peter Ruzanov, Mihail Sarov, Rajkumar Sasidharan, Andrea Sboner, Paul Scheid, Eran Segal, Hyunjin Shin, Chong Shou, Frank J. Slack, Cindie Slightam, Richard Smith, William C. Spencer, E. O. Stinson,

Scott Taing, Teruaki Takasaki, Dionne Vafeados, Ksenia Voronina, Guilin Wang, Nicole L. Washington, Christina M. Whittle, Beijing Wu, Koon-Kiu Yan, Georg Zeller, Zheng Zha, Mei Zhong, Xingliang Zhou, modENCODE Consortium, Julie Ahringer, Susan Strome, Kristin C. Gunsalus, Gos Micklem, X. Shirley Liu, Valerie Reinke, Stuart K. Kim, LaDeana W. Hillier, Steven Henikoff, Fabio Piano, Michael Snyder, Lincoln Stein, Jason D. Lieb, and Robert H. Waterston. Integrative Analysis of the Caenorhabditis elegans Genome by the modENCODE Project. *Science*, 330(6012):1775–1787, December 2010.

[36] Miguel C. Teixeira, Pedro Monteiro, Pooja Jain, Sandra Tenreiro, Alexandra R. Fernandes, Nuno P. Mira, Marta Alenquer, Ana T. Freitas, Arlindo L. Oliveira, and Isabel Sa-Correia. The yeastract database: a tool for the analysis of transcription regulatory associations in saccharomyces cerevisiae. *Nucl. Acids Res.*, 34(suppl_1):D446–451, January 2006.

[37] Rand R. Wilcox. *Introduction to Robust Estimation and Hypothesis Testing, Second Edition (Statistical Modeling and Decision Science)*. Academic Press, 2 edition, January 2005.

[38] Amir Mitchell, Gal H. Romano, Bella Groisman, Avihu Yona, Erez Dekel, Martin Kupiec, Orna Dahan, and Yitzhak Pilpel. Adaptive prediction of environmental changes by microorganisms. *Nature*, 460(7252):220–224, June 2009.

[39] Ron Edgar, Michael Domrachev, and Alex E. Lash. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, January 2002.

[40] Zhijin Wu, Rafael A. Irizarry, Robert Gentleman, Francisco Martinez-Murillo, and Forrest Spencer. A model-based background adjustment for oligonucleotide expression arrays. *Journal of the American Statistical Association*, 99(468):909+, 2004.

[41] Robert Gentleman, Vincent Carey, Douglas Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, Laurent Gautier, Yongchao Ge, Jeff Gentry, Kurt Hornik, Torsten Hothorn, Wolfgang Huber, Stefano Iacus, Rafael Irizarry, Friedrich Leisch, Cheng Li, Martin Maechler, Anthony Rossini, Gunther Sawitzki, Colin Smith, Gordon Smyth, Luke Tierney, Jean Yang, and Jianhua Zhang. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):R80+, 2004.

[42] C. Stark, B. J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. BioGRID: a general repository for interaction datasets. *Nucleic Acids Res.*, 34:D535–539, January 2006.

[43] Itay Tirosh, Sharon Reikhav, Avraham A. Levy, and Naama Barkai. A Yeast Hybrid Provides Insight into the Evolution of Gene Expression Regulation. *Science*, 324(5927):659–662, May 2009.

[44] Leonid A. Mirny. Nucleosome-mediated cooperativity between transcription factors. *Proceedings of the National Academy of Sciences*, 107(52):22534–22539, December 2010.

[45] Warren P. Voth, Yaxin Yu, Shinya Takahata, Kelsi L. Kretschmann, Jason D. Lieb, Rebecca L. Parker, Brett Milash, and David J. Stillman. Forkhead proteins control the outcome of transcription factor binding by antiactivation. *The EMBO Journal*, 26(20):4324–4334, September 2007.

[46] Michael Pierce, Kirsten R. Benjamin, Sherwin P. Montano, Millie M. Georgiadis, Edward Winter, and Andrew K. Vershon. Sum1 and Ndt80 Proteins Compete for Binding to Middle Sporulation Element Sequences That Control Meiotic Gene Expression. *Mol. Cell. Biol.*, 23(14):4814–4825, July 2003.

[47] Brandon S. J. Davies and Jasper Rine. A Role for Sterol Levels in Oxygen Sensing in Saccharomyces cerevisiae. *Genetics*, 174(1):191–201, September 2006.

[48] Nicolas Bouquin, Anthony L. Johnson, Brian A. Morgan, and Leland H. Johnston. Association of the Cell Cycle Transcription Factor Mbp1 with the Skn7 Response Regulator in Budding Yeast. *Mol. Biol. Cell*, 10(10):3389–3400, October 1999.

[49] Najet Amar, Francine Messenguy, Mohamed El Bakkoury, and Evelyne Dubois. ArgRII, a Component of the ArgR-Mcm1 Complex Involved in the Control of Arginine Metabolism in Saccharomyces cerevisiae, Is the Sensor of Arginine. *Mol. Cell. Biol.*, 20(6):2087–2097, March 2000.

[50] D S McNabb, Y Xing, and L Guarente. Cloning of yeast hap5: a novel subunit of a heterotrimeric complex required for ccaat binding. *Genes & Development*, 9(1):47–58, 1995.

[51] Yasmine M. Mamnun, Rudy Pandjaitan, Yannick Mahé, Agnés Delahodde, and Karl Kuchler. The yeast zinc finger regulators Pdr1p and Pdr3p control pleiotropic drug resistance (PDR) as homo- and heterodimers in vivo. *Molecular Microbiology*, 46(5):1429–1440, 2002.

[52] Dipayan Rudra, Yu Zhao, and Jonathan R. Warner. Central role of Ifh1p-Fhl1p interaction in the synthesis of yeast ribosomal proteins. *The EMBO Journal*, 24(3):533–542, February 2005.

[53] M. T. Martínez-Pastor, G. Marchler, C. Schüller, A. Marchler-Bauer, H. Ruis, and F. Estruch. The Saccharomyces cerevisiae zinc finger proteins Msn2p and Msn4p are required for transcriptional induction through the stress response element (STRE). *The EMBO journal*, 15(9):2227–2235, May 1996.

[54] Shirong Zhang, Yitzchak Skalsky, and David J. Garfinkel. MGA2 or SPT23 Is Required for Transcription of the $\Delta 9$ Fatty Acid Desaturase Gene, OLE1, and Nuclear Membrane Integrity in Saccharomyces cerevisiae. *Genetics*, 151(2):473–483, February 1999.

[55] Kevin Wielemans, Cathy Jean, Stéphan Vissers, and Bruno André. Amino acid signaling in yeast: post-genome duplication divergence of the Stp1 and Stp2 transcription factors. *The Journal of biological chemistry*, 285(2):855–865, January 2010.

[56] B. Krems, C. Charizanis, and K. D. Entian. The response regulator-like protein Pos9/Skn7 of Saccharomyces cerevisiae is involved in oxidative stress resistance. *Current genetics*, 29(4):327–334, March 1996.

[57] Marc Larochelle, Simon Drouin, François Robert, and Bernard Turcotte. Oxidative stress-activated zinc cluster protein Stb5 has dual activator/repressor functions required for pentose phosphate pathway regulation and NADPH production. *Molecular and cellular biology*, 26(17):6690–6701, September 2006.

[58] Ronald E. Hector, Michael J. Bowman, Christopher D. Skory, and Michael A. Cotta. The Saccharomyces cerevisiae YMR315W gene encodes an NADP(H)-specific oxidoreductase regulated by the transcription factor Stb5p in response to NADPH limitation. *New biotechnology*, 26(3-4):171–180, October 2009.

[59] Carl G. de Boer and Timothy R. Hughes. YeTFaSCo: a database of evaluated yeast transcription factor sequence specificities. *Nucleic Acids Research*, 40(D1):D169–D179, January 2012.

[60] Timothy L. Bailey, Mikael Boden, Fabian A. Buske, Martin Frith, Charles E. Grant, Luca Clementi, Jingyuan Ren, Wilfred W. Li, and William S. Noble. MEME Suite: tools for motif discovery and searching. *Nucleic Acids Research*, 37(suppl 2):W202–W208, July 2009.

[61] Andrew P. Capaldi, Tommy Kaplan, Ying Liu, Naomi Habib, Aviv Regev, Nir Friedman, and Erin K. O'Shea. Structure and function of a transcriptional network activated by the mapk hog1. *Nature Genetics*, 40(11):1300–1306, October 2008.

[62] J. M. Cherry, C. Adler, C. Ball, S. A. Chervitz, S. S. Dwight, E. T. Hester, Y. Jia, G. Juvik, T. Roe, M. Schroeder, S. Weng, and D. Botstein. SGD: Saccharomyces Genome Database. *Nucleic Acids Res.*, 26:73–79, Jan 1998.

[63] Nicolas E. Buchler, Ulrich Gerland, and Terence Hwa. On schemes of combinatorial transcription logic. *Proceedings of the National Academy of Sciences*, 100(9):5136–5141, April 2003.

[64] Tali Raveh-Sadka, Michal Levo, and Eran Segal. Incorporating nucleosomes into thermodynamic models of transcription regulation. *Genome research*, 19(8):1480–1496, August 2009.

[65] Alex Gilman and Adam P. Arkin. GENETIC "CODE": Representations and Dynamical Models of Genetic Components and Networks. *Annual Review of Genomics and Human Genetics*, 3(1):341–369, 2002.

[66] Todd Wasson and Alexander J. Hartemink. An ensemble model of competitive multi-factor binding of the genome. *Genome Research*, 19(11):2101–2112, November 2009.

[67] Minoru Kanehisa and Susumu Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1):27–30, January 2000.

[68] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene Ontology: tool for the unification of biology. *Nat Genet*, 25(1):25–29, May 2000.

[69] Katia Basso, Adam A Margolin, Gustavo Stolovitzky, Ulf Klein, Riccardo Dalla-Favera, and Andrea Califano. Reverse engineering of regulatory networks in human b cells. *Nature genetics*, 37(4):382–390, 2005.

[70] Nancy Van Driessche, Janez Demsar, Ezgi O Booth, Paul Hill, Peter Juvan, Blaz Zupan, Adam Kuspa, and Gad Shaulsky. Epistasis analysis with global transcriptional phenotypes. *Nature genetics*, 37(5):471–477, 2005.

[71] Achim Tresch, Florian Markowetz, et al. Structure learning in nested effects models. *Statistical Applications in Genetics and Molecular Biology*, 7(1):9, 2008.

[72] Theresa Niederberger, Stefanie Etzold, Michael Lidschreiber, Kerstin C Maier, Dietmar E Martin, Holger Fröhlich, Patrick Cramer, and Achim Tresch. Mc eminem maps the interaction landscape of the mediator. *PLoS Computational Biology*, 8(6):e1002568, 2012.

[73] Holger Fröhlich, Tim Beißbarth, Achim Tresch, Dennis Kostka, Juby Jacob, Rainer Spang, and F Markowetz. Analyzing gene perturbation screens with nested effects models in r and bioconductor. *Bioinformatics*, 24(21):2549–2550, 2008.

[74] Marco Grzegorczyk, Dirk Husmeier, Kieron D Edwards, Peter Ghazal, and Andrew J Millar. Modelling non-stationary gene regulatory processes with a non-homogeneous bayesian network and the allocation sampler. *Bioinformatics*, 24(18):2071–2078, 2008.

[75] Adrian Silvescu and Vasant Honavar. Temporal Boolean Network Models of Genetic Networks and Their Inference from Gene Expression Time Series. *Complex Systems*, 13:54–70, 2001.

[76] T. E. Ideker, V. Thorsson, and R. M. Karp. Discovery of regulatory interactions through perturbation: inference and experimental design. *Pacific Symposium on Biocomputing*, pages 305–316, 2000.

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

[78] Jason A. Papin, Tony Hunter, Bernhard O. Palsson, and Shankar Subramaniam. Reconstruction of cellular signalling networks and analysis of their properties. *Nat Rev Mol Cell Biol*, 6(2):99–111, February 2005.

[79] Benedict Anchang, Mohammad J. Sadeh, Juby Jacob, Achim Tresch, Marcel O. Vlad, Peter J. Oefner, and Rainer Spang. Modeling the temporal interplay of molecular signaling and gene expression by using dynamic nested effects models. *Proceedings of the National Academy of Sciences*, 106(16):6447–6452, April 2009.

[80] Holger Fröhlich, Paurush Praveen, and Achim Tresch. Fast and efficient dynamic nested effects models. *Bioinformatics*, 27(2):238–244, 2011.

[81] Henrik Failmezger, Paurush Praveen, Achim Tresch, and Holger Fröhlich. Learning gene network structure from time laps cell imaging in rnai knock downs. *Bioinformatics*, 29(12):1534–1540, 2013.

[82] Niko Beerenwinkel and Seth Sullivant. Markov models for accumulating mutations. *Biometrika*, 96(3):645–661, 2009.

[83] Niko Beerenwinkel, Nicholas Eriksson, and Bernd Sturmfels. Conjunctive bayesian networks. *Bernoulli*, pages 893–909, 2007.

[84] Natalia Ivanova, Radu Dobrin, Rong Lu, Iulia Kotenko, John Levorse, Christina DeCoste, Xenia Schafer, Yi Lun, and Ihor R. Lemischka. Dissecting self-renewal in stem cells with RNA interference. *Nature*, 442(7102):533–538, August 2006.

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

[86] Qing Zhou, Hiram Chipperfield, Douglas A. Melton, and Wing H. Wong. A gene regulatory network in mouse embryonic stem cells. *Proceedings of the National Academy of Sciences*, 104(42):16438–16443, October 2007.

[87] Holger Fröhlich, Ozgür Sahin, Dorit Arlt, Christian Bender, and Tim Beissbarth. Deterministic Effects Propagation Networks for reconstructing protein signaling networks from multiple interventions. *BMC bioinformatics*, 10(1):322+, October 2009.

[88] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

[89] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[90] Wassily Hoeffding. A non-parametric test of independence. *The Annals of Mathematical Statistics*, 19(4):546–557, December 1948.

[91] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Phys. Rev. E*, 69:066138, June 2004.

[92] David N. Reshef, Yakir A. Reshef, Hilary K. Finucane, Sharon R. Grossman, Gilean McVean, Peter J. Turnbaugh, Eric S. Lander, Michael Mitzenmacher, and Pardis C. Sabeti. Detecting novel associations in large data sets. *Science*, 334(6062):1518–1524, 2011.

[93] Terry Speed. A correlation for the 21st century. *Science*, 334(6062):1502–1503, 2011.

[94] Philipp Eser, Carina Demel, Kerstin C Maier, Björn Schwalb, Nicole Pirkl, Dietmar E Martin, Patrick Cramer, and Achim Tresch. Periodic mrna synthesis and degradation co-operate during cell cycle gene expression. *Molecular Systems Biology*, 10(1), 2014.

[95] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[96] Karl Pearson. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine Series 5*, 50(302):157–175, 1900.

[97] T. W. Anderson and D. A. Darling. Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes. *The Annals of Mathematical Statistics*, 23(2):193–212, June 1952.

[98] Noah Simon and Robert Tibshirani. Comment on" detecting novel associations in large data sets" by reshef et al, science dec 16, 2011. *arXiv preprint arXiv:1401.7645*, 2014.

[99] Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, December 2007.

[100] Jan C. Bryne, Eivind Valen, Man-Hung E. Tang, Troels Marstrand, Ole Winther, Isabelle da Piedade, Anders Krogh, Boris Lenhard, and Albin Sandelin. Jaspar, the open access database of transcription factor-binding profiles: new content and tools in the 2008 update. *Nucl. Acids Res.*, 36(suppl_1):D102–106, January 2008.

[101] Aaron T. Spivak and Gary D. Stormo. ScerTF: a comprehensive database of benchmarked position weight matrices for Saccharomyces species. *Nucleic Acids Research*, 40(D1):D162–D168, January 2012.

[102] Federico Zambelli, Graziano Pesole, and Giulio Pavesi. Pscan: finding over-represented transcription factor binding site motifs in sequences from co-regulated or co-expressed genes. *Nucleic acids research*, 37(Web Server issue):W247–252, July 2009.

[103] C. E. Grant, T. L. Bailey, and W. S. Noble. FIMO: scanning for occurrences of a given motif. *Bioinformatics*, 27(7):1017–1018, April 2011.