Indirect inference of synergistic and alternative signalling of intracellular pathways

DISSERTATION ZUR ERLANGUNG DES DOKTORGRADES
DER NATURWISSENSCHAFTEN (DR. RER. NAT.)
DER FAKULTÄT FÜR BIOLOGIE UND VORKLINISCHE MEDIZIN
DER UNIVERSITÄT REGENSBURG

vorgelegt von

Martin Franz-Xaver Pirkl

aus

Neumarkt i.d. Opf

im Jahr 2016

Der Promotionsgesuch wurde eingereicht am:
07.06.2016

Die Arbeit wurde angeleitet von:
Prof. Rainer Spang

Unterschrift:

Martin Pirkl

Acknowledgements

I like to thank my supervisor Rainer Spang for making the transition from "pure" mathematics (technically applied mathematics, but with paper and pencil instead of a computers) to statistical Bioinformatics as smooth as possible. I would also like to thank my two mentors Michael Boutros from the German Cancer Research Center in Heidelberg and Elmar Lang from the Biophysics department for their cooperation and guidance in RNA interference respectively machine learning. Furthermore, I am very thankful to Michael Boutros' group for generating the vast amount of data we analyzed in chapter 7. The Microarray Data analysed in chapter 6 was produced by Dieter Kube's group in Götting and for that I am also very thankful. I want to thank the author of the original Nested Effects Models Florian Markowetz for the invitation to visit his group in Cambridge and Julio Saez-Rodriguez to take the time to introduce me to his group in Hinxton, also.

Last, but not least I thank the whole group (plus alumni) of the department of statistical Bioinformatics at the University of Regensburg for healthy discussions, work and non-work related, especially the Nested Effects Model and Bayesian Networks people. A special high five goes out to the people of the after lunch table football group.

Publications

Parts of chapters 4, 5, 6 and 7 in this Thesis are published or in preparation for publication.

Pirkl, Martin, Hand, Elisabeth, Kube, Dieter, & Spang, Rainer. 2016. Analyzing synergistic and non-synergistic interactions in signalling pathways using Boolean Nested Effect Models. *Bioinformatics*, **32**(6), 893900.

Kranz, Dominique, Pirkl, Martin, Leible, Svenja, Kerr, Grainne, Spang, Rainer, & Boutros, Michael. 2016. Regulatory networks of Tnfα and Trail induced gene regulation in hepatocellular carcinoma cell lines and primary hepatocytes. *in preparation.*

"I don't think that math is gonna bring you back from the dead."
– Frylock, *Aqua Teen Hunger Force*

Contents

1

Introduction

1.1 Motivation

Cells process input signals to output signals using a network of cellular signalling pathways (Berg *et al.* (2012)). For example, a small molecule binds a membrane receptor. The signal is brought into the cell via structural modification of the receptor. A set of kinases and other signalling molecules propagate the signal through the cytosol. This involves both activation and repression of proteins. Often complexes of multiple proteins must form before a signal propagates. Eventually, the signal enters the nucleus and transcription factors become activated. Finally, the combination of activated transcription factors and regulatory co-factors leads to the transcription of a large set of genes. Eventually, this can lead to changes of the cell phenotype. Some of the involved molecules are also part of other pathways linking multiple pathways together. Understanding the structure and the interplay of pathways is crucial both for understanding the cellular mechanism and for designing novel therapies that target specific pathways.

Inferring networks from molecular profiles is a well developed field in bioinformatics. Transcriptional data can be generated more easily compared to protein activation data. Consequently, many algorithms were developed that focus on the reconstruction of regulatory networks, for example Gaussian graphical models (GGM) (Schäfer & Strimmer (2005)), Bayesian networks (Friedman *et al.* (2000)) or the PC-algorithm (Kalisch & Bühlmann (2007)), the Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE, Margolin *et al.* (2006)). All these methods use observational gene expression data to construct regulatory networks based on different association scores between genes.

It is no problem to quantify the expression of any gene using standard methods like qPCR, microarrays, or RNAseq (Chang (1983); Schena *et al.* (1995); Lashkari *et al.* (1997); Mortazavi *et al.* (2008); Morin *et al.* (2008); Chu & Corey (2012); Logan *et al.* (2009)). Observing signalling networks is more complicated. Protein activation can operate on the levels of protein expression, cellular protein localization, and protein modifications like phosphorylation, ubiquitination etc. While there are assays to assess

activation on any of these levels, those assays are more elaborate, more expensive and less generic. Moreover, for every protein we need to know a priori which type of modification mediates signal transduction.

Biologists have been inferring pathways without formal computations for many years. In this classical approach functional/interventional data is used. Pathways are perturbed by knock-out, knock-down or knock-ins of genes. The consequences of the interventions are observed and interpreted. Markowetz *et al.* (2005) summarize this strategy by "What I cannot break, I cannot understand". Complementing the biology tradition, there are several computational approaches that exploit intervention data. Markowetz *et al.* (2005) introduced Nested Effects Models (NEM) (Markowetz *et al.* (2007); Froehlich *et al.* (2011); Niederberger *et al.* (2012)). This method was designed to infer non-transcriptional signalling pathways by transcriptional downstream effects of pathway perturbation. A pathway is activated in a set of cellular assays where specific pathway genes are silenced. The silencing blocks branches of the pathway. Genes that normally change expression in response to the stimulus no longer react in knock-down assays. Typically the effected gene sets differ from silenced gene to gene. NEMs infer the network structure from the nesting of these sets. In a nutshell: if the effected genes of perturbing gene B are a noisy subset of the effected genes of gene A, then A is upstream of B. This concept has been extended to time series data (Anchang *et al.* (2009); Froehlich *et al.* (2011); Dümcke *et al.* (2014)), evolving networks (Wang *et al.* (2014)) and network inference with hidden confounders (Sadeh *et al.* (2013)).

To date NEMs can infer the upstream/downstream relations of genes in a pathway (Markowetz *et al.* (2005)), they can distinguish activation from repression (Vaske *et al.* (2009)) and they can resolve the flow of information (Anchang *et al.* (2009); Froehlich *et al.* (2011)). However they cannot model the role of complex formation in signalling pathways. If a protein X is activated by a complex, all members of the complex must be present and in the correct activation state. The proteins in the complex operate concertedly and are linked to X by an *AND* gate. In another scenario, X can be activated independently by several proteins. In this case the proteins operate non-synergistically and an *OR* gate links them to X.

Boolean Networks (Kauffman (1969), Gershenson (2004)) can distinguish logical gates. They have been used to simulate signalling pathways (Klamt *et al.* (2006, 2007)) and to reconstruct them from interventional data (Saez-Rodriguez *et al.* (2009)). Its assumed, that the data is produced by an unknown Boolean network, a ground truth network (GTN). Methods for reconstruction aim to infer a network from the data as close to the unknown GTN as possible. Boolean networks impair this goal, because allowing logical gates leads to identifiability problems of network structures (figure 4.6). Especially considering "incomplete" experimental designs. For example if only single knock-down assays are available and the GTN has A go into B and C, while B and C can activate D independently of each other, the GTN is equivalent to the one which has A go into B,C and D directly. To overcome this limitation, prior knowledge on the pathway structures is used. Saez-Rodriguez *et al.* (2009) describe an algorithm called CellNet Optimizer (CNO) to construct signalling pathways from molecular data in the Boolean Network framework. They combine prior knowledge networks, with protein phosphorylation data form interventional assays.

In the future experimental approaches will become more refined and exact. Not only

single perturbations, but double, triple, and so on will become standard. Furthermore noise and unwanted effects from the perturbations will become less prominent. Combinatorial perturbations have already been used to resolve biological network structures Bonneau *et al.* (2006); Nelander *et al.* (2008). This way the use of literature knowledge can be reduced and the complex Boolean structures of the pathways can be reconstructed from scratch.

Here we describe Boolean Nested Effect Models (B-NEM). This method combines advantages from Boolean Network Models and Nested Effect Models. Like Boolean Networks they can distinguish between the synergistic and non-synergistic activation of a protein, and like in Nested Effect Models we do not need direct observations of protein activity. Moreover, B-NEMs can use data from assays, where several pathway genes are perturbed simultaneously. Contrary to the original NEM , B-NEM does not discriminate between stimulation (knock-in) or inhibition (knock-out) of a protein except for the assigned value. Thus the same protein can be overexpressed or stimulated in one (1) and inhibited (0) in another experiment.

1.2 Organization

First, in section 1.3 we will give a short overview of the biological processes of a cell. We explain, why it is important, that these processes remain undisturbed in healthy organisms. We also describe the kind of data we use with our method. The next chapter reviews the Boolean algebra and (hyper-)graph theory, which build the bases of the mathematical concept. In chapter 3 we will give an overview of established methods in the field of network reconstruction from biological data. In chapter 4 we give a detailed description of our novel method (B-NEM) to make inference of protein signalling pathways based on secondary effects from perturbation experiments. We validate B-NEM on simulated data in chapter 5. In chapter 6 we elucidate, how Pi3k and Tak1 mediate B-Cell receptor signalling into the Jnk, p38 and Ikk2 pathways in lymphoma cells. In chapter 7 we apply B-NEM to a dataset derived from hepatocellular carcinoma cells and investigate the crosstalk between apoptotic (Trail) and inflammatory (Tnf-α) signalling. In the last chapter 8 before the conclusion, we use B-NEM to discover cyclic signalling in developmental phases of mouse embryonic stem cells.

If not stated otherwise, all Graphs in this work have been created with the help of the Rgraphviz package (Hansen *et al.* (n.d.)). The R scripts for B-NEM are available at `https://github.com/MartinFXP/B-NEM`.

1.3 Intracellular Signalling Pathways in the Context of Cancer

Protein Signalling Pathways

A protein signalling network or pathway is a set of proteins, which interact inside a cell. These interactions can happen in different ways. For example protein A can change the activity of protein B by processes such as phosphorylation, de-phosphorylation or

ubiquitination. Or two proteins form a complex and together interact with other proteins to propagate a signal.

A pathway is usually stimulated when a protein from outside the cell attaches to a protein in the cell membrane (receptor). For instance the tumor necrosis factor α (Tnfα) binds to the Tnf receptor 1 (Tnfr1). The receptor changes its shape inside the cell. Intracellular proteins such as Traf2 and Rip1 bind to the receptor and become active. They propagate the Tnf-α signal via other proteins such as signalling kinases (e.g. Mekk, Nik, Tak1). At the end of the pathway members of the NfκB protein family (e.g. RelA, RelB) become active and transcribe genes responsible for an inflammatory response which promotes cell survival (Wajant *et al.* (2003); Bradley (2008); Haas *et al.* (2009); Silke (2011); Metzig *et al.* (2011a); Walczak (2011); Darding & Meier (2012); de Almagro & Vucic (2012)).

Signalling in Cancer

Random mutations are regularly introduced into the genome. Most mutations do not alter function and behaviour of a cell. However, sometimes a mutations occurs at a vital part of the genome, like a gene, which can lead to abnormal behaviour of the cell (Kan *et al.* (2013); Tornesello *et al.* (2013)). For example NfκB is constitutively active in some cases of liver cancer, which entails tumor growth (Dufour & Clavien (2005)).

Several processes can cause mutated DNA. For example UV light or radiation directly damages the DNA of the genome. If that happens, automatic repair mechanisms will try to reverse this damage. These mechanisms are stochastic processes and can fail to correctly repair the affected area of DNA. In other words mutations occur in that part of the genome. Another process that can lead to mutations is the division of one cell into two identical daughter cells. Before the cell splits in two, it grows and duplicates its genome. If this duplication is imperfect one or both of the daughter cells become mutated. In general any process involving building or processing the genome entails the danger of mutated cells and tumor development.

Mutations can alter the behaviour of signalling pathways. Every signalling pathway has certain tasks such as the induction of programmed cell death (apoptosis, Brune Bernhard (2003)). If a mutation disturbs such a pathway, the cell won't work properly anymore. For example if cells are damaged beyond repair, normally they will undergo apoptosis and die. The apoptotic signal is propagated via caspase 9 to effector caspases 3 and 7. The effector caspases degrade proteins in the cell, which leads to cell death. However, a mutation in one or both effector caspases 3 and 7 might prevent programmed cell death and promote cancer development (Soung Young Hwa *et al.* (2003); Soung *et al.* (2004)).

Perturbation Biology

We infer the properties of signalling pathways with the help of perturbation experiments. Perturbing a pathway can be done in different ways. Usually a stimulation induces the pathway and it becomes active (Shapiro & Vallee (1991); Boutros *et al.* (2002)). The active pathway propagates the signal. Additionally to the stimulation we can inhibit a member of the pathway. Then we observe the phenotype of a cell type for different combinations of stimulations and inhibitions (Hamilton & Baulcombe (1999); Agrawal

et al. (2003); Boutros *et al.* (2004, 2006); Boutros & Ahringer (2008)). A cell phenotype is an observation of the cells state such as mRNA abundances (Chang (1983); Schena *et al.* (1995); Lashkari *et al.* (1997); Mortazavi *et al.* (2008); Morin *et al.* (2008); Chu & Corey (2012)) or cellular dynamics (live cell imaging, Monya (2010)).

Gene Expression Profiles

We want to make inference on the signalling pathway of a cell by looking at its phenotype after a series of experiments: stimulation of a receptor, knock-down of a gene, inhibition of a protein. In our case the phenotype of a cell in a specific experiment is defined by gene expression profiles. For each experiment and each gene we look at the gene's mRNA abundance (expression). The gene expression of one gene over a series of experiments is the gene expression profile of this specific gene, while the expression of all genes in one experiment is the global gene expression profile of this experiment. Several technologies exist to measure gene expression such as the microarray technology (Chang (1983); Schena *et al.* (1995); Lashkari *et al.* (1997)) or RNAseq (Mortazavi *et al.* (2008); Morin *et al.* (2008); Chu & Corey (2012)).

2

Boolean Hyper-Graphs

2.1 Boolean Algebra

This chapter reviews Boolean algebra (Mendelson (1970); Rosen (2012)), (hyper-)graph theory and Boolean signalling graphs as described in Klamt *et al.* (2006, 2007) and Saez-Rodriguez *et al.* (2009, 2011).

Definition 2.1 (Boolean function)**.** *A function f is Boolean, if it takes a set of n inputs with binary values and returns one binary output.*

$$\mathrm{f} : \{0,1\}^n \to \{0,1\}\,.$$

We write Boolean functions as normal forms. The input variables $x_1, \ldots, x_n \in \{0,1\}$ are called literals. $\wedge$ denotes the AND operator.

$$x \wedge y = 1 \text{ if and only if } x = 1 \text{ and } y = 1.$$

$\vee$ denotes the OR operator.

$$x \vee y = 0 \text{ if and only if } x = 0 \text{ or } y = 0.$$

The negation operator $\neg$ works as follows.

$$\neg x = 1 \text{ if and only if } x = 0.$$

Literals can be combined by the AND operator in what we call a clause:

$$(x_1 \wedge \ldots \wedge x_m) = 1 \text{ if and only if } x_k = 1 \text{ for all } k.$$

These kind of clauses can then be combined by the OR operator. We call combination normal form.

$$\begin{aligned} &\bigvee_j (x_{j_1} \wedge \ldots \wedge x_{j_m}) = 1 \text{ if and only if} \\ &\text{there exists at least one } j \text{ for which } x_{j_k} = 1 \text{ holds for all } k. \end{aligned} \tag{2.1}$$

In other words, a normal form as in (2.1) is 1 if there is at least one clause, in which every literal is 1 and 0 otherwise. The number of literals in each clause can vary. A normal form as in (2.1) is called disjunctive (DNF). Alternatively we define a conjunctive normal form (CNF):

$$\bigwedge_j (x_{j_1} \vee \ldots \vee x_{j_m}) = 1 \text{ if and only if}$$
$$\text{for each } j \text{ there exists at least one } k \text{ for which holds } x_{j_k} = 1. \tag{2.2}$$

In other words a CNF is 1, if in each clause there is at least one literal, which is 1. It is 0 otherwise.

There are several rules to convert normal forms.

Identities:

$$x \wedge 1 = x \tag{2.3}$$
$$x \vee 0 = x \tag{2.4}$$

Commutativity:

$$x \wedge y = y \wedge x \tag{2.5}$$
$$x \vee y = y \vee x \tag{2.6}$$

Associativity:

$$x \wedge (y \wedge z) = (x \wedge y) \wedge z \tag{2.7}$$
$$x \vee (y \vee z) = (x \vee y) \vee z \tag{2.8}$$

Double negation:

$$\neg(\neg x) = x \tag{2.9}$$

De Morgan Laws:

$$\neg(x \wedge y) = \neg x \vee \neg y \tag{2.10}$$
$$\neg(x \vee y) = \neg x \wedge \neg y \tag{2.11}$$

Idempotencies:

$$x \wedge x = x \tag{2.12}$$
$$x \vee x = x \tag{2.13}$$

Annihilators:

$$x \wedge 0 = 0 \tag{2.14}$$
$$x \vee 1 = 1 \tag{2.15}$$

Distributivity Laws:

$$(x \wedge y) \vee z = (x \vee z) \wedge (y \vee z) \tag{2.16}$$
$$(x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z) \tag{2.17}$$

Absorption:

$$(x \wedge y) \vee x = x \tag{2.18}$$
$$(x \vee y) \wedge y = y \tag{2.19}$$

Complementation:

$$x \wedge \neg x = 0 \tag{2.20}$$
$$x \vee \neg x = 1 \tag{2.21}$$

Every DNF can be converted into a CNF and vice versa. This is done by applying the distributivity laws in (2.16)-(2.17). After we have applied the distributivity laws other rules are helpful to reduce the normal form. For example:

$$\begin{aligned}
&(\neg x \wedge \neg y) \vee (z \wedge x \wedge \neg y) \\
&\overset{(2.16)}{=} (\neg x \vee z) \wedge (\neg x \vee x) \wedge (\neg x \vee \neg y) \wedge (\neg y \vee z) \wedge (\neg y \vee x) \wedge (\neg y \vee \neg y) \\
&\overset{(2.21)}{=} (\neg x \vee z) \wedge 1 \wedge (\neg x \vee \neg y) \wedge (\neg y \vee z) \wedge (\neg y \vee x) \wedge (\neg y \vee \neg y) \\
&\overset{(2.13)}{=} (\neg x \vee z) \wedge 1 \wedge (\neg x \vee \neg y) \wedge (\neg y \vee z) \wedge (\neg y \vee x) \wedge \neg y \\
&\overset{(2.3)}{=} (\neg x \vee z) \wedge (\neg x \vee \neg y) \wedge (\neg y \vee z) \wedge (\neg y \vee x) \wedge \neg y \\
&\overset{(2.19)}{=} (\neg x \vee z) \wedge \neg y.
\end{aligned} \tag{2.22}$$

We define the dual form of a normal form by switching $\wedge$ and $\vee$ operators. The dual form includes the dual literals, which have inverse values. The dual literals are marked with a $*$. For example the dual form of the initial form in (2.22) is

$$(\neg x^* \vee \neg y^*) \wedge (z^* \vee x^* \vee \neg y^*). \tag{2.23}$$

For the dual literals it holds that, if $x = 1$, then $x^* = 0$ and vice versa.

Remark 2.1 (Property of the dual form)**.** *Let $X = \{x_i\}_{i \in I \subset \mathbb{N}}$ be a set of literals, N a normal form and N^* its dual form.*
If $(x_i = 1 \; \forall i \in J \subset I \Rightarrow N = 1)$, then $(x_i^* = 0 \; \forall i \in J \subset I \Rightarrow N^* = 0)$.

Proof. Let $N = \bigwedge_j (x_{j_1} \vee \ldots \vee x_{j_m})$ be a CNF. If $N = 1$, then for each j there exists at least one k for which $x_{j_k} = 1$. $N^* = \bigvee_j (x_{j_1} \wedge \ldots \wedge x_{j_m})$ is the dual form of N. If the dual of literals from before are 0 $(x_{j_k}^* = 0)$, every clause in N^* is 0 and therefore N^* is 0.

Let $N = \bigvee_j (x_{j_1} \wedge \ldots \wedge x_{j_m})$ be a DNF. If $N = 1$, there exists at least one clause j for which we have $x_{j_k} = 1$ for all k. $N^* = \bigwedge_j (x_{j_1} \vee \ldots \vee x_{j_m})$ is the dual form of N. If the dual of literals from before are 0 $(x_{j_k}^* = 0)$, there is one clause j, which is 0. Thus N^* is 0.

□

For instance if $x, z, \neg y$ are all 1, then $(\neg x \wedge \neg y) \vee (z \wedge x \wedge \neg y)$ is 1. If $x^*, z^*, \neg y^*$ are all 0, then $(\neg x^* \vee \neg y^*) \wedge (z^* \vee x^* \vee \neg y^*)$ is 0.

2.2 Graphs and Hyper-Graphs

In this chapter we review (hyper-)graph theory (Wilson (1996),Bretto (2013)).

Definition 2.2 (graph)**.** *A set $G = (V, E)$ of vertices $v \in V$ and edges $E = \{e = \{v, w\}, v, w \in V\}$ is a graph.*

Definition 2.3 (walk). *A subset* $\{v_1, \ldots, v_n\} \subset V$ *is a walk, if for all pairs* v_i, v_{i+1} *exists an edge* $e \in E$ *with the property* $e = \{v_i, v_{i+1}\}$.

Definition 2.4 (path). *A walk* $\{v_1, \ldots, v_n\}$ *is a path, if* $(v_i \neq v_j \Leftrightarrow i \neq j)$.

Definition 2.5 (cycle). *A walk* $\{v_1, \ldots, v_n\}$ *is a cycle, if* $v_1 = v_n$.

Figure 2.1 shows a graph with several walks, paths and cycles. There are for example the walks $\{S0, S1, S2, S3, S4, S1\}$ and $\{S0, S1, S2, S3, S4, S5\}$, the paths $\{S1, S2\}$ and $\{S2, S3\}$, and the cycles $\{S1, S2, S3, S4, S1\}$ and $\{S4, S1, S2, S3, S4\}$. The second walk is also a path.

Definition 2.6 (subgraph). $G = (V, E)$ *is a subgraph of a graph* $G^* = (V^*, E^*)$ *if* $V \subseteq V^*$ *and* $E \subseteq E^*$.

Definition 2.7 (directed graph). *A graph* $G = (V, A)$ *is directed with arcs* $a = (v, w) \in A$, *if* (v, w) *denotes a directed relationship between* v *and* w. *Vertex* v *is the parent of* a *and* w *the child. Not that in a directed graph the edge* (v, w) *is different to the edge* (w, v).

Definition 2.8 (connectedness). *A directed graph* G *is connected, if for two vertices* $v, w \in V$ *there is either a path* $\{v, \ldots, w\}$ *or a path* $\{w, \ldots, v\}$.

Definition 2.9 (directed acyclic graph (DAG)). *A directed graph* $G = (V, A)$ *is acyclic, if it contains no cycles.*

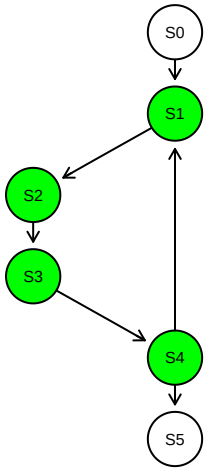


Figure 2.1: **Graph with cycle.** A graph containing the cycle $\{S1, S2, S3, S4, S1\}$ (green). It has one additional incoming and one outgoing edge.

Hyper-graphs generalize normal graphs (Bretto (2013)). Contrary to normal graphs hyper-graphs allow us to depict associations between more than just two vertices. Thus a normal graph is a hyper-graph with edges including exactly two vertices.

Definition 2.10 (hyper-graph). *A set* $H = (V, E)$ *of elements (vertices)* $v \in V$ *and (edges)* $E = \{e = (W_1, W_2), W_1, W_2 \subset V\}$ *is a hyper-graph.*

In the rest of this thesis we only use directed hyper-graphs with cardinality $|W_2| = 1$.

Definition 2.11 (walk). *A subset* $\{v_1, \ldots, v_n\} \subset V$ *is a walk, if for all pairs* v_i, v_{i+1} *exists a hyper-edge* $e \in E$ *with* $e = (W, v_{i+1}), v_i \in W$.

Definition 2.12 (directed hyper-graph). *A hyper-graph* $G = (V, A)$ *is directed with arcs* $(W, v) \in A$, *if* (W, v) *denotes an order between the sets* W *and* $\{v\}$. *Vertices* W *are the inputs (parents) and* v *is the output (child).*

The definitions of cycle, path, connectedness, directed acyclic hyper-graph (DAHG) and sub(hyper-)graph are analog to that of a normal graph.

We consider a special case of hyper-graphs. Those hyper-graphs have arcs representing Boolean functions (Akutsu *et al.* (2003); Klamt *et al.* (2006, 2007); Saez-Rodriguez *et al.* (2009, 2011)).

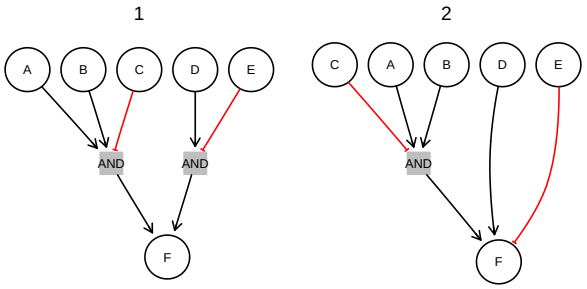


Figure 2.2: **Graphical representation of Boolean hyper-graphs.** Graph 1 corresponds to the disjunctive normal form $F = (A \wedge B \wedge \neg C) \vee (D \wedge \neg E)$. Graph 2 corresponds to $F = (A \wedge B \wedge \neg C) \vee D \vee \neg E$.

Definition 2.13 (Boolean directed hyper-graph (BG))**.** *A directed hyper-graph* $G = (V, A)$ *is Boolean, if every arc* $e \in A, e = (W, v)$, $W = \{w_1, \ldots, w_n\}$ *defines a Boolean function* $e : \{0,1\}^n \to \{0,1\}$, $v = e(w_1, \ldots, w_n)$.

We can now write a Boolean directed hyper-graph as a set of disjunctive normal forms.

$$\Psi = (S_i)_{i \in I \subset \mathbb{N}} \text{ with } S_i = \bigvee_j \left(\bigwedge_{k \in J_j \subseteq I} S_k \right) \tag{2.24}$$

where S_i are the vertices of the graph.

When we talk about graphs, the term Boolean implies directed hyper-edges with one or more parents per edge. Thus, from here on we refer to a Boolean directed hyper-graph as a Boolean graph (BG) or network for short. If not stated otherwise, we visualize a BG as a bipartite graph:

Boolean literals are drawn as circular vertices. AND clauses of a disjunctive normal form are depicted by additional grey rectengular vertices all labeled with AND. We draw an edge from every literal (inputs) in the clause directed into the AND vertex and from the AND vertex into the output. For example the DNF $D = (A \wedge B \wedge C)$ with inputs A, B, C and output D is converted to the normal graph $\{(A, AND), (B, AND), (C, AND), (AND, D)\}$ with the two different vertex sets $\{AND\}$ and $\{A, B, C, D\}$. Several of these bipartite edges combined correspond to the AND clauses combined by OR operators. Figure 2.2, 1 shows the hyper-graph representation of the DNF $F = (A \wedge B \wedge \neg C) \vee (D \wedge \neg E)$. If a literal is negated in the DNF, we illustrate this by a red tee ($\dashv$). If there is only one literal in a clause, we omit the AND vertex (figure 2.2, 2).

In the rest of the thesis we do not distinguish between a clause in a DNF and the corresponding (hyper-edge). For example if we write the edge $C = A \wedge B$, we mean the corresponding hyper-edge.

3

Network Models

A gene regulatory network visualizes relationships between genes as a graph. The graph depicts on how genes regulate each other's mRNA expression. An edge between two genes denotes an association. An arc between two genes denotes a directed causal effect such as a change in expression of gene A causes a change in expression of gene B (A $\rightarrow$ B). However the arc does not imply that a change in expression of B causes a change of expression of A. In this chapter we give a short review of Bayesian Networks to infer gene regulatory networks from gene expression data.

In section 3.2 we review a method, which does not infer a gene regulatory network, but a signalling pathway. However, the inference of the pathway is based on an underlaying gene regulatory network. In section 3.3 we review a method infering a signalling pathway based on how proteins influence each others phosphorylation state.

3.1 Bayesian Networks

Bayesian Networks (BN, Heckerman *et al.* (1995); Neapolitan (2003)) describe causal independences between a set of variables $(X_1, \ldots, X_n)$. For example gene A is indepenent of B given C, if $P(A|B,C) = P(A|C)$. In other words B does not provide additional information on A, if we already know C.

A Bayesian network is parameterized by a directed acyclic graph (DAG) G and a joint probability distribution D.
Notation:

- $I(A; C_1, \ldots, C_n|B)$ denotes A is independent of $C1, \ldots, C_n$ given B.

Figure 3.1 shows an example. Graph 1 defines the independences

$$I(A; E), I(B; D|A, E), I(C; A, D, E|B), I(D; B, C, E|A), I(E; A, D).$$

and joint probability distribution

$$P(A, B, C, D, E) = P(A)P(B|A, E)P(C|B)P(D|A)P(E).$$

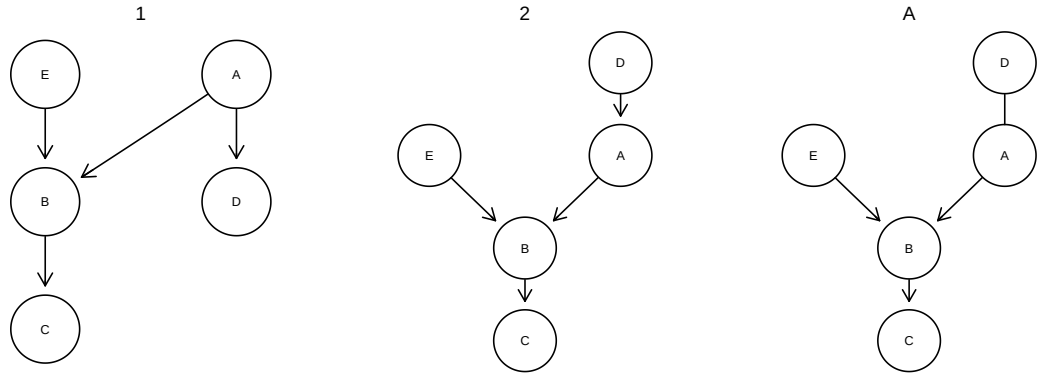


Figure 3.1: **Bayesian Network.** Two equivalent Bayesian networks (1, 2) and their equivalence class (A).

In general, given a graph $G = (\{X_1, \ldots, X_n\}, E)$ we can write the joint probability distribution

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | Pa^G(X_i))$$

with $Pa^G(X_i)$ as the parents of X_i.

Two graphs are equivalent if they define the same probability distribution. From the definition of conditional probability follows:

$$P(A)P(D|A) = P(D)P(A|D).$$

Thus graph 1 is equivalent to graph 2, because both graphs define the same probability distributions. Graph A shows a partially directed acyclic graph (PDAG) denoting the equivalent class for both DAGs 1 and 2.

We score a given graph G against the gene expression data

$$D = (x_{ij}), i = \{1, \ldots, n\}, j = \{1, \ldots, m\}$$

of m samples with the following score

$$S(G : D) = logP(G|D) = logP(D|G) + logP(G)$$

with

$$P(D|G) = \int P(D|G, \Theta)P(\Theta|G)d\Theta$$

and Θ as the conditional probability distributions of each variable X_i on its parents. This marginal likelihood approach regularizes the score for the graph size.

We use search heuristics like genetic algorithms (Mitchell (1999)) or greedy hill-climbing (Cormen *et al.* (2007)) to optimize the score.

3.2 Nested Effects Models

Nested Effects Models (NEM) (Markowetz *et al.* (2005)) are based on the concept of signal flow. A protein signalling pathway is activated via stimulation of a receptor.

The signal flows through the pathway from the membrane and via signalling molecules downstream to transcription factors. The transcription factors become active and execute or repress the transcription of their target genes.

Hierarchical relationships in the pathway are inferred from knock-downs of the pathway players during stimulation. NEM makes the following assumption. The further up a protein A is in the pathway the more target genes change their gene expression during a knock-down of A. If the set of effected genes of protein B is a noisy subset of the set of effects of protein A, we can conclude A upstream of B ($A \rightarrow B$). This relationship implies an indirect knock-down of B every time we perform a knock-down of A.

3.2.1 Original approach

Markowetz *et al.* (2005) call the proteins that are part of the signalling pathway signalling genes or S-genes. Let S_j be an S-gene, then $\phi_{j,k}$ describes the effect state of S_j during knock-down of S_k. $\phi_{j,k} = 1$ denotes that the knock-down of S_k S-gene is affected by the knock-down of S_k, $\phi_{j,k} = 0$ denotes that it is not. In other words if we knock-down S_k, we also knock-down or silence Sj ($\phi_{j,k} = 1$) or we don't ($\phi_{j,k} = 0$). This relationship is described as a silencing scheme via the adjacency matrix $(\phi_{i,j}) = \Phi$. $\phi_{i,j}$ is also called silencing effect.

Activation of S-genes is not directly observed. The data consists of gene expression profiles. Genes in the data which react to the perturbations are called effector genes or E-genes. $E_{i,k}$ denotes the discretized foldchange between knock-down and control during stimulation. $E_{i,k}$ denotes whether E-gene E_i is affected (1) by the knock-down of S_k or not (0). $\Theta = (\theta_i)$ denotes the regulation of E-genes. $\theta_i = j$ denotes E_i is directly regulated by S_j.

An example of (Φ, Θ) is shown in figure 3.2. Due to noise, the observed data includes false positives and false negatives with rates α and β. Markowetz *et al.* (2005) use α and β to calculate the following conditional probabilities:

$$P\left(E_{i,k} \,|\Phi, \theta_i = j\right) =$$

$E_{i,k} = 1$	$E_{i,k} = 0$	
α	$1 - \alpha$	if Φ predicts *no effect*
$1 - \beta$	β	if Φ predicts *effect*

. (3.1)

Different silencing schemes Φ are scored against data $D = (E_{i,k})_{i,k}$ with the marginal likelihood approach.

$$P\left(\Phi \,|D\right) = \frac{P\left(D \,|\Phi\right) \cdot P\left(\Phi\right)}{P\left(D\right)}$$

Let E-gene positions Θ be given. Since $P\left(D\right)$ is the same for every silencing scheme and $P\left(\Phi\right)$ is assumed uniform, we can write

$$P\left(D \,|\Phi, \Theta\right) = \prod_{i=1}^{m} P\left(D_i \,|\Phi, \theta_i\right) = \prod_{i=1}^{m} \prod_{k=1}^{l} P\left(E_{i,k} \,|\Phi, \theta_i\right) \tag{3.2}$$

where D_i is the row of all effects for E-genes i.

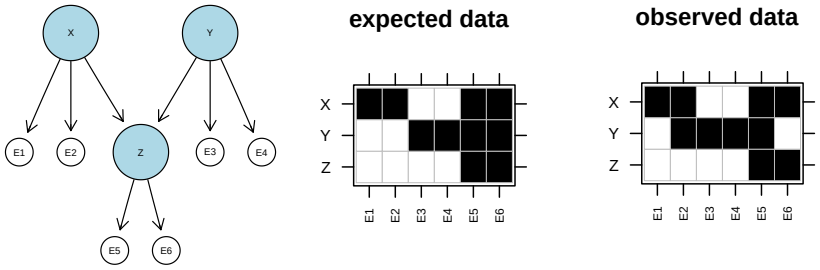


Figure 3.2: **Nested Effects Model.** Network for a silencing scheme of S-genes X, Y and Z with their own exclusive set of E-genes attached to each (left). The effects of the respective knock-downs on the E-genes (right). The observed data differs from the expected due to false positives and false negatives.

Another assumption is the independence of the position of E-genes Θ. In other words the probability of E-gene i to be regulated by S-gene j is independent of the regulation of all other E-genes. It follows

$$P(\Theta|\Phi) = \prod_{i=1}^{m} P(\theta_i|\Phi). \tag{3.3}$$

Moreover we assume an uniform prior probability for the attachment of an E-gene to a S-gene.

$$P(\theta_i = j|\Phi) = \frac{1}{n}. \tag{3.4}$$

Θ is unknown and therefore a marginal likelihood is calculated that averages over Θ.

$$P(D|\Phi) = \int P(D|\Phi,\Theta)\,P(\Theta|\Phi)\,d\Theta \tag{3.5}$$

$$\overset{(3.2),(3.3)}{=} \prod_{i=1}^{m} \int P(D_i|\Phi,\theta_i)\,P(\theta_i|\Phi)\,d\theta_i \tag{3.6}$$

$$\overset{(3.4)}{=} \frac{1}{n^m} \prod_{i=1}^{m} \sum_{j=1}^{n} P(D_i|\Phi,\theta_i = j) \tag{3.7}$$

$$\overset{(3.2)}{=} \frac{1}{n^m} \prod_{i=1}^{m} \sum_{j=1}^{n} \prod_{k=1}^{l} P(E_{i,k}|\Phi,\theta_i = j). \tag{3.8}$$

The error rates α and β are two free parameters.

3.2.2 Optimization

Exhaustive search for the optimal network describing the data is only feasible for a limited number of S-genes. Markowetz *et al.* (2007) and Froehlich *et al.* (2007) developed different heuristics.

Pairwise

The pairwise search (Markowetz *et al.* (2007)) infers pairwise relationships between two S-genes (A, B). The four possible outcomes are A upstream of B $(A \rightarrow B)$, B upstream of A $(B \rightarrow A)$, A and B are indistinguishable $(A \leftrightarrow B)$ and A and B are unconnected/unrelated $(A..B)$. All models are scored and the best is selected.

The computation time of pairwise compared to the exhaustive search decreases especially for larger numbers of S-genes. However contrary to exhaustive search, pairwise does not cover the complete search space and hence does not guarantee the global optimal network. As a result Markowetz *et al.* (2007) also developed the triples search.

Triples

The triples algorithm consists of the following steps.

Algorithm 1 triples

---

1. score each possible network for each triple (A, B, C) and select the best
2. count how often each edge is chosen in all the triples and compute confidence score $f\left(A \rightarrow B\right) = \frac{1}{n-2} \sum_{C \notin \{A,B\}} 1\left[\text{"}A \rightarrow B\text{"} \in M_{ABC}\right]$,
   with indicator function $1\left[.\right]$ for the existence of edge $A \rightarrow B$ in the triplet M_{ABC}
3. edges that exceed a threshold (e.g. 0.5) are chosen for the final graph

---

Triples is computationally more expensive than pairwise, but achieves better results (Markowetz *et al.* (2007)).

Module Networks

Another inference algorithm is called module networks (Froehlich *et al.* (2007)). The method is based on the following notion. Knock-downs of S-genes which are in similar parts of the pathway produce similar expression profiles which cluster together. These clusters of S-genes can be further divided into smaller clusters.

Algorithm 2 module network inference

1. cluster the complete set of S-genes into modules
2. further subdivide until the leaf modules of the cluster hierarchy have at most 4 S-genes
3. exhaustively search for the highest scoring graph for each leaf module
4. connect the leafs by their highest scoring pairwise connection and transitively close the graph for each module
5. recursively do steps $3-4$ until the complete connected network has been established

3.2.3 Factor Graph Nested Effects Models

The original NEMs (Markowetz *et al.* (2005)) are limited to positive interactions. Factor-Graph Nested Effect Models (FG-NEM, Vaske *et al.* (2009)) directly extend the framework by distinguishing between positive and negative interactions between S-genes, as well as positive and negative regulation of E-genes (figure 3.3, A). This is achieved by replacing the binary values of the silencing scheme Φ with six different interaction modes.

1. A activates B; $A \rightarrow B$
2. A inhibits B; $A \dashv B$
3. A is equivalent to B; $A = B$
4. A does not interact with B; $A \neq B$
5. B activates A; $B \rightarrow A$
6. B inhibits A; $B \dashv A$

The novel $A \dashv B$ interaction denotes if A is *not* knocked down ($A = 0$), it has an effect on B ($B = 1$). In other words B is inhibited.

Furthermore interaction modes are inferred from scatter plot profiles (figure 3.3, C). A scatter plot of the observed E-gene data between two knock-downs A and B is divided into nine regions: up/down-regulated by A and B (four regions), not regulated by A and B (one region), regulated by one and not by the other (four regions). The observed modes of E-gene profiles are compared with expected modes to infer the type of regulation between two S-genes. The maximum a posteriori is used as objective function to identify networks which maximize the networks respectively its modes Φ and E-gene regulation Θ.

$$\mathrm{J}(X) = \max_{\Phi} \left\{ P(\Phi) \prod_{e \in E, A, B \in S} \max_{\Theta_{eAB}} \sum_{Y_{eA}, Y_{eB}} P\left(Y_{eA}, Y_{eB} \,|\Phi_{AB}, \Theta_{eAB}\right) P(X_{eAr} \,|Y_{eA}) P(X_{eBr} \,|Y_{eB}) \right\}. \tag{3.9}$$

Y_{eA} is the hidden discrete state of E-gene e during knock-down of A, X_{eAr} is the observed state in replicate r, Θ_{eAB} is the type of regulation for e by the pair A, B and $P(\Phi)$ is the probability of the interaction model Φ.

$$P(\Phi) \propto \left(\prod_{A,B,C \in S} \pi_{ABC} \left(\Phi_{AB}, \Phi_{AC}, \Phi_{BC} \right) \right) \left(\prod_{A,B \in S} \rho_{AB} \left(\Phi_{AB} \right) \right). \tag{3.10}$$

π_{ABC} is the transitivity factor and ρ_{AB} a prior for the interactions. π_{ABC} is zero if a triple A, B, C does not form a valid transitive structure (e.g. $A \to B \dashv C$, $A \to C$) and one otherwise.

Model inference

FG-NEM does model inference on a factor graph (Kschischang *et al.* (2001)). A factor graph is a bipartite graph with factors and variables as vertices. A factor and a variable are connected, if the factor includes the variable. A factor graph representing the example from figure 3.3 is shown in figure 3.4. The factor-graph encodes all possible networks at once. Variables (circles) are

- S-gene associations $\Phi = (\Phi_{AB})$
- hidden E-genes expression states $Y = (Y_{eA})$ with Y_{eA} as the hidden, discrete E-gene state of E-gene e during knock-down A and
- observed E-gene expressions $X = (X_{eAr})$ with X_{eAr} as the expression of E-gene e in replicate r of knock-down A.

Factors (boxes) are expression factors, interaction factors and transitivity factors. Expression factors are Gaussian and connect the observed and the hidden E-gene state. Interaction factors are the probabilities of the form

$$P\left(Y_{eA}, Y_{eB} \,|\Phi_{AB}, \Theta_{eAB} \right). \tag{3.11}$$

The maximum a posteriori is found using equations (3.9) and (3.10) and max-sum message passing (Kschischang *et al.* (2001)).

3.2.4 Dynamic Nested Effects Models

Original NEMs are designed for static experiments. After each experiment, the signalling pathway is assumed to be in a steady state. A steady state means every S-gene is in a final state of either 0 or 1 and there is no further change. As a result original NEMs cannot resolve feed forward loops. For example in case of the network $A \to B \to C$ the original NEM cannot determine if the effect of A on C is direct or indirect via B. Thus a feed forward loop from $A \to C$ is not resolvable under steady state assumptions. Dynamic Nested Effects Models (D-NEM, Anchang *et al.* (2009)) extends NEM and uses timeseries data to infer time dependent signalling processes like feed forward loops.

Novel model parameters in D-NEM are the signalling rates $K = (k_{ij})$ of edges from S-genes i to j (figure 3.5). They are assumed to be exponentially distributed. K replaces

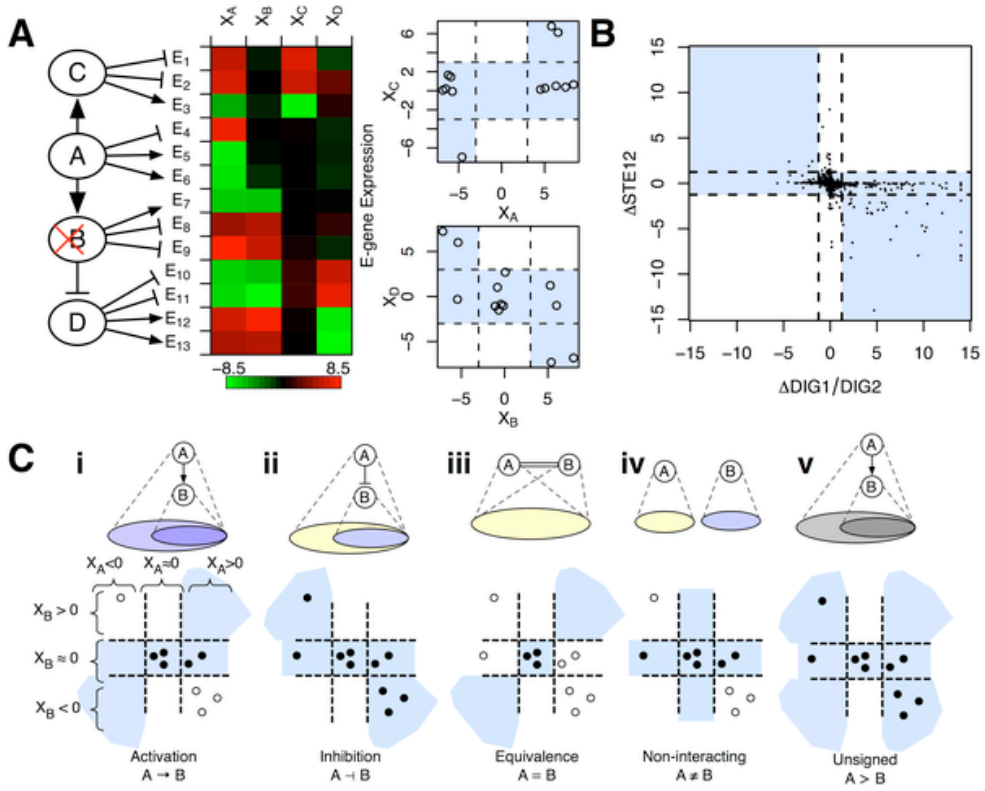


Figure 3.3: **Predicting Pair-wise Interaction Using Quantitative Nested Effects.** (A) Hypothetical example with four S-genes, A, B, C, and D. The graph contains one inhibitory link, BxD (left). A heatmap of E-gene expression under knockdown of each S-gene shows both inhibitory and stimulatory effects (middle). Scatter plots of the C, A, B, and D knock-outs show that expression fits in the shaded preferred regions of each interaction (right). The inhibitory link explains some of the observed data: expression changes under DD (bright red or bright green entries in the heatmap) occur in a subset of the E-genes for which the opposite changes occur in DB. (B) Data from a known inhibitory interaction. Expression levels of effect genes under the DIG1/DIG2 knock-out (y-axis) plotted against their levels under the STE2 knock-out (x-axis) as detected in [17]. Expression changes significant at a = 0.05 indicated in gray lines. DIG1/DIG2 is known to inhibit STE12. (C) Interaction modes. Observed E-gene expression changes are compared to five possible types of interactions between two S-genes, A and B (iv). The top row illustrates the expected nested effects relationship for each type of interaction mode: circles represent sets of E-genes with expression changes consistent with either activation (blue circles) or inhibition (yellow circles). Scatter-plots for each interaction mode show the hypothetical expression changes under DA (x-axis) and DB (y-axis) for all E-genes (circles). E-gene levels are either consistent (filled) or inconsistent (open) with the mode. Shaded regions demark expression levels consistent with each interaction model. The example shows expression changes that most closely match the inhibition mode (indicated by the greatest number of closed circles). This figure was reproduced from Vaske *et al.* (2009).

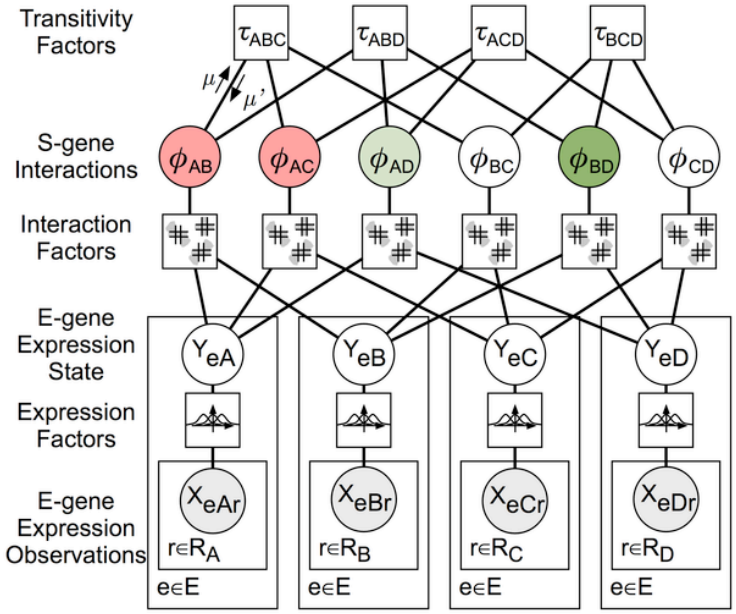


Figure 3.4: **Structure of the factor graph for network inference.** The factor graph consists of three classes of variables (circles) and three classes of factors (squares). XeAr is a continuous observation of E-gene e's expression under ΔA (knock-down of A) and replicate r. YeA is the hidden state of E-gene e under ΔA, and is a discrete variable with domain $\{up, , down\}$. Φ_{AB} is the interaction between two S-genes A and B. Expression Factors model expression as a mixture of Gaussian distributions. Interaction Factors constrain E-gene states to the allowed regions shown in Figure 3.3. Transitivity Factors constrain pair-wise interactions to form consistent triangles. The arrows labeled μ and μ' are messages encoding local belief potentials on Φ_{AB} and are propagated during factor graph inference. This figure was reproduced from Vaske *et al.* (2009).

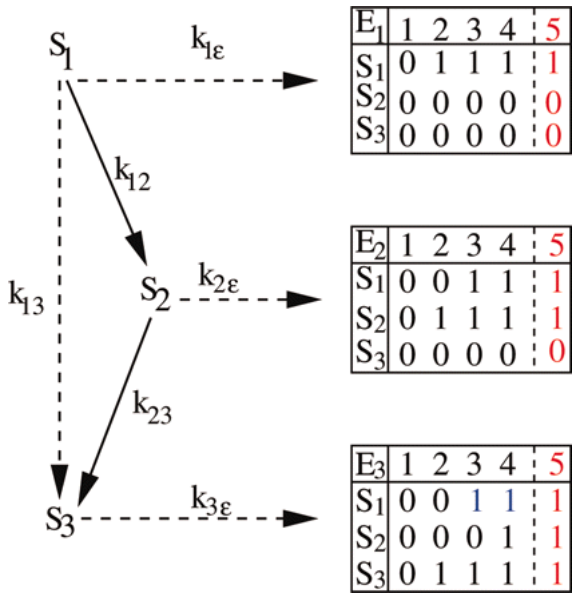


Figure 3.5: **Dynamic Nested effects Model** Elementary example of a D-NEM. Shown is a network of three S-genes together with binary time series tables for typical E-genes connected to the S-genes. Each table holds three rows corresponding to the three possible perturbation experiments of S-genes. A one in column t_i , row S_j of table E_k represents the observation of a downstream effect in E_k , t_i time units after perturbation of S_j. This figure was reproduced from Anchang *et al.* (2009).

the model parameters Φ, which describe the S-gene topology. A large k_{ij} from S-gene i to j corresponds to a high signalling rate and an edge between the S-genes. A small k_{ij} objects any signalling from S-gene i to j. Therefore D-NEMs are parameterized by rate constants K and S-gene to E-gene connectivity Θ as in 3.2.1.

The likelihood to score a candidate set K is defined as

$$P(D|K,\Theta) = \prod_{D=1} P_{S_i \to E_k}(t_s)(1-\beta) + (1 - P_{S_i \to E_k}(t_s))\alpha \\ \times \prod_{D=0} P_{S_i \to E_k}(t_s)\beta + (1 - P_{S_i \to E_k}(t_s))(1-\alpha).$$

α and β are false positive and false negative rates. $P_{S_i \to E_k}(t_s)$ is the probability, that the signal from S_i reaches E_k before timepoint t_s. How this probability is calculated is shown in the following example.

Let's assume we have the path g with a simplified index for the rate constants:

$$S_i \xrightarrow{k_1} S_{j_1} \cdots \xrightarrow{k_{q-1}} S_{j_{q-1}} \xrightarrow{k_q} E_k.$$

Z_g is the sum of of q independent, and exponentially distributed random variables with rate constants $k1, \ldots, kq$. Z_g's targeted probability is $P(Z_g < t_s)$. The density function of Z_g is given by

$$\Psi(t)_g = \int_0^\infty \cdots \int_0^\infty \delta\left(t - \sum_{u=1}^{q} \tau_u\right) \prod_{u=1}^{q} \Psi_u(\tau_u) d\tau_1 \ldots \tau_q$$

with $\Psi_u(\tau) = k_u \exp(-k_u\tau)$ as the density of an exponential with rate k_u. A Laplace transformation returns

$$F_g(t) = \sum_{b=1}^{q} \prod_{a \neq b} \left\{ \frac{k_a}{k_a - k_b} \right\} [1 - \exp(-tk_b)]$$

as a closed form for the cumulative distribution function of Z_g. In the case of two or more k_u with equal values the right side can be undefined. Anchang *et al.* (2009) avoid this by the use of small jitter values. Several paths can connect S_i with E_k. For each path u Z_u is constructed as before and the probability is given by

$$P_{S_i \to E_k}(t_s) = 1 - \prod_u (1 - F_u(t_s)).$$

An optimal network model is inferred via a Gibbs sampling approach (George Casella (1992)).

3.2.5 Partial Nested Effects Models

Usually Nested Effect Models assume, that the pathway is isolated. That is only S-genes included in the model influence the signalling. This is not the case in general. Literature is missing information on hidden players, which are not included in the model. Additionally those players are not known to be missing. Sadeh *et al.* (2013) calls these players unknown unknowns and introduce Partial Nested Effects Models (P-NEM) as an extension to the original NEMs to account for them.

P-NEM looks for patterns in the data, which contradict certain hypotheses (edges) between two S-genes. For example let's assume X is upstream of Y ($M = X \to Y$). E-genes reacting to Y but not X contradict that hypothesis, because if we knock-down X we indirectly knock-down Y given M. These contradicting effects are called a alien patterns. Every alien pattern in the data is tested for significance with a binomial test.

Alien patterns exists for four of five different edge relations between two S-gene (figure 3.6, a). Each relation has its unique set of alien patterns except for the one in which X and Y have a common child (R5). This relation can explain all patterns.

For an example on how to detect significant alien patterns we look at relation R1. Edge $X \to Y$ (R1) has three expected patterns and one alien pattern $(X, Y) = (0, 1)$. The alien pattern can be the result of an expected pattern and noise (expected $\overset{\text{noise}}{\frown\!\!\!\!\!\!\to}$ alien). If we consider a false positive rate α and false negative rate β, we can calculate the probability of each case:

- $(1, 0) \overset{\text{noise}}{\frown\!\!\!\!\!\!\to} (0, 1)$ is the result of one false negative and one false positive and we have a probability of $\gamma_1 = \beta \cdot \alpha$
- $(1, 1) \overset{\text{noise}}{\frown\!\!\!\!\!\!\to} (0, 1)$ is the result of one false negative and one true positive and we have a probability of $\gamma_2 = \beta \cdot (1 - \beta)$
- $(0, 0) \overset{\text{noise}}{\frown\!\!\!\!\!\!\to} (0, 1)$ is the result of one true negative and one false positive and we have a probability of $\gamma_3 = (1 - \alpha) \cdot \alpha$

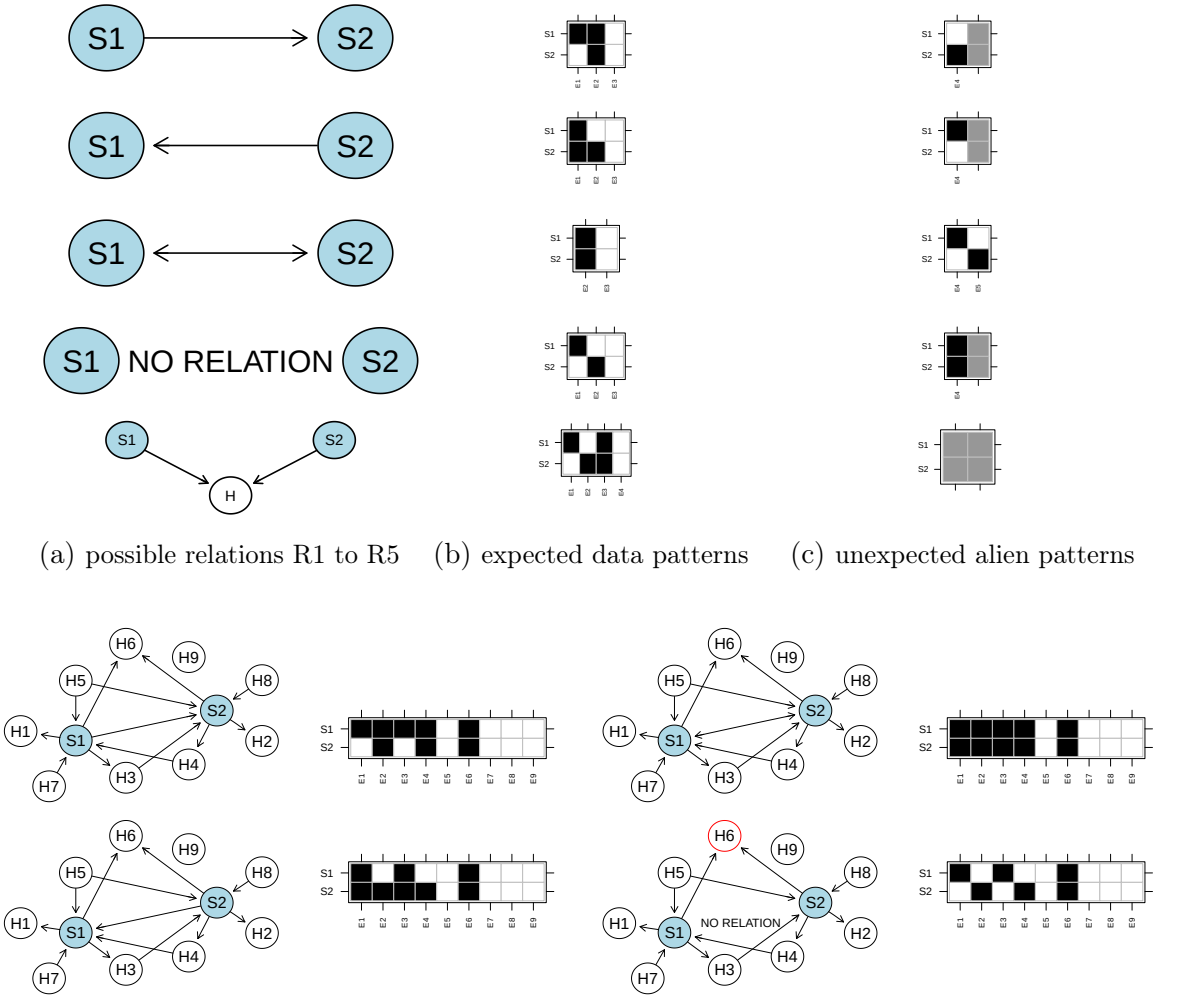
We can now write an upper limit for the probability of more than k alien patterns in the data, given relation R1:

$$P\left(K \geq k \,|X \rightarrow Y\right) \leq \sum_{i=k}^{n} \binom{n}{i} \gamma_{R1}^{i} \left(1-\gamma_{R1}\right)^{n-i} \tag{3.12}$$

n is the total number of E-genes and γ_{R1} an upper bound for the probability of observing an alien pattern. Such a bound for the probability is available for all relations except for R5, since it lacks an alien pattern. The other relations R1, R2, R3 and R4 are rejected if

$$P\left(K \geq k \,|Ri\right) < \kappa$$

with κ as a cutoff for significance, e.g. $\kappa = 0.05$.



(a) possible relations R1 to R5 (b) expected data patterns (c) unexpected alien patterns

(d) hidden nodes for relations R1 and R2 (e) data patterns for the hidden nodes (f) hidden nodes for relations R3 and R4 (g) data patterns for the hidden nodes

Figure 3.6: **Partial Nested Effects Models** Pairwise upstream/downstream relations and their alien patterns: (Top) Shown are the five possible possible relations (R1) . . . (R5) together with their expected silencing patterns and their alien patterns (grey are NA values). R4 are disconnected S-genes without indirect connections. (Bottom) Hidden vertices are introduced in all possible configurations, and the expected patterns of E-genes attached to the hidden vertices are shown. In (R4) the hidden vertex marked in red produces the alien pattern of (R4). Note that this constellation leads to the constellation in (R5). Based on figure 2 in Sadeh *et al.* (2013).

3.3 CellNet Optimizer

The CellNet Optimizer (CNO, Saez-Rodriguez *et al.* (2009, 2011)) uses phosphoproteomics data to infer the signalling logics of a protein pathways. The pathway is modelled as a Boolean graph. Each vertex can be 1 or 0. The corresponding proteins are active (1) or inactive (0). The graph edges correspond to different Boolean functions. The base functions are AND, OR and NOT. If two parent vertices are connected to a child vertex with an AND-gate, both parents need to be active for the child to be active. If they are connected by an OR-gate, one parent is enough to propagate the signal. In case of the NOT-gate the parent needs to be inactive for the child to become active.

The CNO uses a prior knowledge network (PKN). The PKN denotes possible interactions in a simple directed graph (figure 3.7, A). This graph is then processed to a full hyper-graph. Unnecessary vertices are compressed (figure 3.7, C,D). For example if vertex B is unobserved and the PKN includes the edges $A \to B \to C$, the two edges are compressed to $A \to C$. Then all edges are processed to include all possible Boolean functions derived from the PKN (figure 3.7, E). For example if the PKN includes edges $A \to C$ and $B \to C$ the AND-gate A AND $B \to C$ (DNF: $C = A \wedge B$) is added. The search space consists of all networks which are sub-graphs of the extended hyper-graph.

The experimental design of the input data combines stimulations and inhibitions. Pre-set vertices are dependent of the design. For example if a receptor is stimulated in an experiment its model vertex is set to 1. Similarly different players in the pathway can be inhibited and the associated model vertices are set to 0 independently from the rest of the network. The states of all unset vertices, neither stimulated nor inhibited, are calculated based on the pre-set vertices and the Boolean functions in a candidate network P. P is scored against the data with the bipartite objective function (equation (3.13)).

$$\begin{aligned} \Theta(P) &= \Theta_f(P) + \Theta_s(P), \\ \Theta_f &= \frac{1}{n_E} \sum_{k=1}^{s} \sum_{l=1}^{m} \sum_{t=1}^{n} \left(B^M_{k,l,t} - B^E_{k,l,t} \right)^2. \end{aligned} \tag{3.13}$$

$B^M_{k,l,t} \in \{0, 1\}$ is the model predicted state and $B^E_{k,l,t} \in [0, 1)$ the measured state of vertex l at time t in experiment k. Θ_s penalizes the size of the candidate network P. The number of tails of P is divided by the maximal number of tails in the network with all possible hyper-edges. For example the AND-gate $C = A \wedge B$ gets the same size penalty (2) as the OR-gate $C = A \vee B$.

3.3.1 Model inference with a genetic algorithm

Even with a medium sized network with 50 edges the complete search space holds $2^{50} \approx 10^{15}$ candidate networks. Exhaustive search is not feasible anymore and a fast search heuristic is necessary.

A genetic algorithm (GA, Mitchell (1999)) is a stochastic search heuristic and borrows mechanics from evolutionary biology to traverse the search space. GAs reach reasonable convergence with relatively little time consumption. The general outline of a GA is the following:

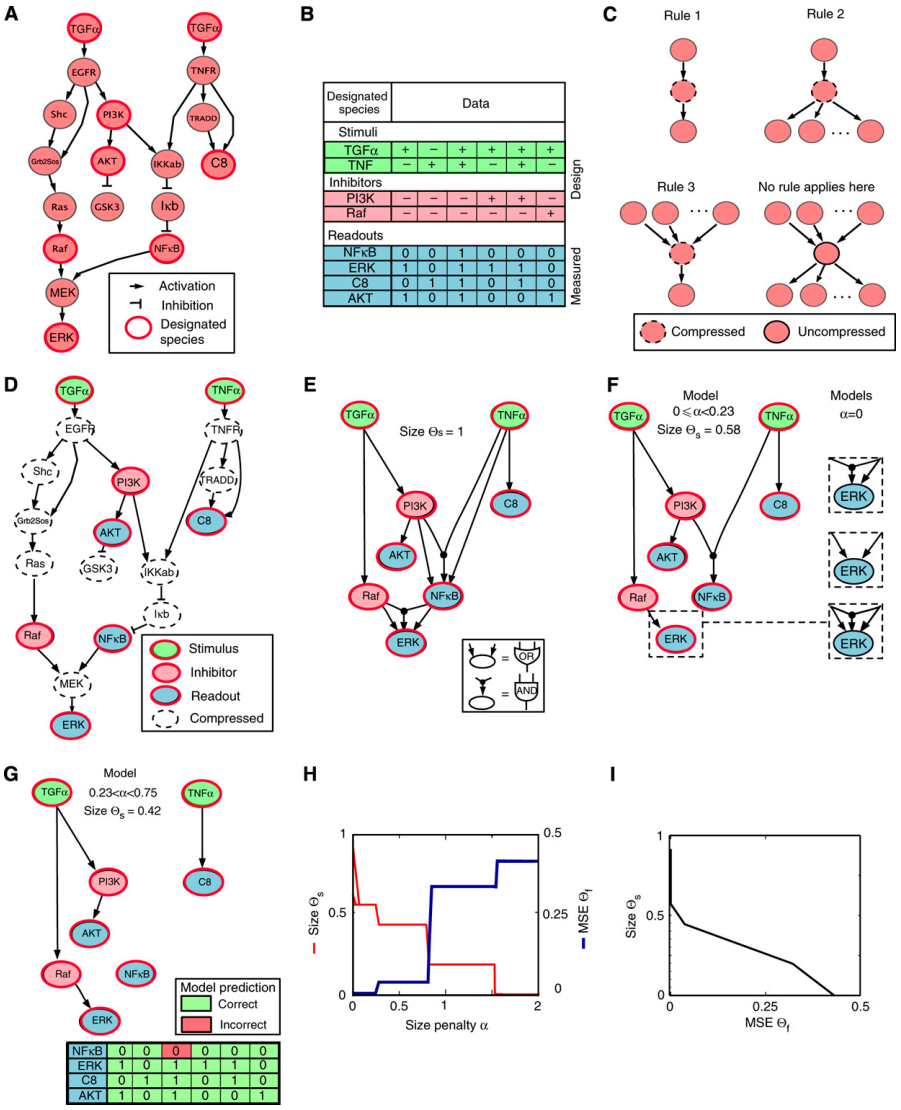


Figure 3.7: **CellNet Optimizer** Assembly, calibration, and analysis of a toy signalling model. (A) Signed directed graph representing a simple pathway as visualized using Cytoscape (Shannon *et al.* (2003)). The topology of the reactions downstream of TGFα and TNFα receptors is imaginary, but it includes real molecules such as Shc, Ras, Raf, MEK, ERK, PI3K, AKT, GSK3, IκK, IκB, NFκB, TRADD, caspase 8 (denoted C8), and the GrbSos complex (denoted GrbSos). (B) The design of the synthetic experiments used to train the graph in panel A. Each column represents an experiment and each row a different designated species as follows: green denotes ligands, red denotes the protein targets of kinase inhibitors, and blue denotes the proteins whose states were assayed (readouts). The presence or absence of ligand or an inhibitor specific to a node is denoted with + and −, respectively. The 0/1 value for the readouts corresponds to the result obtained from simulating the reference model under specific conditions of ligand and inhibitor exposure. (C) Rules applied to graphs to create compressed representations. (D) The experimental design (B) determines which nodes in the graph are designated and which are undesignated. This information, in combination with the rules in panel C was used to create a compressed graph, with nodes eliminated by compression indicated by dashed lines. (E) Superstructure of all models compatible with the graph in panel A. (F) Optimal models for size penalties of $0 \leq \alpha \leq 0.23$. The highlighted panels to the right (boxed with dashed lines) show three different logical structures recovered during model calibration with $\alpha = 0$. The fit to data was perfect for all models ($\Theta_f = 0$). (G) Optimal model for $0.23 \leq \alpha \leq 0.75$. The matrix below shows the single mismatch (in red) between models based simulations and the training data. (H, I) Balance between the fit of the data Θ_f (the MSE deviation from data; see text for details) and size Θ_s for models recovered using different values of the size penalty, α. This figure was reproduced from Saez-Rodriguez *et al.* (2009).

Algorithm 3 genetic algorithm

1. start with initial population of N networks (e.g. random)
2. score each network of the current population
3. create probability distribution for the current generation
4. sample pairs of networks from the current population and combine (e.g. uniform) them to create a new population
5. with a (low) probability randomly change edges (mutation rate)
6. go to step 2 or stop (stop criteria: no improvement of the best score for n generations, timelimit, maximum number of generations, score based convergence).

The networks are encoded as binary vectors with an entry for each hyper-edge in the search space. 1 encodes the hyper-edge as present in the current network and 0 as absent. Two vectors of parent networks A and B are combined to a child C via an uniform selection. In other words for each hyper-edge entry we draw a uniform number r. If r is smaller or equal to 0.5, C gets the entry of A. If r is greater than 0.5, C gets the entry of B.

Stochastic Universal Sampling

The GA algorithm constructs the probability distribution of the network population with stochastic universal sampling (SUS) (algorithm 4). SUS uses a linear fitness function, a special case of rank selection (Mitchell (1999); Baker (1985)), with a selection pressure parameter φ. The chance for the best network to be chosen for reproduction is φ times the chance of the average network.

Algorithm 4 stochastic universal sampling

1. score each network of the current population of size N and rank them from worst (1) to best (N)
2. based on selective pressure φ calculate fitness ν_i for each network $i \in \{1, \ldots, N\}$ $\varphi \in [1, 2]$, $\nu_i = 2 - \varphi + \left(2\left(\varphi - 1\right)\frac{i-1}{N-1}\right)$
3. set interval limits $\omega_0 = 0$, $\omega_i = \sum_{j=1}^{i} \frac{\nu_i}{\sum_{k=1}^{N} \nu_k}$ and breaks $b_i = r \cdot \frac{1}{N} + \frac{i-1}{N}$ with random number $r \sim \mathrm{U}\left(0, 1\right)$
4. $s_i = l$ if and only if $b_i \in [\omega_l, \omega_{l+1})$ for $l \in \{1, \ldots, N\}$

$S = (s_i)$ is now the empirical distribution.

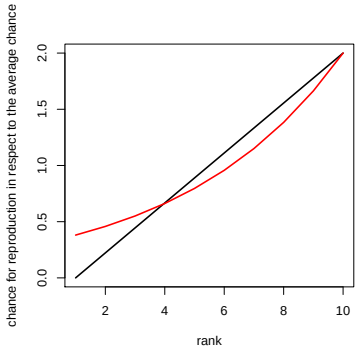


Figure 3.8: **Example for linear and non-liner fitness.** Fitness curves for linear (black) and non-linear (red) fitness with selection pressure 2 for a population of 10 networks. The fitness value (y-axis) describes the chance to be chosen for reproduction in respect to the average network. With a selection pressure of 2 the best network is on average chosen twice as often as the average network.

Non-linear fitness (Pohlheim (1995)) is an alternative to linear fitness and more equally balances exploitation and exploration. Exploitation is the convergence of the current solutions in the local neighbourhood. Exploration introduces more variance in the solutions to investigate a larger search area.

For population size N we can choose selection pressure $\Phi \in [1, N)$. The fitness ν_i is then calculate the following way:

$$\nu_i = \frac{N \cdot x^{i-1}}{\sum_{j=1}^{N} x^{j-1}}$$

for the root x of the polynomial

$$0 = (\Phi - N) \cdot x^{N-1} + \Phi \cdot x^{N-2} + \cdots + \Phi \cdot x + \Phi.$$

An example for the difference between linear and non-linear fitness is shown in figure 3.8. An example for the calculation of the s_i (section 3.3.1) for a population of 10 networks is shown in figure 3.9.

Tournament Selection

Unbiased tournament selection (UTS, Sokolov & Whitley (2005)) is an alternative to stochastic universal sampling (SUS). UTS eliminates the selection pressure parameter and can be computed more efficiently. UTS also guarantees that the sampling set always includes the best network twice and never the worst.

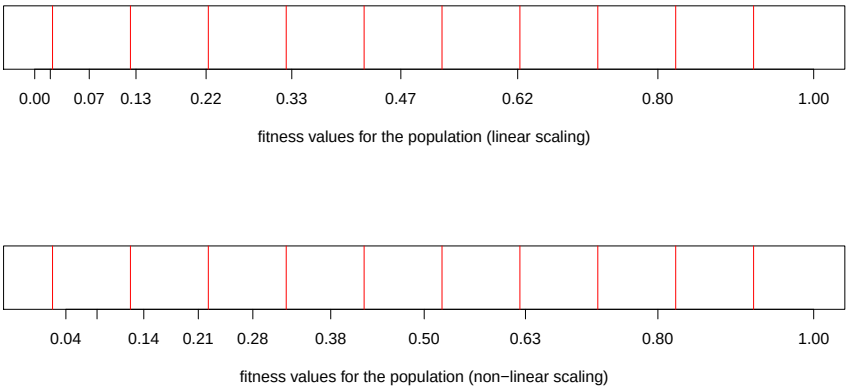


Figure 3.9: **Example for stochastic universal sampling and (non-)linear fitness.** On the x-axis, we see the calculated values for $\omega_i, i \in \{1, \ldots, 10\}$. The red lines mark the values for the randomly selected $b_i, i \in \{1, \ldots, 10\}$. The b_i are the same in both, but the ω_i are scaled differently (top: linear, bottom: non-linear). The medium ranked networks have a smaller chance with non-linear selection pressure, but the lowly ranked networks have a higher chance.

Algorithm 5 tournament selection

---

1. score each network of the current population and rank them
2. permute the ranks with permutation p such that $i \neq \mathrm{p}(i) \ \forall i$
3. add i to the sample distribution $\Leftrightarrow$ i has a higher score than $\mathrm{p}(i)$;
   add $\mathrm{p}(i)$ to the sample distribution $\Leftrightarrow$ $\mathrm{p}(i)$ has a higher score than i.

We ensure that after the permutation in step 2 it holds $i \neq \mathrm{p}(i)$ with the following permutation. First permute the ranks. These randomly ordered ranks are the new unpermuted ranks. Next we shift them by one position with the last rank becoming the first. The tournament size $k \in \mathbb{N}, k > 1$ can be set larger than 2. This way the best network is included k times in the sampling set.

The GA can be divided in an embarrassingly parallel problem. The scoring of the networks in step 2 of the GA (3) is independent. Thus the GA can be parallelized on up to N threads for population size N. We can further parallelize the creation of the probability distribution (algorithm 4 , step 4 and algorithm 5, step 3).

4

Boolean Nested Effects Models

Parts of this chapter have been published (Pirkl *et al.* (2016)).

4.1 Complex and Alternative signalling

It is crucial to understand signalling pathways of cells to battle diseases like cancer (Dufour & Clavien (2005)). Current computational methods are not general enough to reach a high resolution from steady state experiments. We aim to use these experiments to increase pathway resolution and resolve complex signalling structures.

4.1.1 Combinatorial signalling

Signals in a pathway are not just transduced in a straight line from protein A to B to C and so on. Proteins are activated and repressed in many different ways. For instance two or more proteins can form a complex to activate another protein. This complex is depending on all members to function correctly. If one part of the complex is broken or missing, the signal will not be propagated anymore. In contrast during alternative signalling several different pathway members can propagate the signal independently from each other to the target protein. If one of them is broken or missing, the others will cover for it. This redundancy makes the whole signalling pathway more robust against perturbation. These two phenomena can be combined to build arbitrarily complicated signalling pathways.

Boolean networks have been successfully used to model complex formation and alternative signalling (Klamt *et al.* (2006, 2007); Saez-Rodriguez *et al.* (2009, 2011)). The complex formation corresponds to the logical AND-gate and the alternative signalling to the logical OR-gate. Furthermore, the logical NOT-gate is used to model the repression or inhibition of one protein by another.

Any combination of AND-,OR- and NOT-gates is possible to describe the signal transduction. An increasing number of proteins leads to an increasing number of combinations. The number of combinations defines the size of the search space. Additionally, there are

combinations which describe different pathways, but the same Boolean logic. For instance if A activates C either in a complex with B or alone, the corresponding Boolean network as DNF is

$$C = A \vee (A \wedge B) = A.$$

The complex formation of A and B cannot be resolved. This equivalence between different networks is also dependent on the experimental design. If only single knock-downs are available, the two networks

$$G_1 = \{A = S, B = S, C = A \vee B\} \text{ and } G_2 = \{A = S, B = S, C = S\}$$

are equivalent, because the signal from S to C is not blocked by a single knock-down of A or B. A double knock-down on the other hand inhibits A and B and subsequently also C.

Single knock-downs:
$G_1 : C = A \vee B = 0 \vee S = 0 \vee 1 = 1$
$G_1 : C = A \vee B = S \vee 0 = 1 \vee 0 = 1$
$G_2 : C = S = 1$
Double knock-down:
$G_1 : C = A \vee B = 0 \vee 0 = 0$
$G_2 : C = S = 1$

The size of the search space and equivalence classes are the two main challenges. Technologies will become cheaper. Thus a lot more experiments with different combinatorial perturbations can be conducted and the size of equivalence classes will be reduced. Furthermore we use prior knowledge to exclude network hypothesis beforehand, which also reduces the size of equivalence classes and the search space. Then we traverse the search space with a combination of efficient search heuristics.

4.2 Pathway model and score

4.2.1 Signalling Pathways and Deterministic Boolean Networks

Molecular signalling pathways can be described as Deterministic Boolean Networks (Saez-Rodriguez *et al.* (2009)). Networks are encoded as directed acyclic hyper-graphs $\Psi = (S, H)$ consisting of a set of nodes $S = (S_1, \ldots, S_N)$ and a set of Hyper-edges $H = (H_1, \ldots H_M)$. Every node S_i represents a signalling protein that can be either active ($S_i = 1$) or inactive ($S_i = 0$). Hyper-edges describe how the signal is propagated through the network. Every directed hyper-edge H_j connects one or more parent nodes with a single child node. Hyper-edges with one parent node specify whether the child is activated or repressed by its parent. Hyper-edges with more parents specify a unique activation pattern of the parent nodes that is required for activating the child. If a node has multiple incoming hyper-edges, it can be independently activated by all of them. Hence, every hyper-edge with more than one parent node encodes an AND gate and multiple hyper-edges with the same child form OR gates (Figure 4.1). Signalling pathways form AND

gates, if multiple proteins need to be jointly activated to propagate the signal to their target molecule. This is often associated with the formation of larger protein complexes. OR gates in contrast occur when signalling is organized in a redundant manner. As with Bayesian networks and NEMs, we assume that the real graph is acyclic. This limits the scope of the method to models of signalling pathways in which the signal is propagated from receptors via branching cytosolic effector pathways into the nucleus without feedback loops.

4.2.2 Experimental design and data

Our goal is to estimate the signalling pathway model Ψ from a dataset D. The data consist of K gene expression profiles $(D(1), \ldots, D(K))$ from a set of functional assays with specific perturbations of the pathway. We assume that the expression data is on a logarithmic scale. Perturbations include the exogenous stimulation of pathway receptors and the inhibition of signalling components. Every profile $D(k)$ is hence associated with a specific experimental condition $C(k)$ that specifies which receptors were stimulated and which signalling genes were inhibited. This is the typical experimental set-up of NEMs (Markowetz *et al.* (2005, 2007)). Following the NEM literature, we call the signalling pathway components $S_1, \ldots S_n$ S-Genes and the genes that show expression changes in response to perturbations E-Genes. S-genes and E-genes can but need not overlap.

4.2.3 Expected and Observed Response Schemes

For a given hyper-graph Ψ and a given condition $C(k)$, we can calculate the activation states of all nodes in Ψ as follows: (i) root nodes are initialized to zero, (ii) stimulated nodes are set to 1 and inhibited nodes are set to 0 independently of any incoming signals from parent nodes, (iii) all other nodes are determined by propagating activation states through the directed acyclic graph using the Boolean functions defined by the hyper-edges of Ψ. Let $\mathbf{C} = (C(1), ..., C(K))$ be the set of all experimental conditions, and $A \subset \mathbf{C} \times \mathbf{C}$ a set of comparisons between pairs of conditions. For every pair of conditions $i = (C(k), C(l)) \in \mathbf{A}$ we can determine, whether the expected activation of an S-gene is identical under both conditions or not. We set $\Phi_{ij} = 0$, if the predicted state of S_j is identical under C_k and C_l. We set $\Phi_{ij} = 1$, if S_j was switched on, i.e if it is inactive under C_k but active under C_l and we set $\Phi_{ij} = -1$, if S_j was switched off. We call Φ the Expected Response Scheme (ERS) of Ψ (Figure 4.1, middle).
Analogously, we organize the observed responses. For a given E-gene E_j, let $\Delta_{i,j}$ be the expression change of E_j in comparison i. We call Δ the Observed Response Scheme (ORS).

4.2.4 Scoring hyper-graphs

For a given hyper-graph Ψ, we want to score how well its expected responses Φ match the observed responses Δ. This cannot be done directly, because Φ refers to activation states of S-genes, while Δ refers to downstream effects in E-genes. Following the NEM literature (Markowetz *et al.* (2005); Tresch & Markowetz (2008)), we assign E-genes to S-genes. For every E-gene E we search the S-gene for which the expected responses $\Phi(S)$

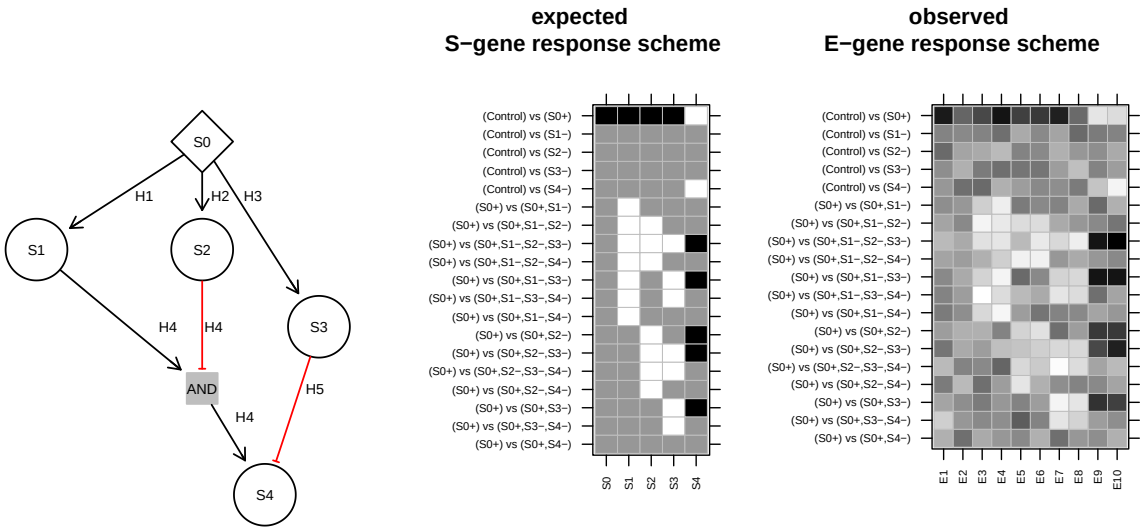


Figure 4.1: **Hyper-graphs and their response schemes** The two matrices are an expected S-gene response scheme of the S-genes and a hypothetical noisy continuous observed E-gene response scheme of attached E-genes for the hyper-graph left. Black matrix entries indicate up-regulation (+1), white down-regulation (-1) and gray no change (0). Each column is a response scheme of an S-gene respectively E-gene. The rows are comparisons between two conditions. In a condition + denotes the activation of the S-gene and − the inhibition independent of the state of the parents. The set of modelled comparisons is restricted to the typical design of a nested effect model. Included are comparisons of stimulation vs. control and stimulations + inhibitions vs. stimulations only. S0 is a receptor that can be activated. The other S-genes propagate the signal and can be inhibited. The edge $H4$ is an AND gate with two parents. S4 is activated by H4, if S1 is active and S2 inactive. Alternatively, the inhibition of S3 can activate S4. Hence $H4$ and $H5$ implicitly form an OR gate.

match observed responses $\Delta(E)$ best. We quantify this match by an association score $\mathcal{A}$ between expected and the observed responses considering negative regulation. That means the fit between S-gene and $-\Delta(E)$. If not said otherwise we use the Spearman rank correlation $\mathcal{A} = \rho$. Finally, we score the hyper-graph by balancing its data fit with its size:

$$\begin{aligned}\mathcal{L}(\Psi) = & \frac{1}{m} \cdot \sum_{E} \max_{S} \{\mathcal{A}(\Phi(S), \Delta(E)), \mathcal{A}(\Phi(S), -\Delta(E))\} \\ & - \zeta \cdot \frac{1}{M} \cdot \sum_{H \in \Psi} \#pa(H)\end{aligned} \tag{4.1}$$

The first sum runs over all E-genes and the second sum runs over all hyper-edges in Ψ. $\#pa(H)$ is the number of parent nodes of hyper-edge H. $\zeta > 0$ is a parameter to calibrate the penalty for network size. The network size penalty is identical to that used in Saez-Rodriguez *et al.* (2009). m is the number of E-genes used in the score and M is the maximal network size possible. This way the score normalizes to $[0, 1]$ and the size penalty to $[0, \zeta]$. This makes ζ independent of the number of E-genes or the overall size of the fully connected network.

4.2.5 Assigning E-genes to S-genes

It is interesting to know, which S-genes influences certain E-genes the most. For example the expected response scheme of NfκB should be most similar to the observed response schemes of known NFκB targets. Thus we calculate the subnetwork of pairwise regulation of E-genes by S-genes.

$\Theta = \{\theta_{i,j} : \theta_{i,j} \in \{-1, 0, 1\}\}$ describes the type of regulation (negative, no, positive) of every E-gene j by every S-gene i. A posteriori we attach every E-gene to the S-gene it is most similar to by setting

$$\theta_{i,j} = \begin{cases} +1 :\Leftrightarrow \max\limits_{(S_k)_k} \{\mathcal{A}(\Phi(S_k), \Delta(E_j)), \mathcal{A}(\Phi(S_k), -\Delta(E_j))\} = \mathcal{A}(\Phi(S_i), \Delta(E_j)) \\ -1 :\Leftrightarrow \max\limits_{(S_k)_k} \{\mathcal{A}(\Phi(S_k), \Delta(E_j)), \mathcal{A}(\Phi(S_k), -\Delta(E_j))\} = \mathcal{A}(\Phi(S_i), -\Delta(E_j)) \\ 0 : \text{ otherwise} \end{cases} .$$

In other words if E-gene j's ORS is most similar to S-gene i's ERS, S_i is defined as the direct regulator of E_j and $\theta_{i,j} = \pm 1$. Otherwise we set $\theta_{k,j} = 0$. If $\Phi(S_i)$ is most similar to the actual foldchanges $\Delta(E_j)$, the sign is positive and if S_i is most similar to $-\Delta(E_j)$, it is negative.

One E-gene can be ambiguously attached to several S-genes, if the association scores for these S-genes are all equal. We can also use prior information to a priori attache E-genes to S-genes or exclude some S-genes as possible regulators for a set of E-genes.

4.2.6 Model adaptive discretization score

The original NEM infers networks from discretized data. It is not trival to find the correct cutoffs for down regulation (-1), up-regulation (1) and no regulation (0). Thus we define a more flexible score which discretizes the data dependent on the ERS.

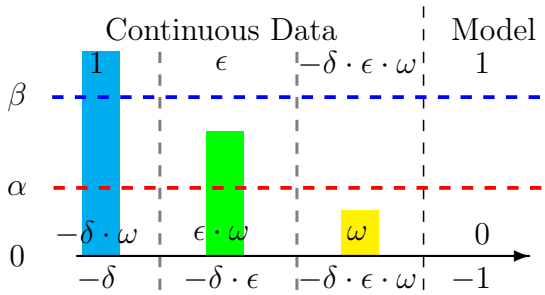


Figure 4.2: **Example: model adaptive discretization.** The blue foldchange larger than β is always discretized as a 1 and either rewarded, if the model predicts a 1 or penalized otherwise. The yellow foldchange is smaller than α and rewarded, if the model predicts a 0 and penalized otherwise. The green foldchange (adaptive effects) between α and β is both rewarded if the model predicts a 1 or a 0 and only penalized if the predicted effect has a different sign. ω is a weight parameter for predicted and observed 0s. ϵ is the weight parameter for the adaptive effects.

For each pair of experiments $l \in \mathbf{A}$ we define the model adaptive discretization ($\mathcal{MAD}$, (4.2)). Let Φ_{il} be the expected response of S_i for l and Δ_{jl} the observed response of E_j. We define

$$\mathcal{MAD}\left(\Phi_{il}, \Delta_{il}\right) =$$

$\Delta_{jl} \backslash^{\Phi_{il}}$	$= -1$	$= 0$	$= 1$
$< -\beta$	1	$-\delta \cdot \omega$	$-\delta$
$\in [-\beta, -\alpha)$	$1 \cdot \epsilon$	$1 \cdot \epsilon \cdot \omega$	$-\delta \cdot \epsilon$
$\in [-\alpha, \alpha]$	$-\delta \cdot \epsilon \cdot \omega$	$1 \cdot \omega$	$-\delta \cdot \epsilon \cdot \omega$
$\in (\alpha, \beta]$	$-\delta \cdot \epsilon$	$1 \cdot \epsilon \cdot \omega$	$1 \cdot \epsilon$
$> \beta$	$-\delta$	$-\delta \cdot \omega$	1

. (4.2)

$\alpha, \beta \in \mathbb{R}, \alpha \leq \beta$ are two cutoffs for the observed responses. If Δ_{jl} is greater than β it is always discretized as a positive (1) respectively lesser than $-\beta$ negative (-1) effect. It is rewarded with 1, if it is in conformity to the predicted state of S_i and penalized with $-\delta$, if it is not. If a foldchange is smaller than $|\alpha|$, it is always discretized as no effect (0) and rewarded respectively penalized the same way. If a foldchange is in the transition phase between β and α respectively $-\beta$ and $-\alpha$, it depends on the model prediction. If the model predicts an effect for the S-gene, the foldchange is also discretized as an effect and rewarded including a weight factor ϵ. If the model predicts no effect the E-gene is discretized as no effect. ω is a weight factor for predicted and observed zeros (no effects). The parameters ω and ϵ are similar to false positive and false negative rates. $\epsilon < 1$: the further the E-gene differs from the predicted S-gene value the smaller the score. $\omega < 1$: false negatives are more likely than false positives. An example is shown in figure 4.2.

Based on the $\mathcal{MAD}$ score we define a new association score:

$$\mathcal{A}\left(\Phi(S_i), \Delta(E_j)\right) = \frac{1}{M} \sum_{(l \in \mathbf{A})} \mathcal{MAD}\left(\Phi_{il}, \Delta_{jl}\right).$$

$M \in \mathbb{N}$ is the maximal number of comparisons made. For example $M = 19$ in figure 4.1.

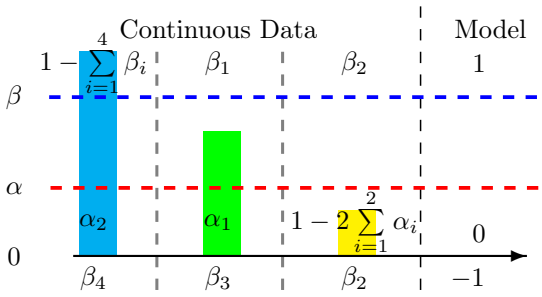


Figure 4.3: **Conditional probability.** Toy example for the conditional probability of (4.3). The figure is analog to figure 4.2 for the $\mathcal{MAD}$ score.

4.2.7 Marginal Likelihood Formulation

Similar to the $\mathcal{MAD}$ score, we can extend the marginal likelihood in Markowetz *et al.* (2005). Let Φ be the expected response scheme, Δ_{il} the observed response of E_i to the pair of experiments $l \in A$ and Θ the regulation of E-genes by S-genes. We define the extended conditional probability by

$$P\left(\Delta_{il} \,|l \in \mathbf{A}, \Phi, \theta_{i,j} = 1\right) =$$

$\Delta_{il} > \beta$	$\beta \geq \Delta_{il} > \alpha$	$\alpha \geq \Delta_{il} \geq -\alpha$	$-\alpha > \Delta_{il} \geq -\beta$	$-\beta > \Delta_{il}$	
$1-\sum_{i=1}^{4}\beta_i$	β_1	β_2	β_3	β_4	if Φ predicts *positive effect*
α_2	α_1	$1-2\sum_{i=1}^{2}\alpha_i$	α_1	α_2	if Φ predicts *no effect*
β_4	β_3	β_2	β_1	$1-\sum_{i=1}^{4}\beta_i$	if Φ predicts *negative effect*

. (4.3)

α_1, α_2 are false positive and $\beta_1, \beta_2, \beta_3, \beta_4$ false negative rates. We assume symmetrically distributed error rates. We can further simplify, if we assume a dependence between the betas respectively alphas, like $\beta_i = \frac{\beta_1}{i},\ \forall\ i \in \{1,2,3,4\}$ and $\alpha_2 = \frac{\alpha_1}{2}$.

Analog to (3.8), we can write the marginal likelihood as

$$P\left(D\,|\mathbf{A}, \Phi\right) = \frac{1}{n^m} \prod_{i=1}^{m} \sum_{j=1}^{n} \prod_{k=1}^{l} P\left(\Delta_{ik}\,|k \in \mathbf{A}, \Phi, \theta_{i,j} = 1\right) \tag{4.4}$$

If we take the absolute of the observed responses Δ_{il} and set the cutoffs $\alpha = \beta$, (4.3) is the same conditional probabilities as (3.1) in the original NEM.

4.2.8 Other Similarity Measures

There are several other measures for similarity, which can be used in equations (4.1) and (4.5). Cosine similarity (Sidorov *et al.* (2014)) makes the score scale invariant. Different types of correlation (Fahrmeir *et al.* (2007)) add shift invariance. We also can think of the E-gene to S-gene relationship as a classification problem. We have n S-genes with n different labels and m unlabeled E-genes. Supervised learning methods (e.g. support vector machines, Chang & Lin (2011); Vapnik (1998)) find the best fitting S-gene for every E-gene and can also calculate a probability as similarity measure.

4.2.9 Automatic E-gene Selection

The previously defined score in equation (4.1) uses every E-gene regardless of goodness of fit. However if an E-gene has no high similarity to any S-gene, it might not be involved in the pathway. Therefore we add an automatic selection process to the algorithm similar to the extra vertex introduced in Tresch & Markowetz (2008). In a filtering step the score includes only E-genes, which exceed a certain threshold of similarity (fitness) to at least one S-gene. For this we change the score from (4.1) to

$$\begin{aligned} \mathcal{L}(\Psi) = & \frac{1}{m} \cdot \sum_{U} \max_{S} \{\mathcal{A}(\Phi(S), \Delta(E)), \mathcal{A}(\Phi(S), -\Delta(E))\} \\ & - \zeta \cdot \frac{1}{M} \cdot \sum_{H \in \Psi} \#pa(H), \\ U = & \Big\{ E : \max_{S} \{\mathcal{A}(\Phi(S), \Delta(E)), \mathcal{A}(\Phi(S), -\Delta(E))\} > \gamma \Big\}. \end{aligned} \tag{4.5}$$

$\gamma \in [0, 1)$ is a cutoff for the minimal fitness. Note that the normalization factor m is independent of γ. Thus a network fitting only one E-gene with a fitness of 1 has the same score as an equal sized network fitting two E-genes with a fitness of 0.5 each.

4.2.10 Local residuals

Even though we give each network candidate a score ((4.1),(4.5)) we do not know exactly how well it explains the data. The network might explain parts of the data perfectly and other parts not at all.

We can check how well a network fits the data by directly comparing the observed response scheme with the expected response scheme. However if we use a scale invariant association score (e.g. correlation), the effects in the ORS might be too small to see with the naked eye. If we have a high number of E-genes, it is also hard to visualize the data. For that reason we introduce an approach to visualize local mismatches (residuals) between ORS and ERS independently of effect strength and number of E-genes. The residuals solely depend on the score. Thus two different scores can produce different residuals. For example Pearson's correlation is shift invariant, but the euclidean distance is not. Let the relation between the ERS ϕ_i of S-gene i and the ORS $\Delta(E_j)$ of E-gene j be $\phi_i = \Delta(E_j) + c$ with a constant c. Thus if we use correlation, it's a perfect fit. If we use distance we get a residual $\phi_i - \Delta(E_j) = c$.

We calculate residuals between a candidate network's ERS and the ORS in the following way. We manipulate a single entry of the ERS and score it again. If the score improves, the network has a residual at the position we changed. For example our network predicts $\phi_{il} = 1$ for S-gene i and the pair of experiments $l \in \mathbf{A}$. We do not change the network but the ERS: $\phi_{il} = 0$ or $\phi_{il} = -1$. If the new score improves, the E-genes' ORS prefers a 0 respectively -1 instead of the network predicted 1.

Let $(E_k)_{k \in \{1,\cdots,l\}}$ be the set of E-genes assigned to S-gene S_i ($\theta_{i,k} = \pm 1$). We calculate

the local fitness score $\mathcal{F}$ for S-gene S_i the following way:

$$\begin{aligned}\mathcal{F}(S_i) &= \sum_{k=1}^{l} \max\left\{\mathcal{A}(\Phi(S_i), \Delta(E_k)), \mathcal{A}(\Phi(S_i), -\Delta(E_k))\right\} \\ &= \sum_{E} |\phi_{i,E}| \max\left\{\mathcal{A}(\Phi(S_i), \Delta(E)), \mathcal{A}(\Phi(S_i), -\Delta(E))\right\}.\end{aligned}$$

We change a single effect Φ_{il} for a pair of experiments $l \in \mathbf{A}$ and S-gene S_i in the ERS. This results in two locally changed ERSs.

$$\Phi^{+}_{ix} = \begin{cases} \Phi_{ix} & : x \neq l \\ 1 & : \Phi_{ix} \in \{0, -1\}, x = l \\ 0 & : \Phi_{ix} = 1, x = l \end{cases}$$

$$\Phi^{-}_{ix} = \begin{cases} \Phi_{ix} & : x \neq l \\ -1 & : \Phi_{ix} \in \{0, 1\}, x = l \\ 0 & : \Phi_{ix} = -1, x = l \end{cases}.$$

In other words we change Φ^{+}_{il} to 1, if it is negative (-1) or zero and to zero, if it positive (1). We change Φ^{-}_{il} to -1, if it is positive or zero and to zero, if it negative (-1). Then we calculate the new local fitness scores for S_i:

$$\mathcal{F}^{+}_{l}(S_i) = \sum_{k=1}^{l} \max\left\{\mathcal{A}(\Phi^{+}(S_i), \Delta(E_k)), \mathcal{A}(\Phi^{+}(S_i), -\Delta(E_k))\right\}$$

$$\mathcal{F}^{-}_{l}(S_i) = \sum_{k=1}^{l} \max\left\{\mathcal{A}(\Phi^{-}(S_i), \Delta(E_k)), \mathcal{A}(\Phi^{-}(S_i), -\Delta(E_k))\right\}$$

If either $\mathcal{F}(S_i) < \mathcal{F}^{+}_{l}(S_i)$ or $\mathcal{F}(S_i) < \mathcal{F}^{-}_{l}(S_i)$, S_i has a residual at position l. $\mathcal{F}^{+} = \left(f^{+}_{il} = \mathcal{F}^{+}_{l}(S_i) - \mathcal{F}(S_i)\right)$ and $\mathcal{F}^{-} = \left(f^{-}_{il} = \mathcal{F}^{-}_{l}(S_i) - \mathcal{F}(S_i)\right)$ is the positive (PRM) respectively negative residuals matrix (NRM). A positive matrix entry denotes the improvement of the fitness by a local change in the ERS. This helps to identify parts of the data which are explained by our optimal network and parts that aren't.

In practice we set all negative matrix values to zero, because changes at these positions do not improve the score. Then we multiply all matrix entries with -1 which have been changed to zero. Figure 4.4 shows an example for a PRM.

4.3 Optimization

Efficient optimization is important for the learning of Boolean networks. Apart from a fast and efficient search algorithm, the design of the experiments has also consequences on the identifiability of the optimal network.

4.3.1 Network Equivalence

Like Bayesian networks and standard NEMs, B-NEMs are affected by likelihood/score equivalence. It is possible that two different networks have the same ERS Φ. If in addition

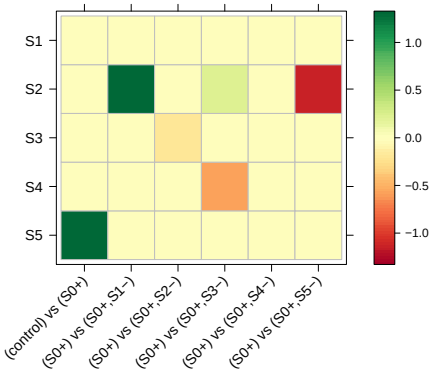


Figure 4.4: **Hypothetical positive residuals matrix.** Columns are pairs of experiments $l \in \mathbf{A}$. Rows are S-genes. Non-zero values indicate a score improvement. Green: the ERS has been changed from 0 or -1 to 1. Red: the ERS has been changed from 1 to 0. For example the E-genes fit better if a network predicts a positive effect for S-gene $S5$ for the pair of experiments "(control) vs (S0+)". They fit worse, if a network predicts no effect instead of a positive effect for S-gene $S2$ at position "(S0+) vs (S0+,S5-)". The E-genes regulated by $S1$ do not produce any residuals. The pair "(S0+) vs (S0+,S5-)" does not produce any residuals either.

the networks have identical size, they yield identical scores no matter what the data looks like. If not the smaller network is chosen. Note that Φ depends on the design of the set of perturbation assays $\mathbf{C}$. Two networks can be distinguished by one experimental design but not by another. Figures 4.5 and 4.6 give examples, how the design can (i) affect score equivalence classes and (ii) affect the optimal scoring network. Interestingly, in figure 4.5 we need an experiment involving only $S3$ during stimulation of $S0$ to correctly identify the signalling logic of S4. In figure 4.6 the double knock-down of $S1$ and $S2$ resolves the equivalence.

The following definitions formally introduce equal networks and (minimal) equivalence classes in the context of B-NEM.

Definition 4.1 (network equality)**.** *Two networks* $\Psi_1 = (S_i)_{i\in I}$ *and* $\Psi_2 = (S_i)_{i\in I}$ *are identical or equal, if and only if both define equal* Boolean functions for every* $S_i, i \in I$*. In this case we write* $\Psi_1 = \Psi_2$*.*
**) Two Boolean functions* f *and* g *are equal, if* $f = g$ *by the rules* (2.3)−(2.21)*.*

For two networks $\Psi_1 = (S_i)_{i\in J_1 \subset I}$ and $\Psi_2 = (S_i)_{i\in J_2 \subset I}$ with an overlapping set of S-genes ($J_1 \cap J_2 \neq \emptyset$) and a set of conditions $\mathbf{C}$, we write $S_{i|\Psi_j,k}$ for the state of S-gene S_i given its Boolean function in Ψ_j and the condition $k \in \mathbf{C}$.

Definition 4.2 (network equivalence)**.** *Two networks* $\Psi_1 = (S_i)_{i\in I}$ *and* $\Psi_2 = (S_i)_{i\in I}$ *are equivalent for a given set of conditions* $\mathbf{C}$*, if and only if* $S_{i|\Psi_1,k} = S_{i|\Psi_2,k}$ $\forall i \in I, k \in \mathbf{C}$*. For the equivalence class of a network* Ψ *we write* $[\Psi]$*.*

A similar definition of equivalence as above but for NEMs was already introduced in Tresch & Markowetz (2008).

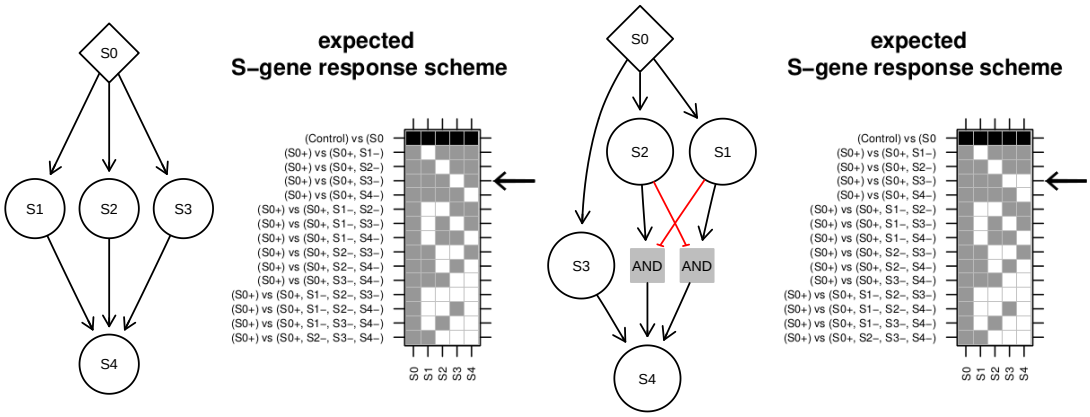


Figure 4.5: **Missing single knock-downs.** The response schemes of the two networks differ only for the experiment marked by the arrow. If that experiment was missing the response schemes would be identical and the left network would score higher due to its smaller size no matter what the data looks like.

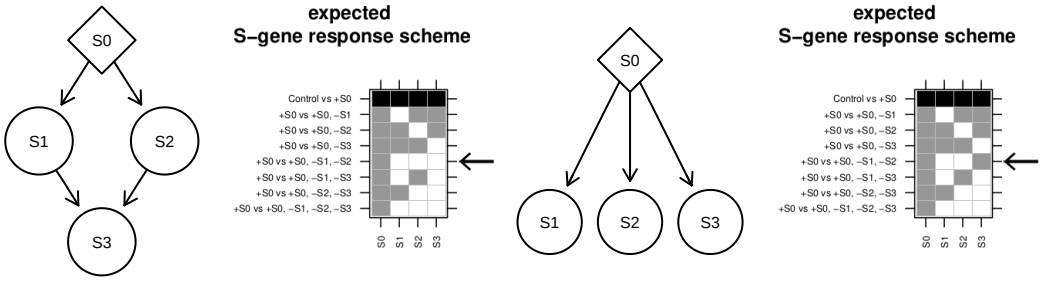


Figure 4.6: **Missing double knock-downs.** The response schemes of the two networks differ only for the experiment marked by the arrow. If that experiment was missing the response schemes would be identical and the right network would score higher due to its smaller size no matter what the data looks like.

Definition 4.3 (minimal equivalence class)**.** *We call* $[\Psi]$ *minimal, if and only if for every pair* $(\Psi_1, \Psi_2) \in [\Psi] \times [\Psi]$ *it follows, that* $\Psi_1 = \Psi_2$*.*

The following definition introduces the concept of standard experiments. These experiments are a generalization of the usual set of experiments used by NEM. NEM is applied to data derived from experiments consisting of positive (stimulation) and negative control, and single knock-downs during the stimulation.

Definition 4.4 (standard experiments)**.** *Experiments in which a subset of S-gene(s) is stimulated and a subset of S-genes knocked down (inhibited) during stimulation are called "standard", if and only if*

- *the inhibited S-genes are downstream of the stimulated S-genes*
- *the stimulated S-genes have no parents*

Stimulations are denoted by a + (stimulation(s)+), knock-downs by a − (S-gene(s)−) and we call the negative control just control. The standard set of contrasts is:

- *(Control) vs (stimulation(s)+)*
- *(stimulation(s)+) vs (stimulation(s)+,S-gene(s)−)*

Restricting the search space using prior knowledge

If the data can not distinguish between competing networks it is still possible that existing domain knowledge can. Like Saez-Rodriguez *et al.* (2009) we represent pathway knowledge by a priori restrictions of the network search space. With this restrictions we do not only reduce network ambiguity due to score equivalence, but also ensure that the constructed networks follow general conventions of modelling signalling pathways (e.g. the signal is propagated from receptors, via cytosolic molecules to nuclear factors). We encode prior knowledge by a directed graph G whose edges are a collection of all links between S-genes that are a priori possible. In other words, it is the missing edges of G that define the search space restriction. We refer to G as a PKN. PKNs are then extended to a Boolean network by adding hyper-edges such that all Boolean functions allowed in G are a priori possible (figure 4.7). Hence, while B-NEM use help from prior knowledge to estimate the network structure or topology they infer logical gates only from data. Using prior knowledge can resolve score equivalence problems, but there is no guarantee that it always does.

A priori set of experiments

Akutsu *et al.* (2003) already have done theoretical work for upper and lower bounds of the number of experiments necessary to resolve the boolean functions of an underlying genetic network. We give an example for a PKN with two S-genes directly regulating a third and the type of perturbations necessary to resolve minimal equivalence classes.

Figure 4.7 shows a simple four S-gene example for a prior network (A) and its extension (B). $S0$ is the stimulation and $S1, S2$ and $S3$ are potential knock-down targets. Graphs 1-5 in figure 4.7 are five subgraphs of graph B representing the minimal equivalence

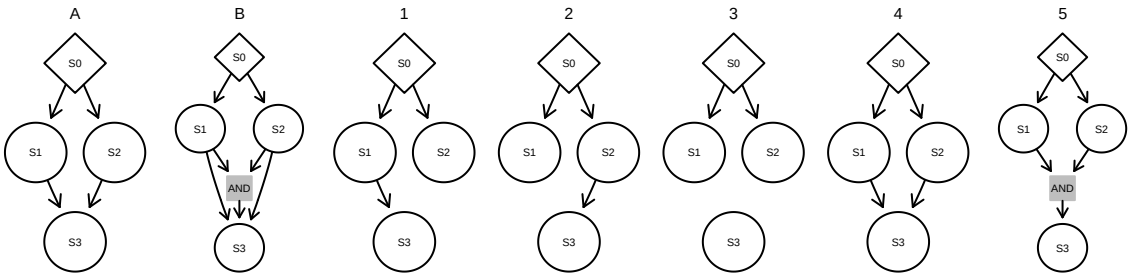


Figure 4.7: **PKN extension and search space.** Prior graph (A) and its extension (B). Equivalence classes representing all possible networks regulating S3 (1-5).

classes. We show that a standard set of experiments with single knock-downs is sufficient to resolve all five classes.

In the following enumerations, when we write regulated by Sn, we mean the stimulation or knock-down and not the S-gene Sn. For example if the S-gene $S1$ positively regulates $S2$ (i.e. $S1 \rightarrow S2$), the knock-down of $S1$ down-regulates $S2$ during the stimulation.

1. $S3$ is up-regulated by the $S0$ stimulation and down-regulated by the $S1$ knock-down, but not the $S2$ knock-down
2. $S3$ is up-regulated by $S0$ and down-regulated by $S2$, but not $S1$
3. $S3$ is not regulated by $S0$ and not regulated by either
4. $S3$ is up-regulated by $S0$ and not regulated by either
5. $S3$ is up-regulated by $S0$ and down-regulated by both

We show an example without feed forward loops, because they are difficult to infer with standard experiments. Figure 4.8 shows an example. The stimulatory signal to $S3$ in networks A and B can be blocked with single knock-downs of $S1$ and $S2$. Thus A and B are equivalent ($[A] = [B]$) for standard experiments. An additional experiment with the stimulation (knock-in) of $S2$ and a knock-down of $S1$ resolves the equivalence. The $S2$ knock-in makes $S3$ independent of $S1$ in network A, but not in B (equation (4.6)). In network B, $S3$ needs active $S1$ to be active itself.

$$\begin{aligned} A &: S3 = S2 = 1 \\ B &: S3 = S1 \wedge S2 = 0 \wedge 1 = 0 \end{aligned} \tag{4.6}$$

The OR feed forward loop (D) is similar. The signal can be blocked by a knock-down of $S1$, but not by $S2$. Thus the networks C and D are equivalent. Once more the single knock-in of $S2$ combined with the knock-down of $S1$ resolves the equivalence (equation (4.7)).

$$\begin{aligned} C &: S3 = S1 = 0 \\ D &: S3 = S1 \vee S2 = 0 \vee 1 = 1 \end{aligned} \tag{4.7}$$

Negative regulation increases the complexity of the search space in example 4.7. If $S1$ and $S2$ have the potential to negatively regulate $S3$ ($S1 \dashv S3$, $S2 \dashv S3$), the number of minimal equivalence classes is more than three times as high (figure 4.9).

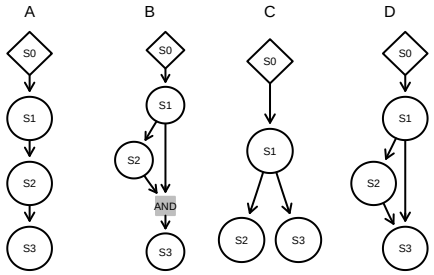


Figure 4.8: **AND-gate feed forward loop.** (A) The signal to S3 can be blocked by S1 or S2. A knock-in of S2 makes S3 independent of S1. (B) The signal can be blocked by S1 or S2. A knock-in of S2 does only activate S3, if S1 is active, too. (C) Only S1 activates S3. (D) A knock-in of S2 activates S3 independently of S1.

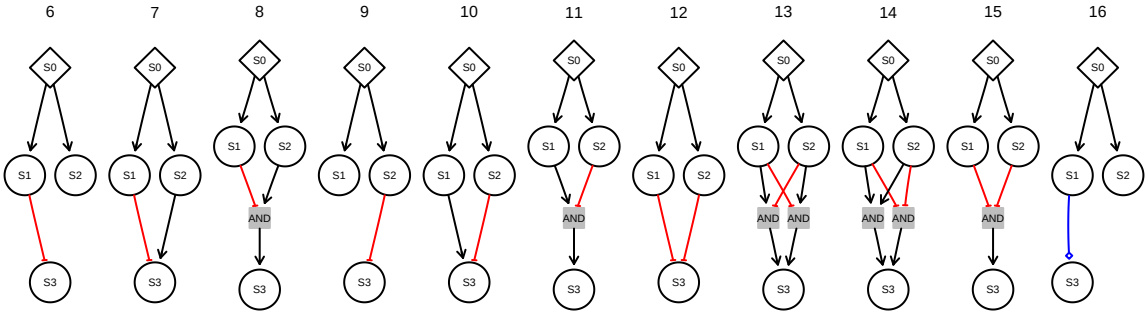


Figure 4.9: **Search space including negative edges.** Minimal equivalence classes representing all possible networks with negative edges into S3. The blue edge on the right denots ambiguous regulation, which means S3 is always active and only down-regulated by its own knock-down.

6. $S3$ is down-regulated by the $S0$ stimulation and up-regulated by the $S1$ knock-down, but not the $S2$ knock-down
7. $S3$ is not regulated by $S0$ and down-regulated by $S2$, but not $S1$
8. $S3$ is not regulated by $S0$ and up-regulated by $S1$, but not $S2$
9. $S3$ is down-regulated by $S0$ and up-regulated by the $S2$, but not $S1$
10. $S3$ is not regulated by $S0$ and down-regulated by $S1$, but not $S2$
11. $S3$ is not regulated by $S0$ and up-regulated by $S2$, but not $S1$
12. $S3$ is down-regulated by $S0$ and up-regulated by both
13. $S3$ is not regulated by $S0$ and up-regulated by both
14. $S3$ is not regulated by $S0$ and down-regulated by both
15. $S3$ is down-regulated by $S0$ and not regulated by either
16. $S3$ is not regulated by $S0$ and not regulated by either

Classes 3 and 16 look liek they have the same response scheme. However in class 3 $S3$ shows no response to its knock-down, but in class 16 $S3$.

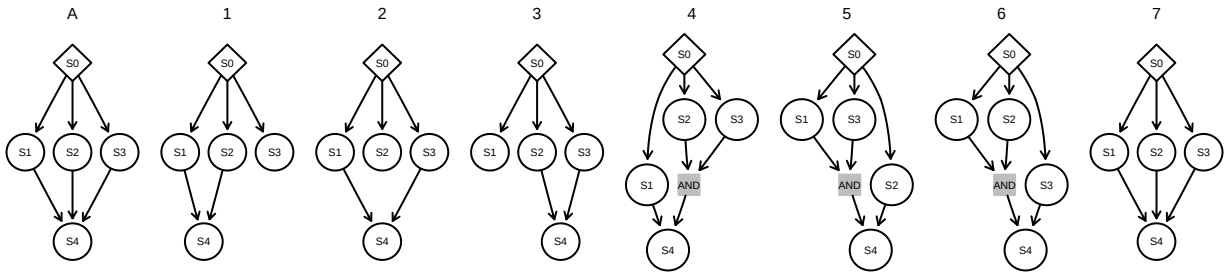
Contrary to the CNO, B-NEM uses indirect measurements (E-genes). E-genes can be positively or negatively regulated by S-genes. Therefore the equivalence classes are not minimal anymore. For instance an E-gene E which is positively regulated by $S3$ in class 1 ($S3 \rightarrow E$) fits perfectly to class 6, because there it is assumed to be negatively regulated ($S3 \dashv E$). This is due to the fact that $S3|_{class\ 1} = \neg\, S3|_{class\ 6}$. This equivalence can be resolved with knock-downs during control. If class 1 is correct, the E-gene shows no effect in the contrast $(Control)\ vs\ (S3-)$. If class 6 is correct, it is up-regulated (> 0).

We have not tackled the general case, but a simple example with two regulatory parents. This example can trivially be scaled up to n parents. However it becomes obvious, that an a priori set of experiments to minimalise equivalence classes is not always feasible. Thus in the next section we describe an alternative approach.

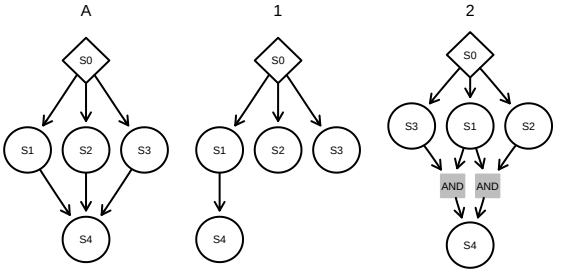
Iterative (a posteriori) set of experiments

Szczurek *et al.* (2009) have introduced the concept of informative experiments to identify redundant experiments. Basically if two different experiments provide the same information, one is redundant and can be omitted. We use the concept of informative experiments to identify those which resolve equivalence classes.

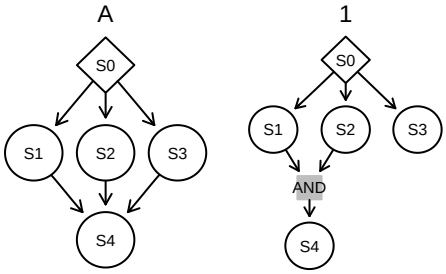
We start with standard experiments with single knock-downs. We use B-NEM to identify high scoring but equivalent networks. For these networks we generate simulated data, including combinatorial perturbations. From the simulated data we deduce the minimal set of additional experiments to resolve equivalences between networks. Three toy examples are shown in figure 4.10. The example in figure 4.10a shows the worst case. If $S4$ is not down-regulated by any single knock-down, we derive seven equivalent networks after the first evaluation with B-NEM. Furthermore three networks have minimal size



(a) A PKN (A) and equivalent networks with minimal size (1-3) and greater size (4-7), given single knock-downs combined with the stimulation of $S0$. The networks are equivalent because $S4$ is not down-regulated by any single knock-down. We can down-regulate $S4$ with one of the three double knock-down in networks 1-3 and two double knock-downs in 4-6. If none of the three double knock-downs regulate $S4$, we infer network 7.



(b) A PKN (A) and two equivalent networks (1-2). The smaller network (1) is preferred. The equivalence comes from the fact, that $S4$ is only down-regulated by the knock-down of $S1$. A double knock-down of S-genes $S2$ and $S3$ resolves the equivalence. If $S4$ is down-regulated by the double knock-down, we infer the 2nd network. If it is not down-regulated, we infer the 1st.



(c) A PKN (A) and one highest scoring network (1). In this case no additional experiments are necessary, since $S4$ is down-regulated by both single knock-downs of $S1$ and $S2$, but not $S3$. Combinatorial knock-downs do not provide additional information.

Figure 4.10: **Iterative experimental design.** Three examples for an iterative approach to experimental designs. Shown are the PKN and highest scoring networks.

and therefore receive the exact same score. Pairwise double knock-downs are necessary to resolve the equivalences. In the example of figure 4.10b $S4$ is down-regulated by the $S1$ knock-down and we only need one additional double knock-down of $S2$ and $S3$. In the best case scenario in figure 4.10c we do not need any additional experiments. $S4$ is down-regulated by the two single knock-downs of $S1$ and $S2$. Thus it would also be down-regulated by all combinatorial knock-downs.

Equivalence due to lack of regulated E-genes

Another aspect which can increases the equivalence classes for B-NEMs is the distribution of E-genes. Let's assume we have the sequential network in figure 4.11 as a GTN and no PKN restriction on the search space. Let's further assume no E-genes are directly regulated by S-gene $S2$. We cannot resolve the GTN, because we have E-genes reacting only to the knock-down of $S1$ and E-genes, which react to all three knock-downs (ERS,

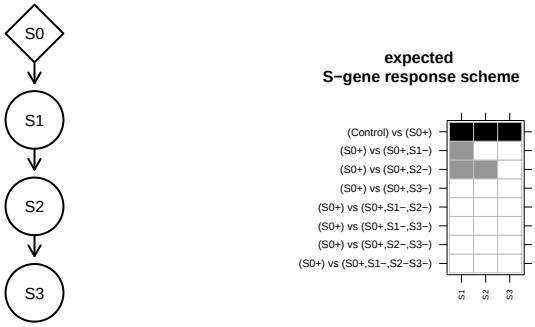


Figure 4.11: **Missing E-genes.** A GTN (left) and its ERS (right).

figure 4.11). That implies that $S1$ is upstream of $S2$ and $S3$. The co-regulated E-genes do not allow for inference on the order of $S2$ and $S3$. However, if we have E-genes directly regulated by $S2$, they are independent of $S3$. Thus $S2$ is placed upstream of $S3$. This has previously been discussed for original NEMs Markowetz *et al.* (2005, 2007)) and HM-NEMs (Wang *et al.* (2014)).

4.3.2 Search Algorithms

B-NEM faces a similar computational challenge as the CNO of Saez-Rodriguez *et al.* (2009). However B-NEM is usually applied to much larger datasets and the of E-gene topology Θ has to be estimated aswell. Thus we implement a modified version of the GA to optimize the model.

Our modifications to the GA include complementary insertion (Louis & Rawlins (1992)), a tournament selection (TS) and non-linear fitness in the SUS. Section 3.3.1 contains details.

(Boolean) Greedy Neighbourhood Search

Besides the genetic algorithm, we also employ a greedy neighbourhood search (GNS) (Cormen *et al.* (2007)).

Algorithm 6 greedy neighbourhood search

1. start with initial network (e.g. random, empty set or fully connected)
2. every edge is evaluated:
   (a) if the current network does not contain the edge, it is added and the new network is scored
   (b) if the current network contains the edge, it is deleted and the new network is scored
3. the edge with the highest score change is then added respectively deleted
4. if there is no improvement stop, otherwise return to step 2

The GNS can be embarrassingly parallelized, since the edge evaluations in step 2 are independent of each other and can be computed simultaneously on N threads, where N is the number of edges.

While this is a fast algorithm, it has one drawback. The search space is not always smooth, but can be contaminated with local optima. The GNS cannot break out of such an optimum. Therefore the greedy search has to be started several times with different initial networks. This increases the probability to find the global optimum. For example lets assume the GTN is

$$\{C = A, D = B, E = C \wedge D\}\,.$$

Let's further assume our starting network is

$$\{C = A, D = B, E = C\}\,.$$

The edge $E = C \wedge D$ does not change the Boolean logic due to edge $E = C$ and the absorption law.

$$E = C \vee (C \wedge D) = C.$$

Thus it does not improve the score. If the greedy algorithm is changed to inspect pairs of edges, this can be resolved. Still there are examples, which demand the inspection of triples and more. Thus we use features of the Boolean functions to more intelligently traverse the search space.

We introduce a third absorption operation (4.8). Instead of removing the clauses which include other clauses (normal absorption), the clauses which are included in other clauses are removed.

$$\begin{aligned} &\text{Inverse absorption:} \\ &x \vee (x \wedge y) \to (x \wedge y) \\ &(x \wedge y) \vee y \to (x \wedge y)\,. \end{aligned} \tag{4.8}$$

The improved GNS is called Boolean Greedy Neighbourhood Search (BGNS).

Algorithm 7 Boolean greedy neighbourhood search

1. start with initial network (e.g. random, empty set or fully connected)
2. every edge is evaluated under the inverse absorption law:
   (a) if the current network does not contain the edge, it is added, inverse absorption is applied and the new network is scored
   (b) if the current network contains the edge, it is deleted and the new network is scored
3. the edge with the highest score change is then either added (inverse absorption is applied) or deleted to get the new network
4. if there is no improvement stop, otherwise return to step 2

The BGNS can resolve the GTN from the previous example. It adds $E = C \wedge D$, simultaneously deletes $E = C$ and scores the new network. In a second example we assume the GTN to be

$$\{C = A \wedge B, D = C\}\,.$$

Let's further assume we start with the network

$$\{D = A \wedge B\}\,.$$

The BGNS can hypothetically resolve the GTN the following way:

1. add $C = A \wedge B$
2. add $D = A \wedge B \wedge C$ (inverse absorption law: delete edge $D = A \wedge B$)
3. add edge $C = D$ (absorption law: delete edge $D = A \wedge B \wedge C$)
4. stop

The GNS algorithm already stops after step 1.

4.4 Nested Effects Models as restricted Boolean Networks

Markowetz *et al.* (2005) already suggested a logical extension of NEMs. Following this notion, we interpret original NEMs as special cases of Boolean graphs.

Let $G = (V, E)$ be the graph of a NEM. A vertice $v \in V$ is 1, if the corresponding protein is inactive. Thus a knock-down of v means $v = 1$. In other words the knock-down has an effect on v. NEM restrict the edges to OR-gates such as $v = (v_1 \vee \ldots \vee v_n)$. Therefore a knock-down of any parent has an effect on v and $v = 1$. The corresponding protein is modelled as inactive.

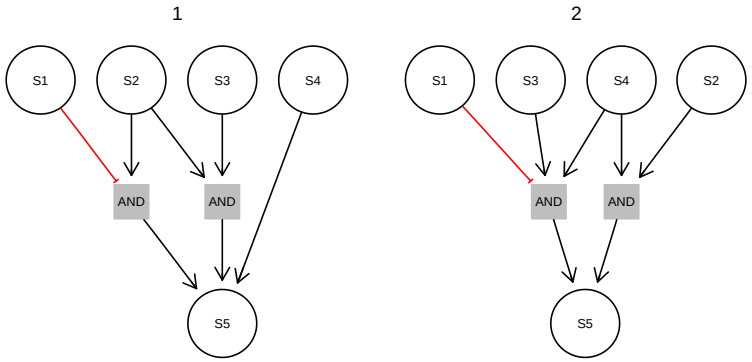


Figure 4.12: **NEM and B-NEM graphs.** Both networks describe the same pathway. The extended NEM (1) models inactive proteins respectively knock-downs as 1. The simplest way to deactivate $S5 = 1$ through its parents is with a knock-down of $S4 = 1$. B-NEM (2) models the active state of a protein respectively a knock-in with 1. The simplest way to activate $S5 = 1$ is to activate $S2 = 1$ and $S4 = 1$. Due to the property of the dual form (2.1), we can use the deduction from the NEM graph for the B-NEM graph. Thus if we knock down $S4 = 0$, we deactivate $S5 = 0$.

B-NEM uses the same Boolean network methodology as Saez-Rodriguez *et al.* (2009). Thus a knock-down of the protein v is modelled with $v = 0$. We can interpret the OR-gate in the NEM case, as an AND-gate in the B-NEM case. For example the NEM OR-gate $C = A \vee B$ is the B-NEM AND-gate $C = A \wedge B$. If we knock down A, it results in a knock-down of C. In the NEM case the knock-down effect (1) is propagated by A to C via the OR-gate. In the B-NEM case the inactivation of A (0) prevents the AND-gate from propagating the signal to C.

$$\text{NEM: } C = A \vee B = 1 \vee B = 1.$$
$$\text{B-NEM: } C = A \wedge B = 0 \wedge B = 0.$$

In general Boolean functions in NEM are the dual forms of the corresponding B-NEM functions.

We further extend NEM by adding the AND-logic and negation (Vaske *et al.* (2009)). For example $C = A \wedge B$ means, the protein C is active ($C = 0$), if we only knock down A or B (1). If we knock down the combination of both, C becomes inactive (1).

In the example of figure 4.12 we start with a graph on the left representing a Nested Effects Model, which we can write as the DNF in (4.9).

$$S_5 = (\neg S_1 \wedge S_2) \vee (S_2 \wedge S_3) \vee S_4 \tag{4.9}$$

We can transform the dual form from its CNF (4.9) into its DNF (4.10) with Boolean

algebra (section 2.1).

$$
\begin{aligned}
S_5^* &= (\neg S_1^* \vee S_2^*) \wedge (S_2^* \vee S_3^*) \wedge S_4^* \\
&\overset{(2.17)}{=} ((\neg S_1^* \wedge S_2^*) \vee (\neg S_1^* \wedge S_3^*) \vee (S_2^* \wedge S_2^*) \vee (S_2^* \wedge S_3^*)) \wedge S_4^* \\
&\overset{(2.17)}{=} (\neg S_1^* \wedge S_2^* \wedge S_4^*) \vee (\neg S_1^* \wedge S_3^* \wedge S_4^*) \vee (S_2^* \wedge S_2^* \wedge S_4^*) \vee (S_2^* \wedge S_3^* \wedge S_4^*) \\
&\overset{(2.12)}{=} (\neg S_1^* \wedge S_2^* \wedge S_4^*) \vee (\neg S_1^* \wedge S_3^* \wedge S_4^*) \vee (S_2^* \wedge S_4^*) \vee (S_2^* \wedge S_3^* \wedge S_4^*) \\
&\overset{(2.18)}{=} (\neg S_1^* \wedge S_3^* \wedge S_4^*) \vee (S_2^* \wedge S_4^*) \,.
\end{aligned} \tag{4.10}
$$

The extend NEM graph is more helpful to answer the question "how can I deactivate a S-gene by deactivating a subset of its parents?". The B-NEM graph is more helpful to answer "how can I activate a S-gene by activating a subset of its parents?".

4.5 A Bayesian Networks view on Boolean Networks

Like NEM, we can interpret an unperturbed Boolean acyclic graph (BAG) as a Bayesian network (3.1, Zeller *et al.* (2009)). Given a Boolean hyper-graph $\Psi = (S, H)$ every S-gene $S_i \in S$ is independent from its non-descendants, all S-genes not downstream of S_i, given its parents $pa(S_i, \Psi)$ in the hyper-graph. The conditional probabilities are calculated by the Boolean functions (equations (4.11)):

$$
\begin{aligned}
S_i &= \bigvee_{j=1}^{m} \left(\bigwedge_{k=1}^{n(j)} S_{k(j)} \right) \\
P\left(S_i = s \in \{0,1\} \mid pa(S_i, \Psi)\right) &= \left| (1-s) - \bigvee_{j=1}^{m} \left(\bigwedge_{k=1}^{n(j)} S_{k(j)} \right) \right| .
\end{aligned} \tag{4.11}
$$

Figure 4.13 shows an example of a BG and a reduction to a PDAG with the following joint probability distribution

$$
\begin{aligned}
&P\left(A=a, B=b, C=c, D=d, E=e\right) \\
&= P\left(A=a\right) \cdot P\left(B=b\right) \cdot P\left(C=c \,|B=b\right) \cdot P\left(D=d \,|A=a, C=c\right) \\
&\cdot P\left(E=e \,|C=c, D=d\right) \\
&\overset{(4.11)}{=} a \cdot b \cdot |(1-c)-b| \cdot |(1-d)-(a \wedge b)| \cdot |(1-e)-(\neg c \vee \neg d)| \,.
\end{aligned} \tag{4.12}
$$

If the Boolean graph is not perturbed, B and C hold the same values and we cannot derive the edge direction (causal effect Pearl (2000)). The following equation holds:

$$
\begin{aligned}
&P\left(C=c \,|B=b\right) \cdot P\left(B=b\right) = \\
&= |(1-c)-b| \cdot b \overset{b=c}{=} |(1-b)-c| \cdot c \\
&= P\left(B=b \,|C=c\right) \cdot P\left(C=c\right) .
\end{aligned} \tag{4.13}
$$

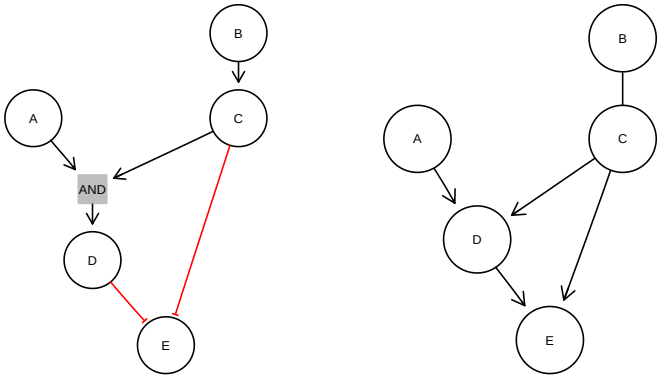


Figure 4.13: **PDAG of a BG.** Example of a BG and its corresponding simplification to a PDAG. Since the PDAG does not explicitly state the Boolean functions defined on the edges, it represent several different BGs.

If C or B is perturbed, they become independent of all other S-genes. If the state of B does not change during a perturbation of C we infer the edge $B \to C$, otherwise $C \to B$.

5

Simulation study

Parts of this chapter have been published (Pirkl *et al.* (2016)).

Before applying B-NEM in practice we check whether the algorithm can reconstruct networks accurately, if the data is generated from known Boolean networks. We refer to underlying data generating networks as Ground Truth Networks (GTN). The experimental conditions consist of controls, single and double stimulations, single knock-downs and the single/double stimulations together with single knock-downs. Finally, observed response schemes were composed by the differences between controls and single/double stimulations/knock-downs and the differences between single/double stimulation and single/double stimulation with a single knock-down.

5.1 Principle simulations

GTNs and matching PKNs are generated by randomly sampling edges from a super PKN shown in figure 5.1. The super PKN has 30 nodes and 144 edges respectively 504 hyper-edges after extension. The nodes fall onto five layers representing ligands, receptors, membrane complexes, cytosolic and nuclear signalling. Edges connect nodes on adjacent layers. 90% of edges are stimulating and 10% are inhibiting. We first draw a PKN and then a GTN. To generate a network of n nodes we randomly choose n nodes from the super PKN, ensuring that their is at least one node at every layer. For this set of n nodes we take all hyper-edges connecting those nodes as the extended PKN. From this PKN we randomly sample 50% hyper-edges, but make sure that the network is at one point stimulated. This means we reject GTNs which do not change their state during any stimulation. Similarly we generate networks of n hyper-edges. Without restricting the GTN to a specific number of nodes. Finally 10 E-genes are attached to every S-gene. Note that the PKN is always consistent with the GTN, no existing edges are a priori excluded. For a given GTN and a set of conditions we calculate the expected E-gene states and add noise by randomly flipping x% of the values with $x \in \{10, 25, 50\}$. This results in a

discrete data set. But since in real applications discretization of the data is not always trivial, we also simulate a continuous data set by adding Gaussian noise $\sim \mathcal{N}(0, \sigma)$ with $\sigma \in \{0.5, 1, 2\}$. Every E-gene profile is generated in triplicates with independent noise. The algorithm's efficiency is scored on simulated data using the runtime, sensitivity and specificity (Loong (2003); Deonier *et al.* (2005))

$$\text{sens} = \frac{TP}{TP + FN}, \quad \text{spec} = \frac{TN}{TN + FP}$$

with *TP, TN, FP, TN* as the true positive, true negative, false positive and false negative rates of the hyper-edges. All simulations were performed on a machine with 12 Intel(R) Xeon(R) CPU X5650 @ 2.67GHz each with 12 megabytes of L1 cache.
For the search algorithm we used

5.1.1 B-NEM accurately estimate the equivalence class of networks with up to 30 S-genes.

We first tested the performance of B-NEM for GTNs with $10, 15, 20, 25$ and 30 S-genes. For each size we generated 10 random GTNs and matching PKNs and run B-NEM on E-gene data generated from these GTNs. The GTNs consisted of 10% of the allowed edges in the corresponding PKN, hence the PKNs were consistent with the GTN and effectively reduce the search space. We then compared the expected response schemes of the estimated networks with that of the GTNs. The top row of figure 5.2 shows the sensitivity and specificity of the estimated networks (solid circle, dashed triangle). The corresponding computation time is shown as the dotted line connecting crosses. In this setting computation is a limiting factor for networks with 30 genes, but reconstruction accuracy is not.

5.1.2 Network reconstruction is sensitive to the strength of the prior knowledge network.

In the previous simulation we checked whether the algorithm finds the correct equivalence class of networks. However, equivalence classes can be large and are hard to interpret. Due to score equivalence multiple networks in the same equivalence class can not be distinguished by data. However, equivalence classes can be shrunk effectively by strong PKNs rendering network reconstruction practical. Thus, we evaluated the accuracy of the estimated networks as a function of the strength of the PKN. For 10 random GTNs with 50 hyper-edges drawn randomly from the full PKN, we run B-NEM using PKNs of $50, 164, 277, 390, 504$ a priori possible hyper-edges. The bottom row of figure 5.2 shows the sensitivity and specificity of the reconstructed networks both on the level of expected response schemes and the actual networks in hyper-edges . While the performance stays very good with respect to response schemes (equivalence classes) it breaks down with respect to network reconstruction if the PKN becomes weak.

If we do not allow for negative regulation, the PKN needn't be a DAG but can have cycles. Cycles in a PKN with negative regulation can lead to undefined expected response schemes. See section A.2 for details. Nevertheless we did the same simulations as above based on a cyclic prior without and with negative regulation (figures D.2 and D.3). Results

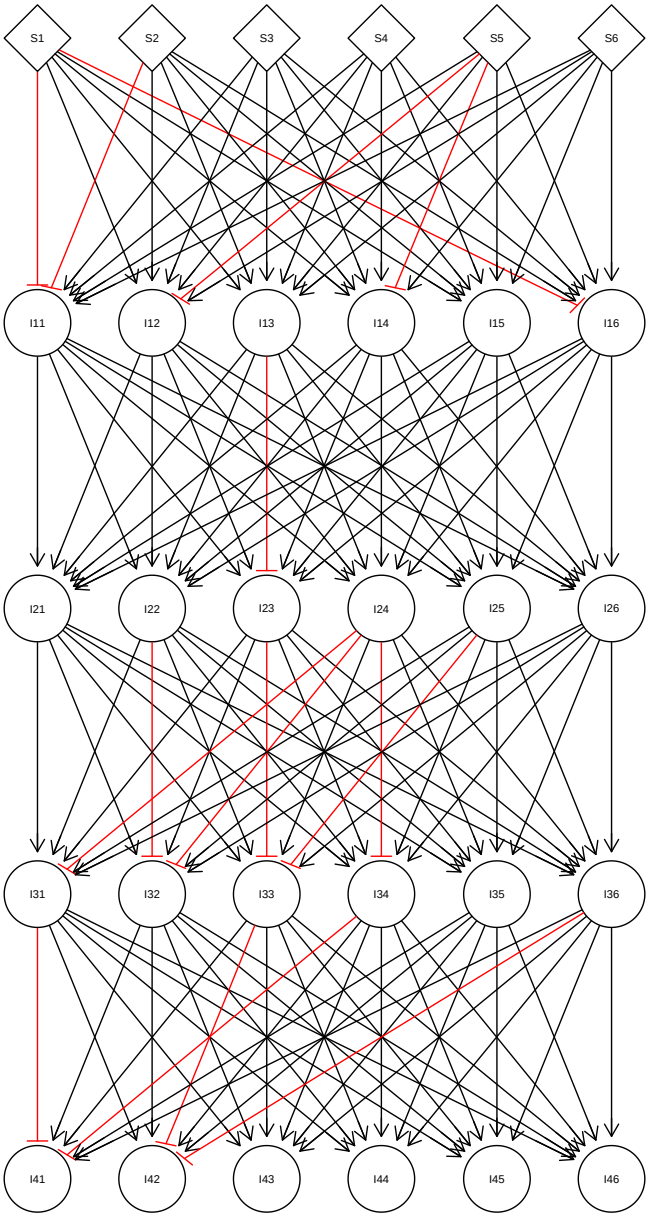


Figure 5.1: **DAG as PKN.** Example of a randomly created Super-PKN with 30 nodes and 144 edges in a normal acyclic graph. $S1$ to $S6$ are nodes which can be set to 0 or 1 as possible stimulations. A node denoted with I can be inhibited and if not is set by the states of its parents. Each edge has a 10% chance of being inhibiting. Extending this PKN with AND gates of size 2 leads to a hyper-graph with 504 hyper-edges.

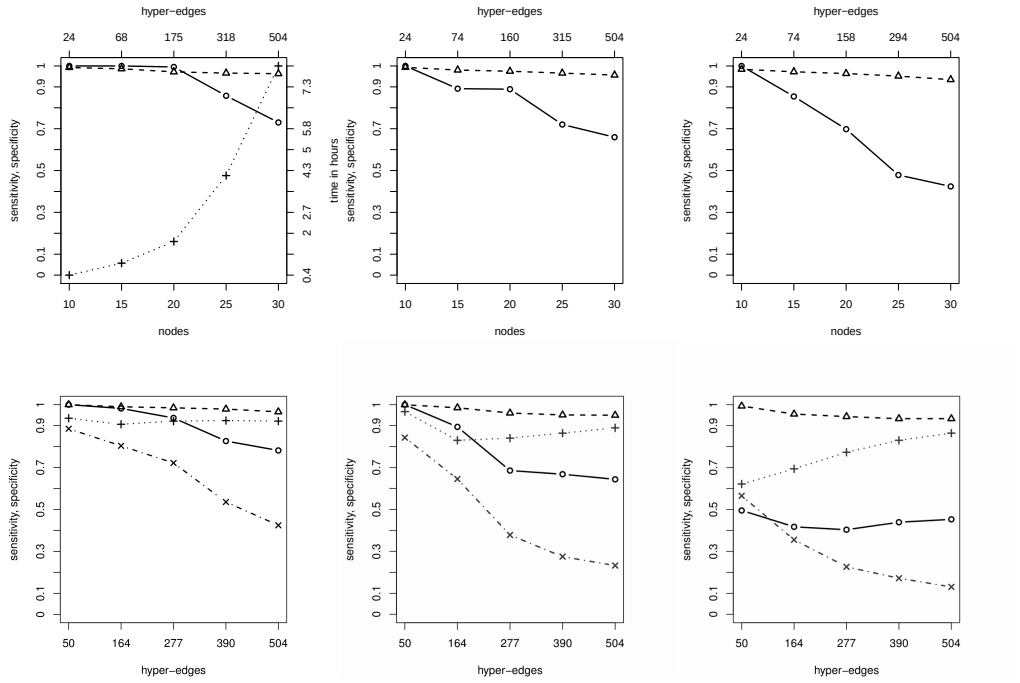


Figure 5.2: **Simulation results.** The three columns show different discrete noise levels $\{0.1, 0.25, 0.5\}$. *Top:* Random GTN of n nodes (x-axis) and the median sensitivity, specificity of the ERS (solid circle, dashed triangle) and running time (dotted cross) for ten runs. The top axis shows the mean PKN size. *Bottom:* Results for ten runs each given a fixed GTN and different PKN sizes (x-axis) including the GTN. Median sensitivity and specificity of the ERS (solid circle, dashed triangle) and the hyper-edges (dashed-dotted x, dotted cross).

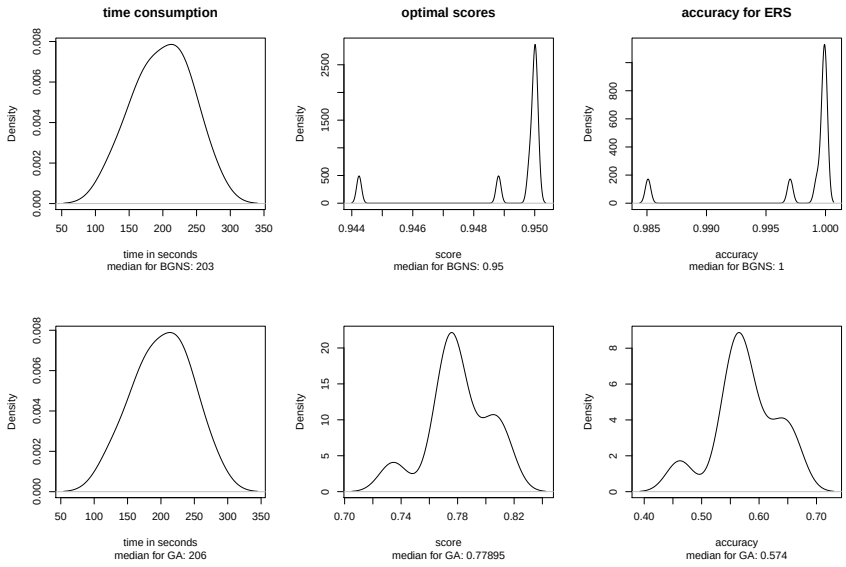


Figure 5.3: **CNO: GA vs BGNS.** *Left:* Distribution of time consumption. BGNS ran first. GA was allowed as much time as the BGNS with empty set needed for each run. *Center:* Distribution of the minimal score. *Right:* Distribution of ERS accuracy.

are shown in figures D.4-D.7. While the results with only positive regulation don't differ much from the results for DAGs, including negative regulation leads to a decrease in sensitivity.

5.2 GA vs BGNS

The scoring of each network takes more time in the B-NEM approach compared to the original CNO (Saez-Rodriguez *et al.* (2009), Terfve & Saez-Rodriguez (2012), Terfve *et al.* (2012)) for several reasons. First of all in B-NEM we have several indirect measurements (E-genes) instead of just the perturbed proteins (S-genes). Hence the dataset is usually larger. Additionally B-NEM has to estimate the E-gene positions Θ. Thus in the CNO setting the GA is already very fast. If the inverse absorption takes longer than scoring a network the BGNS looses its performance advantage. Therefore we compare the GA[1] and the BGNS in the context of the CNO.

We use the PKN from the fundamental simulations in figure D.3. We generate data the same way as before, except that we only have one E-gene (=S-gene) for each S-gene. In an initial simulation run the BGNS ran first and the GA was allowed as much time as the BGNS needed before. In a second simulation run the GA was allowed up to ten times the amount of time of the BGNS.

Figure 5.3 shows the results. The BGNS gets a better score and also a much higher accuracy. If we allow the GA to run up to ten times as long as the BGNS, it achieves similar results (figure 5.4).

[1]population size 100, maximum stall generations 100, elitism 10% and a fully connected network as initialization; empty set has similar results

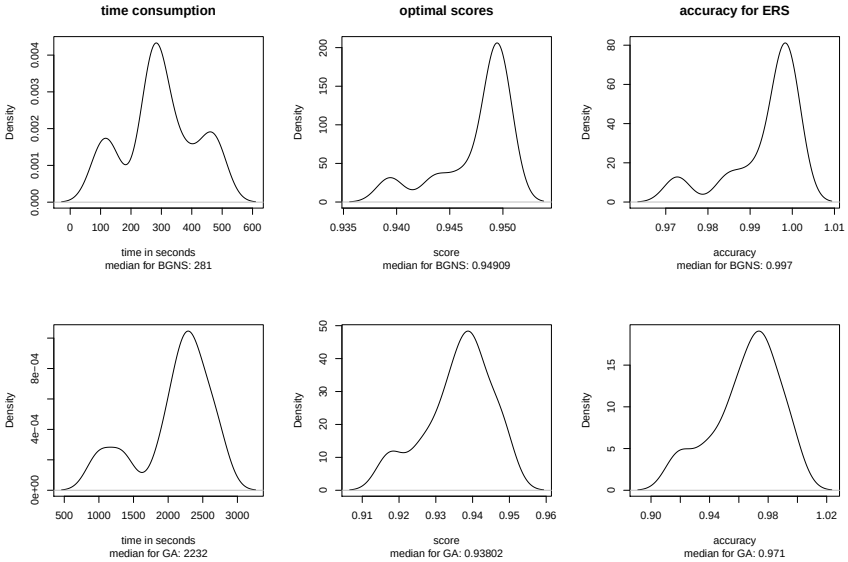


Figure 5.4: **CNO: GA ("no" timelimit) vs BGNS.** *Left:* Distribution of time consumption. BGNS ran first. GA was allowed up to 10 times as much time as the BGNS with empty set needed for each run. *Center:* Distribution of the minimal score. *Right:* Distribution of ERS accuracy.

6

The Role of Pi3k and Tak1 in BCR signalling of Burkitt's Lymphoma Celline BL2

The results of this chapter have been published (Pirkl *et al.* (2016)).

6.1 BCR signalling

We now apply the B-NEM framework to a previously unpublished dataset monitoring gene expression changes in the Burkitt lymphoma cell line BL2 after induction of the BCR. Our analysis explains how BCR signalling propagates to downstream effector pathways like the NFκB, MAP kinase, P38, or JNK pathways through activation of the intermediate messengers TAK1 and PI3K.

B-cell receptor signalling was induced in BL2 cells by cross-linking IgM with an anti-IgM antibody. S-genes were inhibited on protein level using small molecules: 5Z-7-oxozeaenol (TAK1), IKK2 inhibitor VIII (IKK2), Ly294002 (PI3K), SB203580 (P38/MAPK14), SP600125 (JNK), U0126 (ERK1/2). In addition to single inhibitions, IKK2, JNK and P38 were jointly inhibited yielding three double and one triple inhibition. All perturbations were done in triplicate both under BCR stimulation and control conditions and gene expression profiles were generated using Affymetrix hgu133plus2 Genechips. Moreover, profiles of 6 negative controls (unstimulated BL2 cells) and 6 positive controls (BCR stimulated cells) were produced, yielding a dataset of 72 gene expression profiles in total. The dataset was made available at the GEO database[1] under accession id GSE68761.

6.2 Gene expression profiling and preprocessing

The BCR data set was generated using Affymetrix GeneChip Human Genome U133 Plus 2.0 arrays. Profiles were normalized on probe level using the variance stabilization method of (Huber *et al.* (2002)). Batch effects were corrected using ComBat (Johnson *et al.*

[1] http://www.ncbi.nlm.nih.gov/geo/

(2007)). The foldchanges used in the B-NEM modelling were calculated with the limma package 3.16.1 (Smyth (2004)). The following contrasts were calculated: "Stimulation - Control" (stimulation effect), "Inhibition_Stimulation - Stimulation" (silencing of the stimulation during knock-down). See also table 6.1. We calculated general coefficients β_j^i for each gene i and variable (experimental condition) to calculate the gene expression y_i.

$$\begin{aligned} y_i = \; & \beta_0 \cdot \text{Ctrl} + \beta_1 \cdot \text{BCR} + \beta_2 \cdot \text{Jnk} + \beta_3 \cdot \text{p38} + \beta_4 \cdot \text{Ikk2} + \beta_5 \cdot \text{Pi3k} + \\ & \beta_6 \cdot \text{Tak1} + \beta_7 \cdot \text{BCR\&Jnk} + \beta_8 \cdot \text{BCR\&p38} + \beta_9 \cdot \text{BCR\&Ikk2} + \\ & \beta_{10} \cdot \text{BCR\&Pi3k} + \beta_{11} \cdot \text{BCR\&Tak1} + \epsilon_i \end{aligned} \tag{6.1}$$

with a small error ϵ_i. Last we removed "AFFY" control probesets from the data. The affymetrix probeset ids were converted to HGNC gene symbols (Gray *et al.* (2015)) with the hgu133plus2.db, annotate and biomart R packages (Carlson (n.d.); Gentleman (n.d.); Smedley *et al.* (2015); R Core Team (2014)). After the raw data was normalized, observed response schemes were calculated for the comparisons listed in table 6.1. We filtered for E-genes that respond to BCR stimulation by at least an absolute log2 foldchange of 1 and to another comparison by at least an absolute log2 foldchange of $\log2(1.5) \approx 0.58$. This corresponds to a change in expression of at least 100% respectively 50%, leaving us with 602 E-genes.

base level		change level
(control)	vs	(BCR+)
(BCR+)	vs	(BCR+,PI3K-)
(BCR+)	vs	(BCR+,TAK1-)
(BCR+)	vs	(BCR+,ERK-)
(BCR+)	vs	(BCR+,IKK2-)
(BCR+)	vs	(BCR+,P38-)
(BCR+)	vs	(BCR+,JNK-)
(BCR+)	vs	(BCR+,IKK2-,P38-)
(BCR+)	vs	(BCR+,IKK2-,JNK-)
(BCR+)	vs	(BCR+,P38-,JNK-)
(BCR+)	vs	(BCR+,IKK2-,P38-,JNK-)

Table 6.1: **Contrasts.** Contrasts of conditions used to calculate the observed response schemes from the data. + denotes activation of the node and − inhibition in that particular condition.

6.3 Results

6.3.1 Prior knowledge in BCR signalling

The B-cell receptor (BCR) is the cell surface receptor that initiates BCR signalling upon binding of an antigen. BCR signalling leads to the activation of IKK2, P38, ERK, and JNK (DeFranco (1997); Richards *et al.* (2001); Schuman *et al.* (2009); Shinohara & Kurosaki (2009)). These four effector pathways send signals into the nucleus that affect gene expression. The two proteins PI3K and TAK1 are potential mediators of BCR

induced activation of effector pathways. We do not put any restriction on the hierarchical ordering of PI3K and TAK1. PI3K and TAK1 are parts of several other pathways where they are described as activators and not as repressor of signalling. We thus assume that the same holds true in BCR induced signalling. What is not known is which activations depend on which of the two mediators, nor is it known whether they activate downstream pathways independently from each other (OR gate) or jointly (AND gate). Furthermore the combinatorial inhibitions of IKK2, P38 and JNK allow more freedom in the PKN and therefore we do a complete reconstruction on this subnetwork. We summarize this prior knowledge situation in the PKN of Figure 6.1.

6.3.2 Calibrating the sparseness parameter ζ

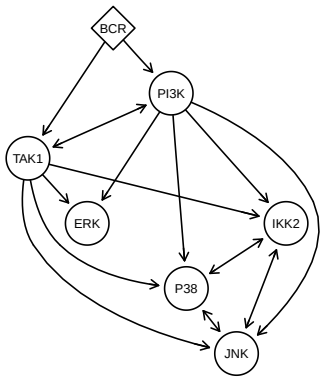


Figure 6.1: **Prior search space restriction.** PKN for BCR signalling into IKK2, P38, JNK and Erk. We do not allow for negative regulation. Naturally, BCR defines the top S-gene. PI3K and TAK1 build the second hierarchical layer but we additionally allow for TAK1 above PI3K or the reverse. The third layer consists of IKK2, P38, JNK and ERK. Since our combinatorial inhibitions reduce the problem of equivalence classes for IKK2, P38 and JNK we allow for the complete reconstruction of the sub network consisting of these three S-genes.

Calibrating ζ is critical to the performance of B-NEM. We randomly split the set of E-genes in half. For various settings of ζ (exponential decrease $\zeta \in \{1, 0.64, 0.36, 0.16, 0.04, 10^{-10}, 0\}$) we learn a network using the first half of the data (training set), and then score this network using the second independent half (test set) but without employing the complexity penalty in equation (4.1). We repeat this step with 100 different random splits of E-genes and take the mean of graph size, connected S-genes (both in percent) and scores of the test sets. Figure 6.2 shows that the score continuously improves as ζ approaches zero. For $\zeta = 0$ the test accuracy drops again. Note that for any positive zeta the smaller network wins in case of likelihood equivalence while for $\zeta = 0$ there is no size penalty operating at all. We thus set ζ to 10^{-10}.

6.3.3 The role of PI3K and TAK1

We run B-NEM on this data using the PKN and the parameter settings described above. Figure 6.3 shows the highest scoring network. The network predicts that the activation of the JNK pathway is only PI3K dependent, while Erk is only TAK1 dependent. IKK2 activation is predicted either as redundant by PI3K or alternatively TAK1. P38 is positively regulated by PI3K via either JNK or alternatively jointly with IKK2. The signal flow to P38 can be stopped either with the inhibition of PI3K or the double inhibition of JNK and IKK2. The observed response schemes side by side with the corresponding expected response schemes can be seen in figures 6.5-6.11 .

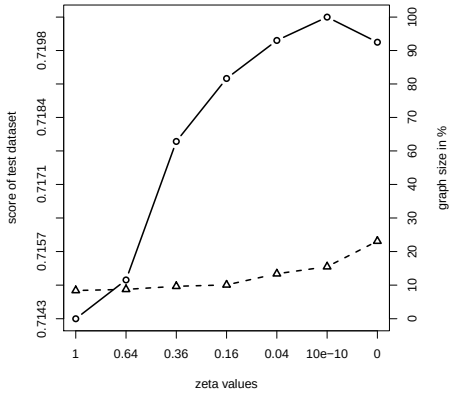


Figure 6.2: ζ **calibration.** Mean cross validated network scores as a function of the complexity parameter ζ. Score on the test dataset (solid circle, log-scale) and graph size in percent (dashed triangle).

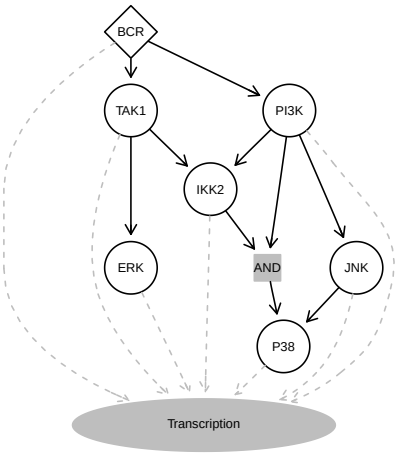


Figure 6.3: **Learned network.** The highest scoring network (black edges). The BCR signal is propagated via PI3K into JNK and P38. IKK2 is alternatively regulated by PI3K or TAK1. PI3K and TAK1 are directly regulated by BCR. TAK1 propagates the signal into the ERK pathway. Additionally P38 is alternatively regulated by JNK or IKK2. The different AND and OR gates are annotated more prominently. Grey dashed edges illustrate the propagation of signals from all molecules into the nucleus to regulate transcription.

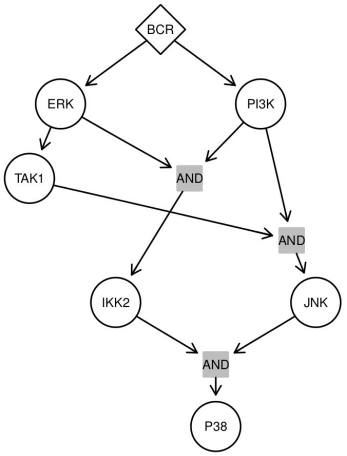


Figure 6.4: Inference on BCR signalling with the original NEM.

That TAK1 alone, as proposed by our model, can not block signalling into IKK2 and JNK has been detected for toll-like receptor 8 (TLR8) signalling in mouse embryonic fibroblasts (MEF, Qin *et al.* (2006)). TAK1 knock-out mice still showed an activated NFκB pathway. TLR8 also seems to be causative in some lymphomas (Ngo Vu N. *et al.* (2011)). Furthermore Matta *et al.* (2012) show that herpes virus encoded viral FLICE inhibitory protein (vFLIP) K13 inducted NFκB activity is not impaired in TAK1 deficient MEFs. Chen & Debnath (2013) give evidence that the IKK complex (IKK1, IKK2, NEMO) acts independently of PI3K in mammary epithelial cells and Xue *et al.* (2000) that ERK can be activated independently from PI3K in nerve growth factor (NGF)-dependent sympathetic neurons. Kloo *et al.* (2011) propose the regulation of IKK2 by PI3K in diffuse large B-Cell like lymphomas. They show in their data, that the PI3K inhibitor only partially blocks IKK2 inhibitor target genes I.e. downstream targets of PI3K are a subset of downstream targets of IKK2, which is not true in our case. In the Nested Effects Model logic this either places PI3K downstream of IKK2 or PI3K and IKK2 have joint downstream targets. A third explanation is, that some NFκB activity is regulated by PI3K, but another alternative regulation is possible as depicted in our network in figure 6.3.

For comparison we used standard Nested Effects Models (Markowetz *et al.* , 2005; Froehlich *et al.* , 2008) to model the BCR data set. Since NEM uses only single perturbations, we discarded all combinatorial perturbations. We discretized the foldchanges using $log2(1.5)$ as cutoff. Figure 6.4 shows the result. In disagreement to established knowledge ERK is placed up stream of all S-genes except PI3K indicating the need to use both Boolean logic and prior knowledge in modelling BCR signalling.

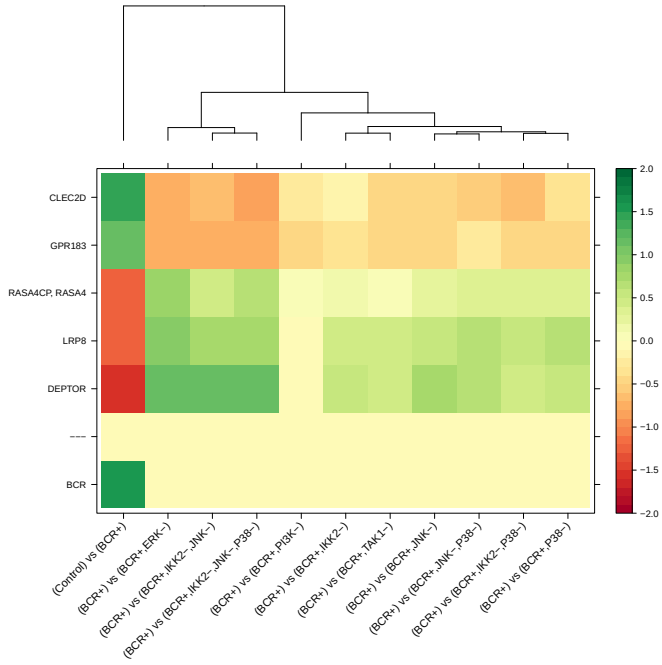


Figure 6.5: E-genes (= affymetrix probesets) regulated by BCR directly. Expected response scheme (bottom row) and observed response schemes (top rows).

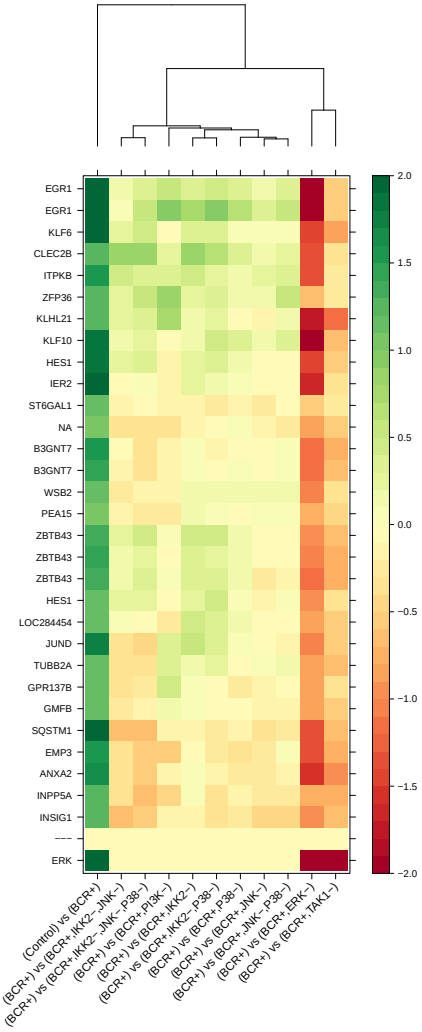


Figure 6.6: Top 30 E-genes (= affymetrix probesets) regulated by ERK directly. Expected response scheme (bottom row) and observed response schemes (top rows).

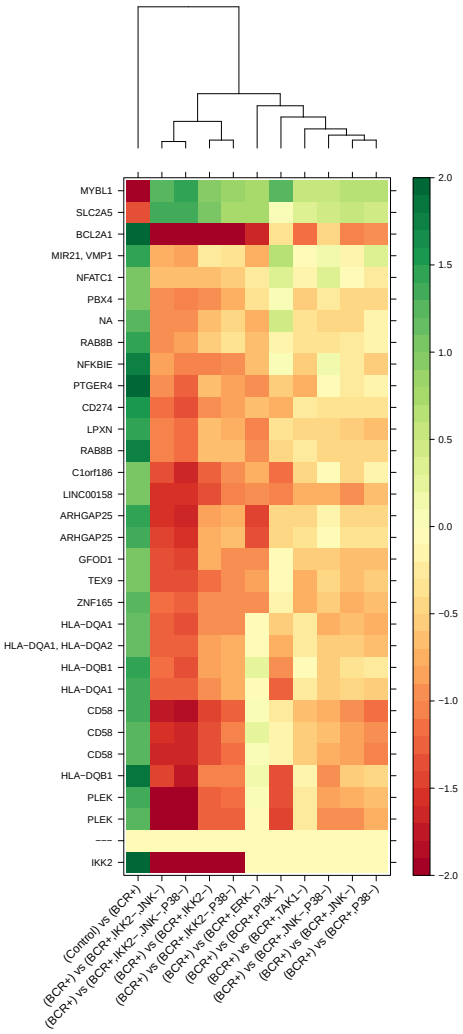


Figure 6.7: Top 30 E-genes (= affymetrix probesets) regulated by IKK2 directly. Expected response scheme (bottom row) and observed response schemes (top rows).

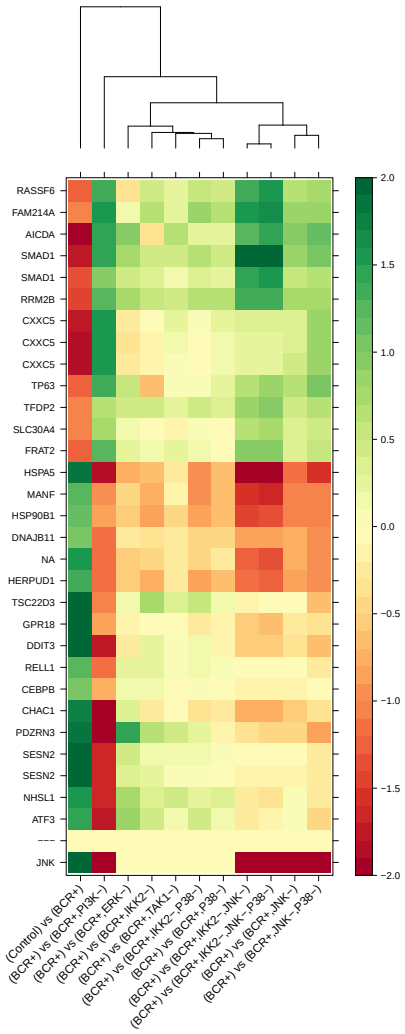


Figure 6.8: Top 30 E-genes (= affymetrix probesets) regulated by JNK directly. Expected response scheme (bottom row) and observed response schemes (top rows).

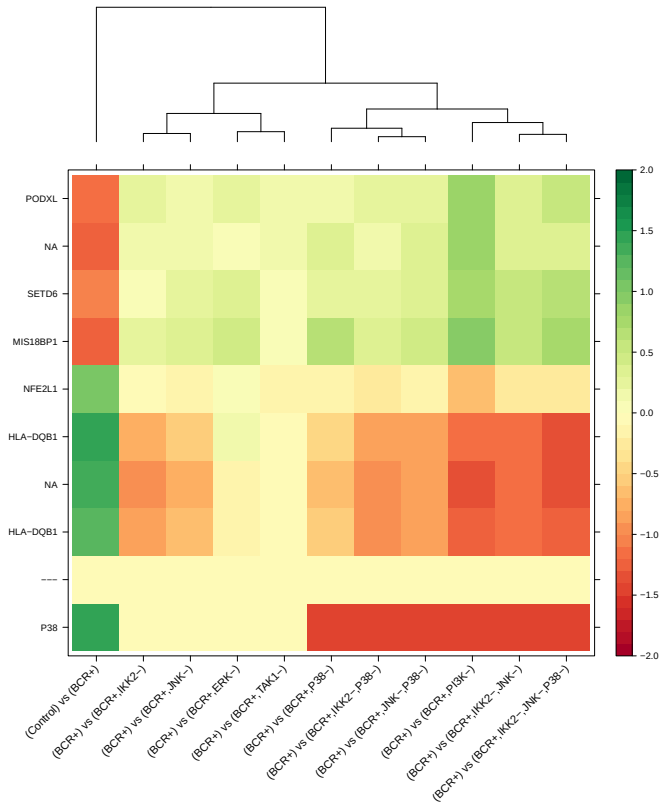


Figure 6.9: E-genes (= affymetrix probesets) regulated by P38 directly. Expected response scheme (bottom row) and observed response schemes (top rows).

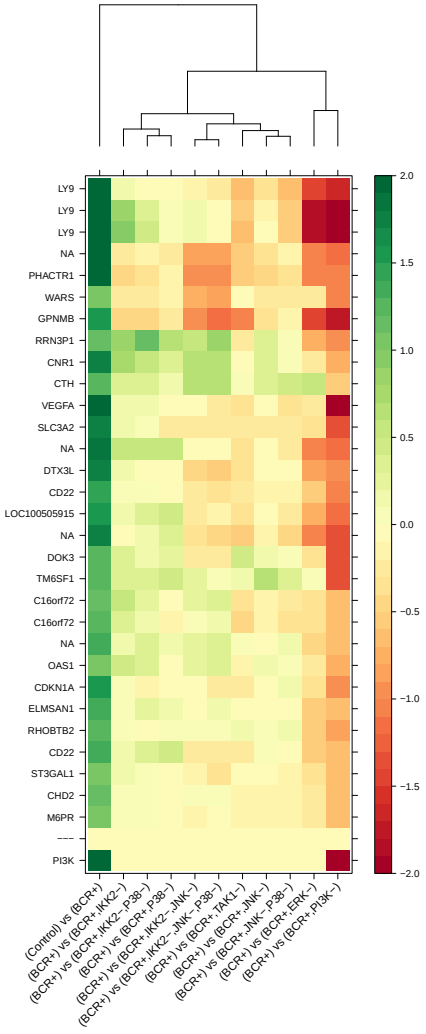


Figure 6.10: Top 30 E-genes (= affymetrix probesets) regulated by PI3K directly. Expected response scheme (bottom row) and observed response schemes (top rows).

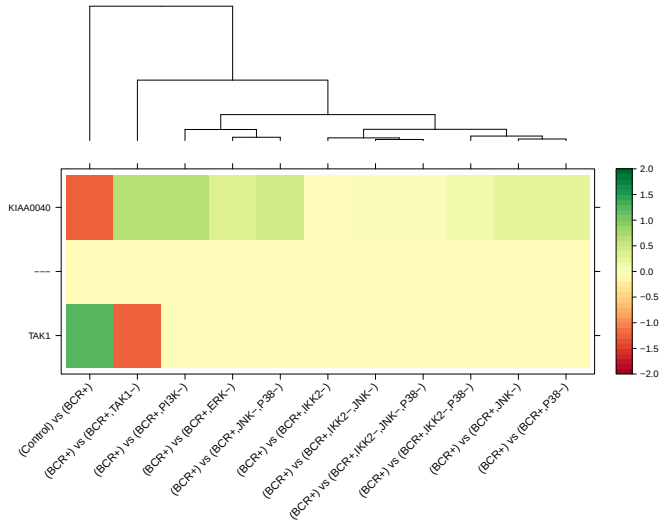


Figure 6.11: E-gene (= affymetrix probeset) regulated by TAK1 directly. Expected response scheme (bottom row) and observed response schemes (top rows).

7

Analyzing Crosstalk of Inflammatory and Apoptotic Signalling in Hepatocellular Carcinoma

7.1 Hepatocellular Carcinoma

Liver cancer (hepatocellular carcinoma, HCC) is one of the most common forms of cancers world wide and has a high mortality rate. The largest risk factors leading to carcinogenesis are hepatitis B and C viruses. Other attributing factors include smoking, alcohol consumption and exposure to aflatoxin B_1 (Gomaa Asmaa Ibrahim *et al.* (2008)).

Important pathways are often de-regulated in HCC (Dufour & Clavien (2005)). For example in some cases NfκB is constitutively active. NfκB signalling has anti-apoptotic effects, like the inhibition of Casp8. This leads to a constant survival signal and helps cancer cells to avoid apoptosis. Thus the cancer can grow and proliferate.

7.1.1 Tnf-α and Trail signalling in HCC

Tumor Necrosis Factor alpha (Tnf-α) is a molecule, which can bind to the receptor Tnfr1 in the cell membrane. Tnf-α regulates a wide range of cellular responses like cell death, proliferation (Bradley (2008)) and in particular the pro-survival pathways regulated by the proteins Jnk and NfκB (Wajant *et al.* (2003), Silke (2011); Metzig *et al.* (2011a)). However many other signalling molecules are involved in the regulation of these pathways and act as mediators such as the the signalling kinases Tak1 and Nik (Haas *et al.* (2009); Walczak (2011); Darding & Meier (2012); de Almagro & Vucic (2012)).

The Tnf-α related apoptosis-inducing ligand (Trail, Schaefer *et al.* (2007)) is a member of a subfamily of Tnf-α related molecules. Trail regulates the extrinsic activation of apoptosis (Falschlehner *et al.* (2009)). It binds to one of the death receptors Dr4 or Dr5. These two propagate the signal to Casp8 which activates Casp3. Then Casp3 degrades proteins, which leads to cell death.

Many pathways communicate with each other via crosstalk (Zucchini-Pascal *et al.* (2013); Imai *et al.* (2014)). Tnf-α and Trail are not isolated pathways either. Both path-

ways share many signalling molecules with each other (Kim *et al.* (2002); Falschlehner *et al.* (2007); Jouan-Lanhouet *et al.* (2012); Azijli *et al.* (2013)). For example they are both known to regulate Casp8 activity.

Several previously unknown molecules have been identified to be involved in Tnf-α or Trail signalling. Among those are Usp2 (Metzig *et al.* (2011b); Mahul-Mellier Anne-Laure *et al.* (2011)), Casp4 (Mao *et al.* (2010); Nickles *et al.* (2012)) and Casp8ap2 (Imai *et al.* (1999); Choi *et al.* (2001); Jun *et al.* (2005); Hummon *et al.* (2012)). Sharpin, Hoip and Hoil1 form the linear ubiquitin chain assembly complex (LUBAC) and are also assumed to mediate Tnf-α and Trail induced signals (Walczak (2011)).

In general of importance when it comes to de-regulated cell function and cancer is the Wingless-Type (WNT) pathway. Not only is this pathway responsible for several pro-survival responses like cell proliferation and migration, but it has also been shown to interact with known Tnf-α and Trail signalling molecules (Lamberti *et al.* (2001); Toyama *et al.* (2010); Mahmoudi Tokameh *et al.* (2009); Zimmerman *et al.* (2013); Hiyama *et al.* (2013)). Additionally there is strong evidence, that the WNT signalling molecule Beta-Catenin (CTNNB1) is mutated in the HCC cell line HepG2, which might alter normal Tnf-α and Trail signalling (Lachenmayer *et al.* (2012); Kan *et al.* (2013); Tornesello *et al.* (2013)).

Since Trail can induce apoptosis in some and not in other cells, the idea is to kill cancer cells but keep normal cells alive. Thus we have to identify potential targets to make cancer cells sensitive and normal cells resistant to Trail induced apoptosis. The WNT pathway and Tnf-α are known to counteract the apoptotic signal. Therefore proteins involved in the crosstalk of WNT, Tnf-α and Trail might help to achieve sensitivity respectively resistance in certain cell types (Falschlehner *et al.* (2007); Papenfuss *et al.* (2008); Russo *et al.* (2010)).

7.2 Data generation and processing

Cells were treated for 24 hours with either Renilla luciferase (Rluc, Shifera & Hardin (2010)), a non-target small interfering RNA (siRNA), both as negative controls, or siRNAs targeting specific mRNAs. Each mRNA was targeted by a pool of four siRNAs to reduce off-target effects (Jackson & Linsley (2010); Hannus Michael *et al.* (2013)). Then the cells were stimulated with Tnf-α, Trail or both and sequenced at time points 0 hours (control, no stimulation), 2 hours and 4 hours. The cells were sequenced on SOLID™ 4 and SOLID™ 5500 machines (`http://www.appliedbiosystems.com`, Wikipedia (2016a)). The sequences were mapped to the human reference genome hg19, build 37 with the Bioscope™ v1.3 software. Counts were generated with the HTSeq python package (Anders *et al.* (2015)).

The data consists of a gene expression count matrix with 21969 rows (genes) and 1268 columns (samples). The experiments are combinations of stimulations and siRNA treatments in triplicate. Overall, three different stimulations (Tnf-α, Trail, Tnf-α&Trail) and 37 genes were targeted. The genes targeted are (hgnc gene symbols, Gray *et al.* (2015)):

APC, ATF2, BIRC2 (cIap1), BIRC3 (cIap2), CASP4, CASP8, CFLAR (c-Flip), CHUK (Ikk1), CTNNB1 (Beta-Catenin), DKK1, DKK4, FLASH, IKBKB (Ikk2), IKBKG (Nemo),

JUN (cJun), MAP2K1 (Mekk), MAP3K14 (Nik), MAP3K7 (Tak1), MAPK8 (Jnk), PIK3CA (Pi3k), RBCK1 (Hoil1), RELA, RIPK1 (Rip1), RIPK3 (Rip3), RNF31 (Hiop), SHARPIN, TAB2, TCF7L2 (Tcf4), TNFRSF10A (Dr4), TNFRSF10B (Dr5), TNFRSF1A (Tnfr1), TNIK, TRAF2, USP2, WLS (Evi), WNT11, WNT5A.

The brackets hold common protein names. We use protein names or the lower case of the gene names when we talk about pathways. When we talk about mRNA or siRNA we use the gene names.

Experimental conditions available are

- control (no stimulation, either no or control siRNA)
- gene knock-down (siRNA)
- stimulation measured at time points 2 hours and 4 hours after stimulation (either no or control siRNA)
- stimulation in combination with siRNA measured at time points 2 hours and 4 hours after stimulation.

Genes that have a mean raw count expression of less than ten over all samples are not used in further analysis.

We calculate normalizing factors to account for varying library sizes among samples with edgeR (Robinson Mark D *et al.* (2009)). We put the normalized counts into the Voom pipeline which log normalizes the data and calculates weights for every entry to account for observational variation. Voom uses a linear model to calculate the weights (Law *et al.* (2014), equation (7.1)). Then the normalized expression values and the weights are put in the Limma pipeline (Smyth (2004)).

$$
\begin{aligned}
y_i = \; & \beta_{0,i} \cdot \mathrm{Ctrl} + \sum_{\mathrm{siRNA}} \left(\beta_{1,i}^{\mathrm{siRNA}} \cdot \mathrm{siRNA}\right) + \\
& \sum_{H \in \{2,4\}} \left(\beta_{1,i}^{H} \cdot \mathrm{Tnf}_H + \beta_{2,i}^{H} \cdot \mathrm{Trail}_H + \beta_{3,i}^{H} \cdot \mathrm{Tnf\&Trail}_H + \right. \\
& \sum_{\mathrm{siRNA}} \Big(\beta_{2,i}^{\mathrm{siRNA},H} \cdot \mathrm{siRNA\&Tnf}_H + \\
& \left. \left. \beta_{3,i}^{\mathrm{siRNA},H} \cdot \mathrm{siRNA\&Trail}_H + \beta_{4,i}^{\mathrm{siRNA},H} \cdot \mathrm{siRNA\&Tnf\&Trail}_H \Big) \right) + \epsilon_i \right. \qquad (7.1)
\end{aligned}
$$

with the expression y_i for gene i, the time point H and a small error ϵ_i.

We calculate the log foldchanges for each gene based on the contrasts

- stimulation - control =(control) vs (stimulation+)
- siRNA&stimulation - stimulation =(stimulation+) vs (stimulation+,gene-)
- siRNA - control =(control) vs (gene-).

For each gene we consider the two time points as two different genes. For example the gene ICAM1 has the above contrasts, except for siRNA - control, for time point two hours and time point four hours. We account for this by introducing the two different genes $\mathrm{ICAM1}_{2h}$ and $\mathrm{ICAM1}_{4h}$. Instead of one gene with identical conditions for two time points, we now have two genes for one time point. From now on we refer to the gene at two hours simply as ICAM1.

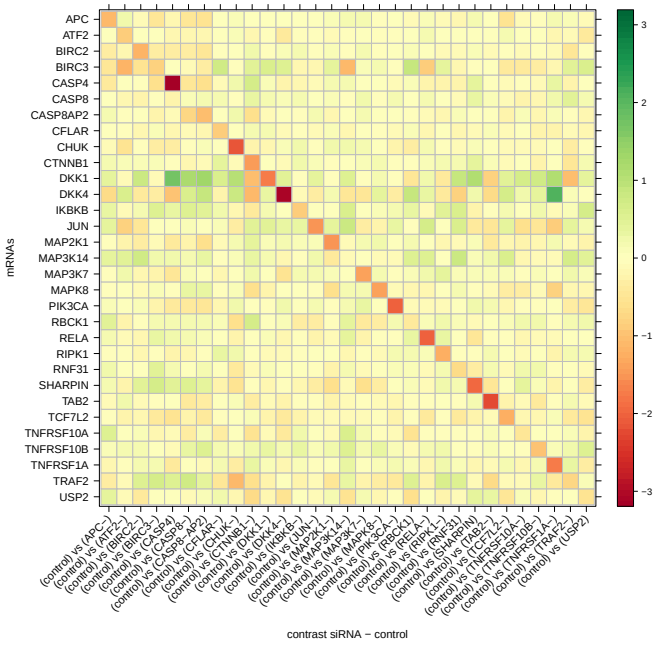


Figure 7.1: **siRNA efficiency.** On the y-axis is the mRNA and on the x-axis the contrast siRNA - control. The diagonal indicates the effectiveness of the siRNA knock-down on its mRNA target.

Gene selection

We define target genes by an absolute log2 foldchange ≥ 1 in any stimulation - control contrast. Additional we demand a differential regulation by at least $log2(1.5) \approx 0.58$ with the opposite sign by the receptor during the associated stimulation. For instance if a gene has a foldchange of ≥ 1 in (control) vs (Tnf-α+), it must have a foldchange of $\leq -log2(1.5)$ in (Tnf-α+) vs (Tnf-α+, Tnfr1−). These criteria reduce the data set to 1376 genes.

Quality of siRNA

We check the siRNA effectiveness for each targeted mRNA (figure 7.1). However some targets like IKBKG are not shown, because they had a too low raw expression. Overall the siRNAs have a high sensitivity, except for CASP8 and MAP3K14.

Including WNT targets

Additionally to the previously selected genes we include genes, which are affected by active Beta-Catenin signalling. Thus we select genes, which have an absolute log2 foldchange of ≥ 1 in (control) vs (Beta-Catenin−) and an absolute log2 foldchange of $\geq log2(1.5)$ in (control) vs (Tcf4−). We discard genes, which are differently regulated by both knock-downs, because we assume positive regulation of Tcf4 by Beta-Catenin (Beta-Catenin $\rightarrow$ Tcf4).

316 genes are identified as WNT targets. Among them we find the known WNT target AXIN2 of two and four hours (Jho *et al.* (2002)). It is down-regulated by both

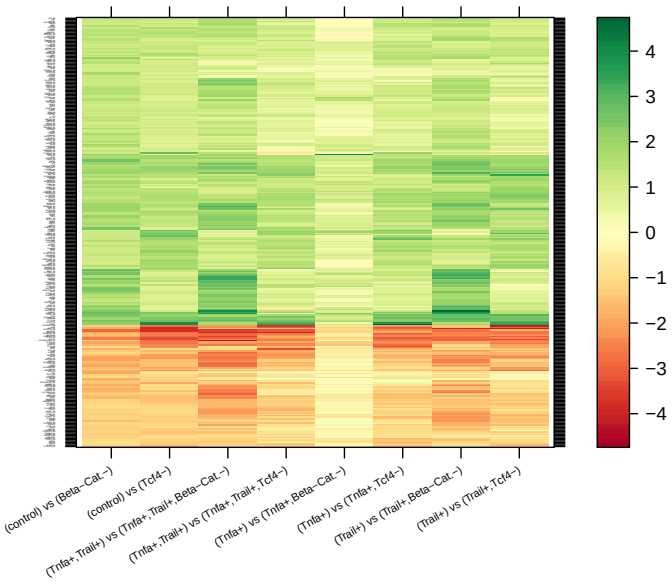


Figure 7.2: **WNT target genes.** Observed response schemes of WNT target genes (rows). The weak knock-down effect of Beta-Catenin during Tnf-α stimulation is clearly visible.

the Beta-Catenin and Tcf4 knock-downs.

The WNT regulated genes react to Beta-Catenin and Tcf4 knock-downs during almost all stimulations (figure 7.2). However during the single Tnf-α stimulation combined with the Beta-Catenin knock-down, the foldchanges have the correct sign but are visibly weaker. We investigate this effect by using B-NEM to estimate the optimal network based on the PKN in figure 7.3, A and the ORS in figure 7.2. The optimum is shown in figure 7.3, B. Beta-Catenin is constitutively active (Beta-Cat. $= \neg FALSE = \neg 0 = 1$). Tnf-$\alpha$ can activate Tcf4 independent of Beta-Catenin. Trail inhibits this activation. Thus during the double stimulation and the single Trail stimulation only Beta-Catenin activates Tcf4. This confirms the weak knock-down effect of Beta-Catenin during the single Tnf-α stimulation (figure 7.2).

B-NEM cannot resolve, which receptor, Dr4, or Dr5, or both negatively regulate Tcf4. However combinatorial knock-downs of Dr4, Dr5 and Beta-Catenin would resolve this uncertainty. For example if only Dr4 inhibits the activation of Tcf4 by Tnf-α, the double knock-down of Dr4 and Beta-Catenin will up-regulate Tcf4 during the double stimulation, the double knock-down of Dr5 and Beta-Catenin won't.

7.3 Prior knowledge

We employ the KEGG pathway database (Kanehisa & Goto (2000); Kanehisa *et al.* (2014)) to define a PKN for the Trail, Tnf-α and WNT pathways (figure D.8). Additionally we place Evi according to Voloshanenko Oksana *et al.* (2013). We also include the interactions of the Tnf-α and Trail receptors with Tcf4 from the previous section.

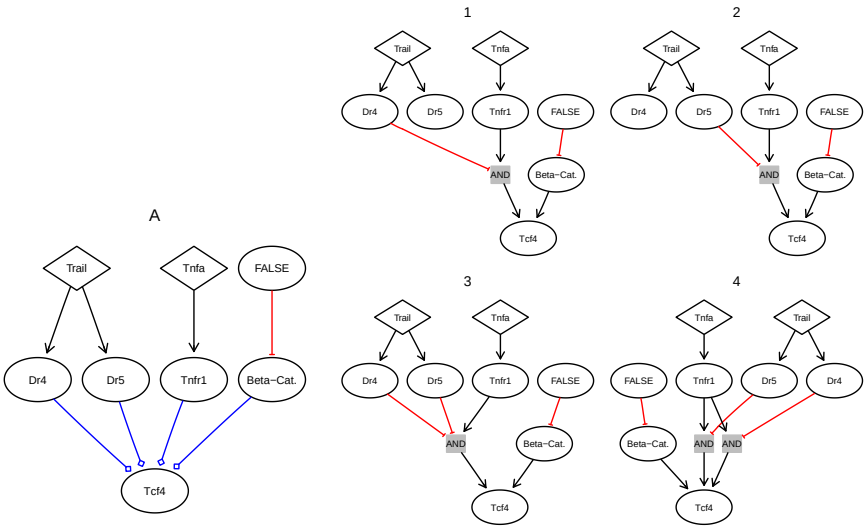


Figure 7.3: **WNT model.** WNT prior network (A), the estimated optimum (1) and three equivalent networks (2-4). Blue edges with a diamond head denote ambiguous regulation ($\rightarrow$ and $\dashv$ possible).

7.4 Results

7.4.1 The core network

For the following analyses, we use the union (1643) of previously selected genes.

Training ζ

Before we search for the optimal network for the large Tnf-α and Trail pathways, we train the ζ parameter. We do this by splitting the dataset in half leaving 821 respectively 822 genes in each. One is the training set and the other is the test set. We then estimate the optimal network for the training set and a size penalty $\zeta \geq 0$. We score the test set on the optimal network with $\zeta = 0$. We do this for several values for ζ (exponential decrease $\zeta \in \{1, 0.64, 0.36, 0.16, 0.04, 10^{-10}, 0\}$) in 100 repetitions. The results are shown in figure 7.4. The score for the test set increases until $\zeta = 10^{-10}$ and decreases again for $\zeta = 0$. Thus we set $\zeta = 10^{-10}$ in further analyses.

Optimal network

The optimal network for Tnf-α and Trail is shown in figure 7.5. We identify the following features:

- Casp8 is activated only by Trail and not Tnf-α.
- Active Tnf-α makes the activation of Casp8 less robust. If only Trail is stimulated the signal to Casp8 can only be blocked by Dr5. In the double stimulation the signal to Casp8 can be blocked by both death receptors Dr4 and Dr5.
- The NFκB pathway (RelA) is only activated by Nemo and not Ikk1 and Ikk2.

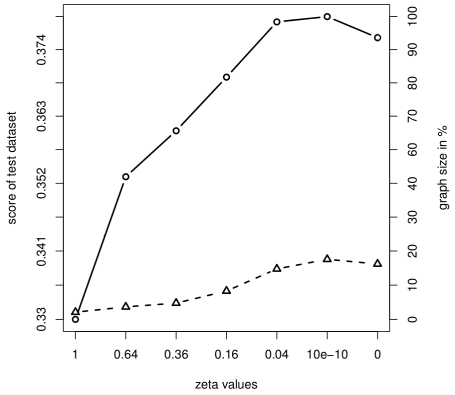


Figure 7.4: **Training ζ.** Mean cross validated network scores as a function of the complexity parameter . Score on the test dataset (solid circle, log-scale) and graph size in percent (dashed triangle).

- Beta-Catenin is active during control and activates Tcf4, which activates Evi, Wnt11 and Wnt5A, and finally Apc. Apc is inactive during control and active during the knock-downs of upstream proteins such as Beta-Catenin.

Modified PKN to remove local residuals

Figure 7.6 shows the residuals in the positive residuals matrix (PRM) and the negative residuals matrix (NRM, section 4.2.10). We marked the interesting residuals in figure 7.6 with blue boxes. We concentrate on two types of residuals: strong effects and systematic effects. For example residuals for Wnt5a (row) are relatively small, but occur always with the Nik knock-down. The following list describes how we intend to account for them with a modified PKN.

- PRM:
  1. (control) vs (Trail+): Trail activates multiple S-genes, thus we add edges Dr5, Dr4 $\rightarrow$ S-gene to the PKN.
  2. Apc (green): Apc E-genes are up-regulated by the Traf2 but not the Tnfr1 knock-down during Tnf-α stimulation. Thus we add edges Traf2 $\dashv$ Apc and Tnfr1 $\rightarrow$ Apc, because without the second edge Tnfr1 would also indirectly inhibit Apc: Tnfr1 $\rightarrow$ Traf2 $\dashv$ Apc.
  3. Apc (red): Evi does not inhibit Apc. Thus we add feed forward loops from Tcf4 to Wnt5a and Wnt11 to bypass Evi.
  4. Casp8: the residuals contradict our model but imply an inhibition of Casp8 by Apc, C-Flip and Pik3.
  5. Wnt5a: E-genes of Wnt5a are up-regulated by the Nik knock-down, thus we add a negative edge from Nik to Wnt5a.
- NRM:

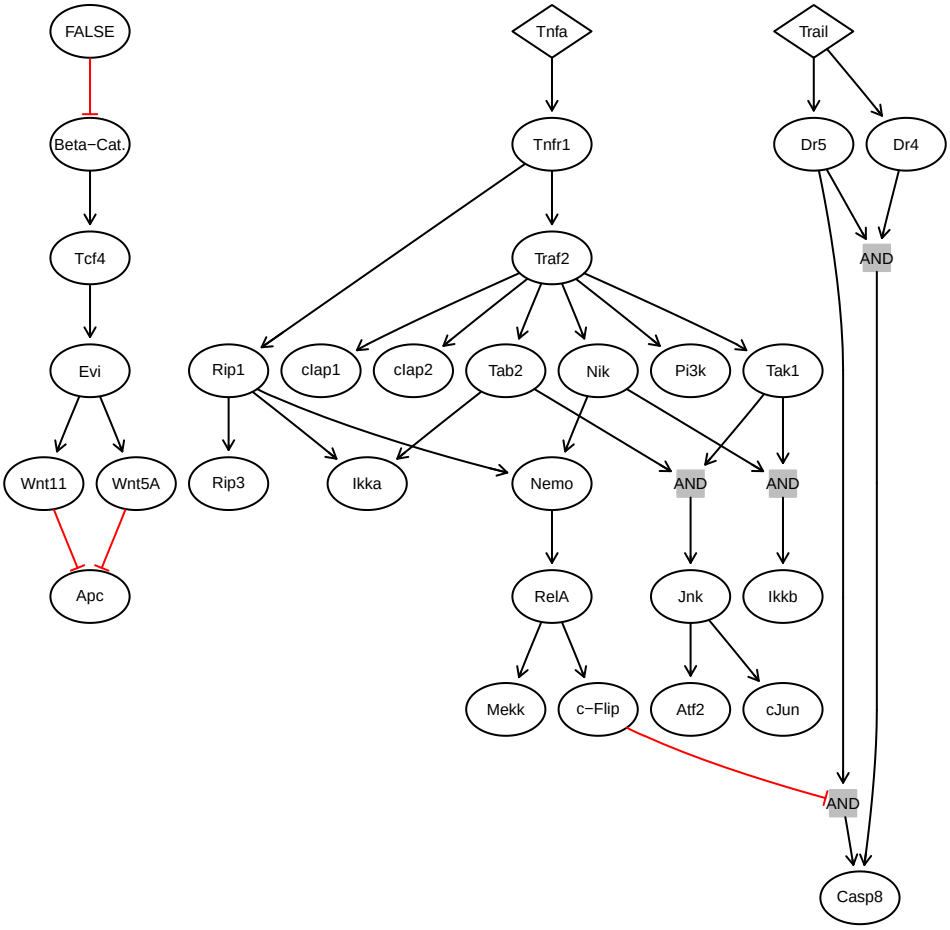


Figure 7.5: **Tnf-α-Trail-WNT result.** Estimated network.

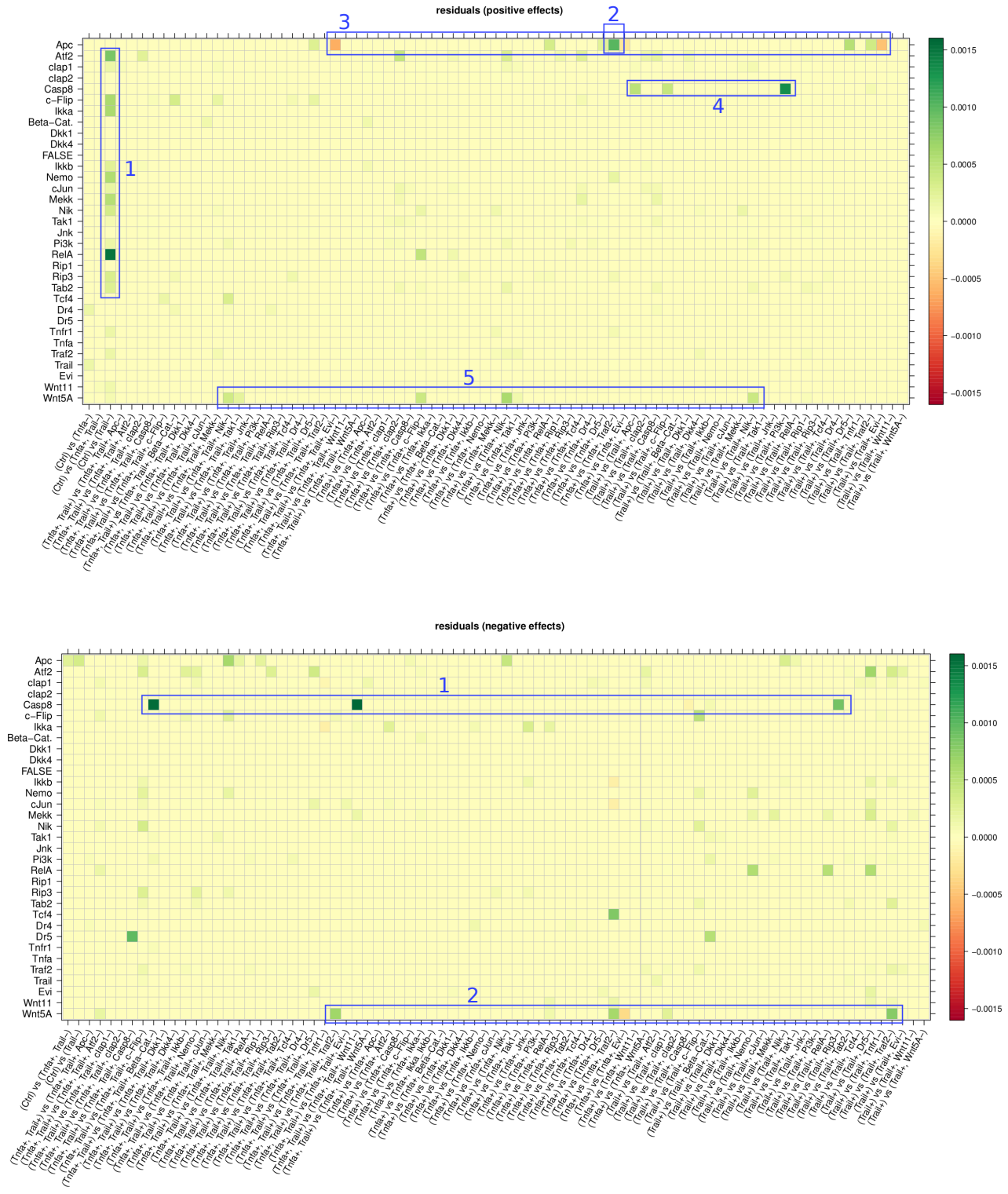


Figure 7.6: **Residuals in the Data.** Residuals in the observed response scheme for the optimal network in figure 7.5. PRM (top): Network predicts no effect, but the E-genes fit better, if one is predicted (green). Network predicts effect, but the E-genes fit better, if none is predicted (red). The same for negative effects is shown in the NRM (bottom).

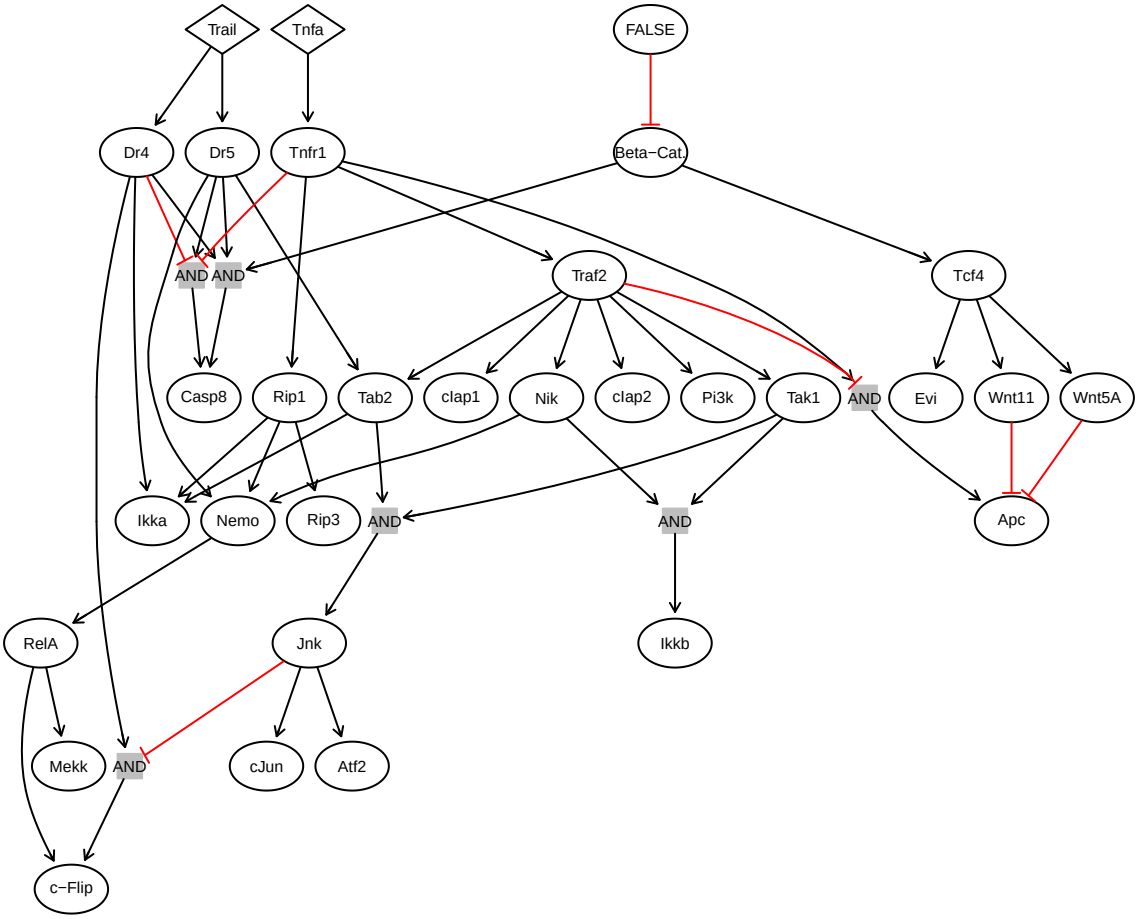


Figure 7.7: **Updated Tnf-α-Trail-WNT result.** Relearned network based on PKN D.9.

1. Casp8: E-genes of Casp8 are down-regulated during knock-downs of Beta-Catenin, Wnt11 and Rip1. Thus we add edges Beta-Catenin $\rightarrow$ Casp8, Wnt11 $\rightarrow$ Casp8, Rip1 $\rightarrow$ Casp8. However Beta-Catenin and Wnt11 are not activated by the receptors. We account for this with edges Tnfr1, Dr4, Dr5 $\dashv$ Casp8.

2. Wnt5a: E-genes of Wnt5a are down-regulated during the knock-down of Traf2, thus we add Traf2 $\rightarrow$ Wnt5a.

Before we relearn the network based on the new revised PKN (figure D.9), we fix E-gene positions Θ in the current optimum, which we also use as start network in the search. The new optimum (figure 7.7) gets a higher score. However it still has some residuals (figure D.10). The ones in the positive effects matrix are not resolvable due to the binary type of our model. For example Casp8 is up-regulated in (control) vs (Trail+) and up-regulated in (Trail+) vs (Trail+, Pik3-). Thus according to the data Casp8 has three different states instead of just 0 and 1.

As an example E-genes regulated by RelA are shown in figure 7.8.

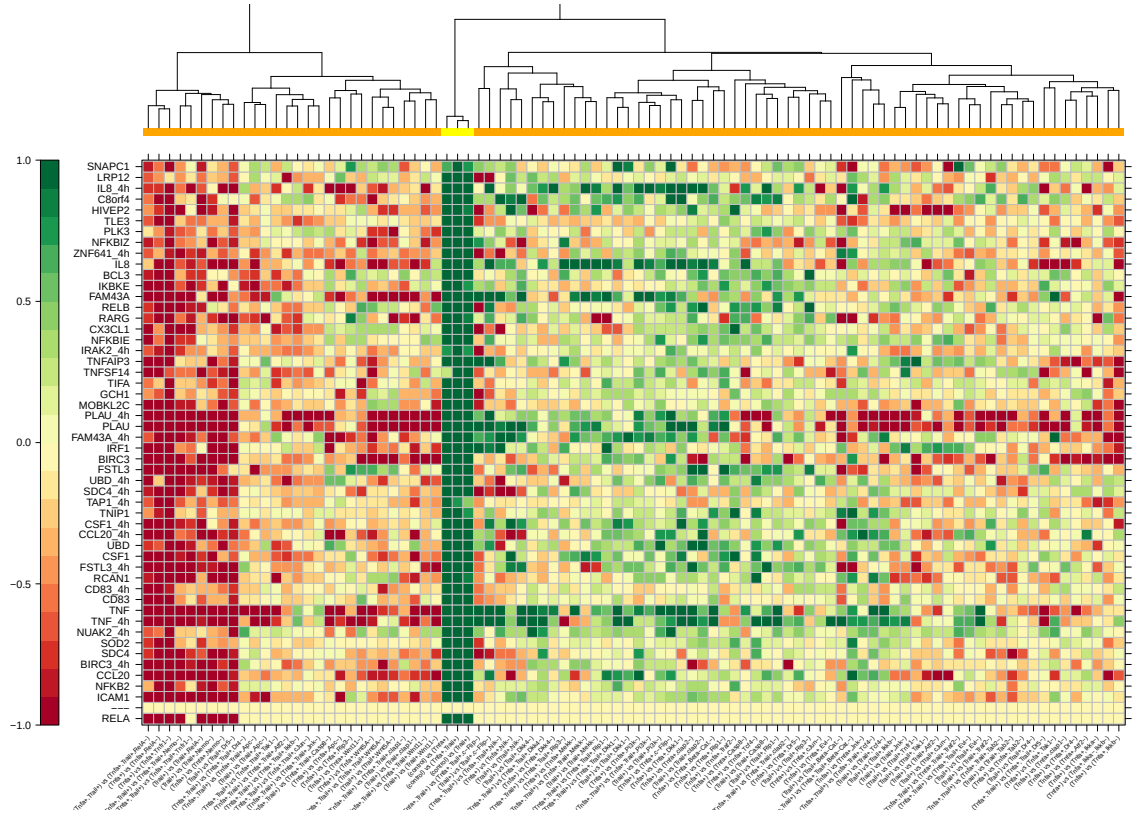


Figure 7.8: **Top RelA regulated E-genes for the updated result.** The bottom row is the expected response scheme of RelA. The other rows show the observed response schemes of attached E-genes.

We account for the non-resolvable residuals with additional manual edges (figure 7.9). We identify the following features:

1. Beta-Catenin is essential for Casp8 activation.

2. Casp8 activation by Trail is less robust during Tnf-α stimulation. It can be deactivated by both Trail receptor knock-downs in the double stimulation, but only by Dr5 knock-down in the single Trail stimulation.

3. The Trail signal to c-Flip cannot be blocked by any knock-down during the single stimulation. During the double stimulation c-Flip can be blocked by the NfκB pathway (i.e. RelA and Nemo).

4. Evi is activated by Beta-Catenin, but it is no mediator between Beta-Catenin and other WNT molecules.

5. Traf2 inhibits Apc.

6. c-Flip, Pik3 and Apc inhibit Casp8 during Trail stimulation.

7. Rip3 and Wnt11 activate Casp8 during Trail respectively the double stimulation.

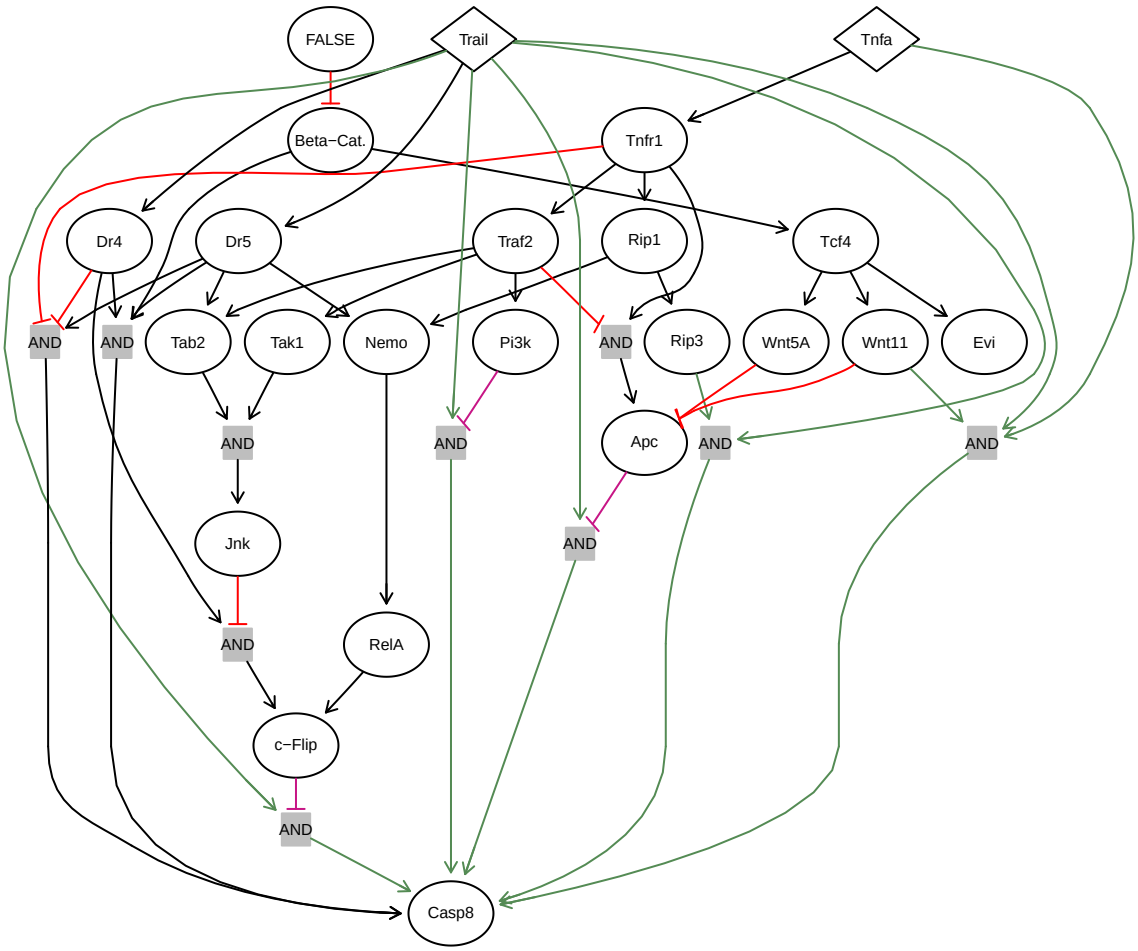


Figure 7.9: **Manually curated network.** Sub-graph of the estimated network of figure 7.7 showing the most important interactions (black and red edges). The green and pink edges show interactions which cannot be modelled with our binary states, but are implied by the residuals.

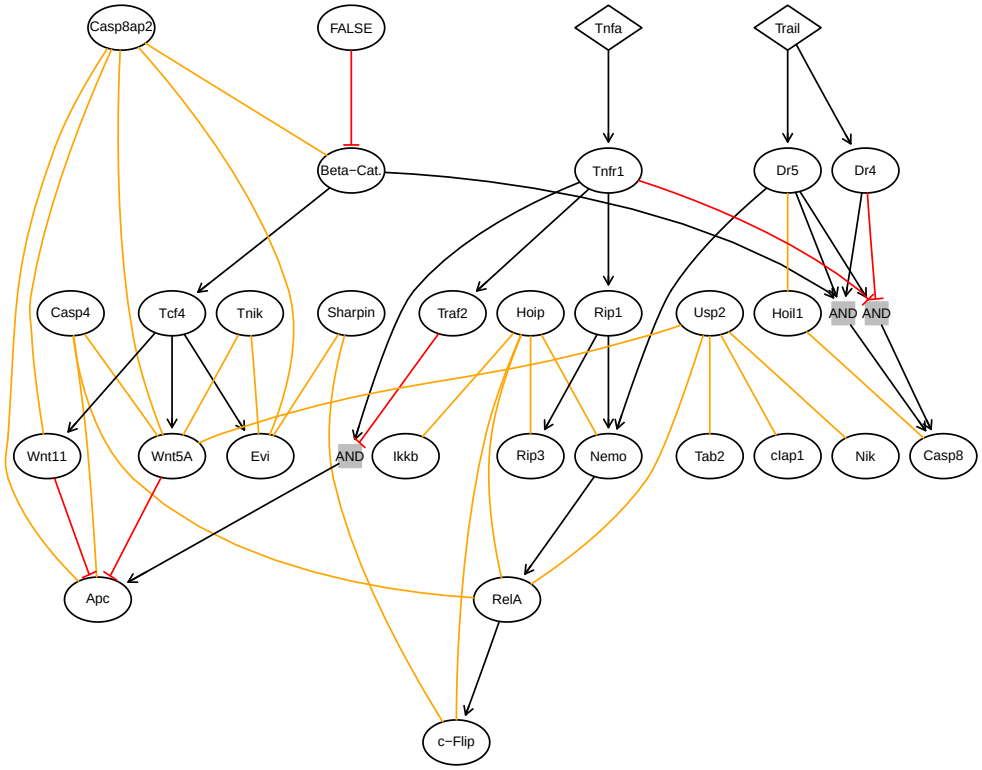


Figure 7.10: **Manually curated network with novel S-genes.** Sub-graph of the estimated network of figure 7.7 showing the most important interactions (black and red edges). The orange edges visualize associations of novel S-genes with the core network.

7.4.2 Testing unknown S-genes for interaction

We have several S-genes for which we have no prior knowledge. We employ Fisher's exact test (Fisher (1922); Agresti (1992)) to infer their position in the network.

Target E-genes of a novel S-gene must fulfill

$$(control)\ vs\ (stimulation+) > \alpha$$

and

$$((control)\ vs\ (stimulation+)) \cdot ((stimulation+)\ vs\ (stimulation+, S\text{-}gene-)) < 0$$

for any stimulation (Tnf-α, Trail, Tnf-α&Trail). We set $\alpha = log2(1.5) \approx 0.58$. For example if an E-gene is up-regulated by Tnf-α stimulation and down regulated by the knock-down of Hoip during Tnf-α stimulation, it is a target of Hoip.

We infer an undirected edge (association) between a S-gene in the network and a novel S-gene, if there is a significant overlap between their target genes. For significance we use Fisher's exact test. P-values are corrected for multiple testing with the false discovery rate (FDR) of Benjamini & Hochberg (1995). We draw an undirected edge between two S-genes, if FDR $< 10\%$. The results are shown in figure 7.10 (orange edges).

The left side of the network shows many association with the WNT pathway. These include not just known WNT molecules (Dkk1, Dkk4), but also previously mostly unknown relations (Usp2 , Casp8ap2, Casp4, Sharpin (Rivkin Elena *et al.* (2013)), Tnik

(Mahmoudi Tokameh *et al.* (2009))). Casp4 and Casp8ap2 specifically are involved with many WNT related molecules. Additionally Casp8ap2 is associated with Beta-Catenin and Casp4 with RelA, but no other RelA regulaters. Hoil1 acts a as a mediator between Dr5 and Casp8, while the other LUBAC member Hoip is involved in NFκB signalling (Nemo, RelA).

8

Applying B-NEM to time series data

B-NEM can infer networks based on different combinatorial experiments: stimulations (knock-ins), knock-downs, individually and in arbitrary combinations. However combinatorial knock-downs and knock-ins are not common practice. Much more so are standard experiments (definition 4.4), especially single stimulations combined with at most single knock-downs. Those experiments are very restrictive to the power of B-NEM, particularly for the inference of OR-gates.

In this chapter we show how B-NEM estimates Boolean features, like OR-gates, from time series data derived from standard perturbation experiments.

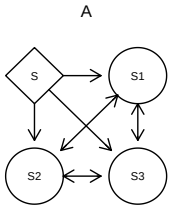
8.1 Algorithm

Let us assume we have a standard set of experiments. It consists of a control experiment without any perturbation, a single stimulation S and experiments with single inhibitions of pathway members $\mathbf{S} = \{S_1, \ldots, S_n\}$ during the stimulation. These experiments are available for several time points $t \in \{t_1, \ldots, t_m\}$. We assume further that the actual stimulation of S-genes depends on the time point t. This means, that at every time point t a certain subset of $\mathbf{S}$ is directly stimulated. See figure 8.1 for a triple example.

Algorithm 8 outlines the steps by means of the example in figure 8.1.

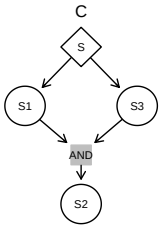
Algorithm 8 inference on time series data

1. estimate the network (figure 8.1, C) for each individual time point t (figure 8.1, B)
2. reannotate the conditions according to the results (figure 8.1, D)
3. combine the reannotated samples for all time points to one dataset
4. estimate the network from the combined data



B

base level		change level
(*control*)	vs	(S^t+)
(S^t+)	vs	(S^t+,S_1^t-)
(S^t+)	vs	(S^t+,S_2^t-)
(S^t+)	vs	(S^t+,S_3^t-)



D

base level		change level
(*control*)	vs	(S_1+,S_3+)
(S_1+,S_3+)	vs	(S_1+,S_3+,S_1-)
(S_1+,S_3+)	vs	(S_1+,S_3+,S_2-)
(S_1+,S_3+)	vs	(S_1+,S_3+,S_3-)

Figure 8.1: **Toy example for algorithm 8.** (A) PKN with three knock-down targets ($S1, S2, S3$) and the stimulation S. (B) Standard contrasts at time point t. + denotes a stimulation (knock-in) and − an inhibition (knock-down). (C) Static network estimated from contrasts of time point t (table A). (D) Reannotated contrasts resolved from time point t in table B. S_1 and S_3 have no parents except for S (network C). Thus they are directly stimulated and we replace S^t+ in the contrasts with S_1+, S_3+. In the case of ambiguity − overrules +. For example (S_1+,S_3+,S_1-) = (S_3+,S_1-).

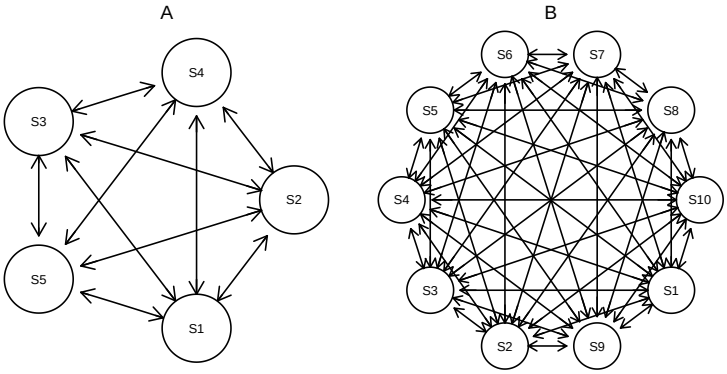


Figure 8.2: **Cyclic PKN for time series data.** Strictly positive PKN with 5 (A) and 10 (B) S-genes.

As outlined in algorithm 8, we first learn the network for each individual time point. For a single time point we model an upstream stimulation S. S positively regulates all other S-genes in the extended PKN ($S \rightarrow S_i$, 8.1, A). For example assume at time point t, S_1 and S_3 do not have parents except for S (figure 8.1, B). We then reannotate the samples of this time point as if S_1 and S_3 are directly stimulated (table 8.1, C). In the next section we validate this algorithm on simulated data.

8.2 Simulation

8.2.1 Data generation

We start with a cyclic PKN of $n \in \{5, 10\}$ S-genes with exclusively positive edges (figure 8.2). We draw a random cyclic GTN and create observed response schemes analog to chapter 5. Instead of the standard experiments, every S-gene can be stimulated or inhibited. Then we simulate up to double stimulations combined with single knock-downs. The resulting contrasts for a triple of S-genes are shown in table 8.1.

We define time points from the full dataset. A time point is defined as a subset of stim-

all contrasts			**time point X**			**time point Y**		
base level		change level	base level		change level	base level		change level
(control)	vs	(S_1+)	(control)	vs	(S_1+)	(control)	vs	(S_1+,S_2+)
(control)	vs	(S_2+)	(S_1+)	vs	(S_1+,S_2-)	(S_1+,S_2+)	vs	(S_1+,S_2+,S_3-)
(control)	vs	(S_3+)	(S_1+)	vs	(S_1+,S_3-)			
(control)	vs	(S_1+,S_2+)						
(control)	vs	(S_1+,S_3+)						
(control)	vs	(S_2+,S_3+)						
(S_1+)	vs	(S_1+,S_2-)						
(S_1+)	vs	(S_1+,S_3-)						
(S_2+)	vs	(S_2+,S_1-)						
(S_2+)	vs	(S_2+,S_3-)						
(S_3+)	vs	(S_3+,S_1-)						
(S_3+)	vs	(S_3+,S_2-)						
(S_1+,S_2+)	vs	(S_1+,S_2+,S_3-)						
(S_1+,S_3+)	vs	(S_1+,S_3+,S_2-)						
(S_2+,S_3+)	vs	(S_2+,S_3+,S_1-)						

Table 8.1: **Contrasts from a cyclic Boolean network.** *Left:* Contrasts of conditions used to calculate the expected and observed response schemes for a positive cyclic Network. + denotes a stimulation (knock-in) and − an inhibition (knock-down). *Center and Right:* Two subsets or "time points X and Y" of the contrasts.

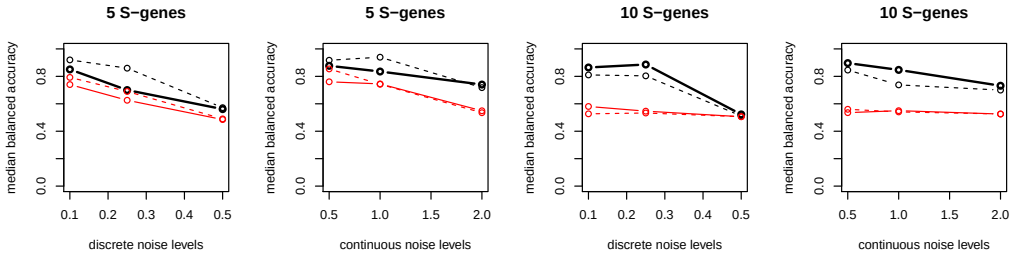


Figure 8.3: **B-NEM simulation results.** Plots for five respectively ten S-genes. Dashed lines show ten random time points and solid lines show five (five S-genes) and 20 (ten S-genes). Black lines show the median balanced accuracy of the ERS and red lines the median balanced accuracy of the hyper-edges. Discrete noise levels are $0.1, 0.25, 0.5$ and continuous noise levels are from a Gaussian distribution $\sim \mathcal{N}(0, \sigma)$ with $\sigma \in \{0.5, 1, 2\}$.

ulated S-genes with the corresponding knock-down conditions (table 8.1, center/right). This results in

$$\binom{n}{1} + \binom{n}{2} = m \in \{15, 55\}$$

different time points for $n = 5$ respectively $n = 10$ S-genes.

8.2.2 Results

We estimate the network for a random subset of five, ten and twenty time points each using algorithm 8. The results over ten independent runs are shown in figure 8.3.

B-NEM reaches a high median balanced accuracy ($= \frac{\text{sensitivity+specificity}}{2}$) for the ERS and for the hyper-edges for five S-genes. For ten S-genes the accuracy for the ERS remains high, while the accuracy for the hyper-edges drops down. Additionally the simulations show that more time points lead to higher accuracy.

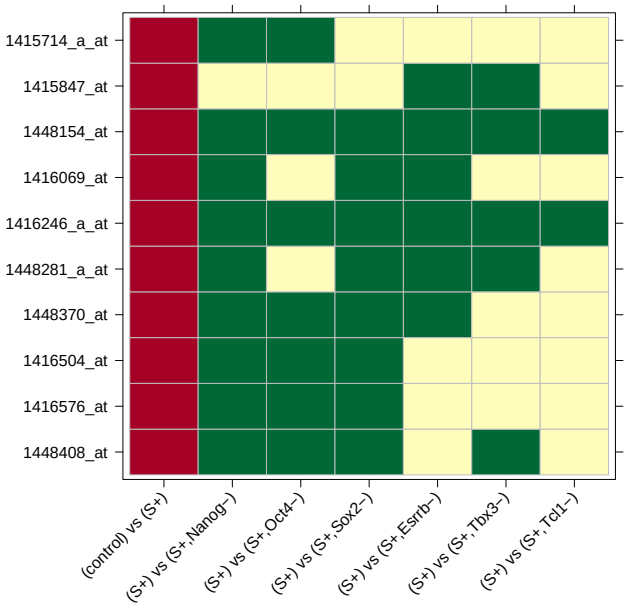


Figure 8.4: **Simulated stimulation signal.** Example of ten E-genes in the data set of Ivanova *et al.* (2006) preprocessed by Anchang *et al.* (2009). Green are the discretized effects (+1) for every knock-down. Red is our in silico added stimulation effect (−1) between positive (S) and negative control. Probe sets 1416246_a_at and 1448154_at have a standard deviation of 0 without the contrast "(*control*) *vs* ($S+$)".

8.3 Self renewal in embryonic stem cells

Ivanova *et al.* (2006) produced gene expression data from experiments with knock-downs by short hairpin RNA (shRNA; Taxman *et al.* (2010)). They target six genes involved in the self renewal of mouse embryonic stem cells: Nanog, Sox2, Oct4, Tbx3, Esrbb and Tcl1 (S-genes). The gene expression for each knock-down is measured at eight different time points with microarrays.

Before we make inference on the time series we compare B-NEM to other NEM extensions (Anchang *et al.* (2009), Froehlich *et al.* (2011), Sadeh *et al.* (2013)). We use the last time point of 122 genes as previously described in Anchang *et al.* (2009) for network reconstruction. The preprocessed data is available in the R package "nem" (Froehlich *et al.* (n.d.)). However there is no "(control) vs (stimulation+)" contrast in the preprocessed data and since the data is discretized our correlation measure could fail. For example E-genes 1416246_a_at and 1448154_at have a constant value of 1 in their observed response schemes. As a results the variation is 0 and we cannot compute a correlation coefficient. We resolve this by simulating a "(control) vs (stimulation+)" contrast which concatenates a -1 to the ORS of every E-gene (figure 8.4). This is valid, because all selected E-genes are differentially regulated between negative and positive controls (stimulation).

We use Spearman's rank correlation and size penalty $\zeta = 10^{-10}$ for the score (4.1) and the BGNS for the optimization. We start from the empty network and the PKN with identical results (figure 8.5). Our result is most similar to the one of Anchang *et al.*

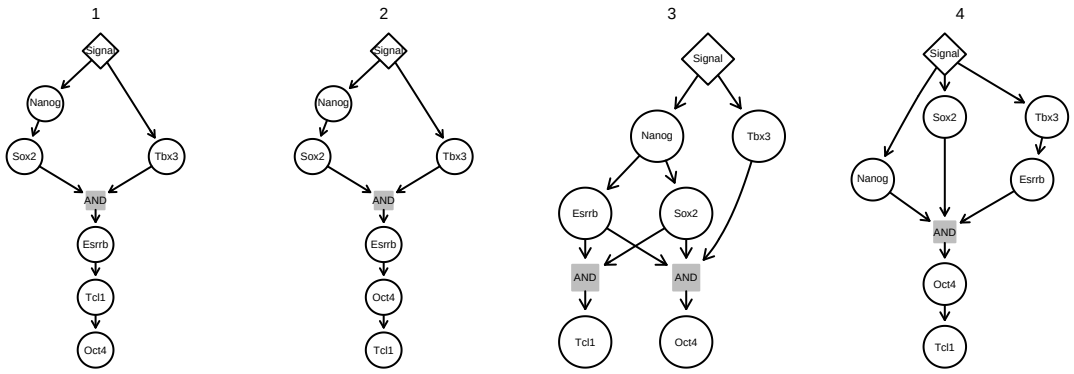


Figure 8.5: **B-NEM compared to other NEM extensions.** The transitive reductions of the results from different methods. **(1)** B-NEM with $\gamma = -\infty$. **(2)** B-NEM with $\gamma \in \{0.8, 0.9\}$. This is in accordance to the estimation of Anchang *et al.* (2009). **(3)** pNEM (Sadeh *et al.* (2013)). **(4)** Fast and efficient dynamic nested effects models (Froehlich *et al.* (2011)).

(2009) except for the switch of Oct4 and Tcl1. However Anchang *et al.* (2009) employ a null S-gene (Tresch & Markowetz (2008)) to exclude E-genes with a bad overall fit. Thus we employ the automatic E-gene selection (4.5) and try different values for γ. Values lower than 0.8 do not exclude any E-genes. For $\gamma \in \{0.8, 0.9\}$ our new optimal network is identical to the one of Anchang *et al.* (2009).

8.3.1 Resolving Dynamic Feedback

In this section, we learn the underlaying Boolean network structure from the eight time points using algorithm 8. Following the last section, we set γ to 0.8

The estimations for each separate time point are shown in figure 8.6. We resolve similar individual time points as Wang *et al.* (2014). Tcl1 is always at the bottom of the network, while Tbx3 is at the top in all eight time points. All other S-genes switch positions. Nanog moves from almost at the bottom in the early time points to the very top. In contrast Oct4 starts at the top and ends up downstream of all but Tcl1. Sox2 and Esrrb do not change their position much.

Figure 8.7, left shows the network estimated with algorithm 8. Tcl1 is concertedly activated through an AND-gate by all other S-genes except Tbx3. However Tbx3 can indirectly activate Tcl1 together with Oct4. Nanog is alternatively activated by Oct4 or Tbx3. Only Sox2 can activate Oct4. Nanog or Oct4 alternatively activate Sox2. Tbx3 is the only activator of Esrrb.

The most prominent subnetwork is the cycle consisting of Nanog, Sox2 and Oct4 (green). Wang *et al.* (2014) also identify this triple as the main feedback mechanism in this pathway. Furthermore the String[1] database shows Oct4, Nanog and Sox2 to be highly connected as well.

[1]STRING, http://www.string-db.org/, Franceschini *et al.* (2013), figure 8.7, right

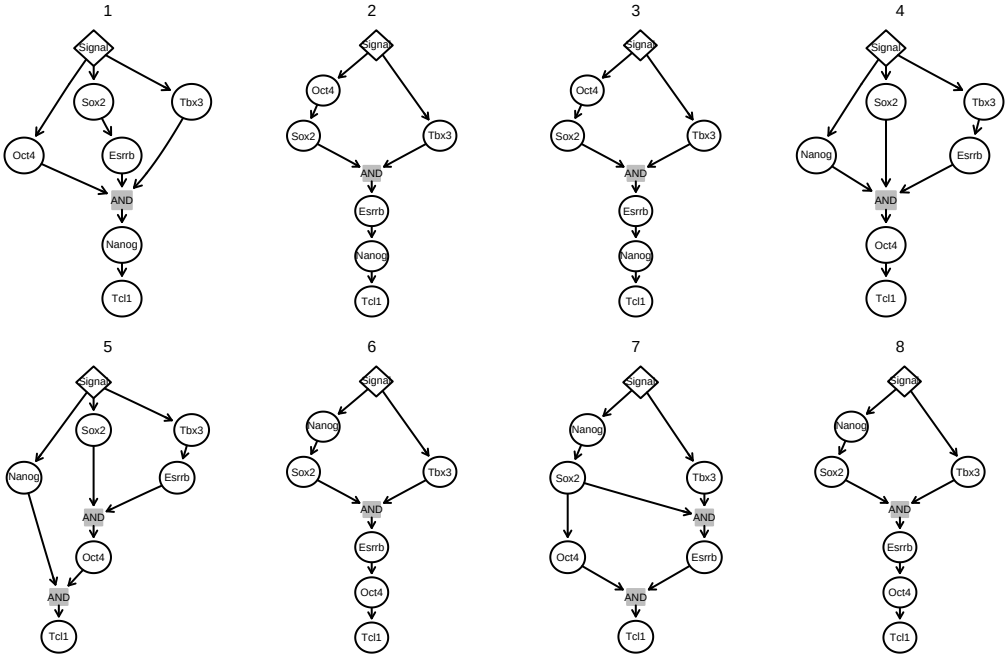


Figure 8.6: **B-NEM estimates isolated time points.** B-NEM estimations for each of the eight separate time points.

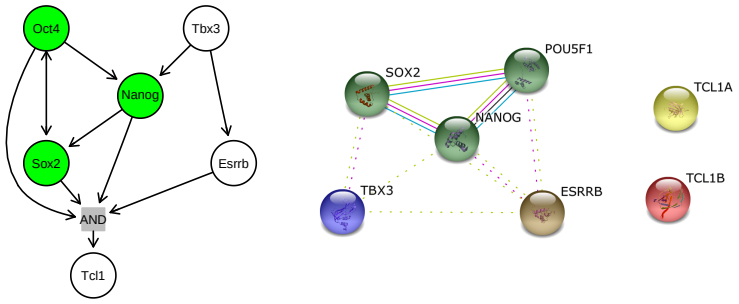


Figure 8.7: **B-NEM estimates a cyclic Boolean network from time series data.** *Left:* B-NEM estimation with algorithm 8. *Right:* String database interactions. $POU5F1$ and $TCL1A/B$ are the HGNC genesymbols (Gray *et al.* (2015)) for $Oct4$ respectively $Tcl1$.

9

Conclusion and Outlook

We introduce a novel method (B-NEM) to infer complex structures of protein signalling pathways based on indirect effects of perturbation experiments. B-NEM combines the CellNet Optimizer of Saez-Rodriguez *et al.* (2009) with Nested effect Models by Markowetz *et al.* (2005, 2007). We employ the Boolean network modelling and prior knowledge integration of the CNO, but like NEM we infer the pathway topology based on indirect transcriptional effects instead of direct measurements of the proteins involved.

We validate B-NEM on simulated data and apply it to three different cancer related datasets. The first application to B Cell receptor signalling shows B-NEMs strength, if the dataset includes combinatorial knock-downs. The double and triple knock-downs of Ikk2, p38 and Jnk allow for reconstruction of the subnetwork.

In the application to Tnf-α and Trail signalling we have only single knock-downs at our disposal. However a double stimulation of Tnf-α and Trail combined with the integration of prior knowledge lets us estimate a Boolean network of the underlaying pathway. For example we infer redundant signalling into NFκB. Nemo mediates the signal to the transcription factor RelA (NFκB), but the signal to Nemo can only be blocked by the receptors. Furthermore Trail needs the constitutively active WNT signal (Beta-Catenin) to induce apoptosis (Casp8).

In the last application we apply B-NEM to time series data derived from mouse embryonic stem cells. We show that we can use the information from each individual time point to estimate a Boolean network without combinatorial knock-downs or prior knowledge.

In the future, data production will become cheaper and predestined for combinatorial perturbation experiments. However it has to be assured that experiments with multiple simultaneous knock-downs and stimulations work as they are supposed to, because as with single knock-downs there will be unwanted effects. For example siRNA have significant off-target effects. The siRNA does not only deplete the target but also other mRNAs. Pools of siRNA, several siRNAs targeting the same gene, can reduce these off-target effects (Jackson & Linsley (2010); Hannus Michael *et al.* (2013)). The assumptions is that every siRNA has its own unique set of off-target effects, while all have the same

on-target effect. Thus the off-target effects will average out and the on-target effect will remain.

siRNA experiments result only in a knock-down of the target gene, since the depletion of the mRNA by siRNA is only partial. Thus the gene and its corresponding protein will still show some activity. However in a knock-out the gene is completely removed and fully depletes the mRNA. Knock-outs are possible with the novel DNA editing method CRISPER-Cas (CRISPER: clustered regularly interspaced short palindromic repeats, Cas: CRISPER associated protein). CRISPER-Cas is used to edit the genome and thus experimental biologists can cut out a whole gene (Beisel *et al.* (2014); Li *et al.* (2014); Shalem *et al.* (2014)). The resulting knock-out data is more suitable for B-NEM, because the Boolean formulation assumes discrete states such as active (1) or inactive (0) as in a knock-out. However B-NEM also works on knock-down data, as we have demonstrated.

Future experiments and designs will increase the power of B-NEM. However the method is still very much extendable. Other static (Vaske *et al.* (2009); Zeller *et al.* (2009); Niederberger *et al.* (2012)) and dynamic modelling approaches (Anchang *et al.* (2009); Froehlich *et al.* (2011)) have already extended the original NEM. Alternatively qualitative Boolean models as in B-NEM can be turned into a quantitative system of ordinary differential equations (ODE, Wittmann *et al.* (2009); Henriques *et al.* (2015)). This system can then be used to exploit more details in time series data. However ODEs are generally more computational expansive and too few time points can lead to overfitting.

APPENDIX A

Signal Propagation

In this chapter we discuss some additional topics regarding the signal flow in BGs and introduce algorithms we use to tackle this problem. We start with the transitive closure of a BG. Then we introduce an algorithms to calculate the activation states of the vertices. Finally we use Boolean networks to model the rules of Rock, Paper, Scissors, Lizard, Spock[1]. We also build a classifier to predict game outcomes.

A.1 Transitivity

Definition A.1 (Implied Boolean function)**.** *Let $G = (V, H)$ be a Boolean directed hyper-graph. A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is implied by G, if there is a rooted tree $T \subset G$ (orientation towards the "root") that corresponds to f: the sources (vertices without parents) of T are the inputs of f and and the sink (root) is the output.*

For example the hyper-graph 1 in figure A.1a implies the function $C = \neg A \wedge B$, because it has a corresponding hyper-edge and the function $E = \neg C \wedge D = \neg(\neg A \wedge B) \wedge D = (A \vee \neg B) \wedge D = (A \wedge D) \vee (\neg B \wedge D)$, because it corresponds to a rooted tree.

Definition A.2 (Transitive closure of a BG)**.** *Let $G = (V, H)$ be a Boolean directed hyper-graph and $\hat{G} = (V, \hat{H})$ a Boolean directed hyper-graph with the following properties:*

1. *$H \subset \hat{H}$*
2. *for every Boolean function f implied by G, there exists a set $\{h_1, \ldots, h_n\} \subset \hat{H}$ which corresponds to f.*

$\hat{G}$ is the transitive closure of G.

This definition of transitive closure is in conformity with the special case of the original Nested Effects Models (Markowetz *et al.* (2005, 2007); Tresch & Markowetz (2008)). For

[1] Sam Kass, Karen Bryla, http://www.samkass.com/theories/RPSSL.html

example consider the graph $G = \{A \to B \to C\}$. We can write $C = B = A$. G implies $C = A$. Thus the transitive closure is $\hat{G} = G \cup \{A \to C\}$.

In the context of B-NEM a Boolean hyper-graph is not necessarily equivalent to its transitive closure. If two vertices are only indirectly connected with each other, the knock-in or knock-down of intermediate vertices makes them independent. For instance if we knock-down the intermediate B in the previous example, we have $C|_G = B|_G = 0$ and $C|_{\hat{G}} = A|_{\hat{G}} \vee B|_{\hat{G}} = A|_{\hat{G}} \vee 0 = A|_{\hat{G}}$. Therefore the graphs are not equivalent for the knock-down of B, $[G] \neq \left[\hat{G}\right]$. Nevertheless, the transitive closure is still a good way to asses signalling properties, given no perturbation of the intermediate vertices. Thus we outline an algorithm (9) to construct the transitive closure or an approximation for a given BG.

Algorithm 9 transitive closure of a Boolean hyper-graph

---

1. input Boolean hyper-graph $G = (V, H)$
2. iterate over n steps ($n = |V| - 2$ is sufficient, but not minimal for the full transitive closure)
   (a) iterate over edges $\hat{h}\,(W, c) \in H$
      i. iterate over vertices $w \in \{w : w \in W, \exists h = (U, w) \in H\}$
         A. replace w in the AND-clause corresponding to $\hat{h}$ with the DNF corresponding to the hyper-edges h with w as child
         B. convert newly formed normal form c^* to a DNF
         C. add the hyper-edges corresponding to the AND-clauses of c^* to the the graph G
   (b) if no novel edges were added during the current iteration: break

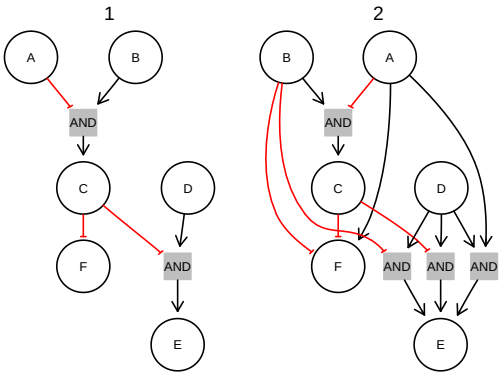
---

Small values for the step parameter n lead to an approximation of the full transitive closure. An example for a BG and its transitive closure is shown in figure A.1a. Basically, what the algorithm does, it replaces the literals in the original DNF of the vertex with other DNF(s) of its parents and uses Boolean algebra to transform the normal form back to a DNF, if necessary. For instance in figure A.1a we compute

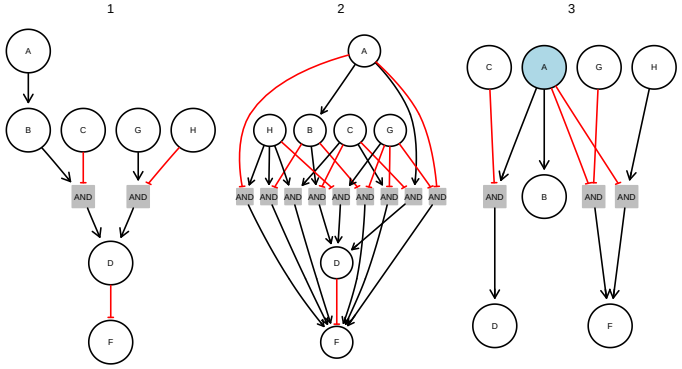
$$\begin{aligned} E &= \neg C \wedge D = \neg\left(\neg A \wedge B\right) \wedge D = (A \vee \neg B) \wedge D = (A \wedge D) \vee (\neg B \wedge D) \\ F &= \neg C = \neg\left(\neg A \wedge B\right) = A \vee \neg B. \end{aligned} \tag{A.1}$$

Figure A.1a, 2 shows the corresponding hyper-edges in the transitive closure. The result can be complex. For example in figure A.1c we add the corresponding hyper-edges to the following DNFs:

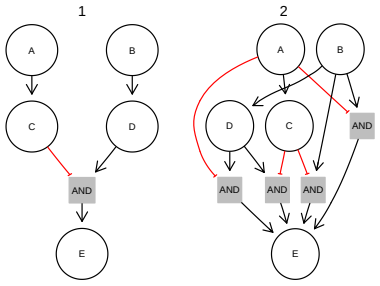
$$\begin{aligned} E &= \neg C \wedge D = \neg A \wedge D \\ E &= \neg C \wedge D = \neg C \wedge B \\ E &= \neg C \wedge D = \neg A \wedge B. \end{aligned} \tag{A.2}$$



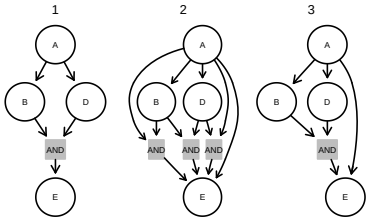
(a) Example of a BAG (1) and its transitive closure (2). If the intermediate vertex C is not perturbed, both graphs are equivalent.



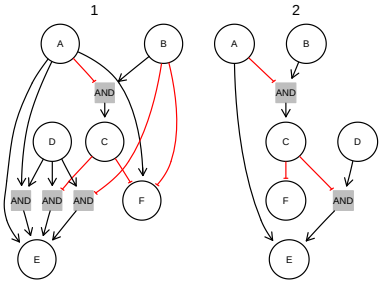
(b) The transitive closure of hyper-graph 1 is hard to read (2). If we draw only edges including A as a parent (3), we can see the (in)-direct influences of A on other vertices.



(c) A BG (1) and its transitive closure (2).



(d) A BG (1), its full transitive closure (2) and its transitive closure reduced by the absorption law (3).



(e) A BG (1) and its transitive reduction (2). Only the feed forward loop $A \rightarrow E$ stays, since $A = 1$ does not indirectly cause $E = 1$.

Figure A.1: **Examples for transitivity.** Examples for the transitive closure (a-d) and the reduction (e) of BGs.

For large graphs the transitive closure can be too complex for an investigation by eye. Therefore it is helpful to look at only a subset of the graph. For example in figure A.1b, we look at the sub-graph including all edges with vertex A in the parent set. This reduction lets us examine the (in)-direct influence of A on all other vertices of the network. Given $C = 0$, A positively regulates D and given $G = 0$ or $H = 1$, A negatively regulates F. Furthermore, the absorption law can also help to make the transitive closure less complex without loosing information (figure A.1d).

Similarly to the closure, we can compute the transitive reduction of a BG. Basically, we remove all feed forward loops of the original graph G and calculate its transitive closure $\hat{G}^*$. G^* is G without any feed forward loops. Only feed forward loops in G, which are not explained by $\hat{G}^*$ are kept and the rest removed (figure A.1e, algorithm 10).

Algorithm 10 transitive reduction of a Boolean hyper-graph

1. input Boolean hyper-graph G
2. make a copy G^* of G
3. remove all feed forward loops in G^*
4. calculate the transitive closure $\hat{G}^*$ of G^*
5. remove all feed forward loops in G which are also in $\hat{G}^*$

A.2 Simulated signal propagation

Given a boolean network Ψ B-NEM compares simulated states of S-gene with measurements of E-genes. Before we can calculate the similarity between E-genes and S-genes, we need to simulate the S-gene states for a given network and conditions. A condition C defines a subset $I \subset V$of vertices we set to 0 or 1 independently of their parents. Vertices not in I have an unknown initial state. This state is calculated based on the states of the vertices in I. We accomplish this by a recursive implementation of the algorithm 11. It is a generalization of depth first search (Tarjan (1972); Wilson (1996)) to BGs.

Algorithm 11 signal propagation

1. start with a random, S-gene S which has unknown states
2. calculate the parent set $pa(S)$
3. if all parents have states $0, 1$ for all experimental conditions, calculate the states of S for all experiments and proceed with step 1
4. if a parent has not been assigned states $0, 1$ over all experimental conditions, proceed with this parent as the new S in step 2

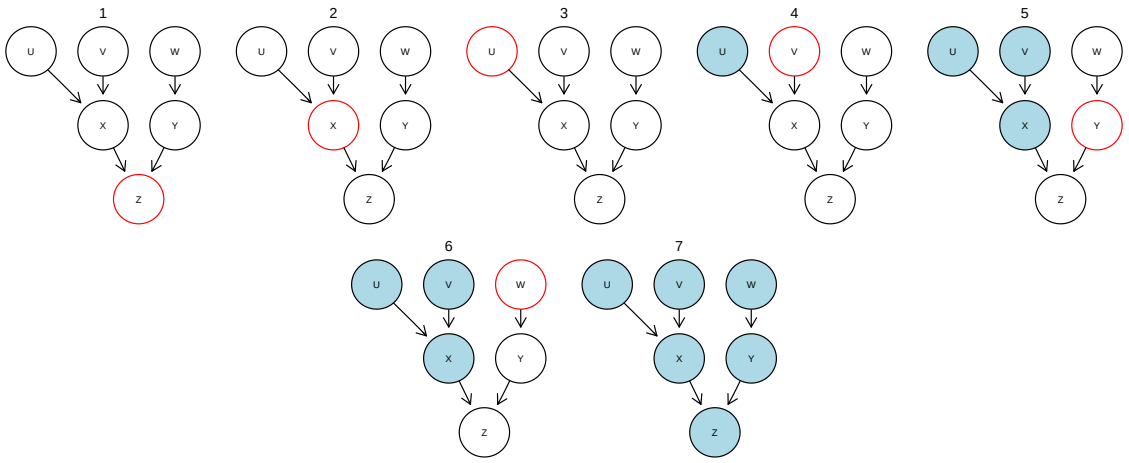


Figure A.2: **State calculation in a DAG.** We resolve the DAG in steps 1-7. Blue vertices have been assigned states 0 or 1. Vertices with a red border are currently under investigation. The starting vertex Z has parents X and Y which in turn have parents U,V and W. They are top vertices without parents and are initiated with 1 or 0. After the initiation of U, V and W, the states of X, Y and subsequently Z are calculated.

This algorithm can only process DAGs (figure A.2). In the next section we present an adjusted algorithm for graphs including cycles.

Cycles

Sometimes we have to deal with BGs that include cycles. For example if we are not sure whether S-gene A is upstream of B or B is upstream of A,

$$\text{i.e. } A \overset{?}{\longleftrightarrow} B,$$

we have to include cycles in the PKN (e.g. figure 6.1). However it is easy to see that every sub network of an extended PKN without negative edges has a well defined ERS. In contrast a PKN with negative edges can lead to an undefined ERS (figure A.5).

If we have a standard set of experiments (definition 4.4), every normal graph without hyper-edges, but with a cycle is equivalent to a DAG (figure A.3 and Zeller *et al.* (2009)). However a more complex set of experiments resolves this equivalence (figure A.4). If the PKN contains cycles and negative regulation, there are cases in which the network does not define an ERS (figure A.5). During the control experiment the stimulation $S0$ is 0. This leads to the following oscillation of states:

$$\begin{aligned}(S1 = 0) \Rightarrow (S2 = 0) \Rightarrow (S3 = 0) \Rightarrow (S4 = \neg S3 = 1) \Rightarrow (S1 = 1) \Rightarrow (S2 = 1) \\ \Rightarrow (S3 = 1) \Rightarrow (S4 = 0) \Rightarrow (S1 = 0).\end{aligned} \tag{A.3}$$

The activation states are well defined for all other experiments.

The following algorithms detect cycles using depth first search (figure A.6, Tarjan (1972)) and assign the correct states to all vertices in a positive Network (A.7, 1-2).

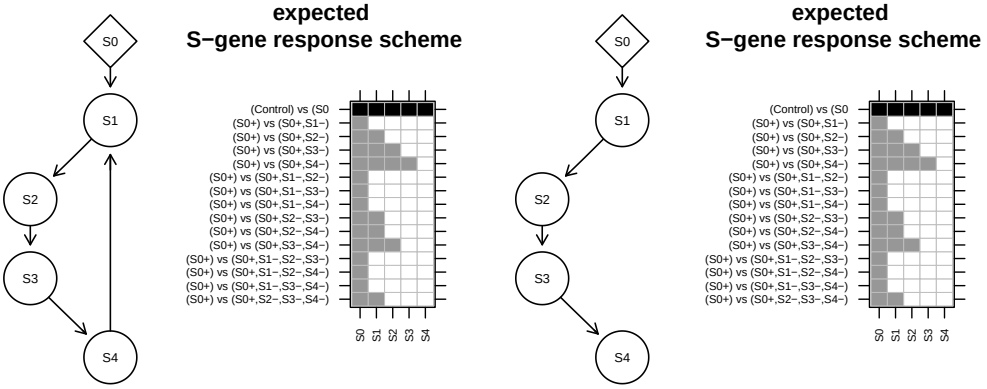


Figure A.3: **Equivalence of DAGs and graphs with cycles.** Example for two equivalent networks given standard experiments. The left network contains a cycle, the right is a DAG. The DAG is preferred due to its smaller size.

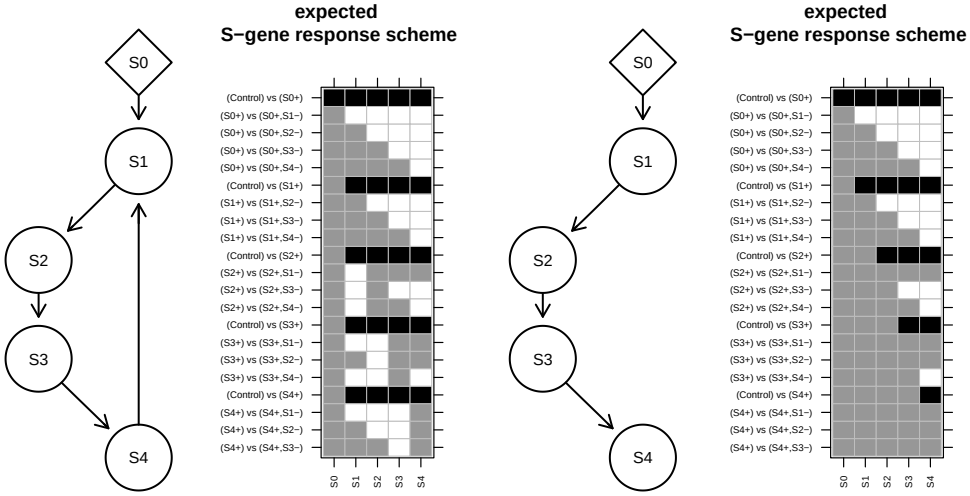


Figure A.4: **Special experiments to resolve cycles.** The same networks as in figure A.3, but with a different set of experiments. The two ERSs clearly differ and the networks are not equivalent.

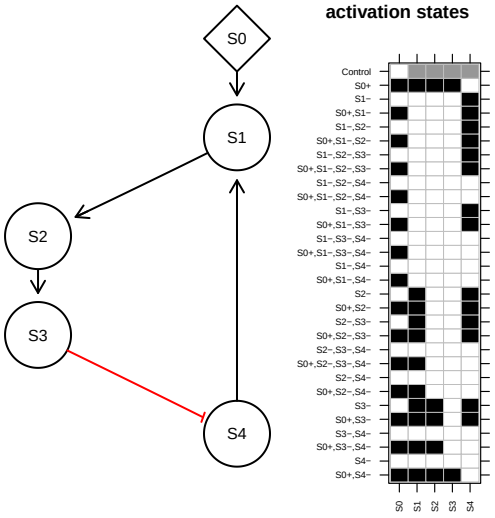


Figure A.5: **Negative cycles lead to oscillation.** Network with a cycle including a negative edge and its activation states. Black is active during the experiment, white inactive and grey undefined (=oscillating). We cannot resolve the states of S-genes $S1-S4$ in the control experiment ($S0 = 0$).

Algorithm 12 cycle detection

1. start with a random S-gene S and empty set C for children/descendents
2. calculate the parent set $pa(S)$ and put S in C
3. if there exists a $S^* \in pa(S)$ with $S^* \in C$, a cycle is detected, if not proceed with step 2 and replace S with the parents $S^* \in pa(S)$

Algorithm 13 resolve cycle

1. cycle detected between parent A and child B
2. temporarily set $B = 0$
3. calculate A
4. unset B
5. calculate B

In the example of figure A.7, 1 we start with W.

$$W = A = S \vee \neg B = S \vee \neg\neg V = S \vee V = S \vee W.$$

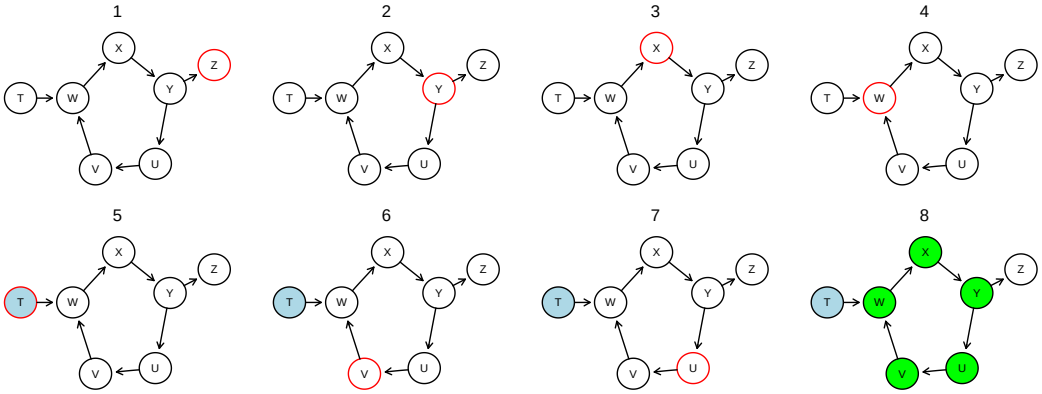


Figure A.6: **Example for cycle detection.** We resolve the cycle in steps 1-8. Let's assume the algorithm starts at vertex Z. The parents are recursively investigated via Y,X and W until vertex T. T is assigned its state (blue). V, the second parent of W, leads to U. U has Y as a parent, but Y has already been a "grand"-child of U and therefore a cycle has been detected (green).

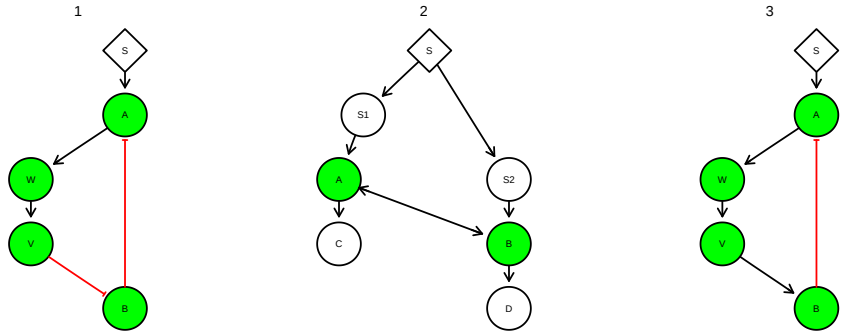


Figure A.7: **Graphs with different kinds of cycles.** Examples of graphs with positive cycles (1, 2), which our signal propagation algorithm can handle. The graph with the negative cycle (3) produces oscillating states for each vertex in the cycle and we cannot resolve a real steady state.

Next we set $V = 0$ and calculate W.

$$W = S \vee V = S \vee 0 = S.$$

Now we unset and calculate V.

In the second example (figure A.7, 2) we calculate

$$A = S1 \vee B = S \vee S2 \vee A = S \vee A.$$

We set $B = 0$ and calculate

$$A = S1 \vee 0 = S, B = A \vee S2 = S.$$

Negative cycles are more complicated, because they can lead to oscillating states. Figure A.7, 3 shows an example. A is defined by its own negation.

$$A = S \vee \neg B = S \vee \neg V = S \vee \neg W = S \vee \neg A.$$

We can detect the cycle and use the algorithm to calculate a steady state, if possible, and a pseudo steady state otherwise. In a pseudo steady state we randomly initiate one vertice of the oscillating cycle with 0 or 1 and compute the remaining unknown states. Fortunately conditions with perturbations lead to few pseudo steady states (figure A.5), because a perturbed vertice is independent of the hyper-graph. For example if a vertice X in the cycle is initiated with a 0 (knock-down) or 1 (knock-in), the states of all vertices in the cycle are well defined and there is no oscillation, because X always remains in its initiated state.

A.3 Different Problem - same Method

Several different problems in science have the same abstracted mathematical formulation. Hence the same method for one problem can be applied to a different problem. For example Ross & Markowetz (2016) transfer the concept of NEM from pathways to oncogenetic trees inferred from single cell data (oncoNEM). Oncogenetic trees model the tumor evolution in specific cancer tissues. The root of the tree is usually the unmutated cell population followed by clones with increasing numbers of mutations. Basically Ross & Markowetz (2016) replace S-genes with clone types, E-genes with single cells and effects with mutations. Thus if clone B evolved from clone A ($A \rightarrow B$), the number of single cells with mutations of type clone B are a noisy subset of single cells with mutations of type clone A.

Conceptually similar to Ross & Markowetz (2016), we use Boolean networks often used for inference and prediction in biology (Saez-Rodriguez *et al.* (2009, 2011); Pirkl *et al.* (2016)) and use them to model the game Rock, Paper, Scissors. Additionally we use Boolean networks as an alternative to established classification methods.

Rock, Paper, Scissors

Rock, Paper, Scissors is a game used to break a gridlocked argument between two people (Wikipedia (2015)). They both count to three and simultaneously make one of three specific signs:

- Rock = fist
- Paper = all fingers extended and held together
- Scissors = index finger and middle finger extended and held apart

The rules are

- Rock crushes Scissors
- Paper covers Rock
- Scissors cuts Paper

For example if person A shows Rock and person B shows Scissors, person A wins ($Rock = 1$) and B loses ($Scissors = 0$). We model those rules as a Boolean network. The literals are $V = \{Rock, Paper, Scissors\}$ and the corresponding Boolean network is

$$\Psi = (Rock = \neg Paper, Paper = \neg Scissors, Scissors = \neg Rock)\,.$$

The sign(s) not used by the two players are set to 0. We introduce the convention, if both players show the same sign, both win.

Rock, Paper, Scissors, Lizard, Spock

Rock, Paper, Scissors, Lizard, Spock (RPSLS, figure A.8, left) is an extension to the original game to lower the probability of a tie. RPSLS introduces additional rules:

- Rock crushes Lizard (hand formed to "sock"-puppet)
- Paper disproves Spock (the vulcan salute[2])
- Scissors decapitates Lizard
- Lizard eats Paper and poisons Spock
- Spock smashes Scissors and vaporizes Rock

If player A shows Rock and player B shows Paper, the literals set to 0 are

$$I = \{Scissors, Lizard, Spock\}\,.$$

The vertices $Rock$ and $Paper$ have to be resolved by signal propagation given a Boolean network. The rules dictate $Rock = 0$ and $Paper = 1$.

We can simulate a dataset given the above rules. The datamatrix $D = (d_{ij} \in \{0, 1\})$ with all possible game combinations consists of

$$\binom{5}{3} + \binom{5}{4} = 15$$

[2]Wikipedia (2016b)

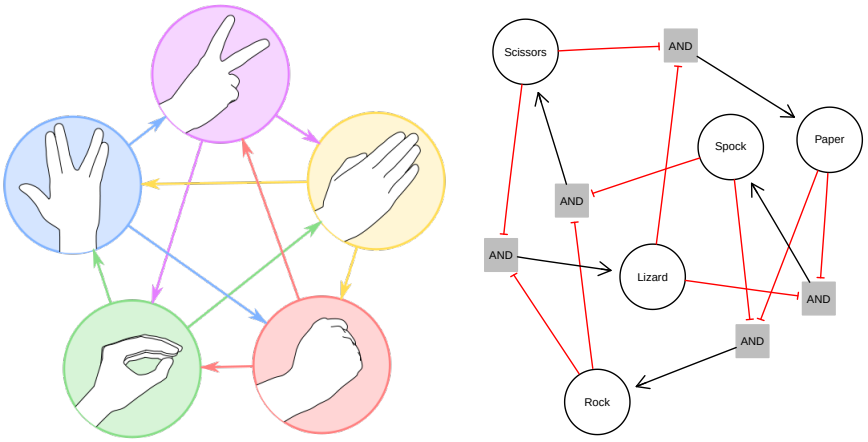


Figure A.8: **Rock, paper, scissors, lizard, Spock.** *Left:* A graphical representation of Rock, Paper, Scissors, Lizard, Spock (Wikipedia (2015)). An edge denotes the winning sign (parent) and loosing sign (child). *Right:* The Boolean hyper-graph denoting the rules.

unique[3] game outcomes (columns) for 5 literals (rows). We do not use any prior network, but the complete set of possible hyper-edges. We resolve the underlaying Boolean network with the CNO (Saez-Rodriguez *et al.* (2009)), the BGNS (algorithm 7 with starts: empty network, PKN) and adjusted state simulation algorithms (11-13) to resolve the Boolean network (figure A.8, right, equation (A.4)).

$$\begin{aligned}\Psi = (&Rock = \neg Paper \wedge \neg Spock, Paper = \neg Scissors \wedge \neg Lizard,\\ &Scissors = \neg Rock \wedge \neg Spock, Lizard = \neg Rock \wedge \neg Scissors,\\ &Spock = \neg Paper \wedge \neg Lizard)\,.\end{aligned} \tag{A.4}$$

Supervised learning with Boolean networks

We use 50% of the random unique experiments (training set) and learn a network as before. We then use this network to predict the outcome of the games (experiments) not used for learning (test set). We compare our method with the R implementations of support vector machines (SVM, Meyer *et al.* (2014)), neural networks (NN, Venables & Ripley (2002)) and self organizing maps (SOM, Wehrens & Buydens (2007)) with default parameters.

The results of 1000 runs are shown in figure A.9. Even though the correct network is hard to learn from incomplete data, the sensitivity and specificity of the predicted winners are high. Specificity is almost equal for all four methods. However Boolean networks reach the highest sensitivity.

[3]Quadruple inhibitions (if both players show the same sign) and triple inhibitions (if both players show different signs) of the five vertices.

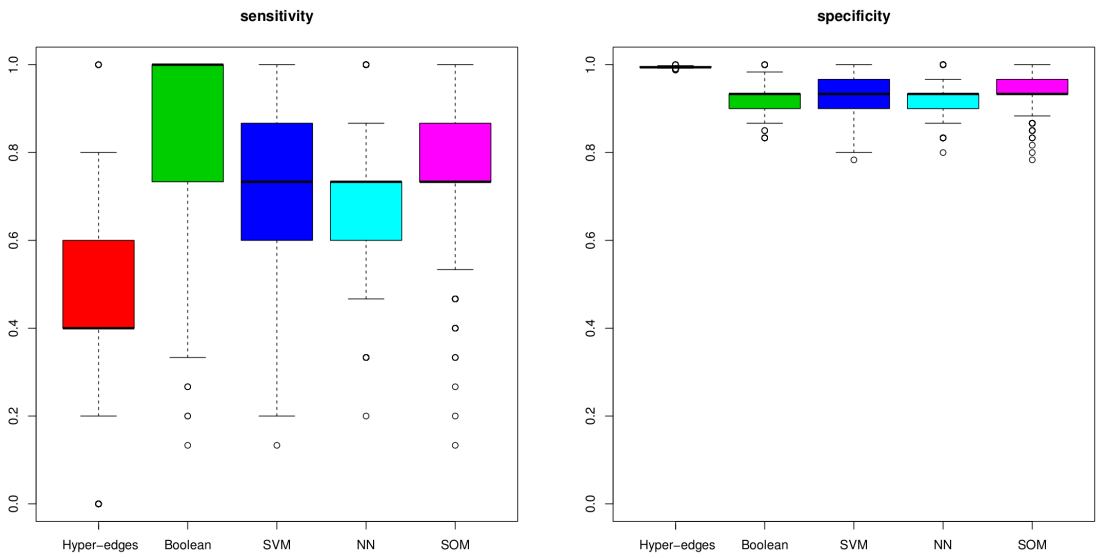


Figure A.9: **Prediction sensitivity and specificity.** Boxplots of sensitivity and specificity for the predicted hyper-edges (only Boolean networks) and the predicted winner(s) for 1000 runs.

APPENDIX B

Similarity Measures

In this section we review several similarity measures which can be used for $\mathcal{A}$ in the score of B-NEM (equations (4.1),(4.5)).

Cosine similarity

Cosine similarity (Singhal (2001); Sidorov *et al.* (2014)) for two vectors $X, Y \in \mathbb{R}^n$ with $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$ uses the euclidean dot product

$$< X, Y >= \sum_{i=1}^{n} x_i \cdot y_i$$

and norm

$$\|.\| = \sqrt{< X, X >}.$$

It is defined as

$$\mathcal{C}(X, Y) = \frac{\langle X, Y \rangle}{\|X\| \cdot \|Y\|} \in [-1, 1] \text{ (Schwarz inequality, Lang (2000))}$$

$\mathcal{C}(X, Y)$ is the cosine of the angle θ between vectors X and Y in the two dimensional euclidean space $\mathbb{R}^2$. The smaller θ, the more similar are X and Y. An angle of 180 degrees corresponds to a cosine similarity of -1 and one of 0 degrees to 1. $\mathcal{C}$ is scale, but not shift invariant.

Correlation

Correlation (Kendall (1938); Fahrmeir *et al.* (2007)) is a general measure of association, which can be interpreted as a similarity measure.

Pearson's r

Pearson's correlation for two variables X and Y is defined as

$$\operatorname{Cor}(X,Y) = \frac{\operatorname{Cov}(X,Y)}{\sqrt{\operatorname{Var}(X)\operatorname{Var}(Y)}} = \frac{\operatorname{E}(X\cdot Y) - \operatorname{E}(X)\cdot\operatorname{E}(Y)}{\sqrt{\left(\operatorname{E}(X^2) - \operatorname{E}(X)^2\right)\left(\operatorname{E}(Y^2) - \operatorname{E}(Y)^2\right)}}$$

with first moment E, variance Var and covariance Cov. If we rewrite the correlation with the sample moment $\bar{X}$ we have

$$\operatorname{Cov}(X,Y) = \tfrac{1}{n}\left\langle X - \overline{X}, Y - \overline{Y}\right\rangle.$$

and analogously

$$\operatorname{Var}(X) = \frac{1}{n}\left\langle X - \overline{X}, X - \overline{X}\right\rangle = \frac{1}{n}\left\|X - \overline{X}\right\|^2.$$

Thus it follows for Pearson's sample correlation

$$r_{XY} = \mathcal{C}\left(X - \overline{X}, Y - \overline{Y}\right).$$

Pearson's correlation is therefore not only scale, but also shift invariant. Any shift is subtracted with the mean.

Spearman's ρ

If a nonlinear relationship between X and Y is assumed, we can use Spearman's rank correlation ρ.

$$\rho_{XY} = r_{\operatorname{rank}(X)\operatorname{rank}(Y)} = \mathcal{C}\left(\operatorname{rank}(X) - \overline{\operatorname{rank}(X)}, \operatorname{rank}(Y) - \overline{\operatorname{rank}(Y)}\right).$$

It has to be taken into consideration, that ranking the ORS for every E-gene takes more time. Fortunately the ORS has to be ranked only once before the optimization. Additionally the length of the intervals between subsequent values $\{1, 0, -1\}$ in the ERS is constant 1. Thus ranking the ERS is just a shift and scale operation (average ranks, Fahrmeir *et al.* (2007)). However Pearson's r is shift and scale invariant. Therefore we do not need to rank the ERS at all. This fact makes Spearman rank correlation practically as fast as Pearson's in the case of B-NEM.

Kendall's τ

For the sake of completeness we have a look at Kendall's rank correlation τ. Even though it takes more time to compute than the others, it might hold some interesting properties such as faster convergence for increasing sample sizes and higher mathematical tractability (Gilpin (1993)).

For the two sample vectors $X = (x_i)_{i\in 1,\ldots,n}, Y = (y_i)_{i\in 1,\ldots,n}$ all pairs i, j are examined for concordance. A pair i, j is concordant, if $x_i < x_j$ and $y_i < y_j$ or $x_i > x_j$ and $y_i > y_j$. A pair is discordant, if $x_i < x_j$ and $y_i > y_j$ or $x_i > x_j$ and $y_i < y_j$. A pair i, j is neither, if $x_i = x_j$ or $y_i = y_j$. Kendall's rank correlation is defined as

$$\tau_{XY} = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{\frac{1}{2}n(n-1)}.$$

Distance

Given two vectors $s, e \in \mathbb{R}^l$ of the S-gene respectively E-gene, a distance measure is an intuitive way to assign a similarity score. For example the euclidean distance is defined as

$$\mathrm{D_e}(s, e) = \|s - e\| .$$

However we must consider, that the E-gene data needs to be normalized or discretized beforehand or we get obscure results. For example the euclidean distance between $S_i = (1, 0, 0, 0)$ and $E_j = (7.1, 0.2, 0.1, -0.3)$ is greater than between S_i and $E_k = (0.4, 0.2, 0.1, -0.3)$ (equation (B.1)), but E_j is clearly the better "fitting" E-gene. Its effects are just not scaled to $[-1, 1]$.

$$\mathrm{D_e}(S_i, E_j) \approx 6.1 > 0.7 \approx \mathrm{D_e}(S_i, E_k) . \tag{B.1}$$

The implementation in the R package "flexclust" (R Core Team (2014); Leisch (2006)) allows for a fast computation. Additionally "flexclust" includes several other distance measures besides euclidean.

APPENDIX C

Normalization to $[0, 1]$

In our B-NEM approach in chapter 4 our input ORS foldchanges from differential gene expression. As an alternative to foldchanges we can normalize the raw expression values to $[0, 1]$ and use the mean squared error from Saez-Rodriguez *et al.* (2009). We achieve this normalization with a method similar to Curry (2013). The method is based on the assumption that a gene, relevant in the context of the experiments, has basically two states 0 and 1. So most of its values are effectively either 0 or 1 or are at least close to them. We assume a small transition phase ($0 << x << 1$) of just a few values further away from both 0 or 1.

Algorithm 14 and equation (C.1) show the details of the normalization. In short we start with a 2-means clustering (MacQueen (1967)) of the log2 normalized expression values of a gene over all samples. Then we normalize the silhouette scores (Rousseeuw (1987)) to $[0, 1]$.

$$x_i^{norm} = \frac{(-1)^{c_i-1}S(x_i) - \min_j \{(-1)^{c_j-1}S(x_j)\}}{\max_j \left\{(-1)^{c_j-1}S(x_j) - \min_k \{(-1)^{c_k-1}S(x_k)\}\right\}}. \tag{C.1}$$

$c_i \in \{1, 2\}$ denotes the cluster of x_i and $S(x_i)$ the silhouette score of x_i given the 2-means clustering $c = (c_i)$. We assume without restriction, that the values in cluster two are the low values and in cluster one the high values. Figure C.1 visualizes the transformation on a toy gene.

Algorithm 14 normalization to $[0, 1]$

1. 2-means clustering of gene expression values
2. replace values with silhouette scores
3. multiply values from cluster two with -1
4. subtract minimum value from all values
5. divide all values by the maximum value

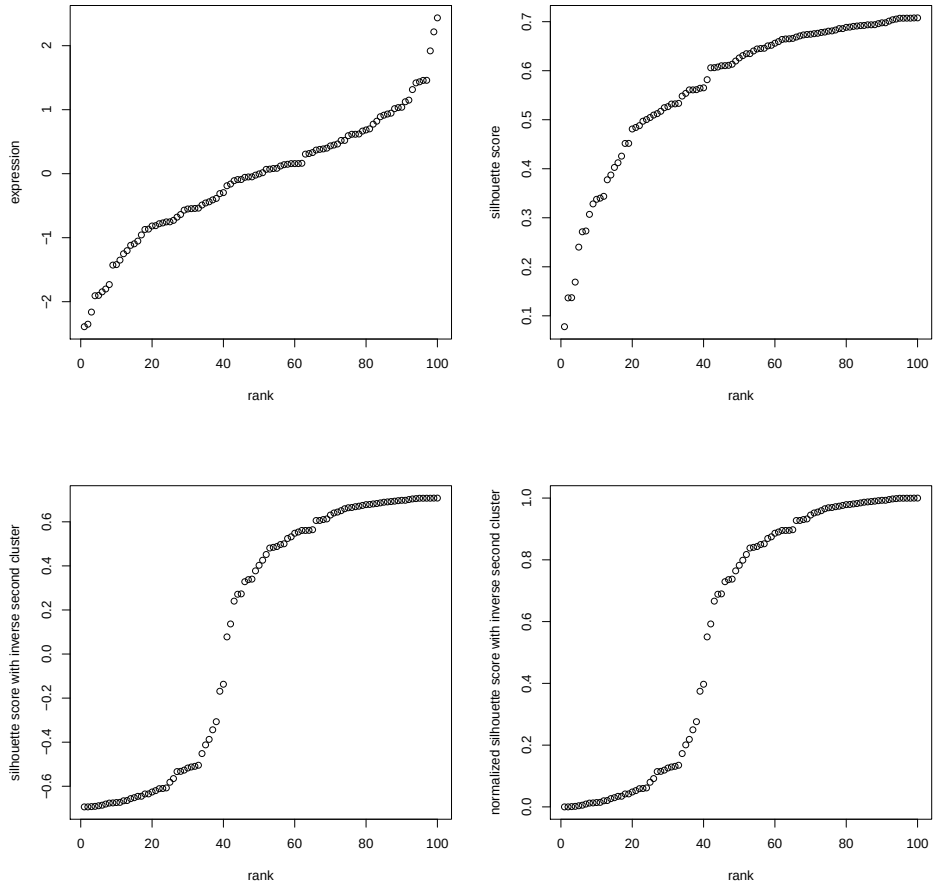


Figure C.1: $[0, 1]$**-normalization.** Toy example. Normalization of the expression values to $[0, 1]$ in three steps. The raw data (top-left), silhouette scores (top-right), silhouette scores of cluster two inverted (bottom-left) and scaled to $[0, 1]$ (bottom-right).

APPENDIX D

Supplementary Figures

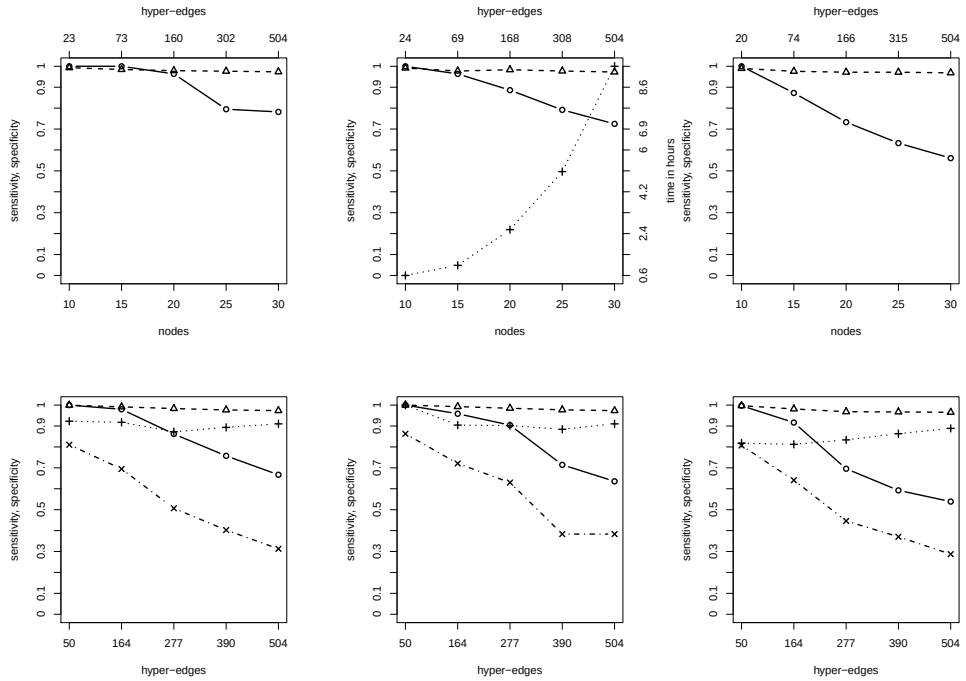


Figure D.1: **Simulation results.** Same as in figure 5.2 except with continuous noise $\sigma \in \{0.5, 1, 2\}$.

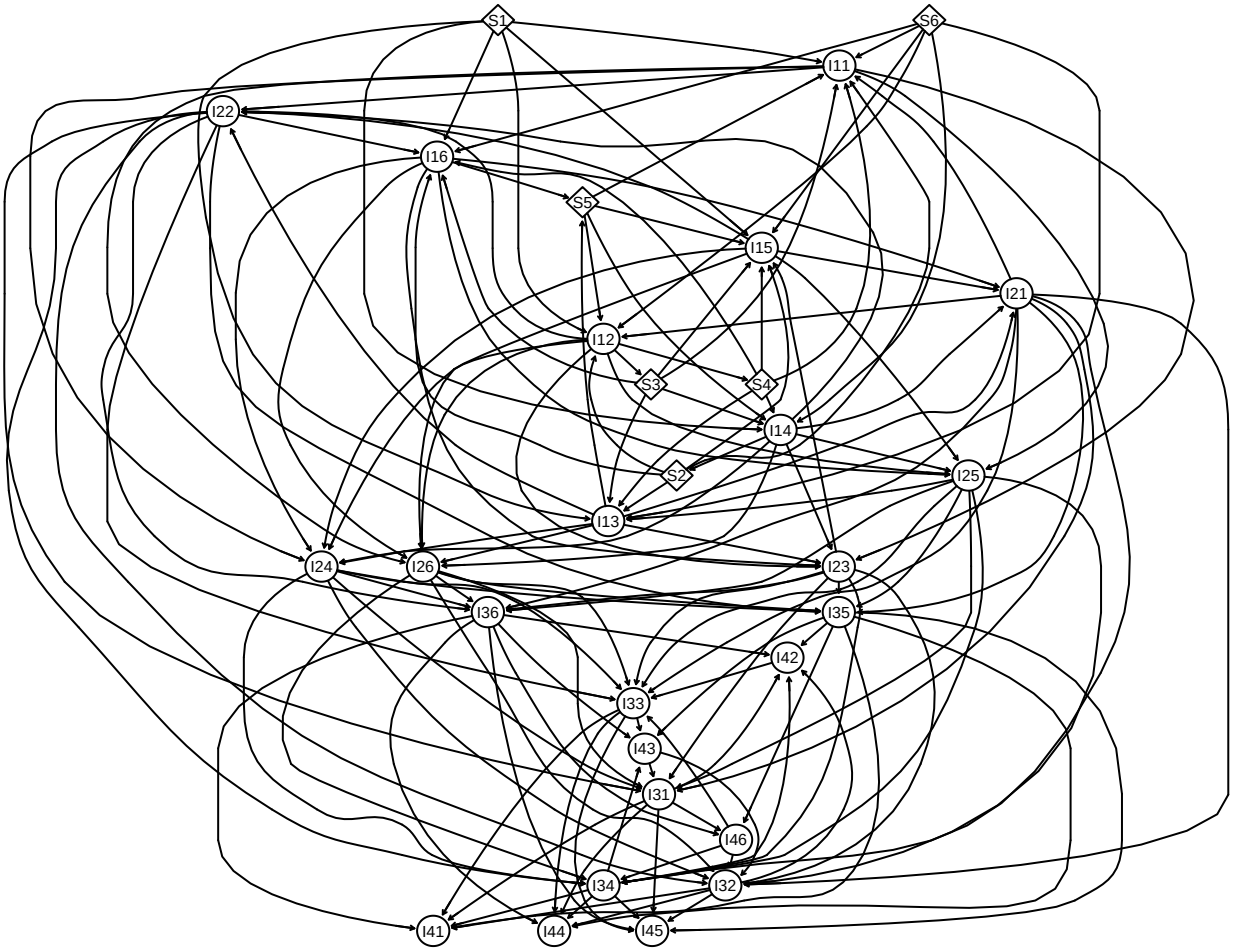


Figure D.2: **Positive cyclic PKN.** Example of a randomly created Super-PKN with 30 nodes and 144 edges in a normal cyclic graph. $S1$ to $S6$ are nodes which can be set to 0 or 1 as possible stimulations. A node denoted with I can be inhibited and if not is set by the states of its parents. Each edge has a 10% chance of being reversed. Extending this PKN with AND gates of size 2 leads to a hyper-graph with roughly 500 hyper-edges.

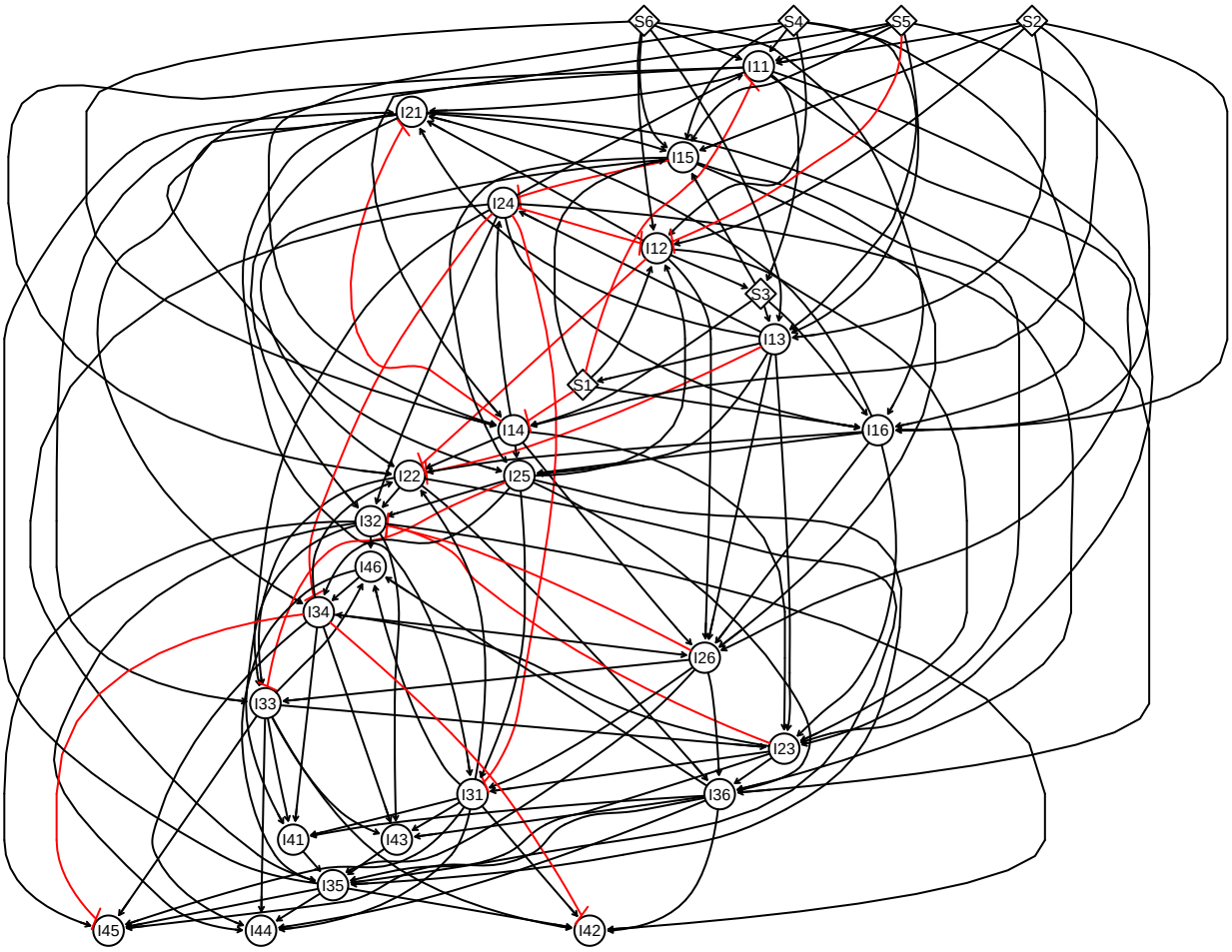


Figure D.3: **General cyclic PKN.**Example of a randomly created Super-PKN with 30 nodes and 144 edges in a normal cyclic graph. $S1$ to $S6$ are nodes which can be set to 0 or 1 as possible stimulations. A node denoted with I can be inhibited and if not is set by the states of its parents. Each edge has a 10% chance of being reversed and another 10% chance of being negative. Extending this PKN with AND gates of size 2 leads to a hyper-graph with roughly 500 hyper-edges.

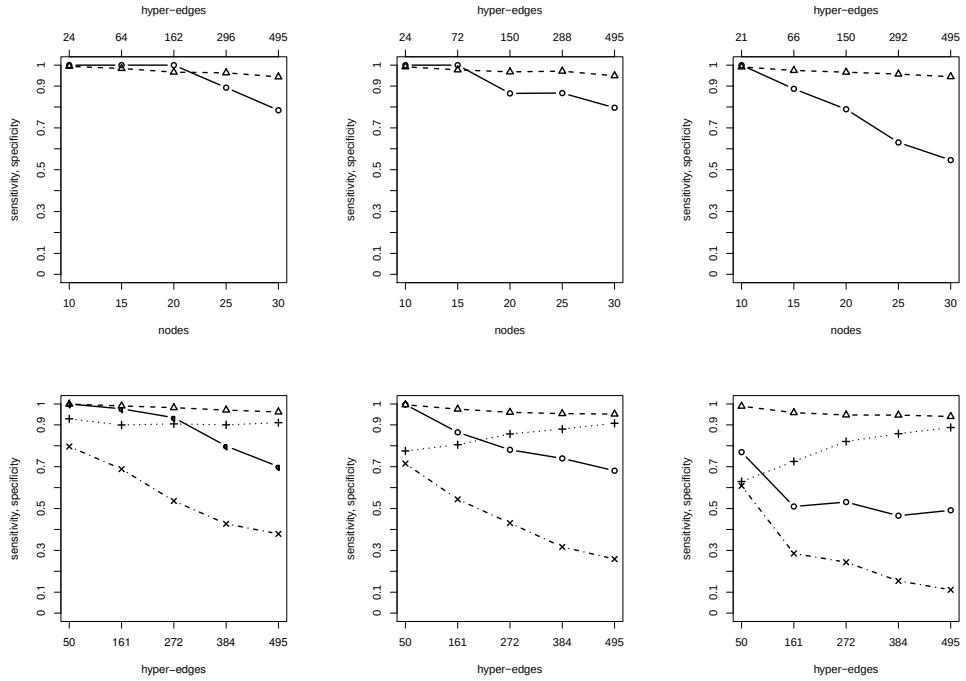


Figure D.4: **Simulation results for cyclic prior.** Same as in figure 5.2 except we used the cyclic PKN (figure D.2) instead of the DAG (figure 5.1).

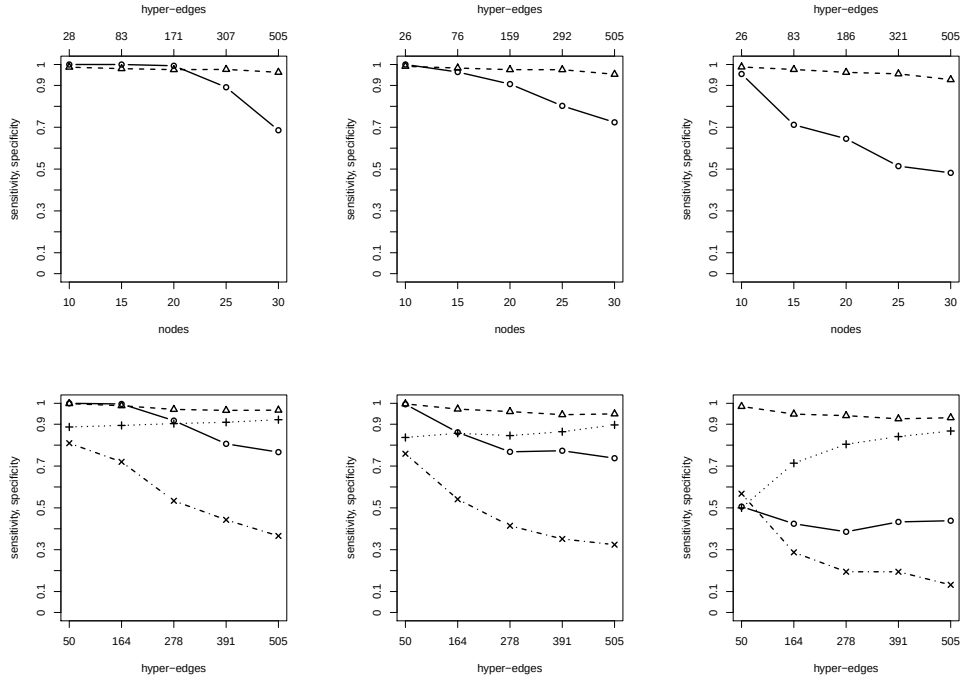


Figure D.5: **Simulation results for cyclic prior with negation.** Same as in figure D.4 except we allowed negative edges in the cyclic PKN (figure D.3).

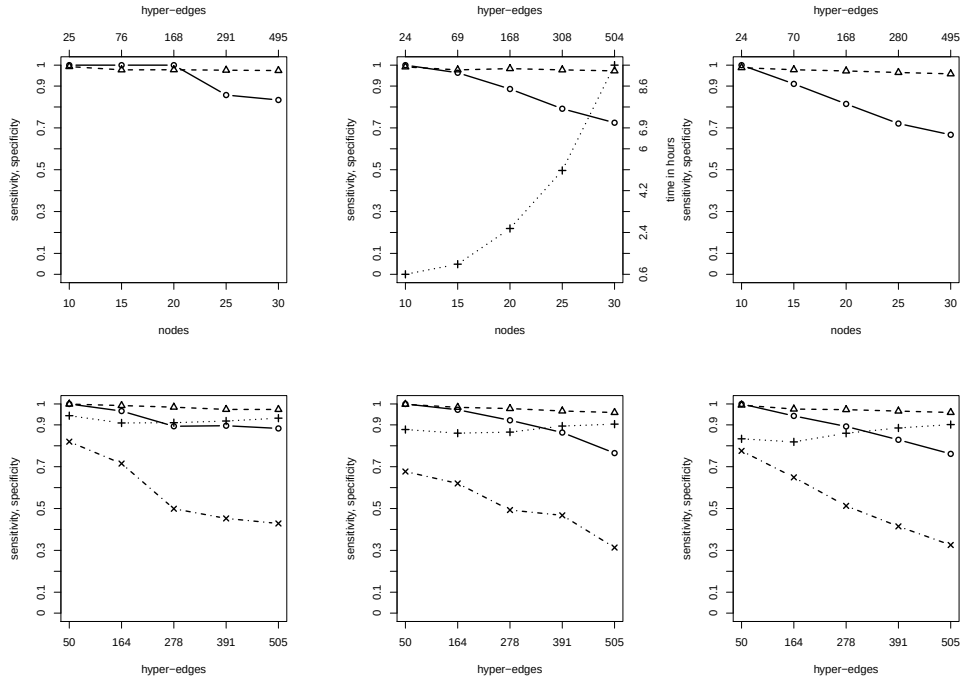


Figure D.6: **Simulation results for cyclic prior.** Same as in figure D.4 except with continuous noise $\sigma \in \{0.5, 1, 2\}$.

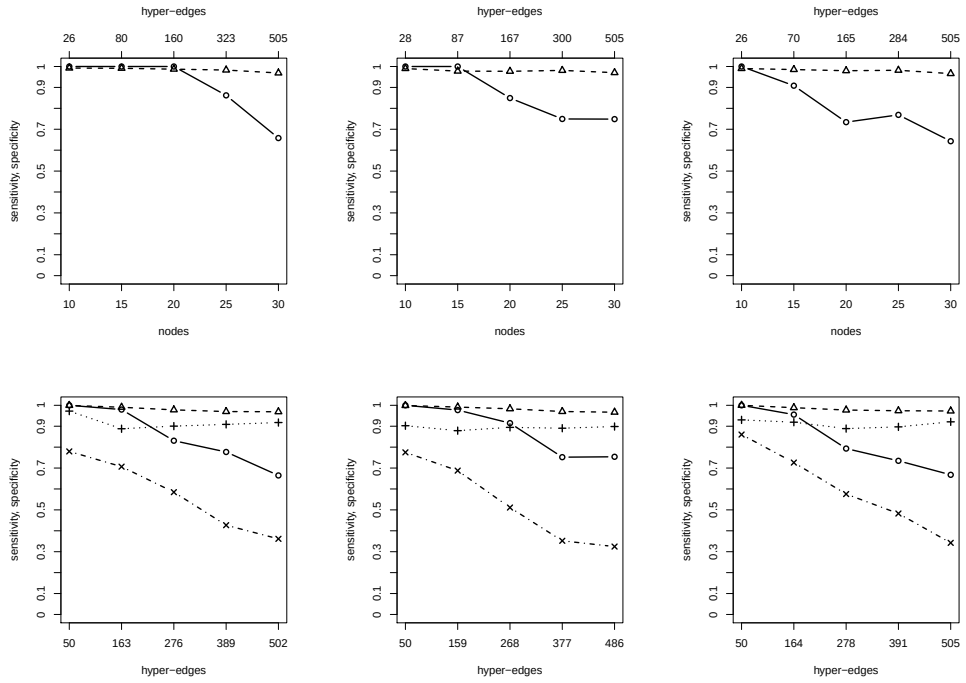


Figure D.7: **Simulation results for cyclic prior with negation.** Same as in figure D.5 except with continuous noise $\sigma \in \{0.5, 1, 2\}$.

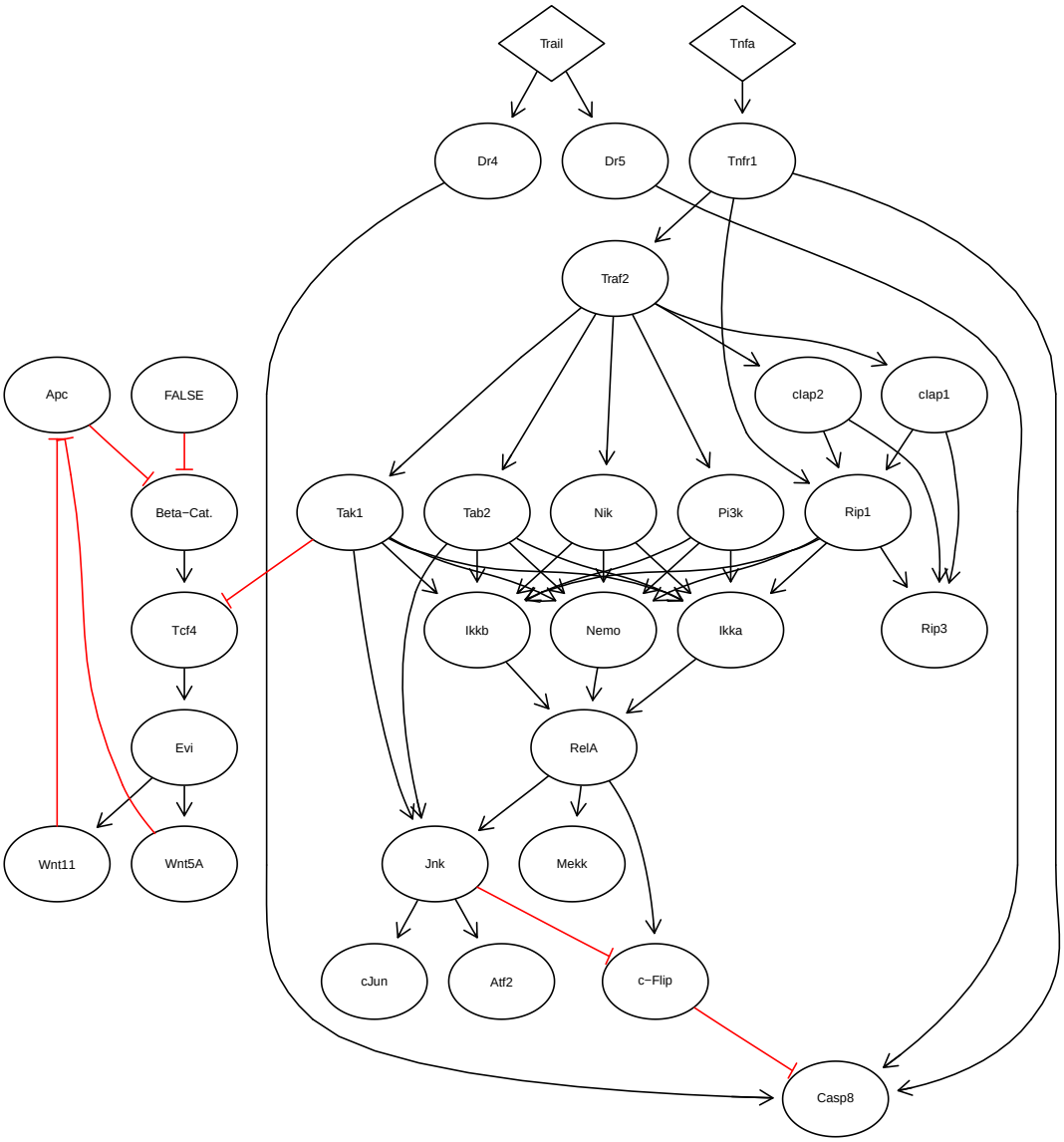


Figure D.8: **Core TNF-α-TRAIL-WNT PKN.** Interpretation of the KEGG pathway database as a directed normal graph (Kanehisa & Goto (2000); Kanehisa *et al.* (2014), http://www.genome.jp/kegg-bin/show_pathway?hsa04668, http://www.genome.jp/kegg-bin/show_pathway?ko04010+K04450, http://www.genome.jp/kegg-bin/show_pathway?hsa04210, http://www.genome.jp/kegg-bin/show_pathway?hsa04310).

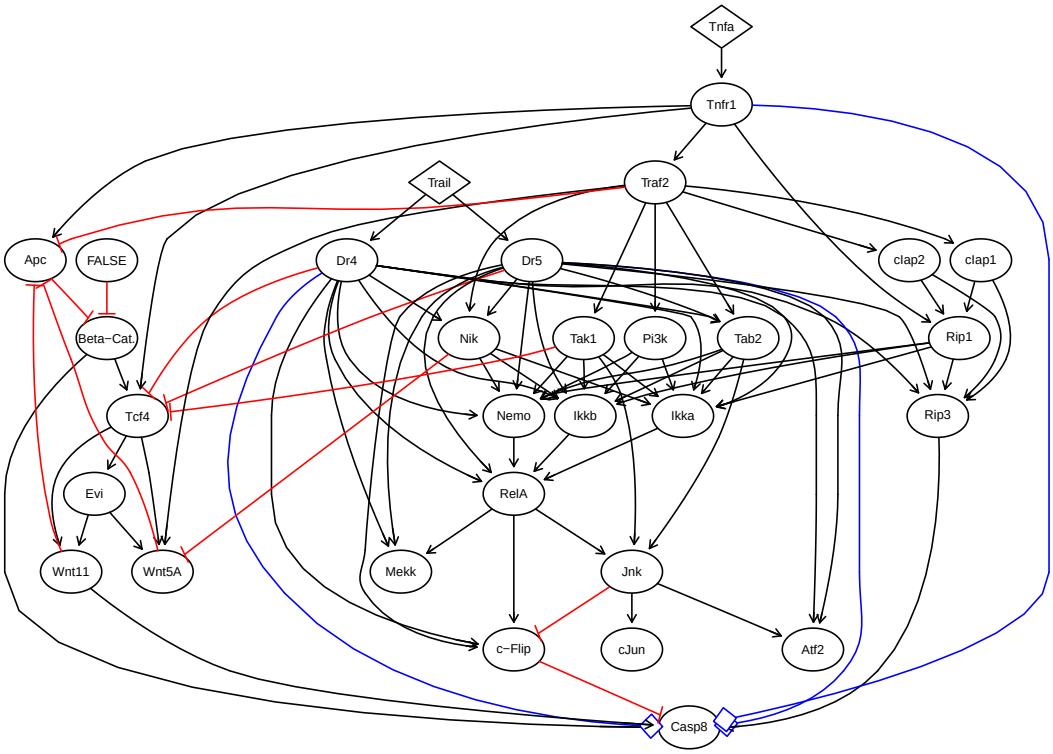


Figure D.9: **Updated PKN.** Modified PKN to account for systematic/strong residuals in figure 7.5. Blue edges with a diamond head denote ambiguous regulation ($\rightarrow$ and $\dashv$ possible).

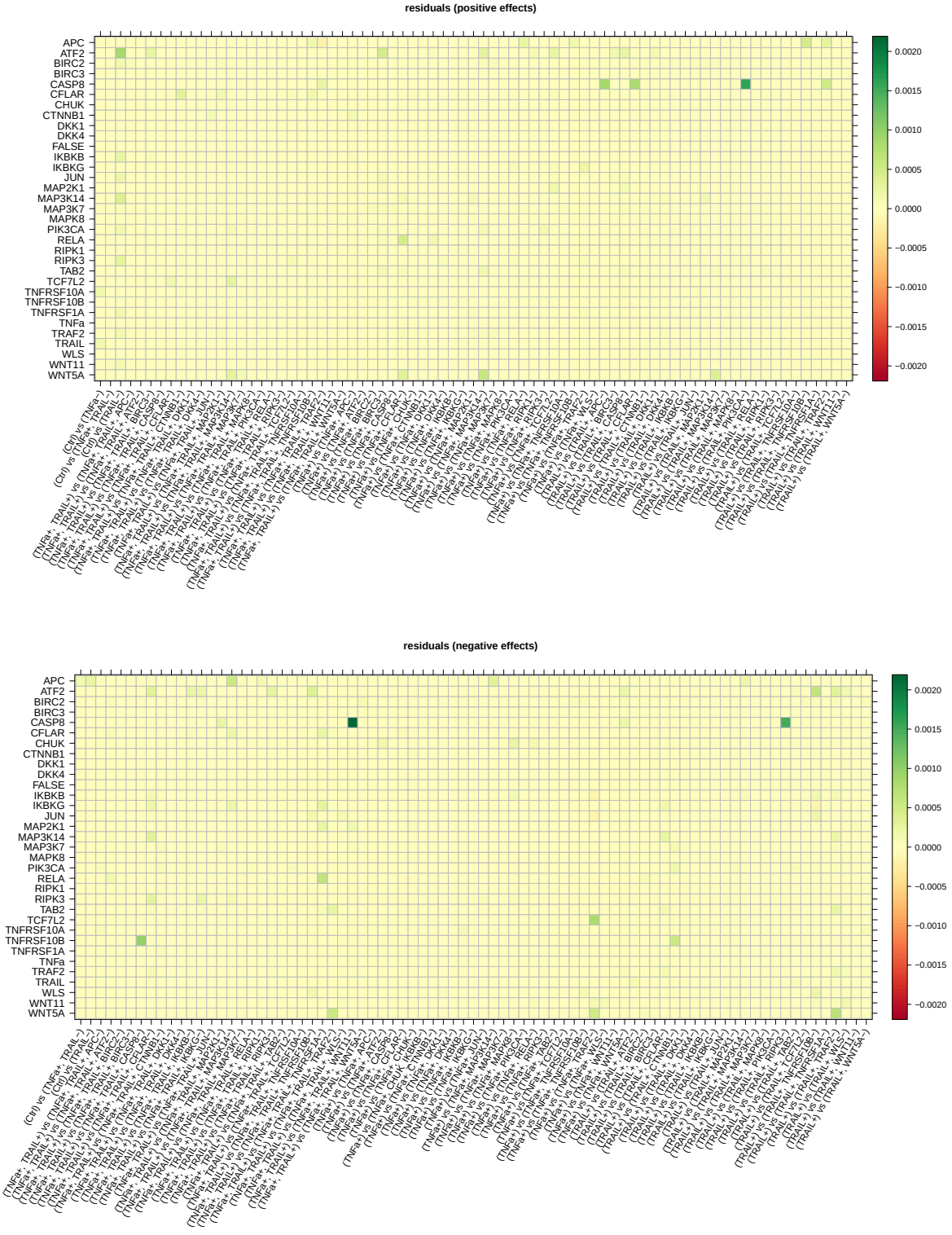


Figure D.10: **Remaining residuals in the Data.** Residuals in the observed response scheme for the improved network in figure 7.7.

Figure D.11: **Top Casp8 regulated E-genes for the updated result.** The bottom row is the expected response scheme of Casp8. The other rows show the observed response schemes of attached E-genes.

Figure D.12: **Top Tcf4 regulated E-genes for the updated result.** The bottom row is the expected response scheme of Tcf4. The other rows show the observed response schemes of attached E-genes.

List of Algorithms

List of Figures

List of Tables

Bibliography

Agrawal, Neema, Dasaradhi, P V N., Mohmmed, Asif, Malhotra, Pawan, Bhatnagar, Raj K., & Mukherjee, Sunil K. 2003. RNA interference: biology, mechanism, and applications. *Microbiol Mol Biol Rev*, **67**(4), 657–685.

Agresti, Alan. 1992. A Survey of Exact Inference for Contingency Tables. *Statist. Sci.*, **7**(1), 131–153.

Akutsu, Tatsuya, Kuhara, Satoru, Maruyama, Osamu, & Miyano, Satoru. 2003. Identification of genetic networks by strategic gene disruptions and gene overexpressions under a boolean model. *Theoretical Computer Science*, **298**(1), 235 – 251. Selected Papers in honour of Setsuo Arikawa.

Anchang, Benedict, Sadeh, Mohammad J, Jacob, Juby, Tresch, Achim, Vlad, Marcel O, Oefner, Peter J, & Spang, Rainer. 2009. Modeling the temporal interplay of molecular signaling and gene expression by using dynamic nested effects models. *Proc Natl Acad Sci U S A*, **106**(16), 6447–6452.

Anders, Simon, Pyl, Paul Theodor, & Huber, Wolfgang. 2015. HTSeqa Python framework to work with high-throughput sequencing data. *Bioinformatics*, **31**(2), 166–169.

Azijli, K., Weyhenmeyer, B., Peters, G. J., de Jong, S., & Kruyt, F A E. 2013. Non-canonical kinase signaling by the death ligand TRAIL in cancer cells: discord in the death receptor family. *Cell Death Differ*, **20**(7), 858–868.

Baker, James E. 1985. Adaptive Selection Methods for Genetic Algorithms. *Pages 101–111 of: Proceedings of the 1st International Conference on Genetic Algorithms*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc.

Beisel, Chase L., Gomaa, Ahmed A., & Barrangou, Rodolphe. 2014. A CRISPR design for next-generation antimicrobials. *Genome Biol*, **15**(11), 516.

Benjamini, Yoav, & Hochberg, Yosef. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Statist. Soc. Ser. B*, **57**(1), 289–300.

Berg, Jeremy M., Tymoczko, John L., Stryer, Lubert, & Jr., Gregory J. Gatto. 2012. *Biochemistry*. 7 edn. New York: W. H. Freeman and Company.

Bonneau, Richard, Reiss, David J., Shannon, Paul, Facciotti, Marc, Hood, Leroy, Baliga, Nitin S., & Thorsson, Vesteinn. 2006. The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biol*, **7**(5), R36.

Boutros, Michael, & Ahringer, Julie. 2008. The art and design of genetic screens: RNA interference. *Nat Rev Genet*, **9**(7), 554–566.

Boutros, Michael, Agaisse, Hervé, & Perrimon, Norbert. 2002. Sequential activation of signaling pathways during innate immune responses in Drosophila. *Dev Cell*, **3**(5), 711–722.

Boutros, Michael, Kiger, Amy A., Armknecht, Susan, Kerr, Kim, Hild, Marc, Koch, Britta, Haas, Stefan A., Paro, Renato, Perrimon, Norbert, & , Heidelberg Fly Array Consortium. 2004. Genome-wide RNAi analysis of growth and viability in Drosophila cells. *Science*, **303**(5659), 832–835.

Boutros, Michael, Brás, Lígia P., & Huber, Wolfgang. 2006. Analysis of cell-based RNAi screens. *Genome Biol*, **7**(7), R66.

Bradley, J. R. 2008. TNF-mediated inflammatory disease. *J Pathol*, **214**(2), 149–160.

Bretto, Alain. 2013. *Hypergraph Theory - An Introduction*. 1 edn. New York: Springer.

Brune Bernhard. 2003. Nitric oxide: NO apoptosis or turning it ON? *Cell Death Differ*, **10**(8), 864–869.

Carlson, Marc. *hgu133plus2.db: Affymetrix Human Genome U133 Plus 2.0 Array annotation data (chip hgu133plus2)*. R package version 2.14.0.

Chang, Chih-Chung, & Lin, Chih-Jen. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, **2**, 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Chang, T. W. 1983. Binding of cells to matrixes of distinct antibodies coated on solid surface. *J Immunol Methods*, **65**(1-2), 217–223.

Chen, Nan, & Debnath, Jayanta. 2013. IB kinase complex (IKK) triggers detachment-induced autophagy in mammary epithelial cells independently of the PI3K-AKT-MTORC1 pathway. *Autophagy*, **9**(8), 1214–1227.

Choi, Y. H., Kim, K. B., Kim, H. H., Hong, G. S., Kwon, Y. K., Chung, C. W., Park, Y. M., Shen, Z. J., Kim, B. J., Lee, S. Y., & Jung, Y. K. 2001. FLASH coordinates NF-kappa B activity via TRAF2. *J Biol Chem*, **276**(27), 25073–25077.

Chu, Yongjun, & Corey, David R. 2012. RNA sequencing: platform selection, experimental design, and data interpretation. *Nucleic Acid Ther*, **22**(4), 271–274.

Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., & Stein, Clifford. 2007. *Introduction to Algorithms*. 7 edn. Cambridge, Massachusetts: The MIT Press.

Curry, E. 2013. ArrayBin: Binarization of numeric data arrays. R package version 0.2.

Darding, M., & Meier, P. 2012. IAPs: guardians of RIPK1. *Cell Death Differ*, **19**(1), 58–66.

de Almagro, M. C., & Vucic, D. 2012. The inhibitor of apoptosis (IAP) proteins are critical regulators of signaling pathways and targets for anti-cancer therapy. *Exp Oncol*, **34**(3), 200–211.

DeFranco, Anthony L. 1997. The complexity of signaling pathways activated by the BCR. *Current Opinion in Immunology*, **9**(3), 296 – 308.

Deonier, Richard C., Tavaré, Simon, & Waterman, Michael S. 2005. Computational Genome Analysis: an Introduction. *Springer-Verlag*, **9**, 534.

Dufour, J.-F., & Clavien, P.-A. 2005. *Signaling Pathways in Liver Diseases*. Berlin: Springer-Verlag.

Dümcke, Sebastian, Bräuer, Johannes, Anchang, Benedict, Spang, Rainer, Beerenwinkel, Niko, & Tresch, Achim. 2014. Exact likelihood computation in Boolean networks with probabilistic time delays, and its application in signal network reconstruction. *Bioinformatics*, **30**(Feb), 414–419.

Fahrmeir, Ludwig, Künstler, Rita, Pigeot, Iris, & Tutz, Gerhard. 2007. Statistik. *Springer-Verlag*, **6**, 610.

Falschlehner, Christina, Emmerich, Christoph H., Gerlach, Bjoern, & Walczak, Henning. 2007. TRAIL signalling: decisions between life and death. *Int J Biochem Cell Biol*, **39**(7-8), 1462–1475.

Falschlehner, Christina, Schaefer, Uta, & Walczak, Henning. 2009. Following TRAIL's path in the immune system. *Immunology*, **127**(2), 145–154.

Fisher, R. A. 1922. On the Interpretation of 2 from Contingency Tables, and the Calculation of P. *Journal of the Royal Statistical Society*, **85**(1), pp. 87–94.

Franceschini, Andrea, Szklarczyk, Damian, Frankild, Sune, Kuhn, Michael, Simonovic, Milan, Roth, Alexander, Lin, Jianyi, Minguez, Pablo, Bork, Peer, von Mering, Christian, & Jensen, Lars J. 2013. STRING v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Res*, **41**(Database issue), D808–D815.

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. 2000. Using Bayesian networks to analyze expression data. *J Comput Biol*, **7**(Jan), 601–620.

Froehlich, Holger, Markowetz, Florian, Tresch, Achim, Niederberger, Theresa, Bender, Christian, Maneck, Matthias, Lottaz, Claudio, & Beissbarth, Tim. *nem: (Dynamic) Nested Effects Models and Deterministic Effects Propagation Networks to reconstruct phenotypic hierarchies*. R package version 2.38.0.

Froehlich, Holger, Fellmann, Mark, Sueltmann, Holger, Poustka, Annemarie, & Beissbarth, Tim. 2007. Large scale statistical inference of signaling pathways from RNAi and microarray data. *BMC Bioinformatics*, **8**, 386.

Froehlich, Holger, Fellmann, Mark, Sueltmann, Holger, Poustka, Annemarie, & Beissbarth, Tim. 2008. Estimating large-scale signaling networks through nested effect models with intervention effects from microarray data. *Bioinformatics*, **24**(22), 2650–2656.

Froehlich, Holger, Praveen, Paurush, & Tresch, Achim. 2011. Fast and efficient dynamic nested effects models. *Bioinformatics*, **27**(2), 238–244.

Gentleman, R. *annotate: Annotation for microarrays*. R package version 1.42.0.

George Casella, Edward I. George. 1992. Explaining the Gibbs Sampler. *The American Statistician*, **46**(3), 167–174.

Gershenson, Carlos. 2004. Introduction to Random Boolean Networks. *arXiv:nlin/0408006v3*.

Gilpin, Andrew R. 1993. Table for Conversion of Kendall'S Tau to Spearman'S Rho Within the Context of Measures of Magnitude of Effect for Meta-Analysis. *Educational and Psychological Measurement*, **53**(1), 87–92.

Gomaa Asmaa Ibrahim, Khan Shahid A, Toledano Mireille B, Waked Imam, & Taylor-Robinson Simon D. 2008. Hepatocellular carcinoma: Epidemiology, risk factors and pathogenesis. *World Journal of Gastroenterology : WJG*, **14**(27), 4300–4308.

Gray, Kristian A., Yates, Bethan, Seal, Ruth L., Wright, Mathew W., & Bruford, Elspeth A. 2015. Genenames.org: the HGNC resources in 2015. *Nucleic Acids Research*, **43**(D1), D1079–D1085.

Haas, Tobias L., Emmerich, Christoph H., Gerlach, Bjoern, Schmukle, Anna C., Cordier, Stefanie M., Rieser, Eva, Feltham, Rebecca, Vince, James, Warnken, Uwe, Wenger, Till, Koschny, Ronald, Komander, David, Silke, John, & Walczak, Henning. 2009. Recruitment of the linear ubiquitin chain assembly complex stabilizes the TNF-R1 signaling complex and is required for TNF-mediated gene induction. *Mol Cell*, **36**(5), 831–844.

Hamilton, A. J., & Baulcombe, D. C. 1999. A species of small antisense RNA in posttranscriptional gene silencing in plants. *Science*, **286**(5441), 950–952.

Hannus Michael, Beitzinger Michaela, Engelmann Julia C, Weickert Marie-Theresa, Spang Rainer, Hannus Stefan, & Meister Gunter. 2013. siPools: highly complex but accurately defined siRNA pools eliminate off-target effects. *Nucleic Acids Research*, **42**(12), 8049–8061.

Hansen, Kasper Daniel, Gentry, Jeff, Long, Li, Gentleman, Robert, Falcon, Seth, Hahne, Florian, & Sarkar, Deepayan. *Rgraphviz: Provides plotting capabilities for R graph objects*. R package version 2.8.1.

Heckerman, D., Geiger, D., & Chickering, D.M. 1995. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, **20**(MSR-TR-94-09), 197–243.

Henriques, David, Rocha, Miguel, Saez-Rodriguez, Julio, & Banga, Julio R. 2015. Reverse engineering of logic-based differential equation models using a mixed-integer dynamic optimization approach. *Bioinformatics*, **31**(18), 2999–3007.

Hiyama, Akihiko, Yokoyama, Katsuya, Nukaga, Tadashi, Sakai, Daisuke, & Mochida, Joji. 2013. A complex interaction between Wnt signaling and TNF- in nucleus pulposus cells. *Arthritis Res Ther*, **15**(6), R189.

Huber, Wolfgang, von Heydebreck, Anja, Sültmann, Holger, Poustka, Annemarie, & Vingron, Martin. 2002. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, **18 Suppl 1**, S96–104.

Hummon, Amanda, Pitt, Jason, Camps, Jordi, Emons, Georg, Skube, Susan, Huppi, Konrad, Jones, Tamara, Beissbarth, Tim, Kramer, Frank, Grade, Marian, Difilippantonio, Michael, Ried, Thomas, & Caplen, Natasha. 2012. Systems-wide RNAi analysis of CASP8AP2/FLASH shows transcriptional deregulation of the replication-dependent histone genes and extensive effects on the transcriptome of colorectal cancer cells. *Molecular Cancer*, **11**(1), 1.

Imai, Y., Kimura, T., Murakami, A., Yajima, N., Sakamaki, K., & Yonehara, S. 1999. The CED-4-homologous protein FLASH is involved in Fas-mediated activation of caspase-8 during apoptosis. *Nature*, **398**(6730), 777–785.

Imai, Yoshinori, Takahashi, Akiko, Hanyu, Aki, Hori, Satoshi, Sato, Seidai, Naka, Kazuhito, Hirao, Atsushi, Ohtani, Naoko, & Hara, Eiji. 2014. Crosstalk between the Rb Pathway and {AKT} Signaling Forms a Quiescence-Senescence Switch. *Cell Reports*, **7**(1), 194 – 207.

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

Jackson, Aimee L., & Linsley, Peter S. 2010. Recognizing and avoiding siRNA off-target effects for target identification and therapeutic application. *Nat Rev Drug Discov*, **9**(1), 57–67.

Jho, Eek-hoon, Zhang, Tong, Domon, Claire, Joo, Choun-Ki, Freund, Jean-Noel, & Costantini, Frank. 2002. Wnt/-Catenin/Tcf Signaling Induces the Transcription of Axin2, a Negative Regulator of the Signaling Pathway. *Molecular and Cellular Biology*, **22**(4), 1172–1183.

Johnson, W Evan, Li, Cheng, & Rabinovic, Ariel. 2007. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, **8**(1), 118–127.

Jouan-Lanhouet, S., Arshad, M. I., Piquet-Pellorce, C., Martin-Chouly, C., Le Moigne-Muller, G., Van Herreweghe, F., Takahashi, N., Sergent, O., Lagadic-Gossmann, D., Vandenabeele, P., Samson, M., & Dimanche-Boitrel, M-T. 2012. TRAIL induces necroptosis involving RIPK1/RIPK3-dependent PARP-1 activation. *Cell Death Differ*, **19**(12), 2003–2014.

Jun, Joon-Il, Chung, Chul-Woong, Lee, Ho-June, Pyo, Jong-Ok, Lee, Kee Nyung, Kim, Nam-Soon, Kim, Yong Sung, Yoo, Hyang-Sook, Lee, Tae-Ho, Kim, Eunhee, & Jung, Yong-Keun. 2005. Role of FLASH in caspase-8-mediated activation of NF-kappaB: dominant-negative function of FLASH mutant in NF-kappaB signaling pathway. *Oncogene*, **24**(4), 688–696.

Kalisch, Markus, & Bühlmann, Peter. 2007. Estimating High-Dimensional Directed Acyclic Graphs with the PC-Algorithm. *J. Mach. Learn. Res.*, **8**(May), 613–636.

Kan, Zhengyan, Zheng, Hancheng, Liu, Xiao, Li, Shuyu, Barber, Thomas, Gong, Zhuolin, Gao, Huan, Hao, Ke, Willard, Melinda D, Xu, Jiangchun, Hauptschein, Robert, Rejto, Paul A, Fernandez, Julio, Wang, Guan, Zhang, Qinghui, Wang, Bo, Chen, Ronghua, Wang, Jian, Lee, Nikki P, Zhou, Wei, Lin, Zhao, Peng, Zhiyu, Yi, Kang, Chen, Shengpei, Li, Lin, Fan, Xiaomei, Yang, Jie, Ye, Rui, Ju, Jia, Wang, Kai, Estrella, Heather, Deng, Shibing, Wei, Ping, Qiu, Ming, Wulur, Isabella H, Liu, Jiangang, Ehsani, Mariam E, Zhang, Chunsheng, Loboda, Andrey, Sung, Wing Kin, Aggarwal, Amit, Poon, Ronnie T, Fan, Sheung Tat, Hardwick, James, Wang, Jun, Reinhard, Christoph, Dai, Hongyue, Li, Yingrui, Luk, John M, & Mao, Mao. 2013. Whole genome sequencing identifies recurrent mutations in hepatocellular carcinoma. *Genome Research*.

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

Kanehisa, Minoru, Goto, Susumu, Sato, Yoko, Kawashima, Masayuki, Furumichi, Miho, & Tanabe, Mao. 2014. Data, information, knowledge and principle: back to metabolism in KEGG. *Nucleic Acids Research*, **42**(D1), D199–D205.

Kauffman, S. A. 1969. Metabolic stability and epigenesis in randomly constructed genetic nets. *J Theor Biol*, **22**(3), 437–467.

Kendall, M.G. 1938. A new measure of rank correlation. *Biometrika*, **30**(1-2), 81–93.

Kim, Young-Soo, Schwabe, Robert F., Qian, Ting, Lemasters, John J., & Brenner, David A. 2002. TRAIL-mediated apoptosis requires NF-kappaB inhibition and the mitochondrial permeability transition in human hepatoma cells. *Hepatology*, **36**(6), 1498–1508.

Klamt, Steffen, Saez-Rodriguez, Julio, Lindquist, Jonathan A., Simeoni, Luca, & Gilles, Ernst D. 2006. A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC Bioinformatics*, **7**, 56.

Klamt, Steffen, Saez-Rodriguez, Julio, & Gilles, Ernst D. 2007. Structural and functional analysis of cellular networks with CellNetAnalyzer. *BMC Syst Biol*, **1**, 2.

Kloo, Bernhard, Nagel, Daniel, Pfeifer, Matthias, Grau, Michael, Düwel, Michael, Vincendeau, Michelle, Dörken, Bernd, Lenz, Peter, Lenz, Georg, & Krappmann, Daniel. 2011. Critical role of PI3K signaling for NF-Bdependent survival in a subset of activated B-celllike diffuse large B-cell lymphoma cells. *Proceedings of the National Academy of Sciences*, **108**(1), 272–277.

Kschischang, Frank. R, Frey, Brendan J., & Loeliger, Hans-Andrea. 2001. Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, **47**, NO. 2.

Lachenmayer, Anja, Alsinet, Clara, Savic, Radoslav, Cabellos, Laia, Toffanin, Sara, Hoshida, Yujin, Villanueva, Augusto, Minguez, Beatriz, Newell, Philippa, Tsai, Hung-Wen, Barretina, Jordi, Thung, Swan, Ward, Stephen C., Bruix, Jordi, Mazzaferro, Vincenzo, Schwartz, Myron, Friedman, Scott L., & Llovet, Josep M. 2012. Wnt-pathway activation in two molecular classes of hepatocellular carcinoma and experimental modulation by sorafenib. *Clin Cancer Res*, **18**(18), 4997–5007.

Lamberti, C., Lin, K. M., Yamamoto, Y., Verma, U., Verma, I. M., Byers, S., & Gaynor, R. B. 2001. Regulation of beta-catenin function by the IkappaB kinases. *J Biol Chem*, **276**(45), 42276–42286.

Lang, Serge. 2000. *Linear Algebra*. 3 edn. New York: Springer-Verlag.

Lashkari, D. A., DeRisi, J. L., McCusker, J. H., Namath, A. F., Gentile, C., Hwang, S. Y., Brown, P. O., & Davis, R. W. 1997. Yeast microarrays for genome wide parallel genetic and gene expression analysis. *Proc Natl Acad Sci U S A*, **94**(24), 13057–13062.

Law, Charity W., Chen, Yunshun, Shi, Wei, & Smyth, Gordon K. 2014. voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol*, **15**(2), R29.

Leisch, Friedrich. 2006. A Toolbox for K-Centroids Cluster Analysis. *Computational Statistics and Data Analysis*, **51**(2), 526–544.

Li, Wei, Xu, Han, Xiao, Tengfei, Cong, Le, Love, Michael I., Zhang, Feng, Irizarry, Rafael A., Liu, Jun S., Brown, Myles, & Liu, X Shirley. 2014. MAGeCK enables robust identification of essential genes from genome-scale CRISPR/Cas9 knockout screens. *Genome Biol*, **15**(12), 554.

Logan, J.M.J., Edwards, K.J., & Saunders, N.A. 2009. *Real-time PCR: Current Technology and Applications*. Caister Academic Press.

Loong, Tze-Wey. 2003. Understanding sensitivity and specificity with the right side of the brain. *BMJ*, **327**(Sep), 716–719.

Louis, Sushil J., & Rawlins, Gregory J. E. 1992. Predicting Convergence Time for Genetic Algorithms. *Pages 141–161 of: Foundations of Genetic Algorithms 2*. Morgan Kaufmann.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. *Pages 281–297 of: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press.

Mahmoudi Tokameh, Li Vivian S W, Ng Ser Sue, Taouatas Nadia, Vries Robert G J, Mohammed Shabaz, Heck Albert J, & Clevers Hans. 2009. The kinase TNIK is an essential activator of Wnt target genes. *The EMBO Journal*, **28**(21), 3329–3340.

Mahul-Mellier Anne-Laure, Pazarentzos Evangelos, Datler Christoph, Iwasawa Ryota, AbuAli Ghada, Lin Bevan, & Grimm Stefan. 2011. Deubiquitinating protease USP2a targets RIP1 and TRAF2 to mediate cell death by TNF. *Cell death and differentiation*, **19**(5), 891–899.

Mao, Zhi Gang, Jiang, Chen Chen, Yang, Fan, Thorne, Rick F., Hersey, Peter, & Zhang, Xu Dong. 2010. TRAIL-induced apoptosis of human melanoma cells involves activation of caspase-4. *Apoptosis*, **15**(10), 1211–1222.

Margolin, Adam A., Nemenman, Ilya, Basso, Katia, Wiggins, Chris, Stolovitzky, Gustavo, Dalla Favera, Riccardo, & Califano, Andrea. 2006. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, **7 Suppl 1**, S7.

Markowetz, Florian, Bloch, Jacques, & Spang, Rainer. 2005. Non-transcriptional pathway features reconstructed from secondary effects of RNA interference. *Bioinformatics*, **21**(21), 4026–4032.

Markowetz, Florian, Kostka, Dennis, Troyanskaya, Olga G, & Spang, Rainer. 2007. Nested effects models for high-dimensional phenotyping screens. *Bioinformatics*, **23**(13), i305–i312.

Matta, Hittu, Gopalakrishnan, Ramakrishnan, Graham, Ciaren, Tolani, Bhairavi, Khanna, Akshat, Yi, Han, Suo, Yulan, & Chaudhary, Preet M. 2012. Kaposis Sarcoma Associated Herpesvirus Encoded Viral FLICE Inhibitory Protein K13 Activates NF-B Pathway Independent of TRAF6, TAK1 and LUBAC. *PLoS ONE*, **7**(5), e36601.

Mendelson, Elliott. 1970. *Theory and Problems of Boolean Algebra and Switching Circuits*. 1 edn. New York: The McGraw-Hill Companies.

Metzig, Marie, Nickles, Dorothee, & Boutros, Michael. 2011a. Large-scale RNAi screens to dissect TNF and NF-B signaling pathways. *Adv Exp Med Biol*, **691**, 131–139.

Metzig, Marie, Nickles, Dorothee, Falschlehner, Christina, Lehmann-Koch, Judith, Straub, Beate K., Roth, Wilfried, & Boutros, Michael. 2011b. An RNAi screen identifies USP2 as a factor required for TNF--induced NF-B signaling. *Int J Cancer*, **129**(3), 607–618.

Meyer, David, Dimitriadou, Evgenia, Hornik, Kurt, Weingessel, Andreas, & Leisch, Friedrich. 2014. *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*. R package version 1.6-3.

Mitchell, Melanie. 1999. *An Introduction to Genetic Algorithms*. 5 edn. Cambridge, Massachusetts: The MIT Press.

Monya, Baker. 2010. Cellular imaging: Taking a long, hard look. *Nature*, **466**(7310), 1137–1140. 10.1038/4661137a.

Morin, Ryan, Bainbridge, Matthew, Fejes, Anthony, Hirst, Martin, Krzywinski, Martin, Pugh, Trevor, McDonald, Helen, Varhol, Richard, Jones, Steven, & Marra, Marco. 2008. Profiling the HeLa S3 transcriptome using randomly primed cDNA and massively parallel short-read sequencing. *Biotechniques*, **45**(1), 81–94.

Mortazavi, Ali, Williams, Brian A, McCue, Kenneth, Schaeffer, Lorian, & Wold, Barbara. 2008. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat Methods*, **5**(7), 621–628.

Neapolitan, Richard E. 2003. *Learning Bayesian Networks*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Nelander, Sven, Wang, Weiqing, Nilsson, Bjoern, She, Qing-Bai, Pratilas, Christine, Rosen, Neal, Gennemark, Peter, & Sander, Chris. 2008. Models from experiments: combinatorial drug perturbations of cancer cells. *Mol Syst Biol*, **4**, 216.

Ngo Vu N., Young Ryan M., Schmitz Roland, Jhavar Sameer, Xiao Wenming, Lim Kian-Huat, Kohlhammer Holger, Xu Weihong, Yang Yandan, Zhao Hong, Shaffer Arthur L., Romesser Paul, Wright George, Powell John, Rosenwald Andreas, Muller-Hermelink Hans Konrad, Ott German, Gascoyne Randy D., Connors Joseph M., Rimsza Lisa M., Campo Elias, Jaffe Elaine S., Delabie Jan, Smeland Erlend B., Fisher Richard I., Braziel Rita M., Tubbs Raymond R., Cook J. R., Weisenburger Denny D., Chan Wing C., & Staudt Louis M. 2011. Oncogenically active MYD88 mutations in human lymphoma. *Nature*, **470**(7332), 115–119. 10.1038/nature09671.

Nickles, Dorothee, Falschlehner, Christina, Metzig, Marie, & Boutros, Michael. 2012. A genome-wide RNA interference screen identifies caspase 4 as a factor required for tumor necrosis factor alpha signaling. *Mol Cell Biol*, **32**(17), 3372–3381.

Niederberger, Theresa, Etzold, Stefanie, Lidschreiber, Michael, Maier, Kerstin C., Martin, Dietmar E., Froehlich, Holger, Cramer, Patrick, & Tresch, Achim. 2012. MC EMiNEM maps the interaction landscape of the Mediator. *PLoS Comput Biol*, **8**(6), e1002568.

Papenfuss, Kerstin, Cordier, Stefanie M., & Walczak, Henning. 2008. Death receptors as targets for anti-cancer therapy. *J Cell Mol Med*, **12**(6B), 2566–2585.

Pearl, Judea. 2000. *Causality: Models, Reasoning, and Inference*. New York, NY, USA: Cambridge University Press.

Pirkl, Martin, Hand, Elisabeth, Kube, Dieter, & Spang, Rainer. 2016. Analyzing synergistic and non-synergistic interactions in signalling pathways using Boolean Nested Effect Models. *Bioinformatics*, **32**(6), 893–900.

Pohlheim, Hartmut. 1995. Ein genetischer Algorithmus mit Mehrfachpopulationen zur Numerischen Optimierung. *at - Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs- und Informationstechnik*, **43**.

Qin, Jinzhong, Yao, Jianhong, Cui, Grace, Xiao, Hui, Kim, Tae Whan, Fraczek, Jerzy, Wightman, Paul, Sato, Shintaro, Akira, Shizuo, Puel, Anne, Casanova, Jean-Laurent, Su, Bing, & Li, Xiaoxia. 2006. TLR8-mediated NF-B and JNK Activation Are TAK1-independent and MEKK3-dependent. *Journal of Biological Chemistry*, **281**(30), 21013–21021.

R Core Team. 2014. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Richards, James D., Davé, Shaival H., Chou, Chih-Hao G., Mamchak, Alusha A., & DeFranco, Anthony L. 2001. Inhibition of the MEK/ERK Signaling Pathway Blocks a Subset of B Cell Responses to Antigen. *The Journal of Immunology*, **166**(6), 3855–3864.

Rivkin Elena, Almeida Stephanie M., Ceccarelli Derek F., Juang Yu-Chi, MacLean Teresa A., Srikumar Tharan, Huang Hao, Dunham Wade H., Fukumura Ryutaro, Xie Gang, Gondo Yoichi, Raught Brian, Gingras Anne-Claude, Sicheri Frank, & Cordes Sabine P. 2013. The linear ubiquitin-specific deubiquitinase gumby regulates angiogenesis. *Nature*, **498**(7454), 318–324.

Robinson Mark D, McCarthy Davis J, & Smyth Gordon K. 2009. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**(1), 139–140.

Rosen, Kenneth H. 2012. Discrete Mathematics and its Applications. *The McGraw-Hill Companies*, **7**, 1071.

Ross, Edith M., & Markowetz, Florian. 2016. OncoNEM: inferring tumor evolution from single-cell sequencing data. *Genome Biology*, **17**(1), 1–14.

Rousseeuw, Peter J. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, **20**, 53 – 65.

Russo, Maria, Mupo, Annalisa, Spagnuolo, Carmela, & Russo, Gian Luigi. 2010. Exploring death receptor pathways as selective targets in cancer therapy. *Biochem Pharmacol*, **80**(5), 674–682.

Sadeh, Mohammad J., Moffa, Giusi, & Spang, Rainer. 2013. Considering unknown unknowns: reconstruction of nonconfoundable causal relations in biological networks. *J Comput Biol*, **20**(11), 920–932.

Saez-Rodriguez, Julio, Alexopoulos, Leonidas G, Epperlein, Jonathan, Samaga, Regina, Lauffenburger, Douglas A, Klamt, Steffen, & Sorger, Peter K. 2009. Discrete logic modelling as a means to link protein signalling networks with functional analysis of mammalian signal transduction. *Mol Syst Biol*, **5**, 331.

Saez-Rodriguez, Julio, Alexopoulos, Leonidas G, Zhang, Mingsheng, Morris, Melody K, Lauffenburger, Douglas A, & Sorger, Peter K. 2011. Comparing signaling networks between normal and transformed hepatocytes using discrete logical models. *Cancer Res*, **71**(16), 5400–5411.

Schaefer, Uta, Voloshanenko, Oksana, Willen, Daniela, & Walczak, Henning. 2007. TRAIL: a multifunctional cytokine. *Front Biosci*, **12**, 3813–3824.

Schäfer, Juliane, & Strimmer, Korbinian. 2005. An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, **21**(6), 754–764.

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

Schuman, James, Chen, Yuhong, Podd, Andrew, Yu, Mei, Liu, Hong-Hsing, Wen, Renren, Chen, Zhijian J., & Wang, Demin. 2009. A critical role of TAK1 in B-cell receptor–mediated nuclear factor B activation. *Blood*, **113**(19), 4566–4574.

Shalem, Ophir, Sanjana, Neville E., Hartenian, Ella, Shi, Xi, Scott, David A., Mikkelsen, Tarjei S., Heckl, Dirk, Ebert, Benjamin L., Root, David E., Doench, John G., & Zhang, Feng. 2014. Genome-scale CRISPR-Cas9 knockout screening in human cells. *Science*, **343**(6166), 84–87.

Shannon, P, Markiel, A, Ozier, O, Baliga, N S, Wang, J T, Ramage, D, Amin, N, Schwikowski, B, & Ideker, T. 2003. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res*, **13**(11), 2498–2504.

Shapiro, R., & Vallee, B. L. 1991. Interaction of human placental ribonuclease with placental ribonuclease inhibitor. *Biochemistry*, **30**(8), 2246–2255.

Shifera, Amde Selassie, & Hardin, John A. 2010. Factors modulating expression of Renilla luciferase from control plasmids used in luciferase reporter gene assays. *Analytical Biochemistry*, **396**(2), 167 – 172.

Shinohara, Hisaaki, & Kurosaki, Tomohiro. 2009. Comprehending the complex connection between PKC, TAK1, and IKK in BCR signaling. *Immunological Reviews*, **232**(1), 300–318.

Sidorov, Grigori, Gelbukh, Alexander, Gomez-Adorno, Helena, & Pinto, David. 2014. Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model. *Computación y Sistemas*, **18**, 491–504.

Silke, John. 2011. The regulation of TNF signalling: what a tangled web we weave. *Curr Opin Immunol*, **23**(5), 620–626.

Singhal, Amit. 2001. Modern Information Retrieval: A Brief Overview. *IEEE Data Eng. Bull.*, **24**(4), 35–43.

Smedley, Damian, Haider, Syed, Durinck, Steffen, Pandini, Luca, Provero, Paolo, Allen, James, Arnaiz, Olivier, Awedh, Mohammad Hamza, Baldock, Richard, Barbiera, Giulia, Bardou, Philippe, Beck, Tim, Blake, Andrew, Bonierbale, Merideth, Brookes, Anthony J., Bucci, Gabriele, Buetti, Iwan, Burge, Sarah, Cabau, Cédric, Carlson, Joseph W., Chelala, Claude, Chrysostomou, Charalambos, Cittaro, Davide, Collin, Olivier, Cordova, Raul, Cutts, Rosalind J., Dassi, Erik, Genova, Alex Di, Djari, Anis, Esposito, Anthony, Estrella, Heather, Eyras, Eduardo, Fernandez-Banet, Julio, Forbes, Simon, Free, Robert C., Fujisawa, Takatomo, Gadaleta, Emanuela, Garcia-Manteiga, Jose M., Goodstein, David, Gray, Kristian, Guerra-Assunção, José Afonso, Haggarty, Bernard, Han, Dong-Jin, Han, Byung Woo, Harris, Todd, Harshbarger, Jayson, Hastings, Robert K., Hayes, Richard D., Hoede, Claire, Hu, Shen, Hu, Zhi-Liang, Hutchins, Lucie, Kan, Zhengyan, Kawaji, Hideya, Keliet, Aminah, Kerhornou, Arnaud, Kim, Sunghoon, Kinsella, Rhoda, Klopp, Christophe, Kong, Lei, Lawson, Daniel, Lazarevic, Dejan, Lee, Ji-Hyun, Letellier, Thomas, Li, Chuan-Yun, Lio, Pietro, Liu, Chu-Jun, Luo, Jie, Maass, Alejandro, Mariette, Jerome, Maurel, Thomas, Merella, Stefania, Mohamed, Azza Mostafa, Moreews, Francois, Nabihoudine, Ibounyamine, Ndegwa, Nelson, Noirot, Céline, Perez-Llamas, Cristian, Primig, Michael, Quattrone, Alessandro, Quesneville, Hadi, Rambaldi, Davide, Reecy, James, Riba, Michela, Rosanoff, Steven, Saddiq, Amna Ali, Salas, Elisa, Sallou, Olivier, Shepherd, Rebecca, Simon, Reinhard, Sperling, Linda, Spooner, William, Staines, Daniel M., Steinbach, Delphine, Stone, Kevin, Stupka, Elia, Teague, Jon W., Dayem Ullah, Abu Z., Wang, Jun, Ware, Doreen, Wong-Erasmus, Marie, Youens-Clark, Ken, Zadissa, Amonida, Zhang, Shi-Jian, & Kasprzyk, Arek. 2015. The BioMart community portal: an innovative alternative to large, centralized data repositories. *Nucleic Acids Research*.

Smyth, Gordon K. 2004. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Stat Appl Genet Mol Biol*, **3**, Article3.

Sokolov, Artem, & Whitley, Darrell. 2005. Unbiased Tournament Selection. *Pages 1131–1138 of: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*. GECCO '05. New York, NY, USA: ACM.

Soung, Young Hwa, Lee, Jong Woo, Kim, Su Young, Park, Won Sang, Nam, Suk Woo, Lee, Jung Young, Yoo, Nam Jin, & Lee, Sug Hyung. 2004. Somatic mutations of CASP3 gene in human cancers. *Human Genetics*, **115**(2), 112–115.

Soung Young Hwa, Lee Jong Woo, Kim Hong Sug, Park Won Sang, Kim Su Young, Lee Jong Heun, Park Jik Young, Cho Yong Gu, Kim Chang Jae, Park Yong Gyu, Nam Suk Woo, Jeong Seong Whan, Kim Sang Ho, Lee Jung Young, Yoo Nam Jin, & Lee Sug Hyung. 2003. Inactivating mutations of CASPASE-7 gene in human cancers. *Oncogene*, **22**(39), 8048–8052.

Szczurek, Ewa, Gat-Viks, Irit, Tiuryn, Jerzy, & Vingron, Martin. 2009. Elucidating regulatory mechanisms downstream of a signaling pathway using informative experiments. *Mol Syst Biol*, **5**, 287.

Tarjan, R. 1972. Depth-First Search and Linear Graph Algorithms. *SIAM Journal on Computing*, **1**(2), 146–160.

Taxman, Debra J., Moore, Chris B., Guthrie, Elizabeth H., & Huang, Max Tze-Han. 2010. *RNA Therapeutics: Function, Design, and Delivery*. Totowa, NJ: Humana Press. Chap. Short Hairpin RNA (shRNA): Design, Delivery, and Assessment of Gene Knockdown, pages 139–156.

Terfve, Camille, & Saez-Rodriguez, Julio. 2012. Modeling Signaling Networks Using High-throughput Phospho-proteomics. *Adv Exp Med Biol*, **736**, 19–57.

Terfve, Camille, Cokelaer, Thomas, Henriques, David, MacNamara, Aidan, Goncalves, Emanuel, Morris, Melody K., van Iersel, Martijn, Lauffenburger, Douglas A., & Saez-Rodriguez, Julio. 2012. CellNOptR: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. *BMC Syst Biol*, **6**, 133.

Tornesello, Maria Lina, Buonaguro, Luigi, Tatangelo, Fabiana, Botti, Gerardo, Izzo, Francesco, & Buonaguro, Franco M. 2013. Mutations in TP53, CTNNB1 and PIK3CA genes in hepatocellular carcinoma associated with hepatitis B and hepatitis C virus infections. *Genomics*, **102**(2), 74–83.

Toyama, Takashi, Lee, Han Chu, Koga, Hironori, Wands, Jack R., & Kim, Miran. 2010. Noncanonical Wnt11 inhibits hepatocellular carcinoma cell proliferation and migration. *Mol Cancer Res*, **8**(2), 254–265.

Tresch, Achim, & Markowetz, Florian. 2008. Structure learning in Nested Effects Models. *Stat Appl Genet Mol Biol*, **7**(1), Article9.

Vapnik, Vladimir N. 1998. *Statistical Learning Theory*. 1 edn. New York: John Wiley & Sons.

Vaske, Charles J, House, Carrie, Luu, Truong, Frank, Bryan, Yeang, Chen-Hsiang, Lee, Norman H, & Stuart, Joshua M. 2009. A factor graph nested effects model to identify networks from genetic perturbations. *PLoS Comput Biol*, **5**(1), e1000274.

Venables, W. N., & Ripley, B. D. 2002. *Modern Applied Statistics with S*. Fourth edn. New York: Springer. ISBN 0-387-95457-0.

Voloshanenko Oksana, Erdmann Gerrit, Dubash Taronish D., Augustin Iris, Metzig Marie, Moffa Giusi, Hundsrucker Christian, Kerr Grainne, Sandmann Thomas, Anchang Benedikt, Demir Kubilay, Boehm Christina, Leible Svenja, Ball Claudia R., Glimm Hanno, Spang Rainer, & Boutros Michael. 2013. Wnt secretion is required to maintain high levels of Wnt activity in colon cancer cells. *Nat Commun*, **4**(oct). Supplementary information available for this article at http://www.nature.com/ncomms/2013/131028/ncomms3610/suppinfo/ncomms3610_S1.html.

Wajant, H., Pfizenmaier, K., & Scheurich, P. 2003. Tumor necrosis factor signaling. *Cell Death Differ*, **10**(1), 45–65.

Walczak, Henning. 2011. TNF and ubiquitin at the crossroads of gene activation, cell death, inflammation, and cancer. *Immunol Rev*, **244**(1), 9–28.

Wang, Xin, Yuan, Ke, Hellmayr, Christoph, Liu, Wei, & Markowetz, Florian. 2014. Reconstructing evolving signalling networks by hidden Markov nested effects models. *The Annals of Applied Statistics*, **8**(1), 448–480.

Wehrens, R., & Buydens, L.M.C. 2007. Self- and Super-organising Maps in R: the kohonen package. *J. Stat. Softw.*, **21**(5).

Wikipedia. 2015. *Rock-paper-scissors — Wikipedia, The Free Encyclopedia*. [Online; accessed 08-July-2015].

Wikipedia. 2016a. *ABI Solid Sequencing — Wikipedia, The Free Encyclopedia*. [Online; accessed 20-April-2016].

Wikipedia. 2016b. *Vulcan salute — Wikipedia, The Free Encyclopedia*. [Online; accessed 1-March-2016].

Wilson, Robin J. 1996. *Introduction to Graph Theory*. 4 edn. Essex, England: Pearson Education Limited.

Wittmann, Dominik M., Krumsiek, Jan, Saez-Rodriguez, Julio, Lauffenburger, Douglas A., Klamt, Steffen, & Theis, Fabian J. 2009. Transforming Boolean models to continuous models: methodology and application to T-cell receptor signaling. *BMC Syst Biol*, **3**, 98.

Xue, Luzheng, Murray, James H., & Tolkovsky, Aviva M. 2000. The Ras/Phosphatidylinositol 3-Kinase and Ras/ERK Pathways Function as Independent Survival Modules Each of Which Inhibits a Distinct Apoptotic Signaling Pathway in Sympathetic Neurons. *Journal of Biological Chemistry*, **275**(12), 8817–8824.

Zeller, Cordula, Froehlich, Holger, & Tresch, Achim. 2009. A Bayesian network view on nested effects models. *EURASIP J Bioinform Syst Biol*, 195272.

Zimmerman, Zachary F., Kulikauskas, Rima M., Bomsztyk, Karol, Moon, Randall T., & Chien, Andy J. 2013. Activation of Wnt/-catenin signaling increases apoptosis in melanoma cells treated with trail. *PLoS One*, **8**(7), e69593.

Zucchini-Pascal, Nathalie, Peyre, Ludovic, & Rahmani, Roger. 2013. Crosstalk between Beta-Catenin and Snail in the Induction of Epithelial to Mesenchymal Transition in Hepatocarcinoma: Role of the ERK1/2 Pathway. *International Journal of Molecular Sciences*, **14**(10), 20768.

Abbreviations

The following list contains often used abbreviations for quick reference.

B-NEM	Boolean Nested Effects Models
BAG	Boolean directed acyclic hyper-graph
BCR	B-Cell receptor
BG	Boolean directed hyper-graph
BGNS	Boolean greedy neighbourhood search
BN	Bayesian network
CNF	conjunctive normal form
CNO	CellNet Optimizer
D-NEM	Dynamic Nested Effects Models
DAG	directed acyclic graph
DNA	deoxyribonucleic acid
DNF	disjunctive normal form
ERS	expected response scheme
FDR	false discovery rate
FG-NEM	Factor Graph Nested Effects Models
GA	genetic algorithm
GGM	Gaussian graphical models
GNS	greedy neighbourhood search
GTN	ground truth network
HCC	hepatocellular carcinoma
KEGG	Kyoto Encyclopedia of Genes and Genomes
$\mathcal{MAD}$	model adaptive discretization (score)
mRNA	messenger RNA
NEM	Nested Effects Models
NN	neural nets
NRM	negative residuals matrix
oncoNEM	oncogenetic Nested Effects Models
ORS	observed response scheme
P-NEM	Partial Nested Effects Models
PDAG	partially directed acyclic graph
PKN	prior knowledge network
PRM	positive residuals matrix
Rluc	Renilla luciferase
RNA	ribonucleic acid
shRNA	short hairpin RNA

siRNA	small interfering RNA
SOM	self organizing maps
SUS	stochastic universal sampling
SVM	support vector machines
Tnf-α	Tumor necrosis factor α
Trail	TNF related apoptosis-inducing ligand
UTS	unbiased tournament selection
WNT	Wingless-Type

Curriculum Vitae

Martin Franz-Xaver Pirkl

Institute of Functional Genomics
Department of Statistical Bioinformatics
University of Regensburg
Josef Engertstr. 9 / Am Biopark 9
93053 Regensburg, Germany
Tel: +49 941 943 5052
martin-franz-xaver.pirkl@ukr.de

Education

2011 - 2016 **University of Regensburg**
Ph.D statistical Bioinformatics
Thesis: Indirect inference of synergistic and alternative signalling of intracellular pathways

2004 - 2010 **University of Regensburg**
Diploma (≈Master) Mathematics
Thesis: Über die Gestalt der Lagrange Multiplikatoren bei einer Navier-Stokes Phasenfeldgleichung (About the form of Lagrange Multipliers for a Navier-Stokes Phasefield Equation)

1995 - 2004 **Willibald Gymnasium Eichstätt**
higher education entrance qualification (Abitur)
emphasis on Mathematics and natural sciences

Working Experience

2011 - 2016 **University of Regensburg**
research assistant in statistical Bioinformatics

2008 **University of Regensburg**
student assistant in applied Mathematics

Conference Presentations

Statistical Learning of Biological Systems from Perturbations, Ascona, 2015 (talk)

BioSysNet Symposium: From Functional Genomics to Systems Biology, Munich, 2014 (poster)

21st Annual International Conference on Intelligent Systems for Molecular Biology, 12th European Conference on Computational Biology, ISMB/ECCB, Berlin, 2013 (poster)

Publication

Pirkl, Martin, Hand, Elisabeth, Kube, Dieter, & Spang, Rainer. 2016. Analyzing synergistic and non-synergistic interactions in signalling pathways using Boolean Nested Effect Models. *Bioinformatics*, **32**(6), 893900.

Skills/Duties

R, command line tools (e.g. RNAseq read mapping/alignment), Bash scripting, Microarray data, next generation sequencing data
applied statistics, Association Networks (Gaussian, Bayesian), Causal Networks (Nested Effects Models), classification, clustering, linear models, survival analysis, differential gene expression analysis
B-Cell receptor signalling in lymphoma, extrinsic apoptosis and inflammation signalling (Tumor necrosis factor α (TNFα) and TNF-related apoptosis inducing ligand (TRAIL)) in hepatocellular carcinoma
teaching assistance and student supervision (undergraduate student practicals/courses/bachelor thesis, PhD student method courses)

References

Rainer Spang:
Institute of Functional Genomics
Statistical Bioinformatics
University of Regensburg
Josef Engertstr. 9 / Am BioPark 9
93053 Regensburg, Germany
Tel +49 941 943 5053
rainer.spang@ukr.de

Michael Boutros:

Signaling and Functional Genomics (B110)
Deutsches Krebsforschungszentrum
Im Neuenheimer Feld 580
69120 Heidelberg, Germany
Tel: +49 6221 42 1951
M.Boutros@Dkfz-Heidelberg.de

Peter Oefner:
Institute of Functional Genomics
Functional Genomics
University of Regensburg
Josef Engertstr. 9 / Am BioPark 9
93053 Regensburg, Germany
Tel: +49 941 943 5014
Peter.Oefner@ukr.de