Freie Universität zu Berlin

Masterthesis

Inference of Boolean Networks considering real-life time course Data

Nina Valery Kersten

Supervisors

Prof. Dr. Heike Siebert
Prof. Dr. Alexander Bockmayr

Advisor

Phd. Robert Schwieger

November 20, 2018

Abstract

The survival of a cell and eventually of its organism depends mostly on the reliable interaction between different kinds of substances. Different functionalities inside and outside a cell like profileration, division and apoptosis are part of different regulatory networks in a system. Small malfunction in these regulatory networks could cause diseases from low impact for the organism to a big one. Therefore it is necessary to learn these regulatory networks, its structure and dynamical behaviour for further drug design approaches. Several efforts have been made to infer a regulatory network.In this work the focus is on inferring Boolean networks from real-life time-course data of breast cancer tissue, which provide a simplified version of the states of substances in a system (a gene is expressed:1, or not:0). The boolean network is validated for inferring the network by three inference different learning algorithms REVEAL, BESTFIT and FULLFIT in combination with different k-means clustering binarization algorithms. The inferred networks are evaluated against a goldstandard to show how well the model predicted the networks structure and its dynamics. Thus, it is shown how the complexity and the size of a network influence the predictive power of the model and the explanatory power of a boolean approach.

Contents

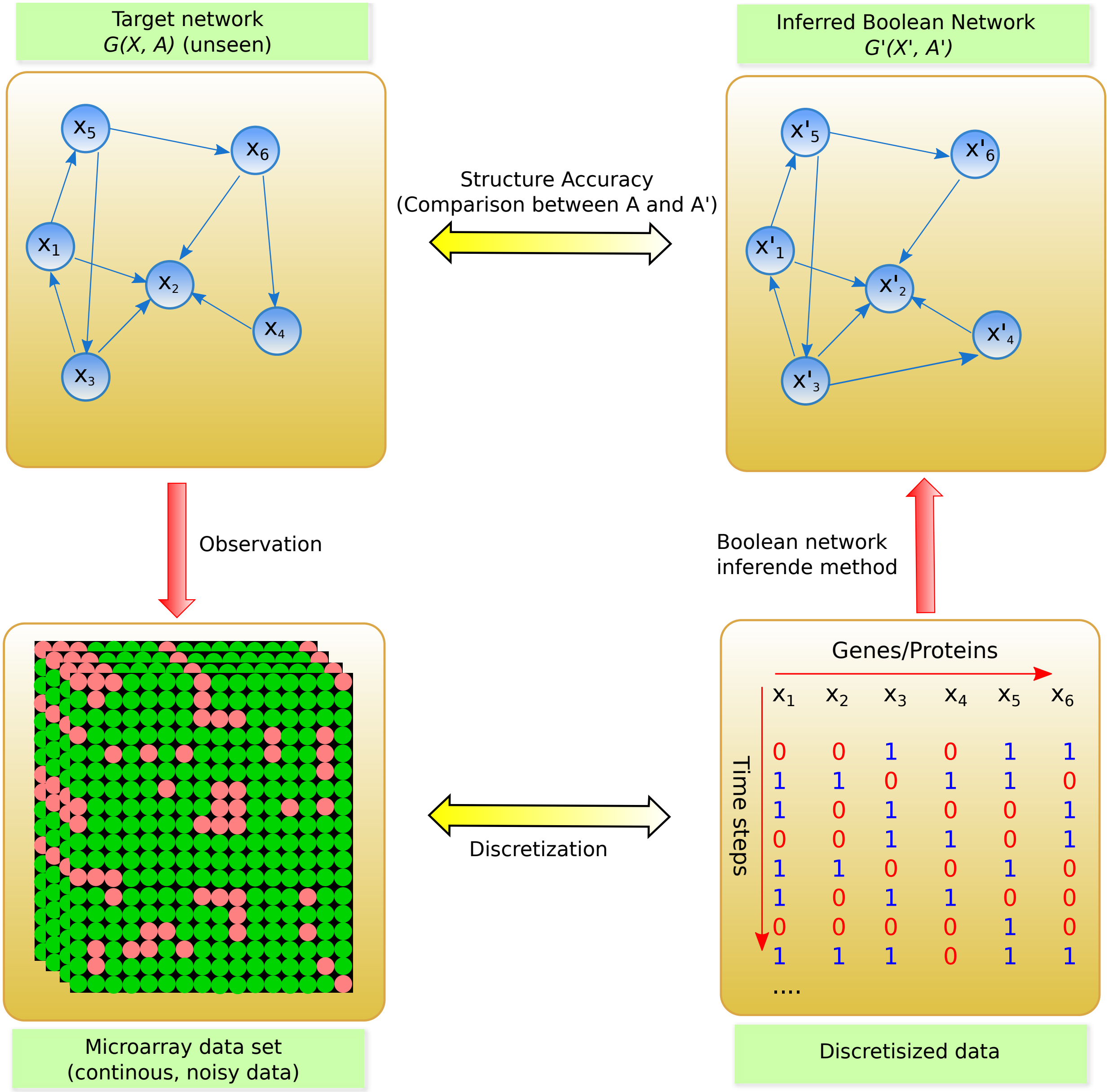
List of Figures

List of Tables

1 Introduction

The development and functioning of a cell and its organism is a product of a complex cellular machinery, where the interaction of genes, proteins, mRNA and many other substances induce a cascade of extracellular signals transducted by mechanisms of the cell membrane,reaching the nucleus of the cell, initiating a transcription process that controls the production and abundance of proteins. Proper functioning of these regulatory networks is essential to the survival and adapation of a cell. Malfunctioning has been identified as the cause of various diseases [Berestovsky & Nakhleh, 2013].To understand the behaviour of a biological system it is necessary to find and analyze the main important processes in a system in a dynamical such that the system could be manipulated systematically by a specific drug. Recent advances in high-throughput techniques provide a big abundance of information about various biological interactions measured over a series of time. It is necessary to handle this big data poperly for significant analysis. Biological information can be considered as different systems (e.g. gene regulation, protein-protein interaction, signal transduction, metabolism) described by a network with certain properties. Several strategies are known to infer a network form biological data like Baysian networks, Neural networks and Boolean networks [Saadatpour & Albert, 2013]. It is desirable to simplify a complex biological system such that the main important parts can be easy analyzed. For this reason the approach of inferring boolean networks is chosen. The simplification starts by converting the continuous biological time-series measurements (e.g. espression value of a gene, concentration of a protein) into discrete values. Therefore three different discretization algorithms are applied: Two cluster and iterative k-means binarization and BASC A binarization. The components of a boolean network of a system are represented by nodes (vertices) and the interactions among the nodes are depicted by edges. The orientation of an edge can be directed such that an edge points from one node to another. This directed edge can be positive or negativ (sign),depending on the influence of one node (the regulator) to another node (the regultee). [Saadatpour & Albert, 2013] In this work three well known inference algorithms are applied (REVEAL,BESTFIT, FULLFIT, MIBNI).[?] Boolean networks are inferred for a example data set, a small real-example and a big real-life data set. Beside the structure the dynamics of a system are important, too. That is why only time-series data is used here.

The big real-life data set is provided by a platform called Dream Challenge and contains concentration measurements of 48 proteins of four cell lines of breast cancer tissue. To determine the predictive power of a model the prediction is compared to a goldstandard data set. The first section is taking a glance on the different kinds of biological input data, the graphtheoretical background of boolean networks and its preprocessing. In the second section it is getting more detailed by giving an overview of different inference algorithms, describing a tool called PyBoolNet and description of the input data. Then the pipeline of application the binarization and inference algorithms is described. Finally the results of the comparison of the prediction against the goldstandard is analyzed and discussed.



x_1	x_2	x_3	x_4	x_5	x_6
0	0	1	0	1	1
1	1	0	1	1	0
1	0	1	0	0	1
0	0	1	1	0	1
1	1	0	0	1	0
1	0	1	1	0	0
0	0	0	0	1	0
1	1	1	0	1	1
....					

Figure 1.1: Pipeline

2 Background

In this section the intention of using different types of biological input data is explained and what potentially will be the occuring problems. Afterwards the binarization algorithms are explained. Then the a graph in general, the boolean network and its synchronous and asynchronous update, and its attractor and basins are defined and explained by a small example. In Network Evaluation the strategies of assessing a classification is accompanied with a real-life example. The whole section is kept quite general, thus in Materials and Methods it is getting more detailed related to the used data.

2.1 Biological Background

Depending on the aim of a network inference the biological input data can be depicted by the interaction of proteins, genes or metabolic substances. The choice of the biological input decides about the information content and thus the structure of the input data for further network inference algorithms. Biological interactions can be observed at different levels of information integration of a cell (gene gene interaction, protein protein interaction and metabolic interaction). In general, a signal (e.g. growth factor) binds to a membran integrated receptor (e.g. receptor-tyrosine-kinase (RTK)) of the cell causing an activation (e.g. phosphorylation) of one or multiple target proteins inducing several signal transduction cascades (Figure 2.1). While the signal is transduced, it is amplified by enzymatic activities or inhibited by feedback pathways down the cascade. Finally transcription factors are activated by the input signal such that the expression of a target gene by generating mRNA (messanger RiboNucleotide Acid) yields a protein. The resulting proteins influence the cell's survival by its profileration, induction of cell differentiation and apoptosis.

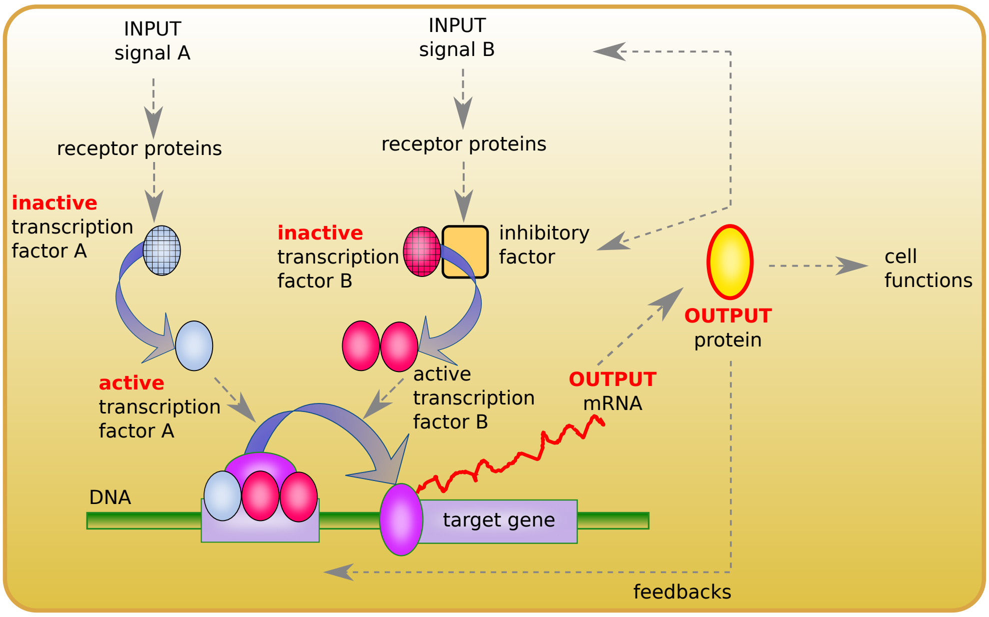


Figure 2.1: Transcriptional Signale Cascade This example shows two different input signals A and B and their transcription factors (blue oval,pink oval) initiating the transcription of a target gene follwed by the creation of its protein. [**?**]

In Figure 2.3 an example of transcriptional gene regulatory etwork is shown. A input signal A and and input signal B (e.g. hormon) bind to a specific receptor protein. The complex of A activates the transcriptionfactor A that binds directly to the gene's cis-regulatory sequence inducing the expression of the target gene. The complex of B initiate a seperation of the inactive B (pink oval) from an ihibitory factor (yellow rectangle). Transcription factor B is then free to bind to the cis-regulatory sequence. Thus the expression level of this target gene is leveled by signal A and B. The mRNA output results in a protein poduct which can be neccessary for cell functions, play a role in the gene transcription process or inluence the signal cascade of the input signals.
The concentration of substances in signalling pathways underlies high fluctuation over time due to transcriptional and translational regulation, such that the inference of a significant network is a challenging task.

Gene Regulatory Networks

In a Gene regulatory network (GRN) the interaction of genes are identified indirectly by the interaction and amount of thier transcriptional products (e.g. mRNA, proteins). The nodes of a GRN are depicted by the genes and the edges are directed by showing whether a gene produces mRNA (transcript of the source gene) which inhibits or activtes the target gene.

Metabolic networks

At the level of Metabolic networks the substances are highly interconnected in a quite complex way (e.g. cell respiration in Figure 2.2). An indiviual's metabolism is determined by its genetics, enviroment and nutrition.[?]. In a metabolic Network the nodes are depited by different biochemical components coneccted by directed edges describing the positive or negative interaction. Biochemical reaction are represented by a metabolic pathway, which consists of a sequence of biochemical reactions that produce a set of metabolites from a set of precursor metabolites and cofactors. The length of a pathway is the number of biochemical reactions between the precursor and the final metabolites of the pathway. The definition of a pathwy is not unique, therefore the length of pathways vary. [?]

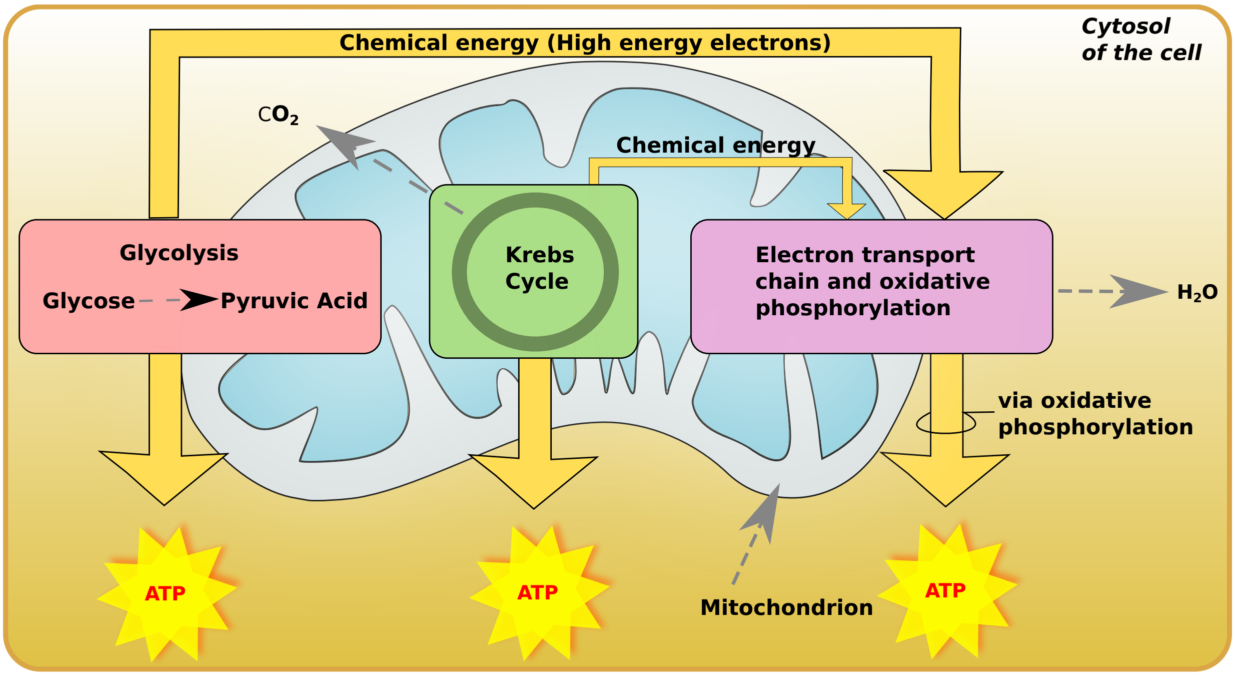

Chemical energy (High energy electrons)
Cytosol of the cell
CO_2
Chemical energy
Glycolysis
Glycose
Pyruvic Acid
Krebs Cycle
Electron transport chain and oxidative phosphorylation
H_2O
via oxidative phosphorylation
ATP
ATP
Mitochondrion
ATP

Figure 2.2: Metabolic Network of the cell respiration. Biochemnical energy from nutritiens is converted into adenosin triphosphate (ATP) by releasing waste product of water H_2O and CO_2.

Protein-Protein-Interaction network

In contrast to the gene regulatory interaction network in a protein-protein interaction (PPI) network the proteins act directly among themselve. Thus the nodes in a network are the interacting proteins. Proteins interact by physical contacts(e.g. electrostatic forces) of high specificity. PPIs play a big role in electron transfer,signal transduction, transport across

Growth factor	Full Notation	description
IGF1		hormone, similar to the insulin function and structure
NRG1		membrane glycoprotein, mediating cell-cell signaling, cri
HGF	hepatocyte growth factor	regulate cell growth,cell motility and morphogenesis
FGF1	Fibroblast growth factor 1	functions as a modifier of endothelial cell migration, pro
Insulin		
EGF		
PBS		
Serum		

membranes and cell metabolism. The underlying assumption is that true interactions are likely to occur between proteins involved in the same biological process, proteins found in the same cell compartment, and proteins whose mRNA are coexpressed. [Pellegrini et al., 2004]

The real-life data set in this work is dealing with PPIs, thus, it is important to know for later understanding and discussion how the data is obtained and which kind of problems might occure in this process.

Referring to the general description of a transcriptional signal cascade we state the receptor being a enzyme-associated receptor and the input signal are growth factors. A enzyme associated receptor has a polypetide chain integrated in the cell's membrane with a tyrosine kinase activity (see Figure below). Growthfactor receptors with a tyrosine kinase activity are called receptor receptor tyrosine kinases (RTKs). These RTKs have the property of autophosphorylation (resp. they amplify their incoming signal). Binds a ligand to this receptor, first the receptor autophosphorylates and then phosphorylates the tyrosin residues of the ligand. By the phosphorylation of the receptor and several other ligands (rep. proteins) a phosphorylating cascade (e.g. signal transduction cascade) is induced. In this Mitogen activated protein phosphorylation cascade the MAP-kinase katalyzes the phosphorylation of effector proteins, such that inactive transcription factors are activated.

The goal is figure out the main important proteins in this phosphorylation cascade by inferring a boolean network based on the measurement of the proteins abundance considering different incoming growth factor (resp. stimuli) displayed in the table below.

Dysregulation of the gene of NRG1 has been linked to diseases such as cancer, schizophrenia, and bipolar disorder (BPD)

One of the most effetive strategy to collect data of protein-protein interaction is a technique so called Reverse phase protein lysate microarray(RPMA, resp. RPPA). RPMA is an antibody-

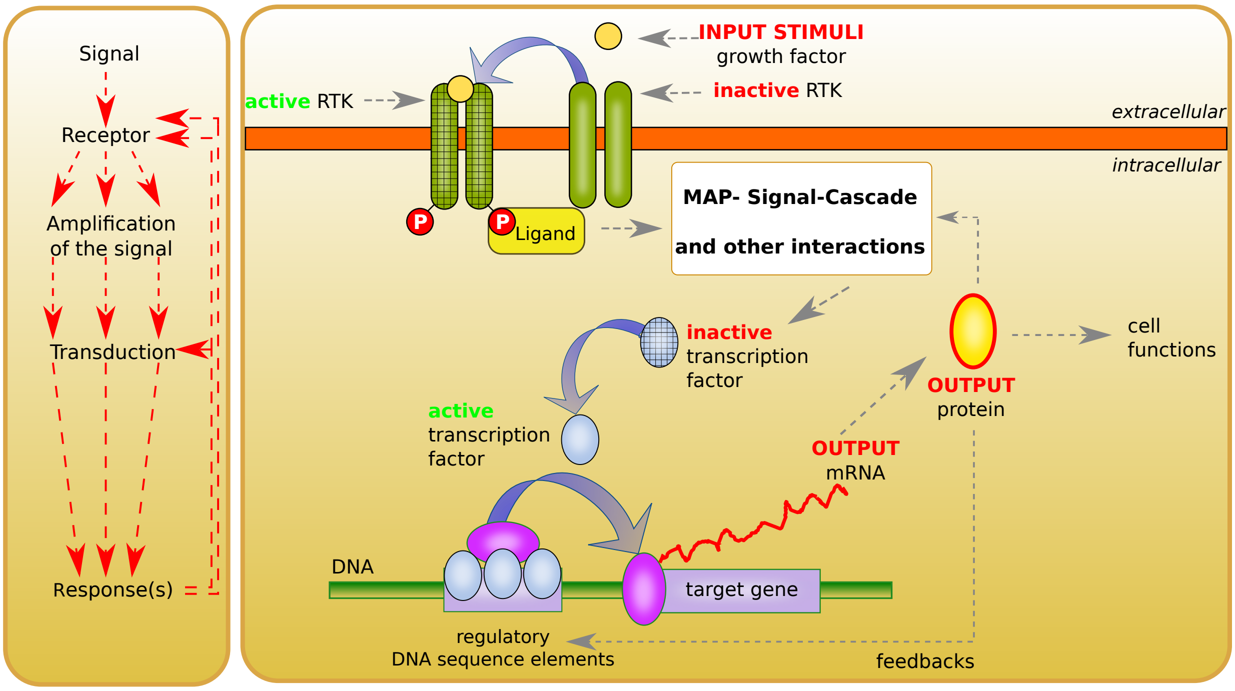


Figure 2.3: Transcriptional Signale Cascade This example shows two different input signals *A* and *B* and their transcription factors (blue oval,pink oval) initiating the transcription of a target gene follwed by the creation of its protein. [?]

based assay that provides quanitative measurements of protein abundance./citepthe HPN DREAM consotium This technique is divided up into 6 parts. First starting with the sample collection. An inhibitor or stimulus in form of drugs is added to a set of celllines at the same time and the celllines are then processed at different time points. Secondly in the Cell Lyses step cell fragments are lysed with a celllysis buffer to obtain high protein concentration.The choice of a buffer decides the quantity of proteins can be lysed out of the cell. Afterwards cellysed probes are diluted.

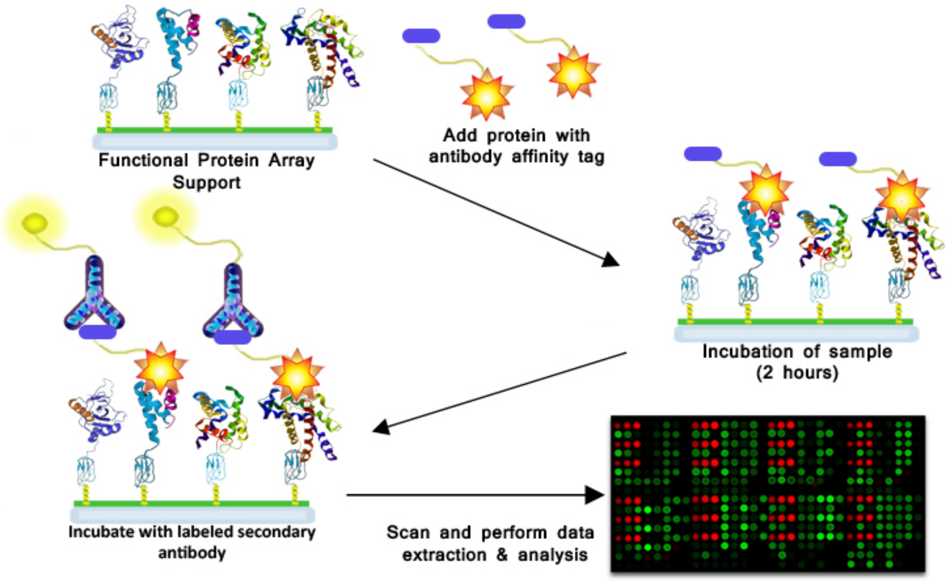

Functional Protein Array Support
Add protein with antibody affinity tag
Incubation of sample (2 hours)
Incubate with labeled secondary antibody
Scan and perform data extraction & analysis

Figure 2.4: RPMA: Antibody binding and fluorometric detection. Proteins are tagged by a specific antibody. After incubation time the secondary antibody is added. Finally the abundance of proteins is determined by a fluorometric detection.

In the Antibody screening the lysates are pooled and resolved by SDS-PAGE followed by western blotting on a nitrocellulose membrane. The membrane is cut into 4mmm strips. Each slide is probed with a different antibody, primary with a secondary antibody. For fluorometric detection a primary and secondary antibody are diluted (Figure 2.4).Detection reagent is put on each slide. Signal amplification and detection is done by an optical flatbed scanner if colormetric technique is used orby laser scanning. Subsequently the data set structure is determined by deleting missing data points and outliers from the set.Then the data set is normalized.

The strenght of RPMA is the high throughput, ulta-sensitive detection of proteins from extremly small numbers of input material which is not possible for western blotting and ELISA. The small spot size on the microarray, ranging in diameter from 85 to 200 micrometres, enables the analysis of thousands of samples with the same antibody in one experiment.The high sensitivity of RPMAs allows for the detection of low abundance proteins or biomarkers such as phosphorylated signaling proteins from very small amounts of starting material such as biopsy samples, which are often contaminated with normal tissue. A great improvement of RPMAs over traditional forward phase protein arrays is a reduction in the number of antibodies needed to detect a protein. The protein isn't detected directly which helps to preserve the proteins. Antibodies, especially phospho-specific reagents, often detect linear peptide sequences that may be masked due to the three-dimensional conformation of the protein. This problem is overcome with RPMAs as the samples can be denatured, revealing any covered epitopes (part of a protein recognized by specific antibody).

The weakness of RPMA are batch effects caused by the choice of the right buffer, quantity of the proteins and the antibody performance. The choice of the right buffer decides the quantity of proteins which can be lysed out of the cell. Little or poor quality of starting material and a long storage time causes low protein. it might be useful to improve the antibody performance by validating it with a smaller sample size under identical conditions before starting with the actual sample collection. Currently the number of signaling proteins for with antibodies exist to get an analyzable signal is small.

All this facts should be considered in later analysing steps.

2.2 Preprocessing

After generating the biological data some preprocessing like normalisation and discretization has to be done. Therefore several strategies are known. Normalization is an essential step, because data can contain outlier, the abundance of some proteins or mRNA is often higher than others and obtaining biological data from the lab could cause several batch effects. In 3.3.2 HPN-DREAM breast cancer data set a normalization method is described more detailed. Before starting inferring a boolean network from real-life time-series data the data has to be discretisized such that each continous value (e.g. concentration measurement) measured at a certain time point of a particular substance (e.g. gene, protein) becomes discrete and has either a value of 1 or 0. The choice of the appropriate discretization algorithm decides about accuracy of a network. In this part of the chapter three discretization algorithms are introduced.

Two clusters k-means binarization

The time-series data is divided into two clusters directly. One cluster contains all the values with the higher mean being set to 1. In the other cluster all the values with the lower mean being set to 0 are combined. This binarization strategie is fast and simple but may exclud some essential information like about oscillations and fluctuations.[Berestovsky & Nakhleh, 2013]

Iterative k-means binarization

A depth d of clustering is set followed by a set of initial number of cluster $k = 2^d$. The input consists of tim-series data $S = \{S_1, ..., S_n\}$ of n species (e.g.different genes or proteins),each of size $m + 1$, where $S_i(t) \in \mathbb{R}^+$ $(0 \leq t \leq m)$ is the concentration of species i at time t. In each iteration the data is classified into k dijoint clusters $C^1_{s_i}, ...C^x_{s_i}$. In each Cluster all its values are replacd by the clusters mean $\mu(C^x_{s_i})$. Here $d = 3$ in the beginning until $d = 1$ such that the two cluster k-means binarization method can be applied.

Example 2.1. Assume we have a time-series data with measurements for a gene A. Starting with a depth of $d = 3$ we have initally $k = 8$ clusters for each gene. Resulting in $\{C^1_{s_A}, ..., C^8_{s_A}\}$, each cluster containing values of time-series data for A. Now for each cluster the mean $\{\mu(C^1_{s_A}), ..., \mu(C^8_{s_A})\}$ is calculated. Afterwards values in a cluster are replaced by its mean. This is done 2 times more such that we get the final cluster with $d = 1$.

In Figure 2.3 the advantages of an iterative binarization in contrast to the direct binarization is shown. The left figure shows the the clustering with only two cluster ending up in the final binarization with a high loss of information. In contrast to the right figure, showing the more clusters are used the more information is kept. Furthermore has iterative clustering the advantage of maintaining the information about oscillations.

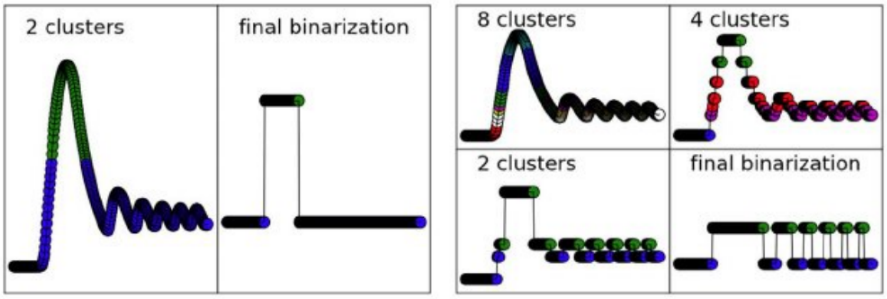


Figure 2.5: Direct binarization vs. Iterative binarization. More detailed binarization values are yielded in iterative binarization with a higher valu of d

BASC A -Binerization

The BASC A binerization algorithm is a bit more complex, thus here a short description is given. For more details have a look in the paper The algorithm is divided up into three parts, starting with computing a series of stepfunctions, then finding the strongest continuity in each step function and estimate the location and variation of the strongest discontinuities. Firstly an intial step function is obtained by rearranging the original time-series measurements in increasing order. Step functions with fewer discontinuities are calculated such that each minimizes the eucledian distance to the initial step function. Afterwards the strongest discontinuity in each step function is found by a high jump size (derivative) in combination with a low approximation error. Finally time-series measurements of gene expression values can be excluded from the network based on the estimations of the location and variation of the strongest discontinuities.

2.3 Boolean Network,Interaction Graph and State Transition Graph

In this section the knowledge of the discretisized biological data is put into a graphtheoretical context of a boolean network. Here mathematical definitions are given, explained by an example.

Definition 2.2. Undirected Graph and Directed Graph
In general, an undirected graph $G = (V, E)$ is defined as a set of vertices V describing the nodes of the system and a set of undirected edges $E = \{(i, j)|i, j \in V\}$ that define a relationship between node i and j. While in a ***directed graph*** *$G = (V, A)$ is an ordered pair, defined as a set of vertices V (nodes) and a set of directed edges A (arcs). A set of directed edges $A = \{(i, j)| \in V\}$ describes the flow of information in network, where (i, j) describes the flow from i (tail)to j (head).*

Definition 2.3. Boolean Network
A boolean network is a directed graph $G(X, F)$, defined by a set of nodes X in a binary vector $X(t) = \{x_1(t), ..., x_n(t)\}$ representing state of a system at time t. Each element $x_i \in X$ corresponds to the state $x_i = 1$ or $x_i = 0$ of a species i. A set $F = \{f_1, ..., f_n\}$ of n transition functions (resp.: set of boolean transition functions $B = \{f_1, ...f_n\}$) contains a particular function for each x_i. Every transition function $f_i \in F$ is therefore a n-variable Boolean function $f : \mathbb{B}^n \leftarrow \mathbb{B}$ which we can represent by a Boolean expression over n input variables.
For every $f_i \in F$ s.t. $1 \leq i \leq n$,

$$f_i(X(t)) = x_i(t + 1)$$

, where $f_i(X(t))$ defines the next state of x_i at time t in the network.

A Boolean network model is a directed graph whose nodes represent the elements of a system, edges represent orientation of regulatory relationships between elements, and every node can have two possible initial states $x_i \in \{0, 1\}$ describing the its activity [12-14]. The activity of a node means a qualtative rate a gene is being transcribed $x_i = 1$ or not $x_i = 0$, a transcription factor is active oder inactive, a protein's concentration is above or below a certain threshold (e.g. phosporylated or un-phosphorylated). Thus a network with n nodes will have 2^n possible states. As time passes the state of each node is determined by the states of its neighbors, through a rule called a transition function, determining the future state of a node. This transition function is represented as a boolean function using the logical operators NOT, AND, OR (resp. &&, ||, !;resp.$\wedge$,$\vee$,$\neg$). For instance a gene x_A could be influenced by another gene x_B positive and by a second gene x_C negative. Then the boolean algebra would look like this:

$$x_A = x_B \text{ AND NOT } x_C$$
$$x_A = x_B \text{ \&\& } !x_C$$
$$x_A = x_B \wedge \neg x_C$$

The application of the transition function to node's state returns a True or False state. Depending on the output of the transition function, the state of the node either stays the same or changes.

The Interaction Graph is some kind of abstraction of boolean network by capturing just the dependencies between the variables and discard further details. These dependencies are visualized by the following graph-based representation.

Definition 2.4. Interaction Graph

The interaction graph of a boolean network $G(X, F)$ is a directed graph $IG(X, \rightarrow)$ that consists of the node set X and the arc set $\leftarrow = \leftarrow_F \subseteq X \times X$ with $(x_i, x_j) \in \rightarrow$ iff f_{x_j} depends on x_i

In an interaction graph for each node being described by another one, this connection is written $x_i \rightarrow x_j$ (resp. x_i 1 x_j) and for the case that there is no connection $x_i \nrightarrow x_j$ (resp. x_i 0 x_j), respectively. Another characteristic of interactions is wether they are activating or inhibiting or both depicted by the *Sign* of an edge.

Definition 2.5. *Sign* of an edge

The Sign of an edge is defined by $Sign(x_i \rightarrow x_j \subseteq \{+, -\})$.

Then the expression $x_i \rightarrow x_j$ is either $x_i \xrightarrow{+} x_j$ (resp. x_i 1 x_j) describing an activating connection, $x_i \xrightarrow{-} x_j$ (resp. x_i -1 x_j) describing an inhibitory connection or both $x_i \xrightarrow{+,-} x_j$ (resp. x_i $1, -1$ x_j).

Furthermore the dynamics of a system can be simulated by repeatedly applying the all transition functions to the variables and updating their states. This computation leads from an Interaction Graph to a State Tansition Graph.

Definition 2.6. State Transition Graph

A State Transition Graph $STG(X, \rightarrow)$ is a directed graph with a set of nodes represented by *a set of binary vectors* $F(t+1) = \{f_1(X(t+1)), ..., f_n(X(t+1))\}$ *representing the updated states of all variables after all of the functions in* F *have executed. The arcs denote possible transitions from one binary state vector to another.*

The State TransitionGraph is either updated *synchronously*, where each variable's state is updated simultaniously after all of the transition functions in F have executed or *asynchronously*, where the states are updated one at randomly choosing a transition functio $f_i \in F$ and updating the state of x_i immediatly.
The following example shows how an Interaction Graph looks like, how the transition functions (resp.boolean rules) are derived from this graph and how the State Transition Graph is constructed for the synchronous and asynchronous case.

Example 2.7. The Interaction Graph in Figure 2.6 show a boolean network with three nodes $X = \{x_1, x_2, x_3\}$ and four activating arcs (black arrow) and two inactivating arcs (red arrow).

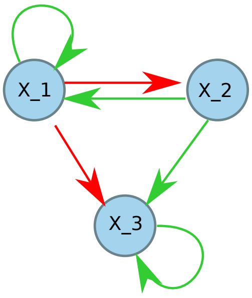


Figure 2.6: Interaction Graph. Three nodes $\{"x1", "x2", "x3"\}$ (blue circle), where activating interactions are represented by black arcs and ihibiting interactions are represented by red arcs.

From the nteraction graph the set of transition functions F (resp. boolean functions) can be derived (2.1) such that the state transition graph can be calculated.

$$f(x_{1,2,3}) = \begin{pmatrix} x_1 & \wedge & x_2 & \wedge & \neg x_3 \\ \neg x_1 & & & & \\ & & x_2 & \wedge & x_3 \end{pmatrix} \tag{2.1}$$

For every possible state of $x_i \in X$ the next state $f_i(X(t)) = x_i(t+1)$ is calculated shown in the computation (2.1)-(2.8) below. This computation provide the new states for a synchronously updated state interaction graph displayed in the left graph in Figure 2.5.[ÃĽlisabeth Remy et al., 2008]

$$x(t) = (0,0,0) \quad \rightarrow \quad f(x(t+1)) = (0,1,0) \tag{2.2}$$

$$x(t) = (0,0,1) \quad \rightarrow \quad f(x(t+1)) = (0,1,0) \tag{2.3}$$

$$x(t) = (0,1,0) \quad \rightarrow \quad f(x(t+1)) = (0,1,0) \tag{2.4}$$

$$x(t) = (1,0,0) \quad \rightarrow \quad f(x(t+1)) = (0,0,0) \tag{2.5}$$

$$x(t) = (1,1,0) \quad \rightarrow \quad f(x(t+1)) = (1,0,0) \tag{2.6}$$

$$x(t) = (1,0,1) \quad \rightarrow \quad f(x(t+1)) = (0,0,0) \tag{2.7}$$

$$x(t) = (0,1,1) \quad \rightarrow \quad f(x(t+1)) = (0,1,1) \tag{2.8}$$

$$x(t) = (1,1,1) \quad \rightarrow \quad f(x(t+1)) = (0,0,1) \tag{2.9}$$

The asynchronous updated state transition graph is computed by adding intermediate computation steps to the synchronus computation. Therefore the red highlighted computations (2.3),(2.7) and (2.9) are split by the amount of state changes. For instance in the computation step (2.2) $x(t) = (0,0,1)$ has two updates of states, x_2 changes and x_3 changes, too. As we know from the descriptin of asynchronous STG's in biological systems processes happen uncommonly at the same time. Thus the initial state $x(t) = (0,0,1)$ provides two possible updates: $f(x(t+1)) = (0,1,1)$ and $f(x(t+1)) = (0,0,0)$. The same procedure is done with (2.6) and (2.8) and finally the asynchronous STG can be drawn, shown by the right graph in Figure 2.5.

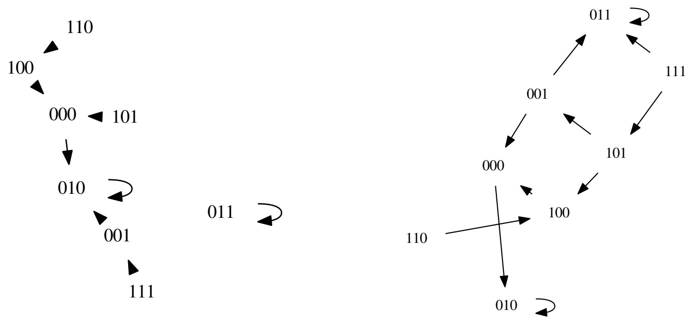


Figure 2.7: Synchronous and Asynchronous State Transition Graph. Left: Synchronous State Transition Graph; Right: Asynchronous State Transition Graph

2.4 Network Evaluation

After inferring a boolean network from biological data the structural performance of this network should be assessed to show how well a prediction of a model fits the observed biological data. For this reason the predicted connections between the nodes of a boolean network $G(V, A)$ are divided up into four possible outcomes displayed in a confusion matrix in Table 2.1. In general, an observed data set and a predicted one is compared providing a True Positive (TP) outcome where the model correctly predicts the positive class, a True Negative (TP) outcome where the model correctly predicts the negative class and False Positive (FP) is an outcome where the model incorrectly predicts the positive class and a False Negative (FN) is an outcome where the model incorrectly predicts the negative class. Referring a boolean network, TP and FP denote the numbers of correctly and incorrectly predicted connections,respectively. And FN denotes the number of non-inferred connections in $G(V, A)$ and TN is the number of correct negative predictions. [?]

True Positive (TP): • Observed Value • Prediction Value • **Number of TP**	**False Posotive (FP):** • Observed Value • Prediction Value • **Number of FP**
False Negative (FN): • Observed Value • Prediction Value • **Number of FN**	**True Negative (TN):** • Observed Value • Prediction Value • **Number of TN**

Table 2.1: Confusion matrix displaying all four possible outcomes.

In the following different formula are defined used for later structural network analysis. To get things straight concerning application and interpretation of the formula in this section an example (**Example 2.16**) regarding real life data is presented at the end of this section.

Definition 2.8. Accuracy is the percentage of correct predictions.

$$\boxed{Accurancy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{TP + TN}{TP + TN + FP + FN}} \tag{2.10}$$

Definition 2.9. Precision returns the proportion of positive indentifiers which was actually correct.

$$\boxed{Precision = \frac{TP}{TP + FP}} \tag{2.11}$$

Definition 2.10. Recall returns the proportion of actual positives identified correctly.

$$\boxed{Recall = \frac{TP}{TP + FN}} \tag{2.12}$$

Which means, *Recall* provides a percentage of inferred connections among the true connenction in $G(V, A)$.
For determining the Receiver-Operating-Characteristic-Curve a True-Positive-Rate (TPR) and a False-Positive-Rate (FPR) is calculated.

Definition 2.11. True-Positive-Rate (TPR). The TPR values are for the y-axis of the ROC.

$$\boxed{TPR = \frac{TP}{TP + FN}} \tag{2.13}$$

Definition 2.12. False-Positive-Rate (FPR). The FPR values are for the x-axis of the ROC.

$$\boxed{FPR = \frac{FP}{FP + TN}} \tag{2.14}$$

Accuracy is not always an appropriate scoring method, because, assigning every object to a larger set achieves a high proportion of correct predictions, but is not generally a useful classification. For this reason Blanced Accuracy and the Mathew Correlation Coefficient are introduced.

Definition 2.13. Balanced Accurancy (BACC). Fraction of predictions our model got right divided by 2.

$$\boxed{BACC = \frac{\frac{\text{Number of correct predictions}}{\text{Total number of predictions}}}{2} = \frac{\frac{TP+TN}{TP+TN+FP+FN}}{2}} \tag{2.15}$$

Definition 2.14. Matthew Correlation Coefficient (MCC). The MCC measures the quality of binary classifications and is a correlation coefficient between observed and predicted binary classifications which returns a value between -1 an 1; MCC $\in [-1, 1]$. Where a value close to 1 denote a perfect prediction, a value close to 0 means that the prediction is not better than a random one and a value close to -1 describes a total disagreement between prediction and observation.

$$\boxed{\text{MCC=} \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}} \tag{2.16}$$

Example 2.15. Given is a model that classified 100 tumors as malignant (the positive calss) or benign (the negative class):

True Positive (TP): • Reality Malignant • Prediction: Malignant • **Number of TP: 1**	**False Posotive (FP):** • Reality: Benign • Prediction: Malignant • **Number of FP: 1**
False Negative (FN): • Reality: Malignant • Prediction: Benign • **Number of FN: 8**	**True Negative (TN):** • Reality: Benign • Prediction: Benign • **Number of TN: 90**

Table 2.2: Confusion matrix displaying all four possible outcomes.

The confusion matrix in **Table 2.2** shows that only one malignant tumor and 90 not malignant tumors were predicted right by the model and 8 tumors were predicted wrongly being benign and one being wrongly malignant. Now the performance of the model is calculated:

$$Accuracy = \frac{1+90}{1+90+1+8} = 0.91 \tag{2.17}$$

$$Precision = \frac{1}{1+1} = 0.5 \tag{2.18}$$

$$Recall = \frac{1}{1+8} = 0.11 \tag{2.19}$$

$$TPR = \frac{1}{1+8} = 0.11 \tag{2.20}$$

$$FPR = \frac{1}{1+90} = 0.01 \tag{2.21}$$

$$BACC = \frac{\frac{1+90}{1+90+1+8}}{2} = 0.46 \tag{2.22}$$

$$MCC = \frac{1*90-1*8}{\sqrt{(1+1)(1+8)(90+1)(90+8)}} = 0.21 \tag{2.23}$$

The *Accurary* (2.16) has a value of 0.91 which means 91% of the 100 total examples are predicted correctly. This result may look good at first sight, but this dataset is class-imbalanced. In a class-imbalanced data set the labels of a binary classification problem have significantly different frequencies. For example, a disease data set in which 0.0001 of the examples have positve labels and 0.9999 have negative labels is a class-imbalanced

problem. But a football game predictor in which 0.51 of example label one team winning and 0.49 label the other team winning is not a class-imbalanced problem.
Thus the significant disparity between the number of positive (here: $TP + TN = 91$) and negative labels (here: $FP + FN = 9$) falsifies the result. This observation is supported by the values of the $BACC$ and the MCC. The $BACC$ (2.21) has a value of 0.46, telling us that the prediction of the model is not that good as the *Accurary* shows and the MCC (2.22) has a value of 0.21 which is quite close to a value of 0. Thus the model predicted rather randomly than significantly. Furthermore the model has a *Precision* of 0.5, meaning when it predicts a tumor is malignant, it is correct 50% of the time. The *Recall* results in a value of 0.11, meaning the model correctly identifies 11% of all malignant tumors. In relation to this example it is worthwhile to identify most of the malignant tumors (high TP value) and get a low number of unidentified malignant tumors (low FN value).

3 Materials and Methods

3.1 Inferencealgorithms

Different approaches of inferring networks are known capturing several advantages and limitations. It is worhtwhile to find a good trade-off between simplicity, scalability, expressiveness and finding the abstraction level for a significant network (model)[?]. Many infernce algorithms are known based on different computational models like the Bayesian approach , the ordinary diffential equation approach (ODE), the artificial neural networks)(ANN) and the Boolean model (BN).[?]

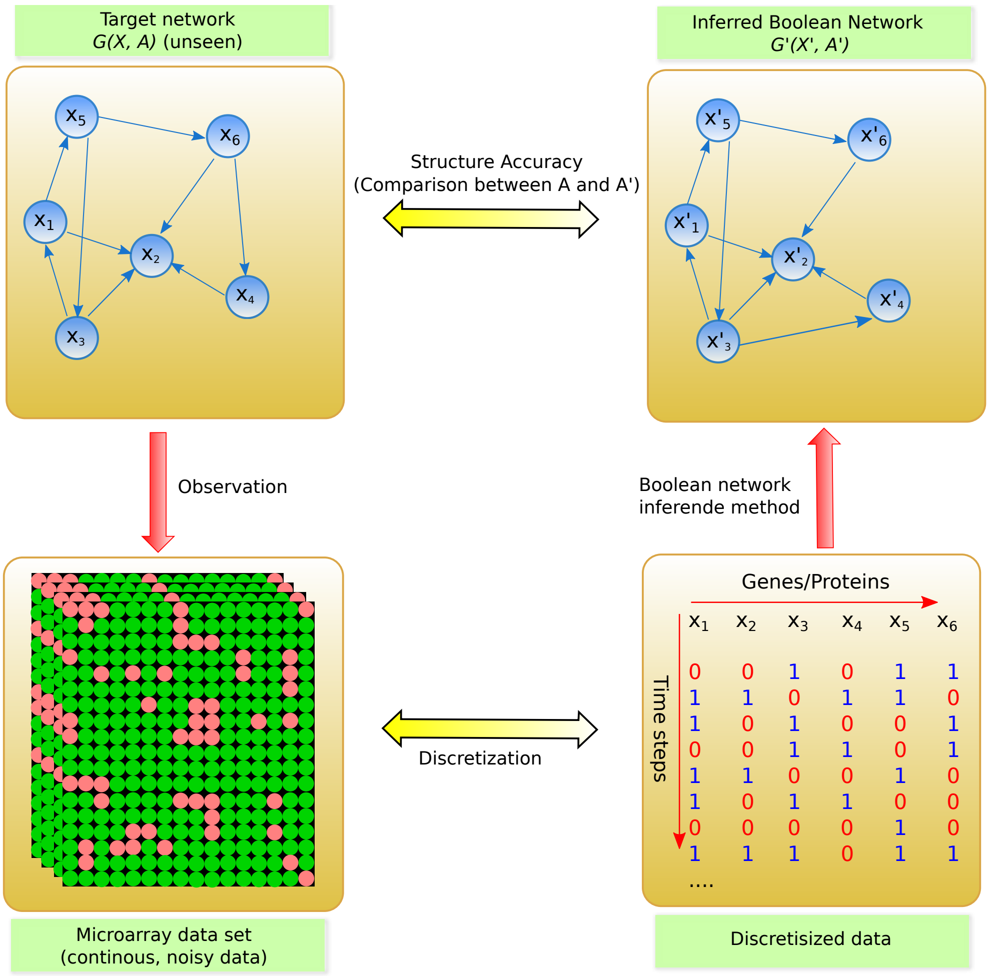
A Bayesian model is a graph-based model of joint multivariate probability distributions that captures properties of conditional independence between variables. (In other words given a gene A and a gene B and a third gene C, then A and B are conditionally independent given C iif, given knowledge that C occurs,knowledge of wether A occurs provides no information on the likelihood of B occuring,and knowledge of wether B occurs provides no information on the likelihood of A occuring.)

The system of differential equations (ODE's) creates networks in consideration with the kinetic properties of a biological system. An ODE is a powerful and flexible model to describe complex relations among components. But the higher the complexity of an unseen network is the more challenging it is to determine an appropriate set of equations which describe the network. [?]

Artificial Neural networks gather their knowledge by detecting patterns and relationships in data and learn through experience. An ANN is constructed by weighted processing elements which constitute the neural layers and are organized in layers. Thus the behaviour of a ANN is determined by a transition function of each variable (neuron), by a learning rule and by the architecture itself. A big advantage, no previous knowledge is needed [?].

Boolean Models are simple Boolean Networks which are well known and an appropriate strategy of inferring the structure, the dynamics (time-series data) and the steady-state (attractor analysis) behaviour of complex data. Analysis of this model provide an axtraction of the important informations of a network. Boolean Network Models do not need any information about kintic parameters [Berestovsky & Nakhleh, 2013]. The relationships in a boolean network model can be derived from a realtively small dataset. Furthermore a boolean model could make qualitative predictions of large complex networks more feasible. For this reason the boolean approach is chosen to show especially the scalibility from a small example data set to a bis real-life data set.Boolean Models are either probabilistic (PBN) or determinstic (DBN). In a PBN the next state of a node is determined by a transition function f_i selected with a certain probability from a set of transition function F. But this approch is limited due to the complexity of computational effort and the state-transitions and steady-state distributions [?]. In a deterministic Boolean Model the the next state of a node is determined by its particular transition function f_i, such that the application of a certain transition function f_i to its corresponding node x_i always yields for an the initial state (0 or 1) the same corresponding updated state.

The input of our problem consists of time-series data $S = \{S_1, ..., S_n\}$ of n species, each of size $m + 1$,where $S_i(t) \in \mathbb{R}^+(0 \leq t \leq m)$ is the concentration of species i at time t, and the output is a Boolean network N on the n species. After the binarization step the time-series data is converted into a set of binary tranjectories $B = \{B_1, ..., B_n\}$ (one per species). The network N is then learned from B



x_1	x_2	x_3	x_4	x_5	x_6
0	0	1	0	1	1
1	1	0	1	1	0
1	0	1	0	0	1
0	0	1	1	0	1
1	1	0	0	1	0
1	0	1	1	0	0
0	0	0	0	1	0
1	1	1	0	1	1

Figure 3.1: Overview of inferrinf a Boolean Network Model. An unseen Target network $G(X, A)$ of a system is perubed and the expressions are measured over a series of time captured in a microarray data set. The continous and noisy data is then normalized, cleared up (redundancy removal, outlier removal) and discretisized to a binary time-series data set. A boolean infernce algorithm infers a boolean network model $G'(X', A')$ which is compared to Goldstandard data network.

[?] In research a variety of boolean model have been published but their implementation efforts are rarely published,too. In [Berestovsky & Nakhleh, 2013] the code of the implementation of the boolean model REVEAL,BESTFIT and FULLFIT is provided with the three different dicretization methods two-k-means and iterative-k-means clustering. In this work this code is validated in such a way that real-life time-series data of the DreamChallenge can be applied and evaluated in a Pipeline including PyBoolNet and an scoring programm.

Redundancy Removal

Real-life time course data could contain redundant information about steady-states and the significance of a transition. A steady-state is a point attractor which is a pair of equal consecutive states indicating a true-steady state iif. it was the last pair in the series. Data measured in a very fine time-scale is often giving false indication of steady-state signal. To overcome this false-steady-state-transition-problem each maximal consecutive sequence of identical states are all removed except for one of the states.
Regarding the significance of a transition the average number of bits needed is determined. If the average number of bits is above 1, this reduction skips informative transitions, which could be important in the learning step.

REVEAL

REVEAL (REVerse Engineering ALgorithm) is an inference algorithm which uses a deterministic transition table to infer boolean relationships between variables. After maximal 2^n iterations of the algorithm a "steady-state" (resp. point attractor) should be found with reprsent the boolean rule (transition function). REVEAL deals with calculation of individual's entropy in combination with the joint entropy and the mutual information.

Definition 3.1. Shannon-Entropy
The Shannon-Entropy is the probaility of observig a particular symbol of event $p(x)$, within a given sequence.

$$\boxed{H = -\sum p(x) log p(x)} \tag{3.1}$$

Example 3.2. Here $p(x)$ is the probability of observing value $x \in \{0, 1\}$ for a variable x_i.

x	0	1	1	1	1	1	1	0	0	0
y	0	0	0	1	1	0	0	1	1	1

$$H(x) = -p(0) * log[p(0)] - [1 - p(0)] * log[1 - p(0)] \tag{3.2}$$
$$H(x) = -0.4log(0.4) - 0.6log(0.6) \quad = 0.97(40\%0, 60\%1) \tag{3.3}$$
$$H(x) = -0.5log(0.4) - 0.5log(0.6) \quad = 1.00(50\%0, 50\%1) \tag{3.4}$$

An element x can have two possible states 1 (on) or 0 (off). H reaches its maximum when both possible states equally probable $H_{max} = log(2) = 1$. Beside the individual entropy of x and y now the combined entropy is consulted.

Definition 3.3. Joint Entropy The joint entropy is defined by the pobability of occurences that x and y occur dependend on each other.

$$\boxed{H(x,y) = -\sum p(x,y) log p(x,y)} \tag{3.5}$$

Example 3.4. The co-occurences of 1 and 0 in x and y are displayed in a quadratic matrix. Afterwards the joint entropy $H(x,y)$ is calculated in (3.6).

$$\begin{array}{cc|c|c|}
\cline{3-4}
\multirow{2}{*}{$\mathbf{y}$} & 1 & 3 & 2 \\
\cline{3-4}
 & 0 & 1 & 4 \\
\cline{3-4}
\multicolumn{2}{c}{} & \multicolumn{1}{c}{0} & \multicolumn{1}{c}{1} \\
\multicolumn{2}{c}{} & \multicolumn{2}{c}{\mathbf{x}}
\end{array}$$

$$H(x,y) = -0.1log(0.1) - 0.4log(0.4) - 0.3log(0.3) - 0.2log(0.2) = 1.85 \tag{3.6}$$

In the last computational step the mutual information is calculated by the combination of the Shannon-Entropy and the Joint-Entropy.

Definition 3.5. Mutual Information The mutual information describes the rate of transmission.

$$\boxed{M(X,Y) = H(X) + H(X,Y) - H(X,Y)} \tag{3.7}$$

This equation can be extended n-times, for n nodes in anetwork.

$$\boxed{M(X,[Y,Z]) = H(X) + H(Y,Z) - H(X,Y,Z)} \tag{3.8}$$

The smallest subset x' that yields $M(x_i, x_i')/H(x_i = 1)$ reflect the set of nodes (resp. genes, proteins) whose states determine the next state of the gene represented by a variable x_i.

Example 3.6. With the knowledge about Shannon entropy, joint entropy and the mutual information the boolean rules (resp. transition functions) for the example of state transition tables with the node set $n = A, B, C$ can be calculated.

Transition table "B":

Input entopies	
H(A)	1.00
H(B)	1.00
H(C)	1.00
H(A,B)	2.00
H(B,C)	2.00
H(A,C)	2.00
H(A,B,C)	3.00

Table 3.1

→

Determine the mutual information for A		
H(A’) 1.00		
H(A’,A) 2.00	M(A’,A) 0.00	M(A’,A)/H(A’) 0.00
H(A’,B) 1.00	M(A’,B) 1.00	M(A’,B)/H(A’) 1.00
H(A’,C) 2.00	M(A’,C) 0.00	M(A’,C)/H(A’) 0.00

Table 3.2

input	at	time (t)	input	at	time ($t+1$)
A	B	C	A’	B’	C’
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	1	1
1	1	1	1	1	1

If $M(A', X) = H(A')$ then $M(A', X)/H(A') = 1$, then X exactly determines A'. This is here the case for B in A', where A' denotes the output’s state shown in the red highlighted line in Table 3.2. The iterationof REVEAL stops here and the boolean rule (resp. transition function)can be inferred.

input	output
B	A
0	0
1	1

Table 3.3

→ $f_A = B$

For further details on the calculation of the transition functions f_B and f_C the reader is referred to the paper [TS2B [REVEAL-PAPER]].

REVEAL works incrementally by first checking single combination of variables, then checking every pair, then every triplet and so on, until the perfect combination of all variables is found.

BESTFIT

The second algorithm BESTFIT (Best-Fit Extension) uses partially defined Boolean functions ($pdBf$). A $pdBf(T,F)$, where $T, F \in \{0,1\}^k$, consists of two vectors T, defines the set of true examples and F, the set of false examples extracted from the binarized time series data. The goal is to find a perfect Boolean classifier. The unique occurences of the pairs $X'(t)$ and $X_i(t+1)$ are added to $pdBf(T,F)$ for each time-step $0 < t < m-1$. Where $X_i(t+1)$ describes the new state of X_i at time step $(t+1)$ explained the best by a set of variables $X'(t) \subseteq \{X_1, ..., X_n\}$ of size $k \leq n$ with the least error size. Here k denotes the indegree value, which describes the number of incoming edges to a node. Thus, a node can have maximally an indegree value of n, neglecting information about the sign of an edge. A $pdBf(T,F)$, where $T, F \in \{0,1\}^k$, consists of two vectors T, defines the set of true examples and F, the set of false examples extracted from the binarized time series data.The goal is to find a perfect Boolean classifier:

$$T = \{X'(t) \in \{0,1\}^n : X_i(t+1) = 1\} \tag{3.9}$$

$$F = \{X'(t) \in \{0,1\}^n : X_i(t+1) = 0\} \tag{3.10}$$

Further, the error size ϵ is defined by the size of the intersection of sets $\epsilon = (T \cap F)$. Now the X' with the lowest error describing $X_i(t)$ the best is chosen. Then the undefined entries in the corresponding $pdBf(T,F)$ are randomly assigned to extract a deterministic function. This algorithm incrementally finds the smallest subset of inputs to explain X_i.

FULLFIT

This algorithm works almost the same as BESTFIT with the only difference that the algorithm only accepts the function with $\epsilon = 0$. Ideally, after all possible, fully consistent, functions are obtained, all resulting networks can be enumerated by choosing a single function for each X_i. In practice this could be become infeasible.

Error Assesement with BooleanNet

The application of a learning algorithms returns multiple solution, depending on the initial states of the nodes. Thus the network fitting the best to the data should be selected. For this reason an error assessment strategy was invented with the help of a Boolean simulator

sp called BooleanNet.
The data set provides a set of binary trajectories $B = \{B_1, ..., B_n\}$ for which an learning algorithm is applied to, to generate boolean network N. N contains the set of transition functions, describing the nodes states. N is used in BooleanNet to generate a new set of binary trajectories Y, whose length is equal to B. The first state in Y is equal to the first state in B. Here BooleanNet simulaniously updates all the states according to asynchronous simulation. Then the error of a boolean network N with respect to B is defined by:

$$\boxed{Error(N, B) = \frac{\sum_{1 \leq t \leq M}[(|B(t) - Y(t)|) * I_n]}{n * M}} \tag{3.11}$$

The difference of B to the simulated Y in dependence on I_n a n-dimensional vector of all ones and M representing the number of binarized states in the reduced time-series. The lower the error the better the model fits the date. For this reason the later implemented code just takes the model with the lowest error for further steps in the pipeline.

3.2 PyBoolNet

3.3 Data Selection

3.3.1 Example data set

3.3.2 HPN-DREAM breast cancer data set

Now it is shown how the Pipeline can be applied to a real-life time course data set.

What is the Dream Challenge?

For a Boolean network inference the data of a platform so-called Dialogue on Reverse Engineering Assessment and Methods (DREAM) - Challenge is used. The DREAM-Challenge is a non-profit, collaborative community effort consisting of contributors from across the research spectrum of questions in biology and medicine. This organization was built in 2006 and publishes crowdsourcing challenges with transparent sharing of data, thus everyone can participate the challenge. The DREAM-Challenge has partnered with Sage Bionetworks, which provide the infrastructure by Sage Bionetworks Synapse platform to get access to the open collaborative data analysis. Overall the DREAM-Challenge is a helpful instrument to get real-life data, comparing results and interact with other researchers all over the world, while contribute solutions to biological and medical questions.[?]. The challenging question was to decide which Dream Challenge data set could be useful for this masterthesis. For inferring a Boolean network and further analysis of the state tarnsition graph the desired data set should contain measurements of experiments with less pertubational information and a in a time course context. The Dream Challenge 5 dealing with gene-gene interaction,providing test and training data sets of gene expressions seemed to be an appropriate candidate. But there was less time course information and a high abundance of pertubation. Thus the Dream8 Challenge is was chosen. This challenge describes protein-protein interaction and measurements for multiple timepoints.

DREAM8

Data Collection

The collection of the HPN8 breast cancer PPI data is done by a technique so called Reverse phase protein array(RPPA). This technique is divided up into 6 parts:

Sample collection An inhibitor or stimulus in form of drugs is added to a set of celllines at the same time and the celllines are then processed at different time points.

Cell Lysis Cell fragments are lysed with a celllysis buffer to obtain high protein concentration.The choice of a buffer decides the quantity of proteins can be lysed out of the cell.

Dilution Dilution of the celllysed probes.

Antibody screening The lysates are pooled and resolved by SDS-PAGE followed by western blotting on a nitrocellulose membrane. The membrane is cut into 4mmm strips. Each slide is probed with a different antibody, primary with a secondary antibody.

Fluorometric detection Primary and secondary antibody are diluted.Detection reagentisput on each slide. Signal amplification and detection is done by an optical flatbed scanner if colormetric technique is used orby laser scanning.

Data set structure Missing data points and outliers are detected and deleted from the data set. The data set is normalized

4 Appilcation of the Pipeline

4.1 Example data set

4.2 HPN-DREAM breast cancer data set

5 Results

5.1

5.2

5.3

6 Conclusion

References

Berestovsky, N., & Nakhleh, L. (2013, 06). An evaluation of methods for inferring boolean networks from time-series data. *PLOS ONE*, *8*(6), 1-9. Retrieved from `https://doi.org/10.1371/journal.pone.0066031` doi: 10.1371/journal.pone.0066031

Chai, L. E., Loh, S. K., Low, S. T., Mohamad, M. S., Deris, S., & Zakaria, Z. (2014). A review on the computational approaches for gene regulatory network construction. *Computers in Biology and Medicine*, *48*, 55 - 65. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0010482514000420` doi: https://doi.org/10.1016/j.compbiomed.2014.02.011

De Las Rivas, J., & Fontanillo, C. (2010, 06). ProteinâĂŞprotein interactions essentials: Key concepts to building and analyzing interactome networks. *PLOS Computational Biology*, *6*(6), 1-8. Retrieved from `https://doi.org/10.1371/journal.pcbi.1000807` doi: 10.1371/journal.pcbi.1000807

Elena S. Dimitrova, J., M. Paola Vera Licona. (2010). Discretization of time series data. *Journal of Computational Biology*(1), 853 - 868. Retrieved from `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3203514/` doi: http://doi.org/10.1089/cmb.2008.0023

Kestler, H. A., Wawra, C., Kracher, B., & Kuehl, M. (2008). Network modeling of signal transduction: establishing the global view. *BioEssays*, *30*(11-12), 1110–1125. Retrieved from `http://dx.doi.org/10.1002/bies.20834` doi: 10.1002/bies.20834

Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., Gerber, G. K., ... Young, R. A. (2002). Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, *298*(5594), 799–804. Retrieved from `http://science.sciencemag.org/content/298/5594/799` doi: 10.1126/science.1075090

Pellegrini, M., Haynor, D., & Johnson, J. M. (2004). Protein interaction networks. *Expert Review of Proteomics*, *1*(2), 239-249. Retrieved from `https://doi.org/10.1586/14789450.1.2.239` (PMID: 15966818) doi: 10.1586/14789450.1.2.239

Saadatpour, A., & Albert, R. (2013). Boolean modeling of biological regulatory networks: A methodology tutorial. *Methods*, *62*(1), 3 - 12. Retrieved from `http://www.sciencedirect.com/science/article/pii/S1046202312002770` (Modeling Gene Expression) doi: https://doi.org/10.1016/j.ymeth.2012.10.012

ÃĽlisabeth Remy, Ruet, P., & Thieffry, D. (2008). Graphic requirements for multistability and attractive cycles in a boolean dynamical framework. *Advances in Applied Mathematics*, *41*(3), 335 - 350. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0196885808000146` doi: https://doi.org/10.1016/j.aam.2007.11.003

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne unerlaubte fremde Hilfe angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen und Hilfsmitteln wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Berlin, den 07. September 2018

. . . .

(Unterschrift .....)