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

Declaration of Originality

I hereby declare that this thesis and this work reported herin was composed by and originated entirely from me.

Information derived from published and unpublished work of others has been acknowledged in the text and references are given in the list of sources.

Berlin, November 20,2018

Nina Valery Kersten

Contents

List of Figures

List of Tables

List of Abbreviations

ATP	adenosin triphosphat
CSV	Comma Seperated Values
e.g.	exempli gratia
ELISA	Enzymatic Immunoassay
GRN	Gene regulatory network
IG	Interaction Graph
MAP	Mitogen Activated Protein
mRNA	messanger RiboNucleotide Acid
PPI	protein protein interaction
resp.	respectively
RPMA	Reverse phase Protein lysate MicroArray
RTK	Receptor Tyrosin Kinases
SDS-PAGE	Sodium Dodecyl Sulfate - Polyacrylamide Gel Electrophoresis
STG	State Transition Graph

Abstract

1. Introduction and Review

This chapter describes the different kinds of biological input data by taking a focus on the biological background of an experimental data set regarding protein-protein interaction of growth factor induced transcriptional signal pathways for inferring regulatory networks. The type of input data and its experimental data collection decides about the choice of an inference model and its interpretation.

Afterwards, biological data is put into a graph-theoretical context of a Boolean network. Additionally, this part shows how to capture the dynamics and structural properties of a Boolean network.

1.1. Motivation

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 (messenger RiboNucleotide Acid) and metabolic substances take place in a cascade of extracellular signals transduced 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 adaptation of a cell. Malfunctioning has been identified as the cause of various diseases [1].

Recent advances in high-throughput techniques provide a big abundance of information about various biological interaction measured over a series of time. It is necessary to handle this big data properly for significant structural and dynamic analysis. Therefore, experimental data is converted into a network structure, where a substance is represented by a node and an interaction is represented by an edge. Mostly substances are part of a big and highly interconnected system where the kinetic information is rarely known [2]. Thus, the overall problem of inferring a regulatory network from biological time-course data is to find a trade-off between simplicity, scalability and explanatory power of a network, such that the main important components of a malfunctioning system can be identified for creating medical solutions [1] [3].

Several models are known of inferring a network from time-course data, which are characterized being either continuous like Ordinary Differential Equation (ODE) models, Baysian models or discrete like Boolean networks (BN)[2].

An Ordinary Differential Equation (ODE) creates networks considering kinetic properties of a biological system. This model is a powerful and flexible model to describe complex relations among substances. But kinetic information are rarly available, such that ODE is only a sufficient strategy when a small well known system is analyzed [2].
A Bayesian model is a graph-based model of joint multivariate probability distributions that captures properties of conditional independence between variables [4].

Boolean network models are well known and do not need any kinetic information of a system. Hence, this model simplifies the representation of a complex system, while capturing the main structure and dynamics as well as the steady-state behaviour [1][2]. Experimental settings are often financially expensive, such that the time course data sets are quite small. Boolean networks are known being able to derive relationships among substances from relatively small data sets (e.g. 50 measurements for one substance) [1]. For this reason the Boolean network model is chosen in this thesis. In a Boolean network model, the initial state of a node is either true (1) or false (0) and the next state is updated according to a Boolean rule composed by other nodes affecting the nodes state. Updating a nodes' state represents the measurement of sample points over a series of time, such that Boolean trajectories of the model are generated.

In general, a pipeline for inferring Boolean networks starts with a normalized time-course data set, which is preprocessed by discretisizing continous values into binary values and remove redundant information. Subsequently, an inference algorithm learns Boolean rules for each node from the data resulting in a set of rules. The predicted model is tested on itself and the predicted network is scored against a gold standard network.

This work is divided up into two parts for each a pipeline is implemented (available on Git: 'github.com/ninakersten/Masterthesis'). In the first part Boolean networks are learned from *in silico* data sets of *E.coli* and the mammilian cell cycle by three well known inference algorithms Best-Fit, Full-Fit and Reveal [3]. The predicted networks are scored against a gold standard, testing the inference algorithms and binarization algorithms on several parameters. In the second part results about the algorithms' performance of the first part are taken into account for inferring a real-life time course data set of phophoprotein abundance measurements in breast cancer cell lines (Figure 1.1). This data is provided by the Reverse Engineering Assessment and Methods (DREAM)-Challenge platform. The DREAM-Challenge is a non-profit, collaborative community effort consisting of participants from across the research spectrum of questions in biology and medicine. On this platform real-life data is provided to everyone, such that everyone can participate by contributing own solutions and compare them to other researchers [5, 6].

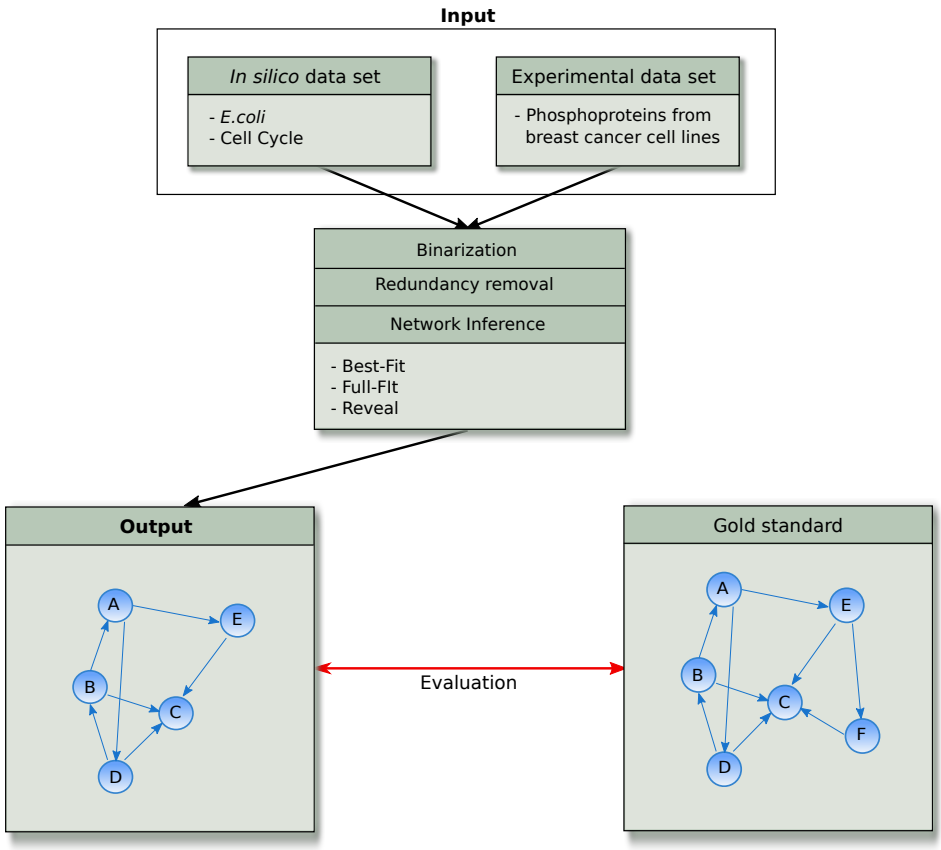

Figure 1.1.: Raw Pipeline. Continuous input data of either E.coli, the cell cycle or of phosphoprotein interaction in breast cancer cell lines are binarized and redundant data is removed. A network is learned from an inference algorithm (e.g.: *Best-Fit, Full-Fit, Reveal*), its predictive power is evaluated by scoring the output against a gold standard network.

This thesis aims to show that a Boolean approach might be a sufficient strategy for inferring big real-life time course data sets by capturing the structure and dynamical properties of a system while emphasizing the limitations which might occur.

1.2. Biological Background

Biological interaction can be observed at different levels of information integration of a cell regarding metabolic interaction, gene-gene interaction and protein-protein interaction. Information integration starts with a signal, which binds to a membrane integrated receptor of the cell causing an activation 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 results in a protein. The resulting proteins influence the cell's survival by its proliferation, induction of cell differentiation and apoptosis.

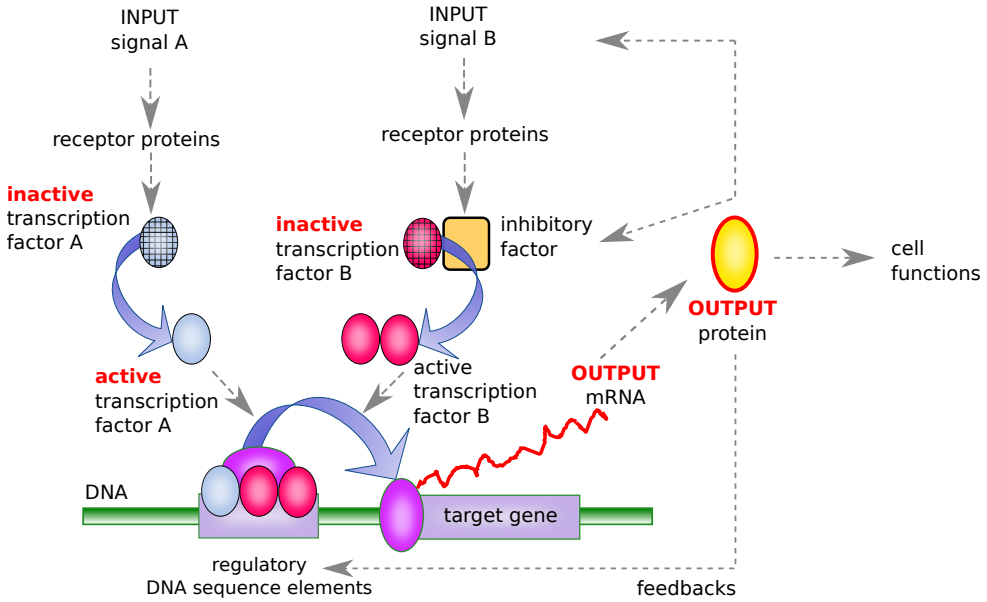

Figure 1.2.: Transcriptional Signal Cascade Two different input signals A and B bind to a specific receptor protein. The complex of A activates the transcription factor A that binds directly to the gene's regulatory sequence inducing the expression of the target gene. Different to A initiates the complex of B a separation of the inactive B (*pink oval*) from an inhibitory 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 product which can take place in several processes of the cell [7].

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 [8, 9].

Metabolic network

At the level of metabolic networks the substances are highly interconnected in a quite complex way (e.g. cell respiration in Figure 1.3). An individual's metabolism is determined by its genetics, environment and nutrition.[10]. In a metabolic network the nodes are depicted by different types of biochemical components connected by directed edges describing a activating or inactivating 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 [11].

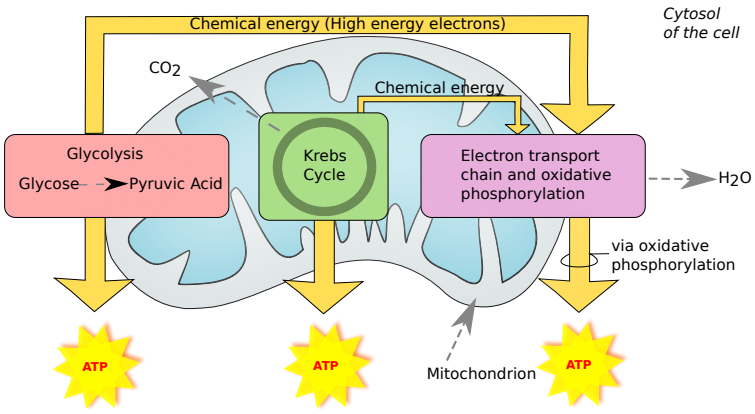

Figure 1.3: Metabolic Network for cell respiration. In a mitochondrion, an essentiell compartment of the cell, converts energy derived from nutrition into biochemical energy ATP(Adenosin TriPhosphate) by releasing waste products of water H_2O and CO_2.

[12]

Gene regulatory network

In a gene regulatory network (GRN) depicted in Figure 1.2 the interaction of genes are identified indirectly by the abundance measurement of their transcriptional products (e.g. mRNA, proteins). The nodes of a GRN are depicted by the genes' names and the edges are directed by showing whether a gene produces proteins which inhibit or activate a target gene [13].

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 the membrane 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 co-expressed [14][15].

The real-life data set of the DREAM-Challenge used in this work is dealing with PPIs, thus, it is important to know for later data collection, data processing and discussion how the data is obtained and which role these PPIs play in a biological context.

Referring to the general description of a transcriptional signal cascade (Figure 1.2) we state the receptor being an enzyme-associated receptor and the input signals are growth factors. An enzyme associated receptor has a polypeptide chain integrated in the cell's membrane with a tyrosine kinase activity (Figure 1.4).

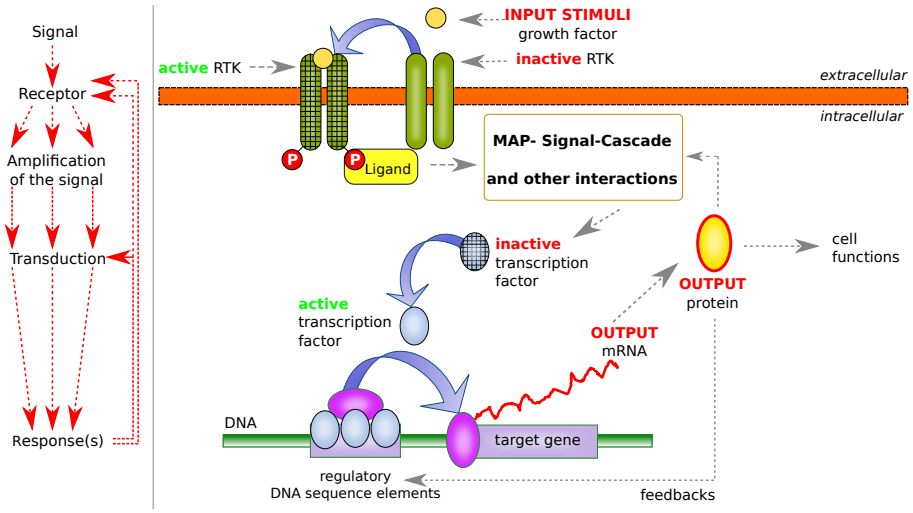

Figure 1.4.: Transcriptional Signale Cascade An incoming stimuli (*yellow circle*) representing a growth factor binds to an inactive RTK (*green*), such that the RTK is activated. The RTK amplifies the signal and initiates a signal transduction cascade, such that an inactive transcription factor (*blue oval*) is activated, which binds to regulatory DNA sequence elements inducing the mRNA transcription of a target gene resulting in a new protein [7].

Growth factor receptors with a tyrosine kinase activity are called receptor tyrosine kinases (RTKs). These RTKs have the property of autophosphorylation, meaning that 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 (resp. proteins) a phosphorylating cascade (e.g. signal transduction cascade) is induced. In this mitogen activated protein (MAP) phosphorylation cascade the MAP-kinase katalyzes the phosphorylation of effector proteins, such that inactive transcription factors are activated starting the transcription process of a target gene and finally resulting in a protein product [16].

In the DREAM-Challenge growth factors are selected depicted as a stimuli by pertubating the cell's signal transduction. Thus, different stimuli cause different signal transduction cascades. Depending on the incoming stimuli particular proteins take place in the signal transduction cascade. The goal is to figure out PPIs in this phosphorylation cascade by inferring a Boolean network based on the measurement of the proteins abundance considering eight different incoming growth factors (resp. stimuli) displayed in Table 1.1. dysregulation of the genes of these growth factors have been linked to diseases such as cancer, schizophrenia, bipolar disorder and many more [17].

Notation	Name	Description
IGF1	Insulin like Growth Factor 1	Hormone, similar to the insulin function and structure [18]
NRG1	Neuregulin 1	Membrane glycoprotein, mediating cell-cell signalling, critical role in growth and developement of the cell [17]
HGF	Hepatocyte Growth Factor	Regulate cell growth, cell motility and morphogenesis [19]
FGF1	Fibroblast Growth Factor 1	Functions as a modifier of endothelial cell migration, proliferation and an angiogenic factor[20]
Insulin	Insulin	Mutations in this gene are associated with type II diabetes and susceptibility to insulin resistance [21]
EGF	Epidermal Growth Factor	This protein acts a potent mitogenic factor that plays an important role in the growth, proliferation and differentiation of numerous cell types [22].
PBS	Translocator Protein (TSPO)	Present mainly in the mitochondrial compartment of peripheral tissues. The protein is a key factor in the flow of cholesterol into mitochondria to permit the initiation of steroid hormone synthesis [23] [23].
Serum	SRF	Member of the MADS box superfamily of transcription factors [24]

Table 1.1.: List of growth factors (stimuli) of DREAM8 Challenge

Reverse phase Protein lysate MicroArray

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-based assay that provides quanitative measurements of protein abundance [25]. 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 cell lines at the same time and the cell lines are then processed at different time points. Secondly in the cell lyses step cell fragments are lysed with a cell lysis buffer to obtain high protein concentration. The choice of a buffer decides the quantity of proteins that can be lysed out of the cell. Afterwards cell lysed probes are diluted. In the Antibody screening the lysates are pooled and resolved by SDS-PAGE (Sodium Dodecyl Sulfate - Polyacrylamide Gel Electrophoresis) followed by western blotting on a nitrocellulose membrane [26]. The membrane is cut into 4mmm strips. Each slide is probed with a different antibody, where a primary antibody is extended by a secondary antibody. For fluorometric

detection primary and secondary antibody are diluted (Figure 1.5). The resulting data is normalized, such that outliers are excluded from the data's structure [27].

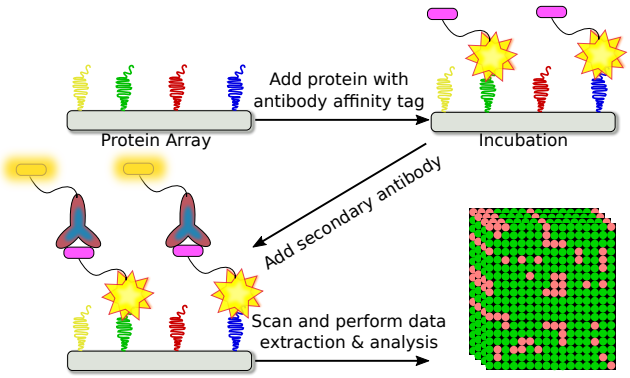

Figure 1.5: RPMA: Antibody binding and fluorometric detection. Proteins are tagged by a specific antibody. After incubation time the secondary antibody is added. Finally the abundance of proteins is determined by a fluorometric detection.

The strength of RPMA is the high throughput, ulta-sensitive detection of proteins from extremely small numbers of input material which is not possible for western blotting and ELISA (Enzymatic Immunoassay) [27]. 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 [28]. The high sensitivity of RPMA 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[29]. A great improvement of RPMAs over traditional forward phase protein arrays is a reduction in the number of antibodies needed to detect a protein [29, 30]. 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 RPMA as the samples can be denatured, revealing any covered epitopes (part of a protein recognized by specific antibody)[30].

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 signalling proteins for which antibodies exist to get an analyzable signal is quite small.

1.3. Graph-theoretical Background

This section is dealing with defining the graph-theoretical properties of a Boolean network N in terms of its structure and dynamics.

Definition 1.1. Boolean Network
A Boolean network N is defined by an n-dimensional binary vector $X = (x_1, ..., x_n)$, where each element $x_i \in X$ corresponds to the state $x_i = 1$ or $x_i = 0$ of a species i. Then a set F of n transition functions contains a particular function f_i for each species x_i [1]. Every transition function $f_i \in F$ is therefore a n-variable Boolean function $f : \mathbb{B}^n \to \mathbb{B}$, which is represented by a Boolean expression over n input variables [31].
For every $f_i \in F$ s.t. $1 \leq i \leq n$,

$$f_i(X(t)) = x_i(t+1) \tag{1.1}$$

, where $f_i(X(t))$ defines the next state for each x_i at time $(t+1)$ in the network.

In a biological context the activity of a node is a qualitative rate whether a gene is being transcribed $x_i = 1$ or not $x_i = 0$, a transcription factor is active or 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. [2][32][**?**].
Then a transition f_i describes a rule of n node defining by their activating or inhibitory influence on a target node the next state of a that node [1].

For instance, a gene x_A is activated by another gene x_B and inhibited by a second gene x_C. Then the transition function of x_A is $f_{x_A} = x_B \wedge \neg x_C$ and the Boolean algebra looks like this;

$$x_A = x_B \;\&\&\; !x_C \tag{1.2}$$

, where a Boolean function is a composition of nodes $x_i \in X$ and logical operators represented by '&&', '||', '!' (resp. 'NOT', 'AND' and 'OR' ;resp. '$\wedge$' ,'$\vee$' , '$\neg$') [2][1].

A Boolean network can be abstracted into a graphical representation of an interaction graph capturing the structural properties of it. Therefore the notation of a directed graph is introduced.

Definition 1.2. Directed Graph
A directed graph $G = (V, A)$ *is an ordered pair, defined as a set of nodes* $V = \{v_1, v_2, ..., v_n\}$ *and a set of directed edges* A *denoted as 'arcs'. A set of directed edges* $A = \{(i, j)|i, j \in V\}$ *describes the flow of information in a network, where* (i, j) *describes a flow from* i *(tail) to* j *(head)* (Figure 1.6).

[33]

Definition 1.3. Interaction Graph
The interaction graph (IG) (resp. dependency raph) of a Boolean network N *is a directed graph* $IG(X, A)$ *that consists of the node set* X *and the arc set* $A = A_F \subseteq X \times X$ *with* $(x_i, x_j) \in A$ *iff* f_{x_j} *depends on* x_i*, then:*

$$a_{ij} = \begin{cases} 1 & \text{if there is an edge from node i to node j} \\ 0 & \text{otherwise} \end{cases} \tag{1.3}$$

[**?**]

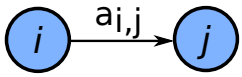

Figure 1.6: Directed Graph *G*. $a_{i,j}$ is a directed edge (arc) depicting the flow of information from node i to node j.

In an interaction graph for each node being described by another one, this connection is written 'x_i 1 x_j' and for the case that there is no connection 'x_i 0 x_j', respectively.

Furthermore an interaction graph provides information about nodes having a positive or negative influence on other nodes depicted by the *Sign* of an edge. This term is introduced for completion and is not necessary for later investigation in this thesis.

Definition 1.4. *Sign* of an edge
The sign of an edge is defined by $Sign(x_i \to x_j) \subseteq \{+, -\}$.

Then the expression $x_i \to x_j$ 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).

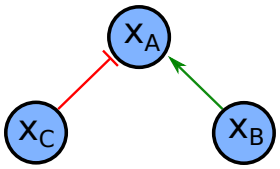

Figure 1.7: Interaction graph *IG*. Relating to the Boolean rule (1.2) a node x_A is activated (*green*) by x_B and inhibited (*red*) by x_C.

The number of incoming edges of a node denoted by the ***in-degree*** of a node is a sufficient parameter for complexity analysis in a network.

Definition 1.5. ***In-degree***
The *in-degree* of a node is the number of incoming edges to a node determining its state.

$$k_i^{in} = \sum_j a_{ij} \tag{1.4}$$

[3**?**] Hence, the out-degree describes the number of outgoing edges of a node. Nodes with only outgoing edges (in-degree= 0) are called sources, and nodes with only incoming edges (out-degree= 0) are sinks of the network. The higher the *in-degree* of a node, the more nodes are part of its transition function f and the complexity increases. Here, the *in-degree* of x_A in Figure 1.7 is $k_{x_A}^{in} = 2$.

Furthermore the dynamics of a system (resp. the state change behaviour of a node over a series of time) can be simulated by repeatedly applying the set F of transition functions to the corresponding set of nodes X and updating their 'current' state [1].
In a ***synchronous*** simulation, the states of all nodes are updated simultaniously after all transition function of F have been applied to all nodes X. In contrast to the ***asynchronous*** simulation, the states are updated by randomly choosing a transition function $f_i \in F$ and updating the state of x_i in the exact time [34, 35, 32, 36]. In biological processes interaction of substances rarely happen at the same time, thus dynamical analysis is an important factor of detecting interacting substances.

These two terms of updating a node's state are abstracted in a so called *state transition graph* [2].

Definition 1.6. State Transition Graph
A state transition graph (STG) 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.*

[2, 37**?**]
Starting from an initial state in the state transition graph and iteratively updating the state of the nodes, the state of the system evolves over time by following a trajectory of states until the nodes' states do not change anymore: $X(t) = X(t+1)$, so called ***steady-state*** [2, 35]. These steady-states describe the 'true' state of the nodes in a system, thus the 'true' connection of the nodes to each other.

The following example demonstrates deriving F from an interaction graph IG, calculating the corresponding set of binary trajectories $B = \{B_1, ..., B_n\}$ (one per species) updated in a synchronous and asynchronous manner resulting in two state transition graphs.

Example 1.1 The interaction graph in Figure 1.8 shows a Boolean network with three nodes $X = \{x_1, x_2, x_3\}$ (*blue circle*), where positive (resp. activating) edges (*green*) and negative (resp. inhibiting) edges (*red*) represent the interaction between the nodes. The *in-degree* of $k^{in}_{x_1} = 2$, $k^{in}_{x_2} = 1$ and $k^{in}_{x_3} = 3$.

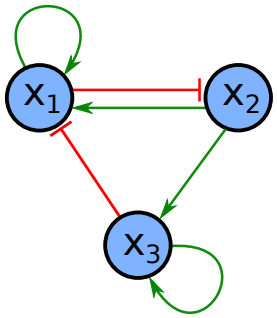

Figure 1.8.: Interaction Graph

From the interaction graph the set of transition functions F (resp. Boolean functions) can be derived (1.6) such that the state transition graph can be calculated.

$$F(x_{1,2,3}) = \begin{pmatrix} x_1 & \wedge & x_2 & \wedge & \neg x_3 \\ \neg x_1 & & & & \\ & & x_2 & \wedge & x_3 \end{pmatrix} \tag{1.5}$$

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 (1.7)-(1.14) below [38].

$$x(t) = (0,0,0) \quad \rightarrow \quad f(x(t+1)) = (0,1,0) \tag{1.6}$$
$$x(t) = (0,0,1) \quad \rightarrow \quad f(x(t+1)) = (0,1,0) \tag{1.7}$$
$$x(t) = (0,1,0) \quad \rightarrow \quad f(x(t+1)) = (0,1,0) \tag{1.8}$$
$$x(t) = (1,0,0) \quad \rightarrow \quad f(x(t+1)) = (0,0,0) \tag{1.9}$$
$$x(t) = (1,1,0) \quad \rightarrow \quad f(x(t+1)) = (1,0,0) \tag{1.10}$$
$$x(t) = (1,0,1) \quad \rightarrow \quad f(x(t+1)) = (0,0,0) \tag{1.11}$$
$$x(t) = (0,1,1) \quad \rightarrow \quad f(x(t+1)) = (0,1,1) \tag{1.12}$$
$$x(t) = (1,1,1) \quad \rightarrow \quad f(x(t+1)) = (0,0,1) \tag{1.13}$$

As we know from the description of asynchronous STG's in biological systems processes happen uncommonly at the same time, which is observed by comparing state transition graph of the synchronous (A) and asynchronous (B) model in Figure 1.9. In contrast to the synchronous STG where each state has a unique successor, in the asynchronous STG multiple successors of a trajectory are possible (e.g. (1.8),(1.12),(1.14)) [2].

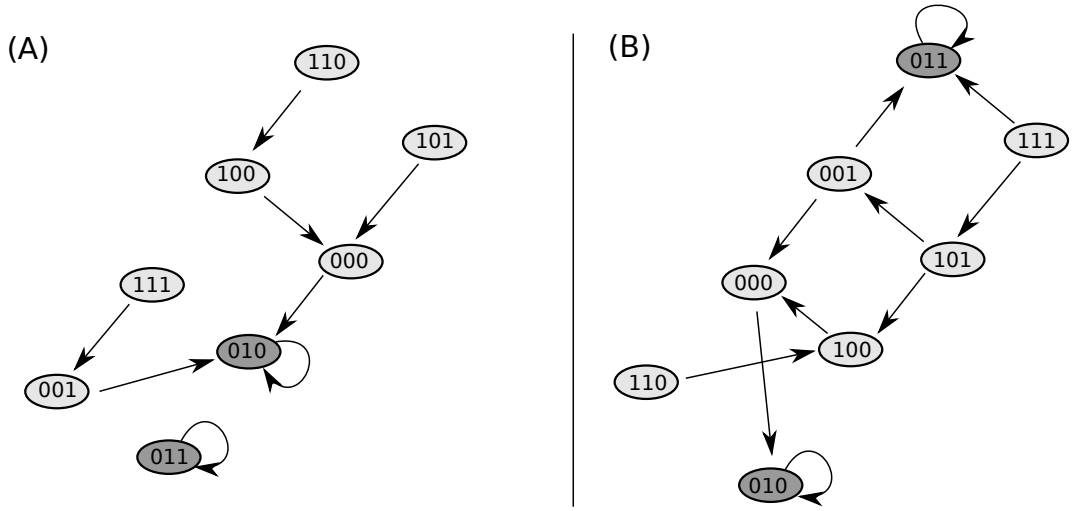

Figure 1.9.: Synchronous and Asynchronous State Transition Graph. (A): Synchronous state transition graph; and (B): Asynchronous state transition graph of a Boolean network. Binary digits from left to right depict the state of the nodes x_1,x_2,x_3. The dark gray states are possible steady-states in the system.

2. Materials and Methods

A common method in network inference (resp. machine learning) is to split an experimental data set into a training data set and a smaller test data set which provides the 'gold-standard' used to evaluate the model. Thus, the test set provides an unbiased evaluation of a final model. For tuning the parameter of a model a validation set so called '*in silico*' data set is additionally taken into account. Experimental settings often contain complex abundance of information (e.g. many nodes in an observed system, adding multiple drugs causing different pertubations), such that assessing the performance for improving the parameter selection of an inference algorithm is quite pretentious. Hence, the *in silico* data contains unbiased data covering the main properties of the training data obtained through computational simulation [39].

Here, an *in silico* data set is created from *E.coli* for investigating the runtime and influence of the *in-degree* considering performance of an inference algorithm. Additionally a second *in silico* data set derived from the cell cycle network which is used to assess the performance of the inference algorithms regarding different cluster depths for binerization of continuous data and the number of sample points.

The DREAM8-HPN-DREAM Breast Cancer Network Inference Challenge provides the training data set and a test data set. Participants of this challenge used the test data for assessing their models, but in this thesis two different gold standard data sets are selected instead (3.3). The aim of selecting gold standard different from the actual challenge is to get a more sufficient evaluation.

Inferring a Boolean networks is a complex combination of data preprocessing (e.g. normalization, discretization, redundancy removal) and choosing an appropriate inference algorithm as well as a sufficient evaluation strategy. An implementation of a pipeline published by Natalie Berestosky [1] containing discretization methods (2-k-means, iterative k-means), redundancy removal, three well known inference algorithms (Best-Fit,Full-Fit,Reveal) and error assessment is extended and validated, such that a real-life data set can be applied and the predicted network structure can be assessed (Figure 2.1).

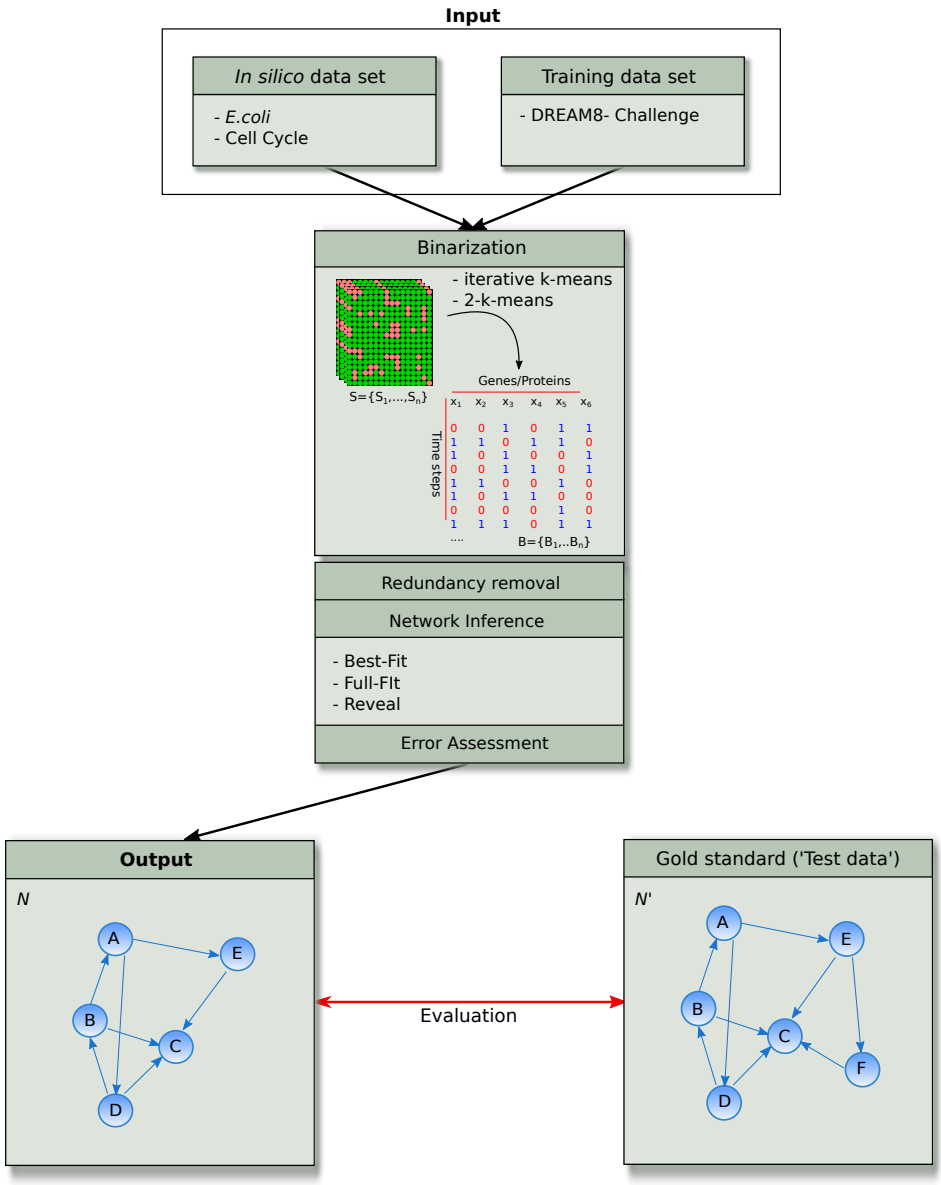

Figure 2.1.: Extended Pipeline. Continous data sets S of *in silico* and a training set are discretisized to binary trajectories B. Redundant information is removed and the network fitting the best to the data is returned. The inferred network N is scored against a gold standard network N'.

2.1. Data collection: *In silico* data set

Several strategies are known to generate an *in silico* data set, such that it was first tried to generate the *in silico* data set independent on a real life organism, by applying the Barabasi-Albert (BA) model or generate multiple sets from one example network [40].
The more sufficient method turned out to be creating an *in silico* data set by extracting subnetworks from the *E.coli* network by a tool, so called *GeneNetWeaver*. *E.coli* (*Escherichia coli*) is a well studied bacterium consisting of 1565 genes (resp. nodes) with 3758 interaction (resp. edges) [41]. For assessing the performance of the inference algorithms regarding an increasing number of incoming links a set of four networks with 10 to 14 nodes, each extended to 9 subnetworks are generated, such that 45 networks are yielded. For example, a subnetwork of E.Coli can have a set of 10 nodes with a maximal *in-degree* of $k_i^{in} \in \{1, ..., 9\}$. The range of 10 to 14 is selected due to the fact that *Reveal* is not performing by a system of 15 nodes (computational limited by: 8 RAM and a Core i5 processor) and starting by 10 is for better comparison of the performance measurement to literature [3].

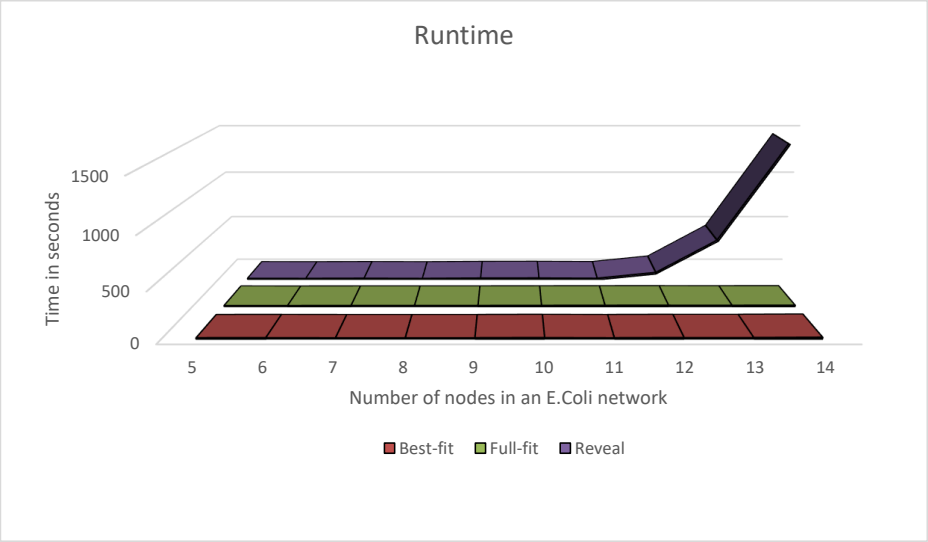

Figure 2.2.: Starting by subnetworks of *E.coli* of 5 nodes to a network of 14 nodes *Reveal* is running out of time.

[42][43] In addition a small real-life network of the mammalian cell cycle (Figure 2.3) is used to asses the performance of the algorithms regarding the number of sample points and the clustering depth in the binerization step. The cell cycle network is taken from the repository

of PyBoolNet. PyBoolNet is a python package for the generation, analysis and visualization of Boolean networks.
[42] The cell cycle is a process of signal transduction leading to the reproduction of the genome of a cell (Synthesis or s phase) and its division into daughter cells (Mitosis, or M phase). Positive signals or growth factors cause the activation of Cyclin D (CycD) in the cell, which inhibits the retinoblastoma protein Rb. Rb is a key tumor suppressor, which is mutated in large variety of cancer cells [44]. This cell cycle network consist of 10 interacting transcriptional counterparts of genes with 35 edges and has a maximal in-degree of 5.

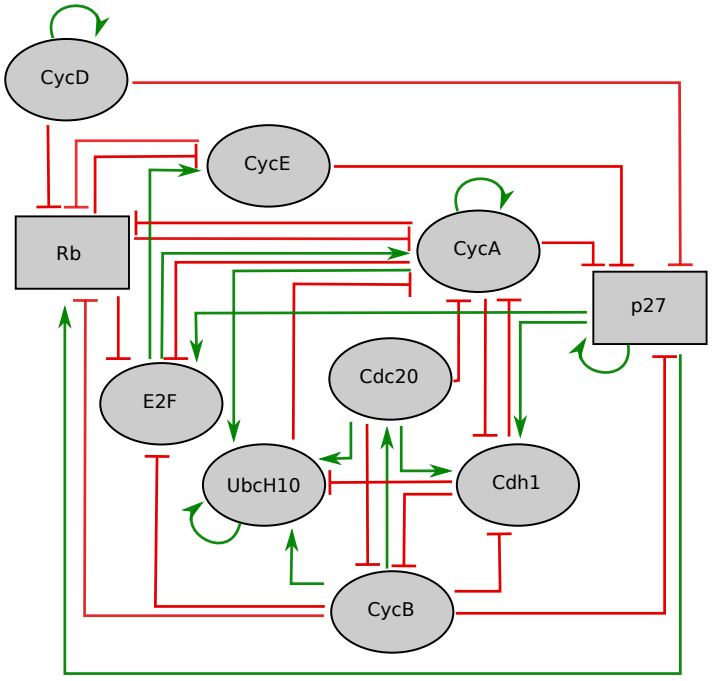

Figure 2.3.: Cell Cycle. Logical regulatory directed graph for the mammalian cell cycle network. Each node represents a key regulatory element and each edge the interaction between them. Arrows describe activating activity (green) while blunt arrows describe inhibitory activity (red).

2.2. Data collection: Training data set

The challenging question is to decide which Dream Challenge provides appropriate data for testing a Boolean model. The prevalent requirements for an experimental data are measurements of experiments with less pertubational information in a time-course context with at least 50 sample points, such that all of the three algorithms are applicable.
The DREAM5-Challenge is dealing with gene-gene interaction, providing test, training data sets and a gold standard of gene expressions seemed to be an appropriate candidate. But there is less time-course information and a high abundance of pertubation (e.g.. knock out experiments, gene deletion experiments, applied drugs and environmental pertubations and

dosages of the dugs), such that inferring a network by considering these additional information is quite challenging.
In contrast to the DREAM5 Challenge the DREAM8 Challenge provides micro-array data with less pertubational information (eight stimuli), but enough sample points by about ∼85 for in each data set. Therefore this Dream Challenge is selected.

2.2.1. DREAM8 Challenge

The "DREAM 8 - HPN-DREAM Breast Cancer Network Inference Challenge" took place in 2016 and was running for 3 month. The challenge focuses on inferring causal signaling networks by detecting phosphoproteins on signalling downstream of receptor tyrosine kinase (RTK) in human cancer cell lines. Figure 2.4 shows an example.

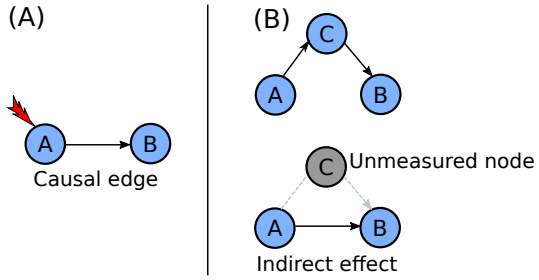

Figure 2.4: Causal edges. (a) Inhibition of a parent node A (red arrow) can change the abundance of child node B. (b) If node A influences B causally by measuring node C, then the causal network should contain edges from A to C and from C to B, but not from A to B.

The real-life time-course data is provided by the Heritage Provider Network (HPN). More than 2000 networks were submitted by challenge participants. The networks spanned 32 contexts and were scored in terms of their structure and dynamics. The challenge shows that a significant network can be obtained by merging certain submitted network (submissions with high performance) to an aggregated network such that the community based approach proves that an aggregated network yields a higher performance in contrast to a single submission [45].

The challenge comprised three sub-challenges: Causal network inference (SC1), time-course prediction(SC2) and visualization (SC3). The sub-challenge SC1 is divided up into two parts A and B. In A the interaction graph with information about edge occurence is inferred and confidence score (resp. edge weights) indicating the strength of evidence in favour of each possible edge is calculated. In B the causal network is created and in the other two sub challenges the phosphoprotein time-course data is predicted under further perturbations and in the last challenge methods are developed to visualize these complex, multidimensional data sets. This work focuses on sub-challenge SC1-A considering the inference of the interaction graph without taking the edge weight or *Sign* into account. Thus the scalability of the three inference algorithms from a small network to a big one can be assessed.

2.2.2. Data structure

The challenge spanned 32 different contexts, each defined by a combination of 4 cell lines (BT20, UACC812, BT549, MCF7) and 8 stimuli (Insulin, Serum, HGF, NRG1,EGF, FGF1,GF1,IGF1) (Table 1.1) and three kinase inhibitors and a control DMSO (Dimethyl sulfoxide). The inhibitors inhibit the kinase activity of their target, which means they inhibit the ability of the target to catalyse phosphorylation of its substrates but not necessarily inhibit the phophorylation of the target itself [45]. All cell lines are provided by the American Type Culture Collection (ATCC) and where chosen because they represent the major subtypes of breat cancer (basal, luminal, laudin-low and HER2-amplified) and known to have different genomic aberrations [46] [47][48]

Experimental setting

Protein arrays were carried out using RPMA, an antibody detection method described in the biological background in chapter 1. Each cell line was serum-starved for 24hours and then treated for 2 hours with an inhibitor (or combination of them). Cells were then either harvested (0 time point) or stimulated by one of the eight stimuli for 5, 15, 30 or 60 minutes or for 2 or 4 hours (Figure 2.5) [49][45]. *adapted from* [45]

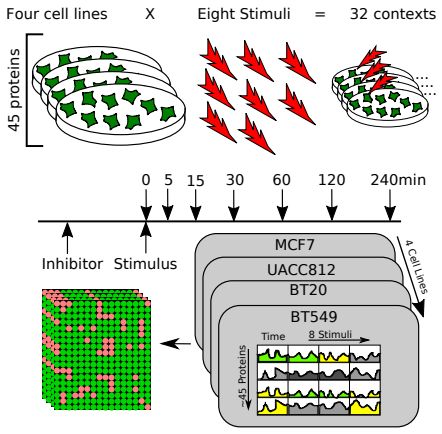

Figure 2.5: Data Collection of the DREAM8 Challenge data set. Four cell lines (MCF7, UACC812, BT20, BT549) are treated with eight stimuli resulting in 32 contexts. Every experiment starts by adding an inhibitor followed by a stimulus. The concentration of $\sim$ 45 phosphoprotein detecting antibodies is measured after 5,15,30,60,120 and 240 minutes.

In each of the 32 contexts time-course data for $\sim$ 45 phophoproteins measured up to four hours depicted as the 'main-data set'. The number of measured phophoproteins varies across the cell lines due to the antibodies, which evolve over time during the experiments. Participants who were interested in more phosphoproteins and more sample points were referred to a 'full-data set' with measurements up to 72 hours and an amount of up to 125 phophoproteins [45]. As in the challenge the inference is focused on the 'main data set', which is here the training data set. The provided data (for each of the 4 cell lines) is contained in a Comma Separated Values (CSV) file format.

Normalization

Normalization is an essential step, because data can contain outliers, the abundance of some proteins or mRNA is often higher than others and obtaining biological data from the lab could cause several batch effects. Therefore each data set is already normalized by the laboratory by converting raw data from log_2 values to linear values. For each antibody across the sample set the median is determined. The median value denote the middle position when all the observations are arranged in an ascending or descending order. It divides the frequency distribution exactly into two halves. Fifty percent of observations in a distribution have scores at or below the median. Hence median is the 50th percentile and also known as the 'positional average'[50].
Each raw linear value is divided by the median to get the median-centered ratio. Then the median of the median-centered ratio is calculated for each sample across the entire amount of antibodies. This median functions as a correction factor, thus each sample has its own correction factor. If the correction factor is above a value of 2.5 or below a value of 0.25 then the sample is considered as an outlier and extracted from the data set. Finally each median-centered ratio is divided by the correction factor resulting in normalized linear values [45] [51, 52].

Training data

A CSV file is structured by four headers starting with the 'Slide ID' containing information about the protein name, it's phospholylation site, antibody type, antibody validation status, and the antibody slide number (Table 2.1).

			Slide ID	4E-BP1_pT37_T46-R-V_GBL9026591	...
			Antibody Name	4EBP1_pT37_pT46	...
			HUGO ID	EIF4EBP1_pT37_pT46	...
Cell Line	**Inhibitor**	**Stimulus**	**Timepoint**		
BT20	Inhibitor		0	3.0724988347	...
BT20	Inhibitor		0	3.168004721	...
BT20	Inhibitor		0	3.0629789682	...
BT20	...		...	...	...
BT20		Insulin	5	3.3041031492	...
BT20		FGF1	5	4.315396736	...
...	...	...	...	...	...

Table 2.1.: Training data: CSV structure

The 'Antibody Name' describes the protein name, the phosphorylation site and antibody type (e.g.,Antibody name: '4EBP1_ pT37_pT46', where '4EBP1' depict the phophoprotein). The third header is a 'HUGO ID' an approved nomenclature of the proteins in combination with

the the phosphorylation site. For the network inference the antibody names are depicted as the node names in a network. The last header shows the type of a cell line, the inhibitor, the stimulus, the time point of measurement followed by the concentration measured for each phosphoprotein detecting antibody.

2.3. Binarization Algorithms

Normalized continous time course data $S = \{S_1, ..., S_n\}$ of n species (e.g. genes, 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. A binarization algorithm aims to categorize this data into discrete values, which simplifies the input data for applying an Boolean inference algorithm. Hence, the data set S is turned into a set of binary trajectories $B = \{B_1, ..., B_n\}$ (one per species), where the state of a species (e.g. gene or protein) is binarized to a value of 1 or 0, such that a Boolean network N can be inferred from B (Figure 2.1)[1]. It is worthwhile to find good trade-off between simplification of the data and preserving information about the system.

Two clusters k-means binarization

Given a set of time course data $S = \{S_1, ..., S_n\}$, where each observation $S_i(t) \in \mathbb{R}^+$ $(0 \leq t \leq m)$ is a d-dimensional real vector, two k-means binarization aims to partition the n observations into $k = 2$ cluster $C = \{C_1, C_2\}$ by setting an observation's value $S_i(t)$ to 1 if it is above the overall mean $\mu(S)$ and setting to 0 if an observation's value $S_i(t)$ is below (2.1).

$$S_i(t) = \begin{cases} 1 & \text{, if } S_i(t) \geq \mu(S) \\ 0 & \text{, if } S_i(t) \leq \mu(S) \end{cases} \tag{2.1}$$

[53]
This binarization strategy is fast and simple but may exclude some essential information like about oscillations and fluctuations in a system (e.g. cell cycle). Therefore the iterative k-means binarization method is introduced. [1]

Iterative k-means binarization

An initial depth d of clustering is set followed by a set of initial number of cluster $k = 2^d$. Then the method is divided up into two parts:

(1) In each iteration the data of each species S_i is classified into k disjoint clusters $C^1_{S_i}, ...C^x_{S_i}$. In each Cluster all its values are replaced by the clusters mean $\mu(C^x_{s_i})$.

(2) In the next iteration step d is decremented by one and the clustering is repeated.

This iteration continues until $d = 1$, where the data in the cluster with lower values of the mean are replaced by 0 and higher values are replaced by 1.

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 initially $k = 8$ clusters for each gene in the data set. This results in eight cluster $\{C^1_{s_A}, ..., C^8_{s_A}\}$ containing the time course data of gene A. Now for each cluster the mean $\{\mu(C^1_{s_A}), ..., \mu(C^8_{s_A})\}$ is calculated. Afterwards values in each cluster are replaced by its mean. This is done 2 times more by decrementing d, such that $d = 1$ and all values of A being higher the overall mean are set to 1 the one lower are set to 0.

2.4. Redundancy Removal

Detecting the *steady-state* in a Boolean network is necessary to indicate the significance of a transition. Measuring on fine time-scale and binarization of the data could cause false indication of *steady-states* by the inference algorithm. This could lead to wrong interpretation of node interaction in a Boolean network. For this reason false *steady-states* have to be removed from the binarized data set B. Thus, except the last pair in the time-course data, each maximal consecutive sequence of states is removed, except one state. The remaining last pair in the time-course data set should indicate the true steady-state [1].

2.5. Inference algorithms

Several Boolean network based inference algorithms are known [54],[55],[56], [57], [58] ,here three well known deterministic Boolean models are applied: REVEAL, Best-Fit and Full-Fit. In a deterministic Boolean model 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 initial state (0 or 1) the same corresponding updated state. Whereas in a probabilistic Boolean network 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 approach is limited due to the complexity of computational effort and the state-transitions and steady-state distributions [35],[59],[55].

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 which is represented by a Boolean rule (transition function). REVEAL is dealing with the calculation of a node's entropy in combination with a joint entropy and the mutual information [35].

Definition 2.2. Shannon-Entropy
The Shannon-Entropy is the probability of observing a particular symbol of event $p(x)$, within a given sequence.

$$H = -\sum p(x) log p(x) \tag{2.2}$$

Example 2.3. Here $p(x)$ (resp. $p(y)$) is the probability of observing a value $x \in \{0, 1\}$ (resp. $y \in \{0, 1\}$) for a node x (resp. y), where x and y can take two possible states 1 (on) or 0 (off). In a Boolean context Table 2.1 shows binarized time-course data with states for a node x and y.

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

Table 2.2.: Transition table B

$$H(x) = -p(0) * log[p(0)] - [1 - p(0)] * log[1 - p(0)] \tag{2.3}$$
$$H(x) = -0.4 log(0.4) - 0.6 log(0.6) \qquad = 0.97(40\%\ 0, 60\%\ 1) \tag{2.4}$$
$$H(y) = -0.5 log(0.5) - 0.5 log(0.5) \qquad = 1.00(50\%\ 0, 50\%\ 1) \tag{2.5}$$

H reaches its maximum when both possible states are equally probable $H_{max} = log(2) = 1$ (2.4). Beside the individual entropy of x and y now the combined entropy is consulted.

Definition 2.4. Joint Entropy
The joint entropy is defined by the probability of occurrences that x and y occur depended on each other.

$$H(x, y) = -\sum p(x, y) log p(x, y) \tag{2.6}$$

Example 2.5. Referring to Table 2.2: The co-occurrences of 1 and 0 in x and y are displayed in a quadratic matrix. The Joint Entropy $H(x, y)$ each combinatorial occurrence of x with y is summed up (2.6).

y			
	1	3	2
	0	1	4
		0	1
		x	

$$H(x,y) = -0.1log(0.1) - 0.4log(0.4) - 0.3log(0.3) - 0.2log(0.2) = 1.85 \tag{2.7}$$

In the last computational step the Mutual Information is calculated by the combination of Shannon-Entropy with Joint-Entropy.

Definition 2.6. Mutual Information
The mutual information describes the rate of transmission.

$$M(X,Y) = H(X) + H(X,Y) - H(X,Y) \tag{2.8}$$

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

$$M(X,[Y,Z]) = H(X) + H(Y,Z) - H(X,Y,Z) \tag{2.9}$$

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 2.7. With the knowledge about Shannon-Entropy, Joint-Entropy and the Mutual-Information the Boolean rules (resp. transition functions) for a node set $n = \{A, B, C\}$ can be calculated. In Table 2.3 all initial possible combinatorial occurrences 2^n of the node set n are represented in the left table. The right table shows the states of the nodes after one transition $(t+1)$ for the node set $\{A', B', C'\}$.

Transition table "B":

input	at	time (t)
A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

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

Table 2.3.: Left: Table of initial possible states for the variable set A,B,C. Right: Table of states after one transition step $(t+1)$ for the variable A',B',C'

For the initial states in Table 2.2 the Shannon-Entropy and the Joint-Entropy is calculated (Table 2.3).

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

→

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 2.5.

Table 2.4.

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 2.4. The iteration of REVEAL stops here and the Boolean rule (resp. transition function) can be inferred (Table 2.5).

input	output
B	A
0	0
1	1

→ $f_A = B$

Table 2.6.

For further details on the calculation of the transition function f_B and f_C the reader is referred to the paper.

REVEAL calculates simple network quickly and works incrementally by checking every possible combination of nodes and starting with a single node, then checking every pair and so on. Thus REVEAL is searching for the 'perfect' combination of nodes. Less computational expensive algorithms are Best-Fit and Full-Fit [3, 1].

Best-Fit

The second algorithm Best-Fit (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 occurrences 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 in-degree value, which describes the number of incoming edges to a node. Thus, a node can have maximally

an in-degree 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 (2.9), defines the set of true examples and F (2.10), the set of false examples extracted from the binarized time series data. The goal is to find a perfect Boolean classifier:

$$T = \{X'(t) \in \{0,1\}^n : X_i(t+1) = 1\} \tag{2.10}$$

$$F = \{X'(t) \in \{0,1\}^n : X_i(t+1) = 0\} \tag{2.11}$$

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 [57].

Full-Fit

This algorithm works almost the same as Best-Fit 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 [60].

2.6. Error Assessement

The application of an inference algorithm returns multiple solution, depending on the initial state of the nodes. Thus, the network fitting the best to the data should be selected. For this reason an error assessment strategy is provided with the help of a Boolean simulator so called BooleanNet [36].

The data set provides a set of binary trajectories $B = \{B_1, ..., B_n\}$ for which an inference 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 simultaneously updates all the states according to a synchronous simulation. Then the error of a Boolean network N with respect to B is defined by:

$$Error(N, B) = \frac{\sum_{1 \leq t \leq M}[(|B(t) - Y(t)|) * I_n]}{n * M} \tag{2.12}$$

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 data. For this reason the model with the lowest error is selected for further analysis.

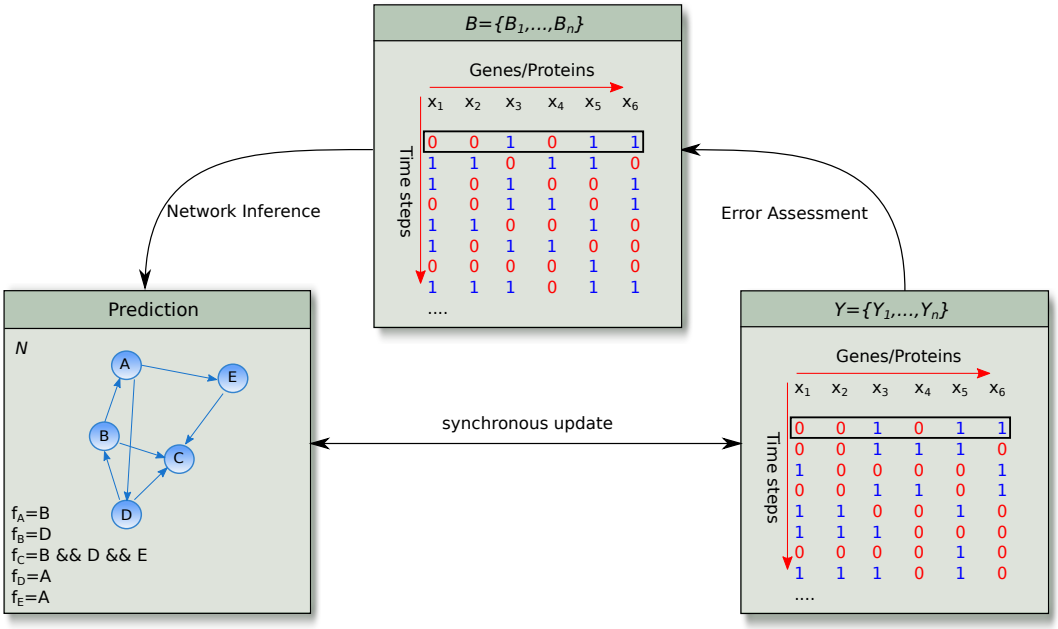

Figure 2.6.: Principle of error assessment. A network N is inferred from a set of binary trajectories B. *BooleanNet* synchronously updates the initial states (*black rectangle*) of B resulting in a set of binary trajectories Y. B and Y are compared

2.7. 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 a Boolean Network N is converted into its Interaction Graph IG. The edges of the predicted IG are compared to the edges of a selected gold standard network IG. The comparison is divided up into four possible classes displayed in the confusion matrix in Table 2.7.

True Positive *(TP)*: • Observed Value • Prediction Value • **Number of *TP***	**False Positive *(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 2.7.: Confusion matrix

In general, a true positive (TP) class the model correctly predicts the positive class and in a true negative (TN) class the model correctly predicts the negative class and in the false positive (FP) class the model incorrectly predicts the positive class and in the false negative (FN) class the model incorrectly predicts the negative class.

Referring to a Boolean network N, TP and FP denote the numbers of correctly and incorrectly predicted connections, respectively. And FN denotes the number of non-inferred connections of a gold standard N' in the prediction N, while TN denote the number of correctly non-inferred connections [3].

In the following different scoring metrics are defined used for later structural network analysis. To get things straight concerning application and interpretation of the metrics an example (**Example 2.13**) is presented at the end of this section.

Definition 2.8. Precision
Precision is the proportion of correctly inferred connections out of all predictions:

$$Precision = \frac{TP}{TP + FP} \tag{2.13}$$

[3]

Definition 2.9. Recall

Recall (Sensitivity) is the proportion of inferred connections among the true connections in the gold standard N.

$$Recall = \frac{TP}{TP + FN} \tag{2.14}$$

[3]

Definition 2.10. Accuracy

Accuracy is the percentage of correct predictions.

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

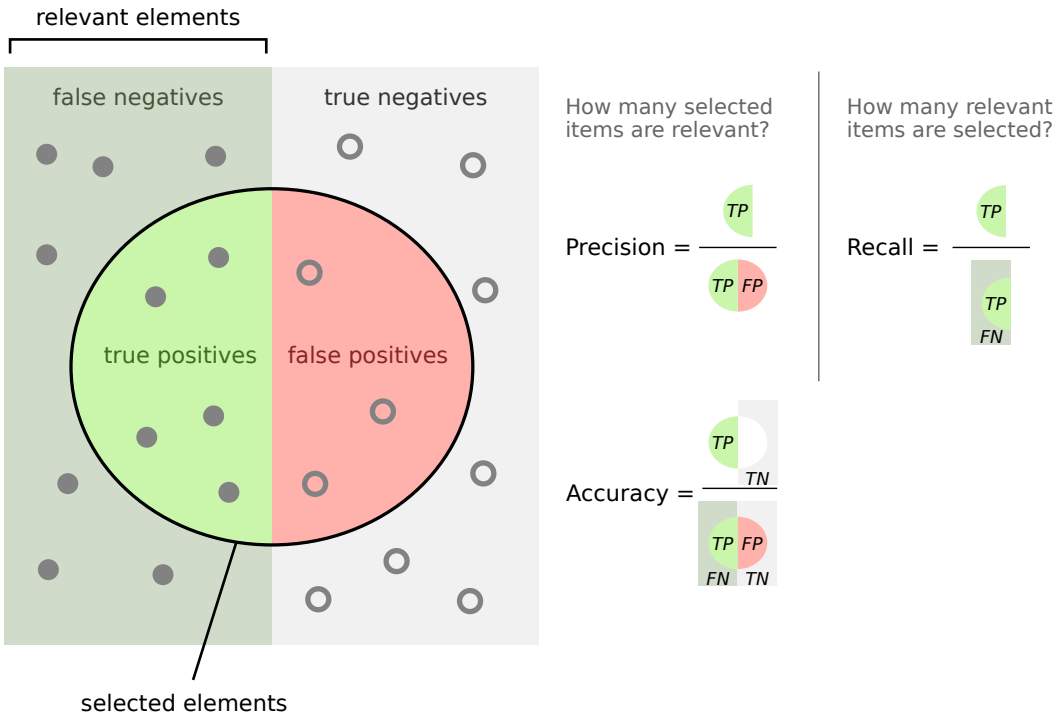

Figure 2.7.: Precision,Recall and Accuracy. Left: positive elements (*gray filled circles*) TP and FN; Right: negative elements (*empty circles*) FP and TN. Big circle: Elements predicted by the model.

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 balanced accuracy and the Matthew correlation coefficient are introduced.

Definition 2.11. Balanced Accurancy (BACC)
Balanced Accuracy (BACC) is the Fraction of predictions our model got right divided by 2.

$$BACC = \frac{\frac{\text{Number of correct predictions}}{\text{Total number of predictions}}}{2} = \frac{\frac{TP}{TP+FN} + \frac{TN}{FP+TN}}{2} \tag{2.16}$$

[61]

Definition 2.12. Matthew Correlation Coefficient (MCC)
The MCC is a correlation coefficient between the observed and the predicted of binary classifications which returns a value between -1 *and* 1*;* $MCC \in [-1, 1]$*. A value close to* 1 *indicates a perfect prediction, a value close to* 0 *means that the prediction is not better than an average random one and a value close to* -1 *describes a total disagreement between prediction and observation (inverse prediction).*

$$\text{MCC}= \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{2.17}$$

[62]

Example 2.13. Given is a model that classified 100 tumors as malignant (the positive class) or benign (the negative class): [63]

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 2.8.: Confusion matrix displaying all four possible outcomes.

The confusion matrix in Table 2.3 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:

$$Accuracy = \frac{1+90}{1+90+1+8} = 0.91 \tag{2.18}$$

$$Precision = \frac{1}{1+1} = 0.5 \tag{2.19}$$

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

$$TPR = \frac{1}{1+8} = 0.11 \tag{2.21}$$

$$FPR = \frac{1}{1+90} = 0.01 \tag{2.22}$$

$$BACC = \frac{\frac{1+90}{1+90+1+8}}{2} = 0.46 \tag{2.23}$$

$$MCC = \frac{1*90-1*8}{\sqrt{(1+1)(1+8)(90+1)(90+8)}} = 0.21 \tag{2.24}$$

The *Accurary* (2.18) 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. Data imbalance said to exist when all classes of the confusion matrix are not equally proportioned [64]. For example, a disease data set in which 0.0001 of the examples have positive 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 inequality 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 $BACC$ (2.23) has a value of 0.46, which means that the prediction of the model is not that good as the *Accurary* (2.18) shows and the MCC (2.24) 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 (2.19), 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. Pipeline and Results

This chapter introduces a pipeline of the *in silico* data set (Figure 3.3) and a pipeline for the DREAM8 Challenge data set (Figure 3.8) describing the processing of the data from discretization to inferring a network and finally scoring the predicted network against a gold standard network. The results of the *in silico* data set are necessary to set the parameter for the DREAM8 Challenge pipeline. Both pipelines can be executed from the command line by a bash script and are available on Git: "github.com/ninakersten/Masterthesis".

3.1. Pipeline of the *in silico* data set

For both the sub-networks of *E.coli* and the cell cycle network each set F of transition function is executed to sets of continuous data ($S = \{S_1, S_2, ..., S_n\}$), which are generated with *odefy*, a MATLAB- and Octave-compatible toolbox for the automated transformation of Boolean models into systems of ordinary differential equations (Figure 3.3, Figure 3.1). With *odefy* the number of sample points and the time interval for a data simulation can be determined. The time interval is set to a range of 1 to 50. The *in silico* data sets are converted into the *csv* format of the structure of the DREAM8 Challenge input data (Table 2.1) and for the discretization and learning step converted into a *text* file format (Figure 3.1). Names of the species are anonymized by single characters depicted in the first header of a *txt* file and original names are stored in a header below followed by the time course data set S. Information about cell line, inhibitor and stimulus are neglected (Figure 3.1).

CSV data structure

			Slide ID	GeneX
			Antibody Name	GeneX
			HUGO ID	GeneX
Cell Line	**Inhibitor**	**Stimulus**	**Timepoint**	
Cell Line	no	no	0	0.547689876756 ...
Cell Line	no	no	1	0.647689876754 ...
Cell Line	no	no	2	0.747689876754 ...
Cell Line	no	no	...	...
...	...	...	... 50	...

Text data structure

NON $ % + ...				
#NON	GeneX	GeneY	GeneZ	...
0	0.6279850	3.5199156	0.8539050	
1	0.7181498	4.0246494	0.8374815	...
2	0.8288568	3.1841118	0.8957388	...
3	0.9202167	3.6352309	1.1001242	...
...	...	...	...	...
... 50	...	...	...	...
...	...	...	...	...

Figure 3.1.: *CSV* **to** *TXT*: Anonymized single character represent the species and information about cell line, inhibitor and stimulus is neglected.

After discretizing continuous time course data into a set of binary values ($B = bin(S)$, where $bin \in \{2\text{-}k\text{-}means$, iterative $k\text{-}means\}$) redundant values are removed from this set. Boolean networks ($N = learn(B)$) are learned from this data by each inference algorithm ($learn \in \{$Best-Fit, Full-Fit, Reveal$\}$). A value for the minimal error $MinError$ is set, such that an inference algorithm runs i times (here: $i = 5000$) until a network with an error ($Error(N, B)$) lower than the minimal error is achieved. It is worthwhile to get an error of 0, meaning the Boolean model describes the data perfectly. The amount of returned solutions is set to a value of 3, such that in each inference process three solutions (resp. Boolean Networks) are inferred, all with an error lower than the minimal error. A single Boolean Network with the lowest error across all iterations is selected for further structural evaluation.

Inference settings

For investigating the impact of the *in-degree* in a network on the algorithms performance, sub-networks of *E.coli* are processed by the iterative *k-means* binarization algorithm with a cluster depth of $d = 3$ in combination with Best-Fit, Full-Fit or REVEAL. The cluster depth with $d = 3$ is selected due to previous research proving its reliability regarding the trade-off between simplicity and loss of information (e.g. oscillations).
[1] For assessing the dependence of an inference algorithm to the number of sample points, continuous data for the cell cycle is generated for $m \in \{50, 100, 150, 200, 250, 300, 350, 400$,450, $500\}$. The whole set of inference algorithms only runs starting by 50 sample points. Complex systems, like a cell cycle are oscillating, thus a large number of sample points are needed to detect 'real' steady-states and achieve a sufficient Boolean network.

For measuring the impact of the clustering depth the cell cycle's continous data is generated with 100 sample points (similar to the abundance of sample points of ~ 85 in the DREAM8 Challenge) and inferred with a clustering depth of $d \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, where $d = 1$ denote the two clusters *k-means* binarization algorithm and from $d = 2$ denote the iterative *k-means* binarization algorithm. Table 3.1 shows a summarized overview of the settings for the *in silico* pipeline.

Settings: Pipeline	*in-degree*	sample points	cluster depth (k)
# sample points	100	[50 : 500]	100
time interval	[1 : 50]	[1 : 50]	[1:50]
bin-method	$k = 3$	$k = 3$	$k \in [1, 10]$
REVEAL	✓	✓	✓
Best-Fit	✓	✓	✓
Full-Fit	✓	✓	✓
$MinError$ (ϵ)	0.6	0.6	0.6
max. iteration (i)	5000	5000	5000
solutions	3	3	3
Network properties			
network source	*E.coli*	Cell cycle	Cell cycle
# networks	45	10	10
max. *in-degree*	$d \in [9 : 14]$	6	6
# nodes	$n \in [10 : 14]$	10	10
# edges	[10 : 126]	35	35

Table 3.1.: Settings: Pipeline *in silico* and network properties

Prediction processing

The predicted Boolean networks are converted (from *.bnet*-format) into Interaction graphs (to a *.sif*-format) by *PyBoolNet* (Figure 3.2). Each interaction graph of a predicted network is scored against its gold standard interaction graph generated from the initial Boolean network with *PyBoolNet*.

Hence, each line in a *.sif*-file of an interaction graph represents an edge in a Boolean network (Figure 3.2). The edges of the gold standard and the prediction are compared resulting in a confusion matrix for computing precision, recall, accuracy, balanced accuracy and the Matthew correlation coefficient (Table 2.7).

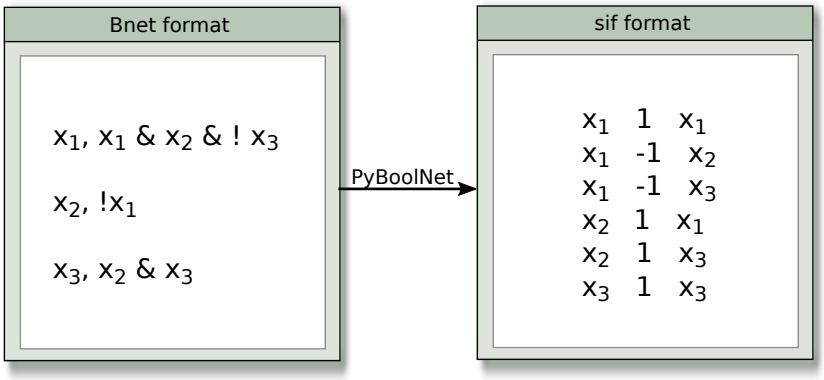

Figure 3.2.: Boolean Network to Interaction Graph: The predicted Boolean network N (*.bnet*-format) is converted into an interaction graph IG (*.sif*-format).

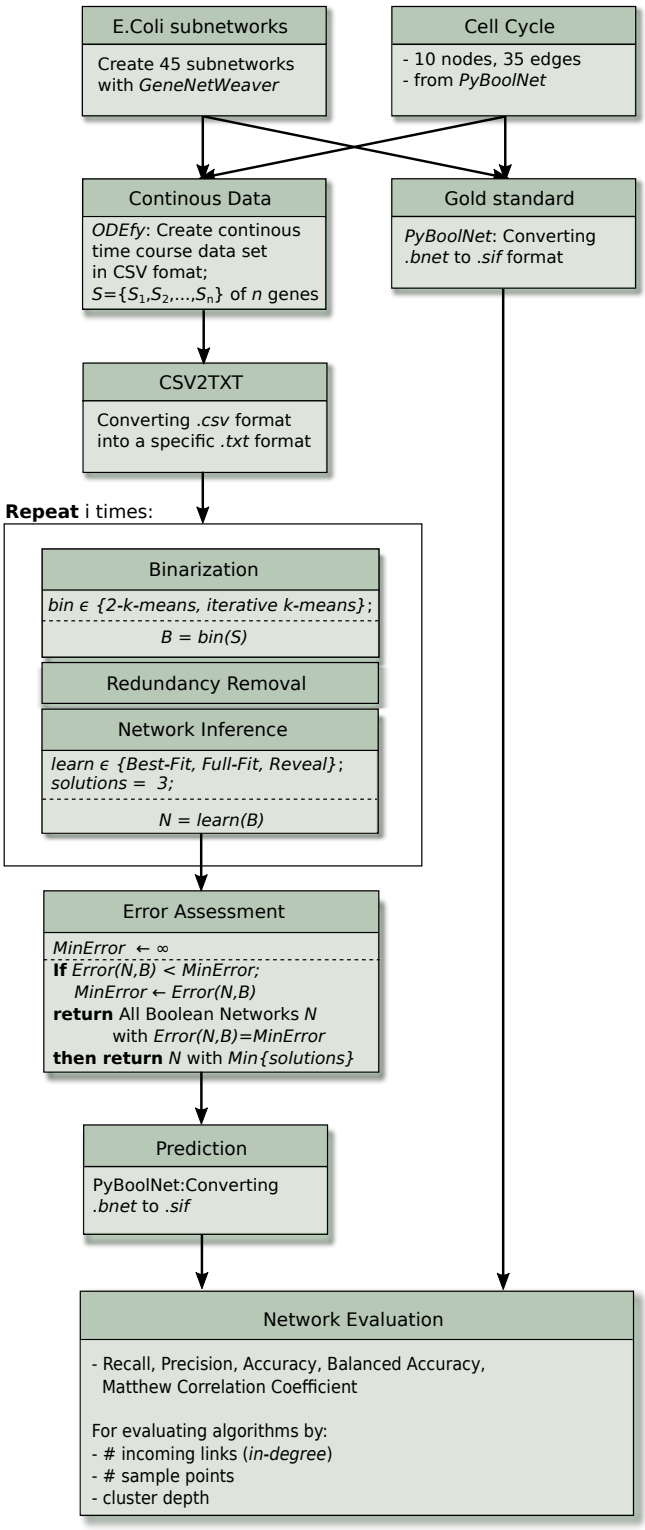

Figure 3.3.: Pipeline *in silico*. This pipeline shows the final setup for assessing the algorithms performance regarding the number of incoming links of nodes in a network, the number of sample points and the cluster depth for binarization.

3.2. Results and Discussion of the *in silico* data set

In-degree

Figure 3.4 shows 45 sub-networks of *E.coli* grouped into nine categories, each containing five sub-networks with n nodes; $n \in |V|$, where $|V| = \{10, 11, 12, 13, 14\}$. Each category on the x-axis depicts the *in-degree* $k_i^{in} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$) of a set of nodes in a network. Thus, the inference algorithms Best-Fit, Full-Fit and REVEAL predict five networks in each category where the mean accuracy value is obtained by scoring each predicted interaction graph against a corresponding gold standard interaction graph, derived from the initial network N' (Figure 3.3).

Starting with an *in-degree* of $k_1^{in} = 1$ the mean accuracy values of Best-Fit, Full-Fit and REVEAL are ~ 0.854, ~ 0.844 and ~ 0.849. With increasing *in-degree* the mean accuracy value decreases for all three inference algorithms. Thus, with an *in-degree* of $k_9^{in} = 9$ the mean accuracy value for Best-Fit, Full-Fit and REVEAL are ~ 0.399, ~ 0.407 and ~ 0.399. None of the three inference algorithms show an outstanding significantly higher or lower mean accuracy. Hence, on average about $\sim 85,2\%$ of relevant links are predicted correctly by all three inference algorithms when the *in-degree* is $k_1^{in} = 1$ and an average of about $\sim 40,2\%$ of relevant links are predicted correctly by an *in-degree* of $k_9^{in} = 9$.

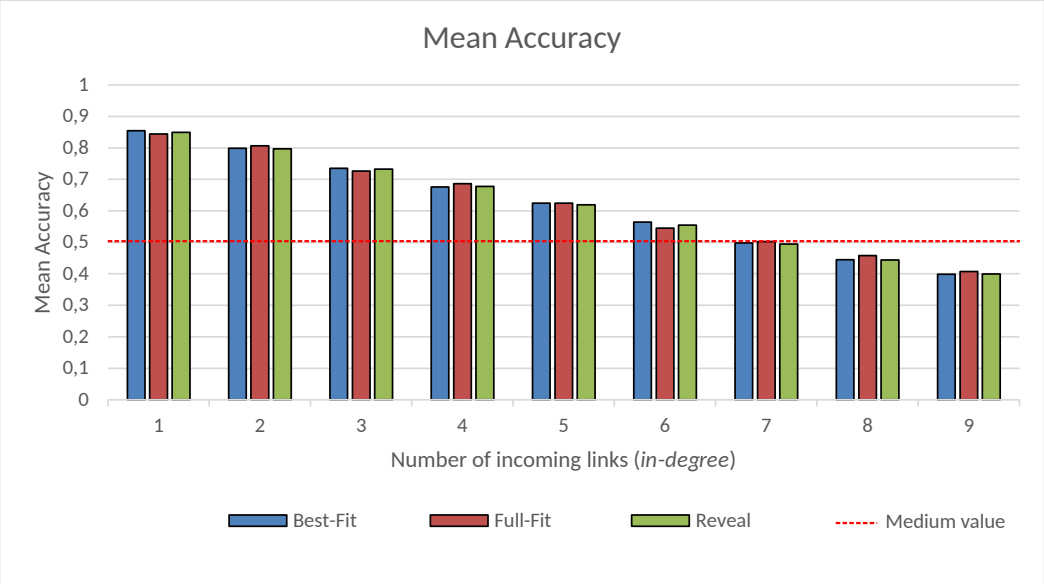

Figure 3.4.: *In-degree*:Mean Accuracy. Sub-networks of *E.coli* are grouped into nine groups definded by the *in-degree* of nodes in each sub-network. With increasing *in-degree* the mean accuracy of all three inference algorithms decreases.

Thus the more nodes occur with a high *in-degree* the more complex is the system and the worse the inference algorithm is able to detect the whole information. This observation captures

the observation of Shohang Barman and Yung-Keun Kwon [3] who state that the number of incoming links represent the degree of complexity of the inference problem. They introduced a novel mutual information-based Boolean network inference (MIBNI) method and tested its structural and dynamic performance. Therefore, about ~ 300 sub-networks of *E.coli* of different network sizes ($|V| = 10, 20, ..., 100$) were randomly generated by *GeneNetWeaver* and nodes of these networks were grouped by their *in-degree*. Their new method yielded a slightly better performance in contrast to Best-Fit and REVEAL. Nevertheless, with incresing number of incoming links the mean accuracy decreases for all measured inference algorithms assessed in this article, too.

Number of sample points

Continuous data sets of the mammilian cell cycle network are generated with different number of sample points $m \in \{50, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$ and the predicted interaction graph is scored against the interaction graph of a gold standard cell cycle interaction graph derived from the initial network (Figure 3.3).
In Figure 3.5 (a) recall values of Best-Fit, Full-Fit and REVEAL are ranging between 0 (REVEAL) and ~ 0.231 (Best-Fit). The values fluctuate across the number of sample points for each of the three inference algorithms and regarding the overall mean recall, about $\sim 9.49\%$ of relevant links (TP and FN) are predicted.
Taking the precison in Figure 3.5 (b) of Best-Fit, Full-Fit and REVEAL into account the values fluctuate among the increasing number of sample points, too. Here, values range from 0 (REVEAL) to ~ 0.428 (Best-Fit). The mean precision value of Best-Fit is ~ 0.236 and for Full-Fit is ~ 0.252 and for REVEAL is ~ 0.219.
Thus, averaging across Best-Fit, Full-Fit and REVEAL about $\sim 23,6\%$ of the predicted links are predicted correctly.

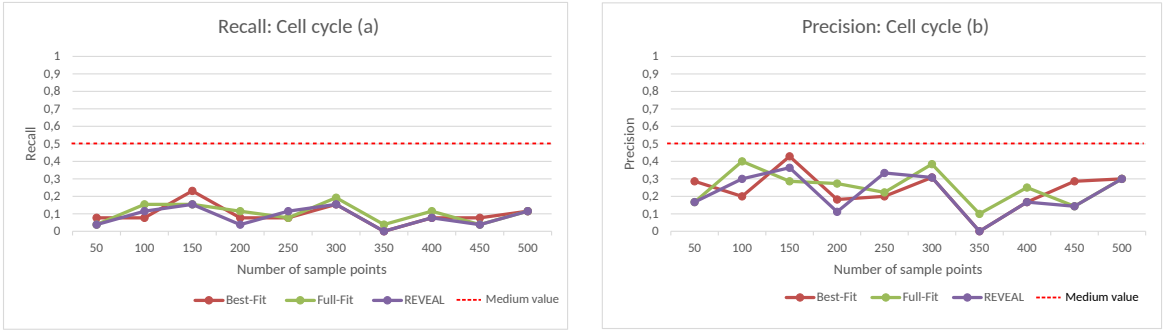

Figure 3.5.: Number of sample points. (a) Recall for Best-Fit, Full-Fit and REVEAL; (b) Precision for Best-Fit, Full-Fit, REVEAL

Fluctuations across the increasing number of sample points in Figure 3.5 may be a result of the oscillating property of the cell cycle system, which is not captured in each network. The number of false negative predicted links is tremendeously high in constrast to the number of true positive predicted links for all inference algorithms, which explaines the recall. While precision is taking the true positive and false positive predicted links into account, which are not that highly class im-balanced like the classes for the recall are.

For this reason the balanced accuracy is considered (Figure 3.6). Here, the class imbalance is relativized, such that all three inference algorithms range around a value of ~ 0.5 among the increasing number of sample points. This means, each inference algorithm predicts the links in a network rather randomly.

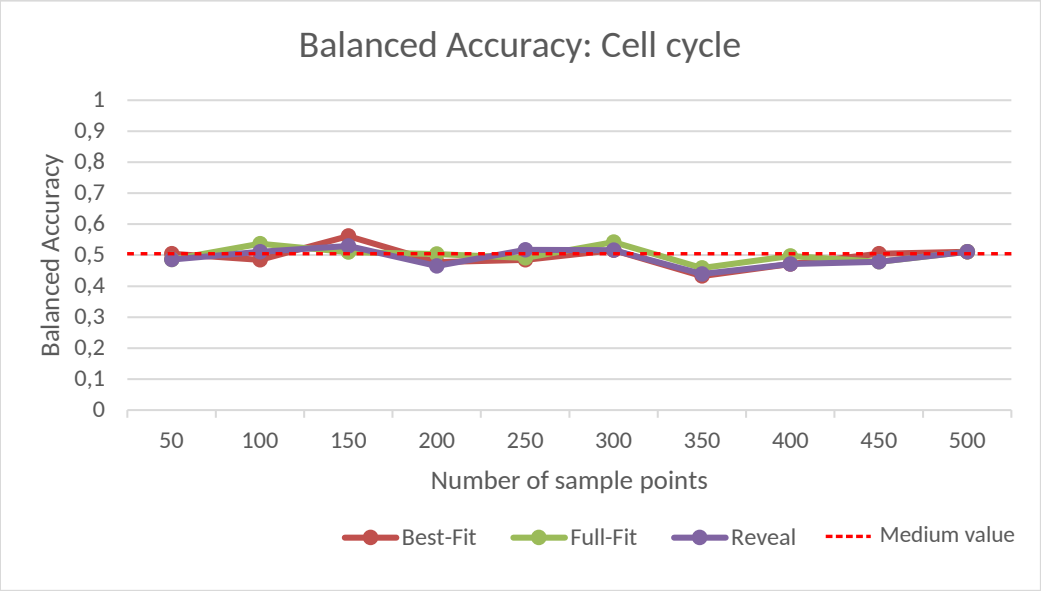

Figure 3.6.: Balanced Accuracy: Number of sample points.

Regarding recall, precision and balanced accuracy in this setting context it is observed that the number of sample points has no significant impact on the inference algorithms' performance. The cell cycle is with 35 links a highly interconnected network and due to its' oscillating property quite challenging to infer. Thus, taking prior knowledge from literature into account it is important to mention, that this observation is not true for less complex networks of smaller size [1].

Cluster depth

In Figure 3.7 the cluster depth d is increased from $d = 1$, which represents the two clusters *k-means* binarization algorithm and from $d = 2$ to $d = 10$ describing the cluster depth of the iterative *k-means* binarization algorithm. In (a) the Matthew correlation coefficient starts with the highest value of ~ 0.258 for REVEAL for $d = 1$ followed by Full-Fit with a value of ~ 0.106 and Best-Fit with ~ 0.106. Proceeding with $d = 3$ Best-Fit stands out with a value of ~ 0.258 whereas Full-Fit and REVEAL range around a value of 0. With increasing cluster depth the performance of all three inference algorithms approximates around a mean value of ~ 0.048.
In comparison with balanced accuracy in (c) when $d = 3$ Best-Fit perfoms the best with a value of 0.588 as well. Increasing the cluster depth yields an approximation of all inference algorithms around a mean value of ~ 0.516.
In (b) Best-Fit yields the best performance with a precision value of 0.6 and $d = 3$. The value highly fluctuates until $d = 5$ and approximates from $d = 6$ to a precision value of ~ 0.287. Regarding the recall in (d) none of the inference methods is significantly outstanding. Taking the mean recall approximatly $\sim 12,3\%$ of relevant links are predicted by the methods. While $\sim 28,7\%$ of predicted links are correctly predicted.

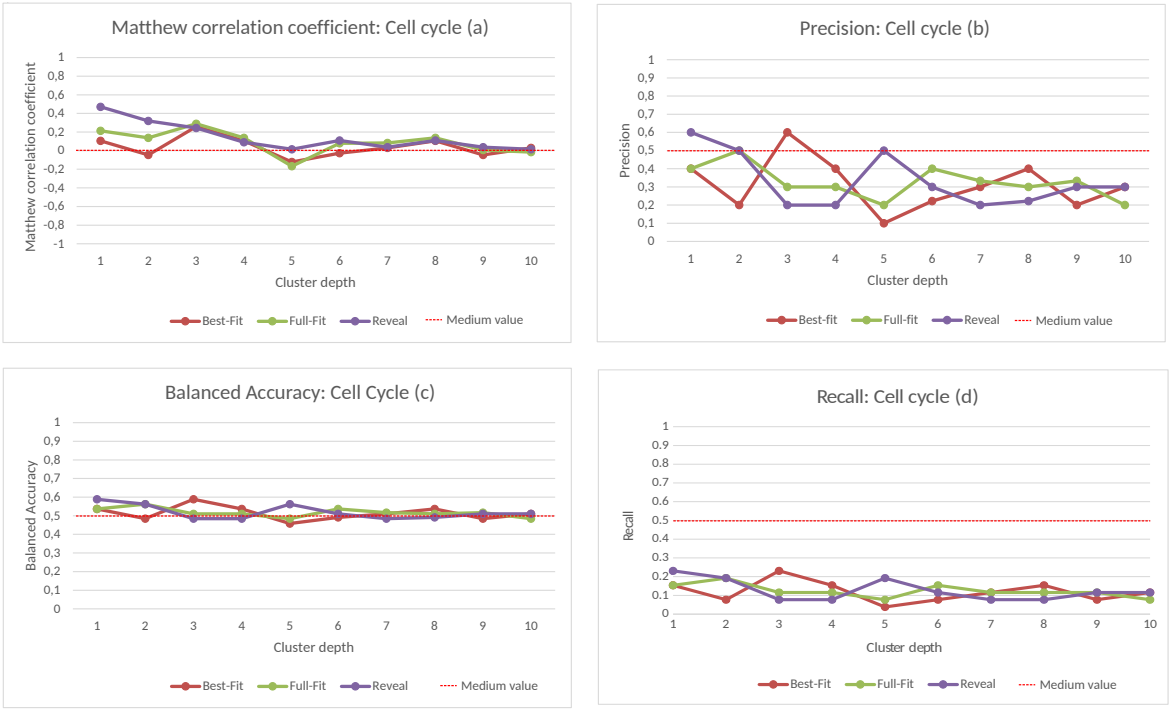

Figure 3.7.: Performance considering cluster depth d

Results of the performance measurement in dependence on the number of sample points are similar to the results of performance measurement in dependence on the cluster depth

regarding fluctuations in precision, Best-Fit is performing the best across precision and recall and balanced accuracy and Matthew-correlation-coefficient show no outstanding performance of a method. Futhermore, regarding the runtime performance is REVEAL computuationally expensive (Figure 2.2) and therefore not an appropriate method for big real-life data network inference. Taking prior knowledge from literature into account then Best-Fit in combination with the iterative *k-means* binarization algorithm captures the structural and dynamic complexity in a system the best [1]. For these reasons, this combination is applied to the real-life data set of the DREAM8-Challenge described in the next section.

3.3. Pipeline of the DREAM8 Challenge data set

In this section it is shown in which way the *in silico* pipeline is validated, such that the DREAM8-Challenge data set can be processed, resulting in a DREAM8-Challenge pipeline depicted in Figure 3.8.
This pipeline starts with the 'main' training data set of the DREAM8-Challenge, which is converted into a *txt* file format (Table 2.1, Figure 3.1) by splitting each data set of a cell line into eight text files depending on each stimulus. Each of the resulting 32 *text*-files contain about ~ 85 sample points and about ~ 45 antibody names.

Setting selection

The higher the *in-degree* of each node (Figue 3.4) and the abundance of nodes in a network (Figure 2.2) the higher is the complexity in a system and hence, the higher is the computational cost of an inference algorithm [3]. The aim of the *in silico* pipeline is to investigate which parameter has a major influence on the algorithms performance, such that the best performing algorithm can be selected for the DREAM8-Challenge data set.
As mentioned in 2.1 technical requirements limit (8 RAM and a Core i5 processor) the application of REVEAL to the DREAM8-Challenge data set resulting an infeasible task (Figure 2.2). This observation is confirmed by Shohang Barman and Yung-Keun Kwon [3] who state that mutual information based approaches like REVEAL are computationally expensive, because they are implemented to compute the exact mututal information values over all possible combination of nodes. Therefore, REVEAL is not considered in this pipeline.

Furthermore, results of the *in silico* pipeline regarding the choice of abundance of sample points (Figure 3.6) and cluster depth (Figure 3.7 (c)) show no significant influence on the algorithms' performance. An increasing *in-degree* (Figure 3.7) shows an decreasing performance for all three algorithms, such that no algorithm can be excluded by this investigation either.

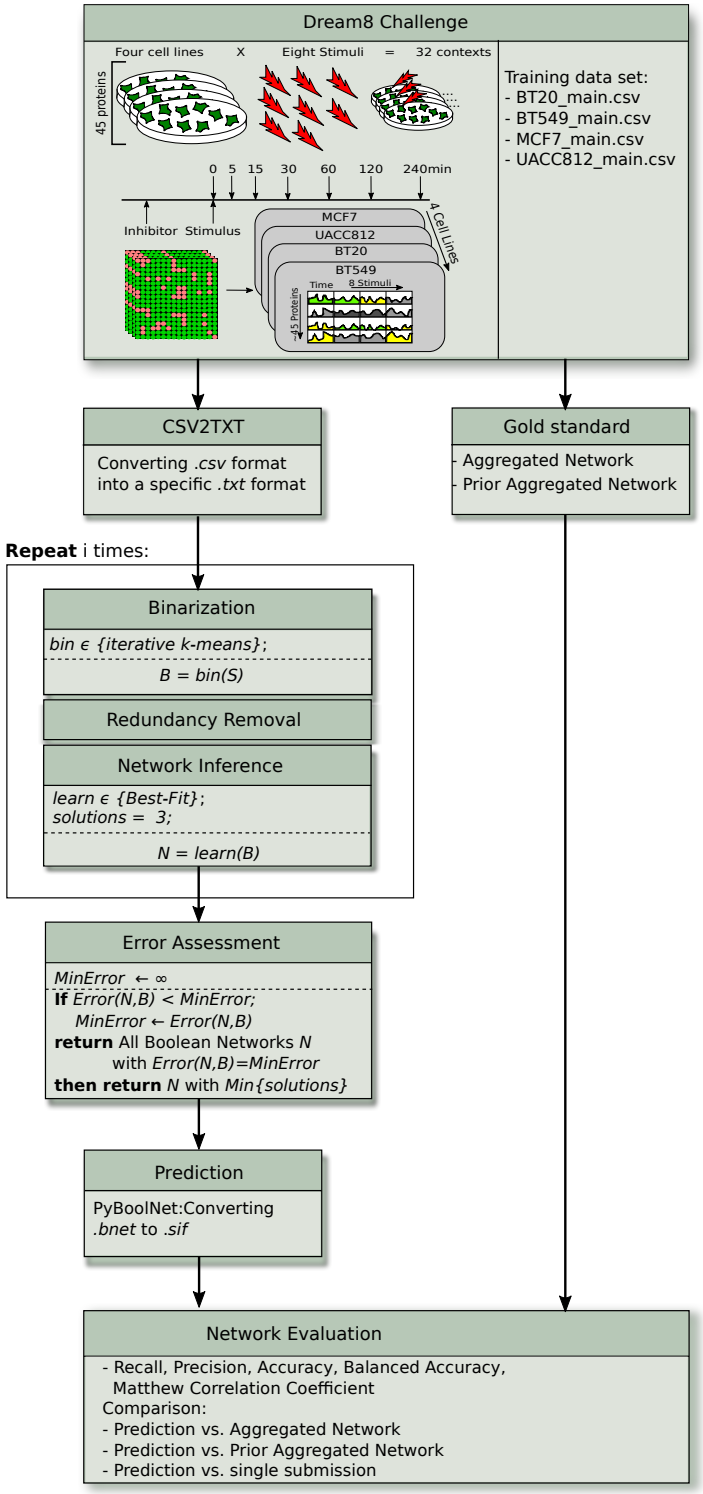

Figure 3.8.: Pipeline DREAM8-Challenge. This pipeline shows the final setup for the DREAM8-Challenge input data. Settings are selected by previous investigation of the *in silico* data set and prior knowledge from literature.

Thus, results regarding error assessment of Best-Fit, Full-Fit and REVEAL in combination with two cluster *k-means* binarization algorithm and iterative *k-means* binarization algorithm of Natalie Berestovsky and Luay Nakhleh are taken into account. In this paper based on minimal error, convergence and uniqueness the recommended combination of Best-Fit with the iterative *k-means* algorithm (with $d = 3$) for complex systems is applied to the DREAM8 Challenge data set.

As in the *in silico* pipeline the minimal error is set to a value of ????? and maximally runs 5000 times by returning 3 networks and selecting the solution with the lowest error. Predicted Boolean networks (in *bnet*-format (Figure 3.2)) are converted into an interaction graph (in *sif*-format (Figure 3.2)) and scored against two aggregated networks each representing an aggregated network and an aggregated prior network each representing a gold standard.

For simplification of notation the prediction is defined by the set of networks predicted by the DREAM8-Challenge pipeline with the inference algorithm Best-Fit and the iterative *k-means* binarization algorithm with $d = 3$.

Gold standard and Evaluation selection

Since gold standard networks are often based on prior knowledge from literature, learning novel connections in a network, such that specific biological systems can be truly mimicked is restricted [45]. Therefore, the DREAM8-Challenge did not provide a gold standard to the participants. Thus, a provided test data set was used as an abstracted representation of a gold standard to asses the algorithms' performance resulting in prior knowledge independent networks.

Due to varying strategies of evaluating predictions, e.g. choice of the scoring metric and implementation strategy, it is quite challenging to compare the resulting performances between the participants reliably. For this reason the DREAM8-Challenge provides a standard scoring tool: 'DREAMTools' python package [65].
This tool needs as input a *sif*-file and an *eda*-file, while an *eda* (electronic design automation) contains information of edge occurrences in a network like in a *sif*-file and additionally a confidence score for each edge. A confidence score (resp. edge weight) describes the probabilty of an edge being existent or not [25].

DREAMTools compares the confidence scores of a predicted network against confidence scores of a gold standard network returning values of the Area Under the Receiver Operating Characteristic curve (AUROC) and the Area Under the Precision Recall Curve (AUPR) [65] [45].

Both evaluation metrics are commonly used when there is an imbalance between the number of positives and negatives in the gold standard [67]. For further details regarding definition and application of these metrics the reader is referred to section A.1.'DREAM8-Challenge scoring priciples'. In subchallenge SC1A of the DREAM8-Challenge each participating group was ranked by their scoring results, revealing which inference performs the best [66].

In contrast to the DREAM8-Challenge this work makes use of an aggregated network and an aggregated prior network each representing a gold standard. These networks were submitted after finishing the challenge in 2016. The aggregated network is a compendium of 66 submissions of the participants with the best performance reduced by correlated submissions [45]. Thus, this network comprises prior knowledge networks with prior knowledge independent networks. The aggregated prior network is a combination of 10 prior knowledge networks that participants used as part of their submission [45]. Table 3.2 gives an overview of the properties of the aggreagted network, aggregated prior network and the prediction reagarding the number of nodes and links in each network and the number of networks each network is composed are shown.

Network	# nodes	# edges	# networks
Prediction (Best-Fit)	~ 45	~ 140	1
Aggregated Network	~ 45	~ 2200	66
Aggregated Prior Network	~ 45	~ 1400	10

Table 3.2.: Network properties

Predictions of all 74 participating groupes of the DREAM8-Challenge for the SC1A challenge and the prediction generated by the DREAM8-Challenge pipeline are scored against the aggregated network and against the aggregated prior network. This results in a new ranking including the pipelines' prediction, revealing how a Boolean approach is performing in relation to the participants' submission.

Instead of taking AUPR and AUROC, balanced accuracy, Matthew correlation coefficient, precision and recall are implemented in this pipeline.

3.4. Results of the DREAM8-Challenge data set

In this section first it is emphasized why it is recommanded to use balance accuracy instead of accuracy for an imbalanced set. Then the prediction and all submissions of the DREAM8-Challenge are ranked by scoring these networks against the aggregated network and aggregated prior network.

Prediction versus Aggregated Network and Aggregated Prior Network

In Figure 3.9 the prediction is scored against the aggregated network and the aggregated prior network among the stimuli for each cell line opposing balanced accuracy to accuracy. The mean accuracy of scoring the prediction against the aggregated network yields a value of ~ 0.126 and for scoring the prediction against the aggregated prior network yields a value of ~ 0.386. The mean balanced accuracy value for scoring the prediction against the aggregated network is ~ 0.505 and for scoring the prediction against the aggregated prior network is ~ 0.495. Regarding the major distance between accuracy and balanced accuracy it is observed that the data is class imbalanced. Especially for the case of the scoring the prediction against the aggregated prior network.

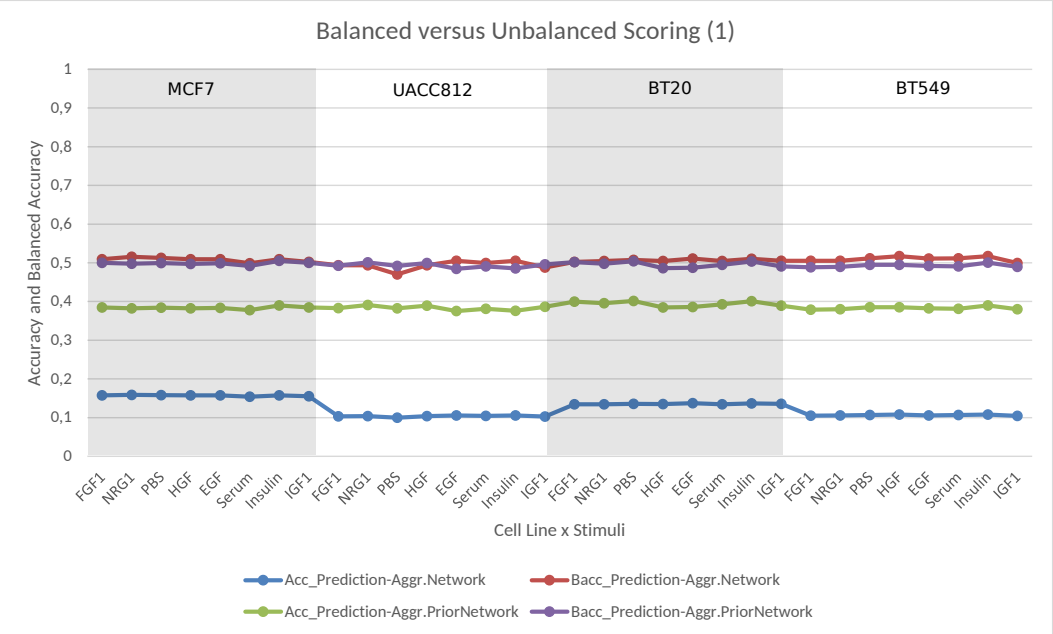

Figure 3.9.: Imbalanced classes (1). Accuracy and Balanced Accuracy for the prediction scored against the aggregated network and aggregated prior network.

This observation is emphasized by considering the size of each class in the confusion matrix for both evaluation measurements. Comparing the prediction with the aggregated network predominantly many false negatives are detected, thus the accuracy is worse. Whereas com-

paring the prediction with the aggregated prior network the number of false negatives is much lower and the number of true negatives is much higher such that the accuracy yields better result.

For emphasizing this observation the prediction is scored in Figure 3.10 instead against the aggregated prior network against a submission of the last participant (rank: 74., $\sim$ 600 edges among the 32 contexts) of the DREAM8-Challenge leader board [66]. Here, scoring the prediction against the last participant shows that the disparity of the size between these networks is not that big as between the prediction and the aggregated network. Thus scoring the prediction against the last participant results in a predominant abundance of true negatives thus the mean accuracy of $\sim$ 0.716 is much better than scoring the prediction against the aggregated network.

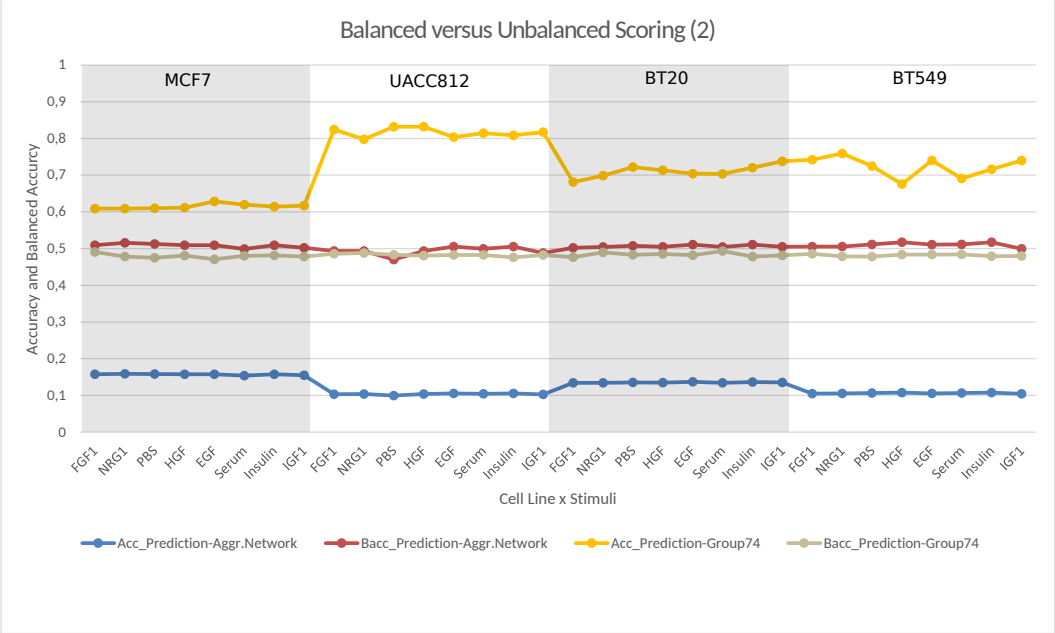

Figure 3.10.: Imbalanced classes (2). Accuracy and Balanced Accuracy for the prediction scored against the aggregated network and aggregated prior network.

The impact of true negatives and false negatives can be put into perspective by applying the balanced accuracy metric. Therefore accuracy is neglected and further evaluation are perfomed with precision, recall, balanced accuracy and matthew correlation coefficient.

In Figure 3.11 the recall (a) and the precision (b) values are shown. It is important to indicate that the recall is presented in a higher resolution such that each evaluation is easier to distinguish.

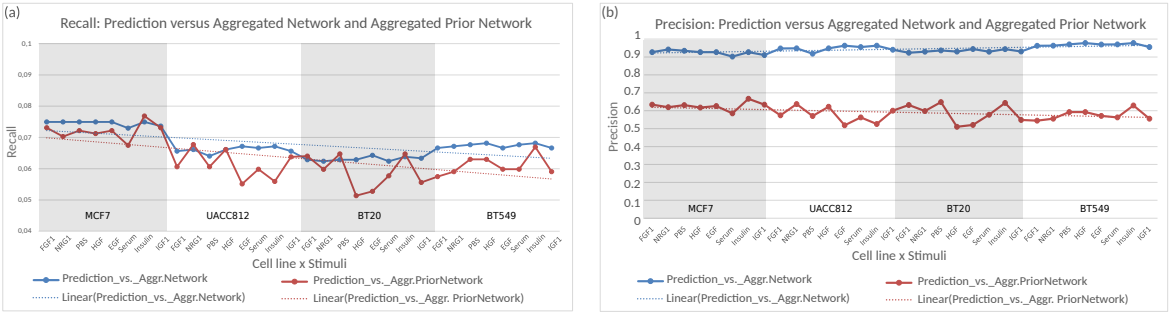

Figure 3.11.: Recall: Prediction versus Aggregated/Prior Network

In (a) the prediction is scored against the aggregated network yielding a mean recall value of $\sim 6,7\%$ and scored against the aggregated prior network resulting in a mean recall value of $\sim 6,3\%$. Beside the fact that scoring against the aggregated prior network causes a few fluctuations among cell lines and stimuli the overall mean recall value is not significantly different to the mean recall value of scoring the prediction against the aggregated network. All in all, the prediction predicted $\sim 6,5\%$ of the releveant elements.
In (b) the prediction scored against the aggregated network yields a mean precision value of ~ 0.943, meaning about $\sim 94,3\%$ edges are predicted correctly. Regarding the prediction beeing scored against the aggregated prior network regarding the mean precision value about $\sim 59,1\%$ edges are predicted correctly.

New Ranking: Aggregated Network and Aggregated Prior Network

A total of 74 submissions of participants of the DREAM8-Challenge in combination with the predicted network of the DREAM8-Challenge pipeline are scored against the aggregated network and the aggregated prior network. The networks are ranked by their mean value of balanced accuracy, Matthew correlation coefficient (Mcc), precision and recall such that a ranking of the prediction is achieved.
In Figure 3.12-3.14 names of the participating groups of the DREAM8-Challenge are pseudonymized by their original ranking of the SC1A leaderboard. For this reason the prediction of the DREAM8-Challenge is pseudonynomized by a value of 0, which is highlighted in each figure by a red rectangle. In Table 3.3. the ranking of the prediction for each evalutaion method and its corresonding value is shown.

Scoring metric	Type	Aggregated Network	Aggregated Prior Network
Mean Balanced Accuracy	rank	43	49
	value	~ 0.0628	~ 0.4950
Mean Mcc	rank	33	39
	value	~ 0.0097	~ 0.0195
Mean Precision	rank	33	45
	value	~ 0.9436	~ 0.5909
Mean Recall	rank	42	47
	value	~ 0.0677	~ 0.0632

Table 3.3.: Ranking of the prediction

In Figure 3.12 mean balanced accuracy values are descend ordered by the values of all submissions and the prediction scored against the aggregated prior network. In relation to the DREAM8-Challenge submissions the prediction has a rank of 49 with a value of ~ 0.495. Mean balanced accuracy values ordered by the aggregated network yields a rank of 43 for the prediction with a value of ~ 0.063.
Regarding the mean Matthew-correlation-coefficient in Figure 3.13 the networks are descend ordered by the results of scoring against the aggregated network. For this case the prediction yields a new rank of 33 with a value of ~ 0.0097 and values ordered by scoring against the aggregated prior network yield a rank for the prediction of 39 and a corresponding value of ~ 0.0195.
Regarding the mean recall ranking in Figure 3.13 the ranking is ordered by the scoring against the aggregated prior network, and the prediction predicts $\sim 6,8\%$ of relevant edges of the aggregated network with a rank of 42 and predicts $\sim 6,3\%$ of relevant edges of the aggregated prior network with a rank of 47.
Furthermore, the precison shows that regarding the aggregated network the prediction predicts $\sim 94.36\%$ of the relevant edges correctly with a rank of 33 and regarding the aggregated prior network the prediction predicted $\sim 59,1\%$ of the relevant edges correctly with a rank of 45.

Averaging across the whole evaluation set for scoring against the aggregated network the prediction yields a rank of 49 and for scoring against the aggregated prior network the prediction yields a rank of 50. It is noticed that averging among the whole evaluation set captures the ranking of the best performing group in the DREAM8-Challenge. This control emphasizes the reliability of evaluation methods used in this thesis.

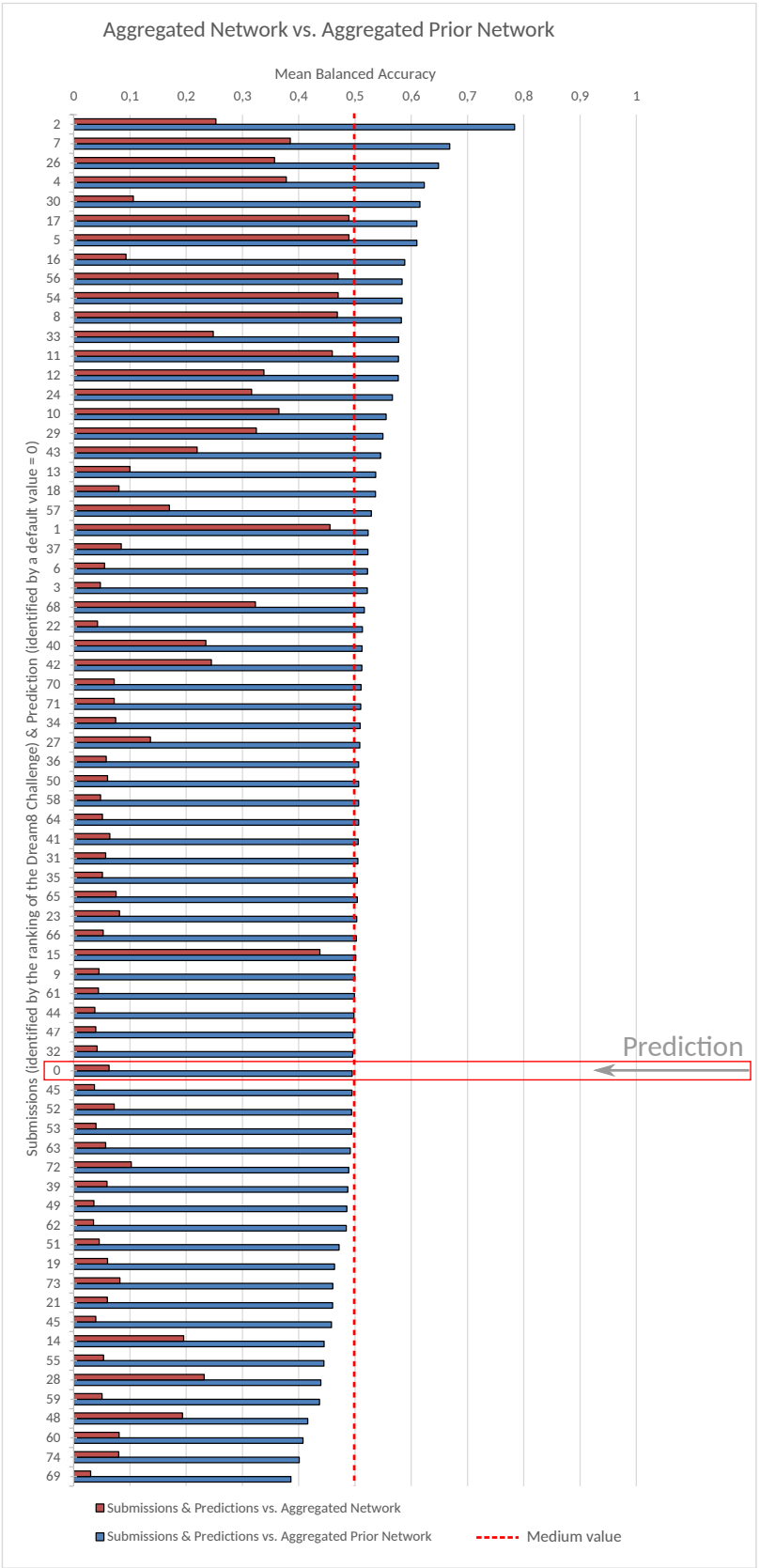

Figure 3.12.: New Ranking (Balanced Accuracy): Aggr. Network and Aggr.Prior Network

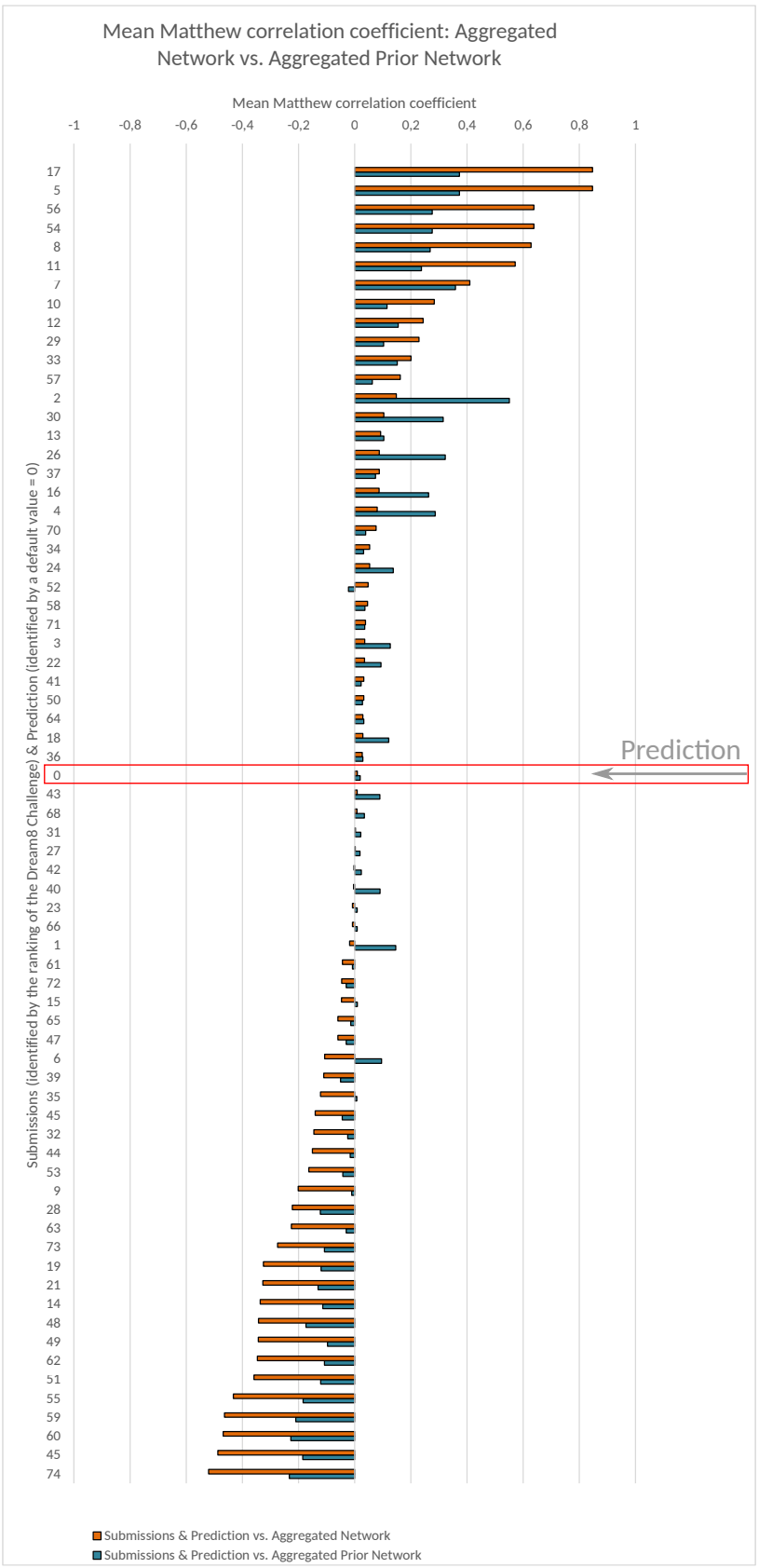

Figure 3.13.: New Ranking (Matthew correlation coefficient): Aggr. Network and Aggr.Prior Network

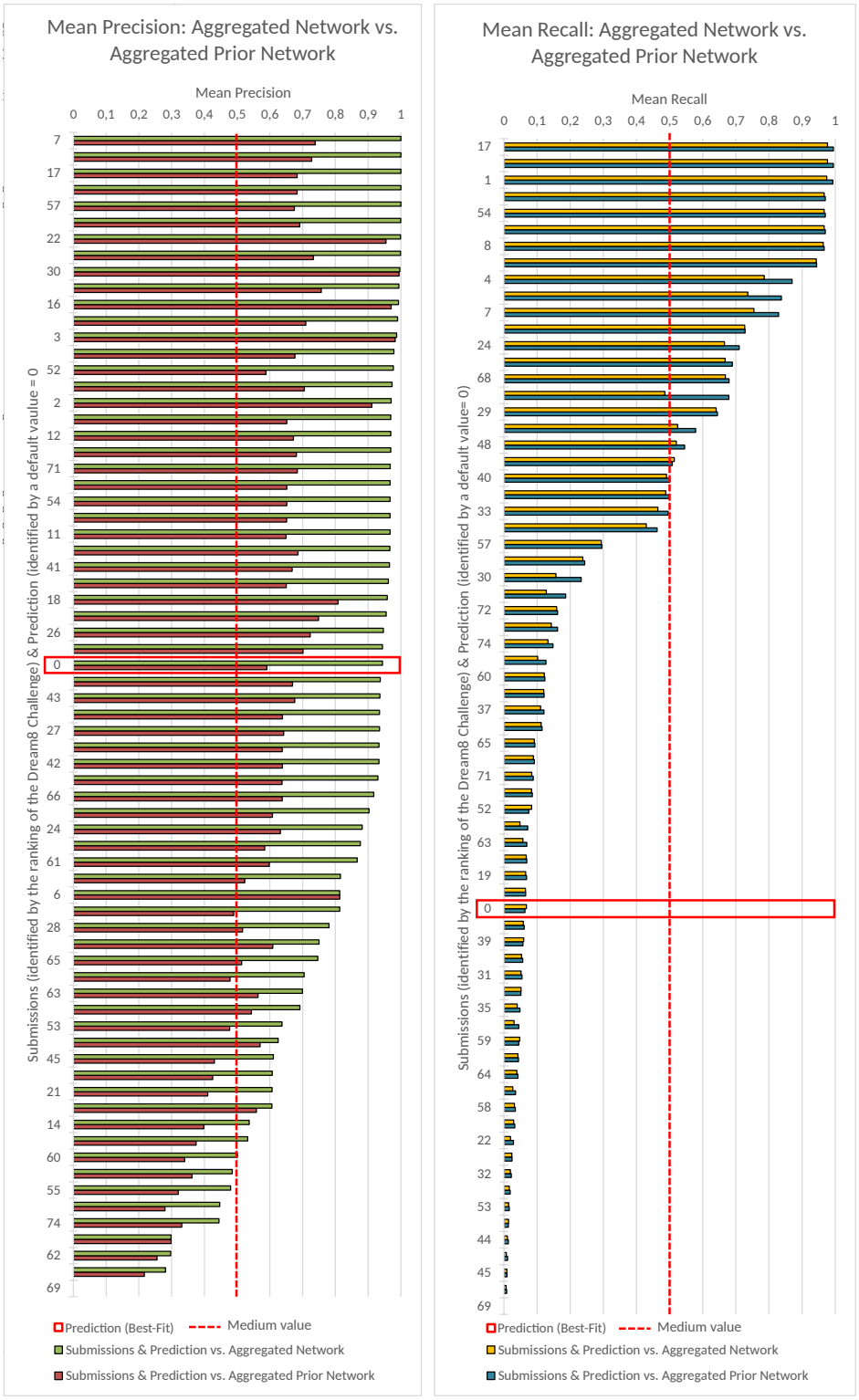

Figure 3.14.: New Ranking (Precision and Recall): Aggr. Network and Aggr.Prior Network

4. Discussion

Results

- Hier wuerde ich die Werte der Resulate diskutieren:Also woran es liegen koennte, dass der eine wert so und der andere so ist...
- Dann ein Fazit: Es konnte gezeigt, werden, dass der Boolsche Ansatz mit den anderen Teilnehmern der Challenge mithalten konnte

Verbesserungen

Dann wuerde ich mich an der Pipeline entlang hangeln und eroertern bei welchen Schritt welche Verbesserungen sinnvoll sind.

Nachteile von BooleanModels:
- Recent survey say: "Faithfulness to biological reality", "ability to model dynamics"is low
- TS2B needs a relatively large number of time points in the time-series data, otherwise we have the problem of "too many variables, too few equations", which a very large degree of freedom, thus a nonuniqueness of solutions.
- In practice the regulatory network is unknown, thus judging whether a BN provides a close approximation or notis not easy.
- ODE's are used to simulate the "true dynamics", but ODE's are not necessarily unique with respect to the dynamics they generate
- It is important to mention,that often the goldstandard network is constructed from multiple informtion (literature, mature information).In this case just the raw time-series data was provided without additional information.
- Parameter of the model have a big effect:e.g. Initial concentration in the time-series data set: Has significant influence on the topology and functions of the inferred BN. For instance: if the reaction rate is quite low the dependecy of two genes cant be observed in the synchronous documented time-series.

Vorteil:

- Very closely reflecting the topology of a regulatory network (upstream/downstream relationships)
- faithfulness to biological reality: Binarization, redundancy removal -> varying degrees
- number of Boolean functions is exponential in the number of genes in the system

General:
- The more complex a system is (heavily oscillatory,large variability) the more time points are needed.
- BN have a good predictive power for small networks
Fast convergence of BESTFIT can be explained by its lack of requirement that the candidate functions f' be complete.
Slow Convergence: REVEAL only accepts complete functions, thus R. produces more intuitive BN

Finally:
- The amount of time-points at which concentrations maust be sampled may be very large (with disagrees with commonly stated claim that BN require very little data to learn or train.)

Improvements and Outlook:
- There are several other boolean algorithms for network modeling:
- Baysian inference approach for a Boolean Network(BIBN): The maximum number of regulatory genes is bounded by two and used to infer a Boolean network by maximizing a joint posterior probability. To select the most informative regulatory genes an approximated multivariate mutual information measure is inccorporated.
- ARCANE: Time-delay algorithm for the reconstruction of accurate cellular networks (ARCANE) method is used to compute the mutual information by considering a time gap between gene expression values.
MIDER: Mutual information distance and entropy reduction method (MIDER),which defines a mutual information-based distance between genes to specify the directionality.
- MIBNI: Mutual information-based Boolean network inference method:The method first identifies a et of initial regulatory genes using mutual information-based feature selection, andthen improves the dynamics prediction accurary by iteratively swapping a pair of genes betweensets of the selected regulatory genes and the other genes. (good for structural and dynamic analysis)
- Kombination aus Boolean Approach und ODE: First break down the complex system to low level informative network. Attractor analysis. Then focus on the nodes of the attractor and the basinsand then just look by ODE on the smaller subset of nodes.(e.g. kinetic information)
- GROÃ§ER NACHTEIL:

These mutual information-based methods are computationally expensive, because they are implemented to compute exact mutual information values over all possible combination of genes.

- In insilico data network inference were almost better inferred when teams did not use prior knowledge.
- Conversely, notusing a prior network did not necessarily result in poor performance; mean AUROC scores ranged from 0.49-0.71, with prior knowledge: 0.49-0.78.
- Similar previous DREAM challenges showed that there is no clear relationship between method (inference algorithm) and performance. Pre-processing and implementation are important.
- With the aggregated prior network potentially novel changes can be highlighted.
- Although causal network inference mayfail for many theoretical and practical reasons, the results of this challenge showed, that it is feasible inferring significant large-scale networks in complex mamalian settings by a community effort.
- it is emphasized that further work needed to clarify the theoretical properties of the score.
Result: Their analysis revealed contexts that deviate from known biology, such deviations are likely to be particularly important for understanding disease-specific dysregulation and therapeutic heterogenity. Furthermore, it is possible that the literature is biased toward cancer, and for that reason priors based onthe literature may be less effective in other disease settings.

DREAM Challenge:Additional Data Details
- Some phopsho antobodies have low quality and should be xcluded from the set
- Annotation error might cause wrong scoring against the gold standard
- Antibodies are NOT comparable. Each antibody has varying degrees of affinity and avidity towards its target protein. Assuming two proteins have the same concentration, they may not result in the same level of data values. Therefore, they are not directly comparable.
- Batch Effect: Samples in different cell lines are NOT comparable (e.g. the datapoint for AKT_pS473 in MCF7 (serum,5min,DMSO) can not be compared with the corresponding datapoint in BT20).Normalization procedures could be used to reduce the batch effect.
- The antibodies available for this assay have evolved over time, so the proteins measured are not identical across all datasets. ($BT20 = 48, BT549 = 45, MCF7 = 41, UACC812 =$ 45phophoproteins)
- Some antibodies target more than one isoform of a protein (eg: the antibody for AKT targets AKT1, AKT2, AKT3)

-Normalization of microarray data important: Choice of normalization method important
–> Welche noramilisierungmethoden gaebe es noch?

It is desireable to determine the nodes whos *in-degree* is the highest among other nodes, whose removal can break down the network into isolated clusters [68].

Scoring metric selection: AUROC, AUPR
This is done, because computing the confidence score is computationally
limited, thus DREAMtools could not be applied.
For calculating the edge scores the pipeline has to run approximately a 100 times for obtaining a set of predictions, then counting the occurrences of each edge in each prediction which would return a probability for each edge. One execution of the pipeline needs about 5 hours. Of course it was taken into account to use a cluster (e.g. Allegro), due to a lack of globally implemented Bioconda for installing Pycluster, this approach failed.

conclusion

Hier wuerde ich den Schluss ziehen, dass man den Boolschen Ansatz durchaus verwenden kann, jedoch eine kombination verschiedener ansÃďtze wohl am besten eigenen wÃijrde. Zum Bespiel, bei groÃ§en komplexen Netzwerken: Am besten erst mit einem boolschen Ansatz grob betrachten dann ODE based approaches fÃijr detaillierter analysen verwenden...

References

[1] Natalie Berestovsky and Luay Nakhleh. An evaluation of methods for inferring boolean networks from time-series data. *PloS one*, 8(6):e66031, 2013.

[2] Assieh Saadatpour and Réka Albert. Boolean modeling of biological regulatory networks: a methodology tutorial. *Methods (San Diego, Calif.)*, 62(1):3–12, 2013.

[3] Shohag Barman and Yung-Keun Kwon. A novel mutual information-based boolean network inference method from time-series gene expression data. *PloS one*, 12(2):e0171097, 2017.

[4] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. *Journal of computational biology : a journal of computational molecular cell biology*, 7(3-4):601–620, 2000.

[5] DREAM Challenges. About dream: The dream challenges are crowdsourcing challenges examining questions in biology and medicine, 02.10.2014.

[6] Eric Bender. Challenges: Crowdsourced solutions. *Nature*, 533(7602):S62–4, 2016.

[7] U.S. Department of Energy Office of Science. Genomes to life program roadmap, 2001.

[8] Hans A. Kestler, Christian Wawra, Barbara Kracher, and Michael Kühl. Network modeling of signal transduction: establishing the global view. *BioEssays : news and reviews in molecular, cellular and developmental biology*, 30(11-12):1110–1125, 2008.

[9] Chris J. Oates, Bryan T. Hennessy, Yiling Lu, Gordon B. Mills, and Sach Mukherjee. Network inference using steady-state data and goldbeter-koshland kinetics. corrected. *Bioinformatics (Oxford, England)*, 28(18):2342–2348, 2012.

[10] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, 2002.

[11] Ilya Shmulevich, Ilya Gluhovsky, Ronaldo F. Hashimoto, Edward R. Dougherty, and Wei Zhang. Steady-state analysis of genetic regulatory networks modelled by probabilistic boolean networks. *Comparative and functional genomics*, 4(6):601–608, 2003.

[12] RBPAonline. Cell energy flow chart photosynthesis and cellular respiration key, 2018.

[13] Guy Karlebach and Ron Shamir. Modelling and analysis of gene regulatory networks. *Nature reviews. Molecular cell biology*, 9(10):770–780, 2008.

[14] Javier de Las Rivas and Celia Fontanillo. Protein–protein interactions essentials: Key concepts to building and analyzing interactome networks. *PLoS Computational Biology*, 6(6), 2010.

[15] Matteo Pellegrini, David Haynor, and Jason M. Johnson. Protein interaction networks. *Expert review of proteomics*, 1(2):239–249, 2004.

[16] Werner Müller-Esterl and Ulrich Brandt. *Biochemie: Eine Einführung für Mediziner und Naturwissenschaftler*. Spektrum Akad. Verl., Heidelberg, korrigierter nachdr. 2009 der 1. aufl. 2004 edition, 2009.

[17] National Center for Biotechnology Information, U.S. National Library of Medicine. Nrg1 neuregulin 1 [homo sapiens(human)], 04.11.2018.

[18] National Center for Biotechnology Information, U.S. National Library of Medicine. Igf insulin like growth factor [homo sapiens (human)], 28.10.2018.

[19] National Center for Biotechnology Information, U.S. National Library of Medicine. Hgf hepatocyte growth factor [homo sapiens (human)], 05.11.2018.

[20] National Center for Biotechnology Information, U.S. National Library of Medicine. Fgf1 fibroblast growth factor 1 [homo sapiens (human)], 04.11.2018.

[21] National Center for Biotechnology Information, U.S. National Library of Medicine. Irs1 insulin receptor substrate 1 [homo sapiens (human)], 04.11.2018.

[22] National Center for Biotechnology Information, U.S. National Library of Medicine. Egf epidermal growth factor [homo sapiens (human)], 21.20.2018.

[23] National Center for Biotechnology Information, U.S. National Library of Medicine. Tspo translocator protein [homo sapiens (human)], 02.10.2018.

[24] National Center for Biotechnology Information, U.S. National Library of Medicine. Srf serum response factor [mus musculus (house mouse)], 08.10.2018.

[25] author name not available. Hpn-dream breast cancer network inference challenge.

[26] A. A. Al-Tubuly. Sds-page and western blotting. *Methods in molecular medicine*, 40:391–405, 2000.

[27] Stefanie Boellner and Karl-Friedrich Becker. Reverse phase protein arrays-quantitative assessment of multiple biomarkers in biopsies for clinical use. *Microarrays (Basel, Switzerland)*, 4(2):98–114, 2015.

[28] Anitha Ramaswamy, E. Lin, Iou Chen, Rahul Mitra, Joel Morrisett, Kevin Coombes, Zhenlin Ju, and Mini Kapoor. Application of protein lysate microarrays to molecular marker verification and quantification. *Proteome science*, 3:9, 2005.

[29] Katherine M. Sheehan, Valerie S. Calvert, Elaine W. Kay, Yiling Lu, David Fishman, Virginia Espina, Joy Aquino, Runa Speer, Robyn Araujo, Gordon B. Mills, Lance A. Liotta, Emanuel F. Petricoin, and Julia D. Wulfkuhle. Use of reverse phase protein microarrays and reference standard development for molecular network analysis of metastatic ovarian carcinoma. *Molecular & cellular proteomics : MCP*, 4(4):346–355, 2005.

[30] Lance A. Liotta, Virginia Espina, Arpita I. Mehta, Valerie Calvert, Kevin Rosenblatt, David Geho, Peter J. Munson, Lynn Young, Julia Wulfkuhle, and Emanuel F. Petricoin. Protein microarrays: Meeting analytical challenges for clinical applications. *Cancer cell*, 3(4):317–325, 2003.

[31] Hannes Klarner. *Contributions to the Analysis of Qualitative Models of Regulatory Networks*. Dissertation, Freie Universität Berlin, Berlin, November 2014.

[32] Harri Lähdesmäki. On learning gene regulatory networks under the boolean network model. *Machine Learning*, 52(1/2):147–167, 2003.

[33] Georgios A. Pavlopoulos, Maria Secrier, Charalampos N. Moschopoulos, Theodoros G. Soldatos, Sophia Kossida, Jan Aerts, Reinhard Schneider, and Pantelis G. Bagos. Using graph theory to analyze biological networks. *BioData Mining*, 4:10, 2011.

[34] Martin Hopfensitz, Christoph Mussel, Christian Wawra, Markus Maucher, Michael Kuhl, Heiko Neumann, and Hans A. Kestler. Multiscale binarization of gene expression data for reconstructing boolean networks. *IEEE/ACM transactions on computational biology and bioinformatics*, 9(2):487–498, 2012.

[35] S. Liang, S. Fuhrman, and R. Somogyi. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 18–29, 1998.

[36] István Albert, Juilee Thakar, Song Li, Ranran Zhang, and Réka Albert. Boolean network simulations for life scientists. *Source Code for Biology and Medicine*, 3:16, 2008.

[37] Lee et al. Transcriptional regulatory networks in saccharomyces cerevisiae. *Science (New York, N.Y.)*, 298(5594):799–804, 2002.

[38] Élisabeth Remy, Paul Ruet, and Denis Thieffry. Graphic requirements for multistability and attractive cycles in a boolean dynamical framework. *Advances in Applied Mathematics*, 41(3):335–350, 2008.

[39] Warren S. Sarle, Cary, NC. What are the population, sample, training set, design set, validation set, and test set?, 17.05.2002.

[40] Barab&aacute and A. si. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[41] Thomas Schaffter, Daniel Marbach, and Dario Floreano. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics (Oxford, England)*, 27(16):2263–2270, 2011.

[42] Hannes Klarner, Adam Streck, and Heike Siebert. Pyboolnet: a python package for the generation, analysis and visualization of boolean networks. *Bioinformatics (Oxford, England)*, 33(5):770–772, 2017.

[43] Adrien Fauré, Aurélien Naldi, Claudine Chaouiya, and Denis Thieffry. Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics (Oxford, England)*, 22(14):e124–31, 2006.

[44] Cooper GM. The cell: A molecular approach: Tumor suppressor genes, 2000.

[45] Hill et al. Inferring causal molecular networks: empirical assessment through a community-based effort. *Nature methods*, 13(4):310–318, 2016.

[46] Neve et al. A collection of breast cancer cell lines for the study of functionally distinct cancer subtypes. *Cancer cell*, 10(6):515–527, 2006.

[47] Garnett et al. Systematic identification of genomic markers of drug sensitivity in cancer cells. *Nature*, 483(7391):570–575, 2012.

[48] Barretina et al. The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603–607, 2012.

[49] Hennessy et al. A technical assessment of the utility of reverse phase protein arrays for the study of the functional proteome in non-microdissected human breast cancers. *Clinical proteomics*, 6(4):129–151, 2010.

[50] Mark Lund Adam Lund. Measures of central tendency, 2018.

[51] Selim Aksoy and Robert M. Haralick. Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern Recognition Letters*, 22(5):563–582, 2001.

[52] Wenbin Liu, Zhenlin Ju, Yiling Lu, Gordon B. Mills, and Rehan Akbani. A comprehensive comparison of normalization methods for loading control and variance stabilization of reverse-phase protein array data. *Cancer informatics*, 13:109–117, 2014.

[53] J. MacQueen. Some methods for classification and analysis of multivariate observations. *University and California Press, Los Angeles*, (volume 1):281–297, 1967.

[54] Shengtong Han, Raymond K. W. Wong, Thomas C. M. Lee, Linghao Shen, Shuo-Yen R. Li, and Xiaodan Fan. A full bayesian approach for boolean genetic network inference. *PloS one*, 9(12):e115806, 2014.

[55] Pietro Zoppoli, Sandro Morganella, and Michele Ceccarelli. Timedelay-aracne: Reverse engineering of gene networks from time-course data by an information theoretic approach. *BMC bioinformatics*, 11:154, 2010.

[56] Alejandro F. Villaverde, John Ross, Federico Morán, and Julio R. Banga. Mider: network inference with mutual information distance and entropy reduction. *PloS one*, 9(5):e96732, 2014.

[57] Haseong Kim, Jae K. Lee, and Taesung Park. Boolean networks using the chi-square test for inferring large-scale gene regulatory networks. *BMC bioinformatics*, 8:37, 2007.

[58] Jeremiah J. Faith, Boris Hayete, Joshua T. Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J. Collins, and Timothy S. Gardner. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS biology*, 5(1):e8, 2007.

[59] Xiujun Zhang, Xing-Ming Zhao, Kun He, Le Lu, Yongwei Cao, Jingdong Liu, Jin-Kao Hao, Zhi-Ping Liu, and Luonan Chen. Inferring gene regulatory networks from gene expression data by path consistency algorithm based on conditional mutual information. *Bioinformatics (Oxford, England)*, 28(1):98–104, 2012.

[60] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE transactions on neural networks*, 5(4):537–550, 1994.

[61] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M. Buhmann. The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition*, pages 3121–3124. IEEE, 23.08.2010 - 26.08.2010.

[62] Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari. Optimal classifier for imbalanced data using matthews correlation coefficient metric. *PloS one*, 12(6), 2017.

[63] GoogleDevelopers. Classification: Thresholding | machine learning crash course | google developers, 01.10.2018.

[64] Ashish Anand, Ganesan Pugalenthi, Gary B. Fogel, and P. N. Suganthan. An approach for classification of highly imbalanced data using weighting and undersampling. *Amino acids*, 39(5):1385–1391, 2010.

[65] Cokelaer et al. Dreamtools: a python package for scoring collaborative challenges. *F1000Research*, 4:1030, 2015.

[66] author name not available. Hpn-dream breast cancer network inference challenge.

[67] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In William Cohen, editor, *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, New York, NY, 2006. ACM.

[68] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.

A. Appendices

A.1. DREAM8-Challenge scoring priciples

For determining the Receiver-Operating-Characteristic-Curve a True-Positive-Rate (TPR) and a False-Positive-Rate (FPR) is calculated.

Definition A.1. True-Positive-Rate (TPR).
The TPR values are for the y-axis of the ROC.

$$TPR = \frac{TP}{TP + FN} \tag{A.1}$$

Definition A.2. False-Positive-Rate (FPR).
The FPR values are for the x-axis of the ROC.

$$FPR = \frac{FP}{FP + TN} \tag{A.2}$$

Threfore, a threshold τ is set and the confidence scores are classified by this threshold. Confidence scores below this threshold take a value of 0 indicating an edge is less likely to occur and a confidence score above τ is taking a value of 1 indicating an edge is potential. By increasing the threshold $\tau \in [0, 1]$ the amount false positve decrases and false negative increase. Resulting calasses are put into a context of True Positive Rate ($TPR = \frac{TP}{TP+FN}$) and False Negative Rate ($FPR = \frac{FP}{FP+TN}$). This yields a set of values returning a value of the area under the receiver operating characteristic curve (AUROC).

Similar to balanced accuracy the AUPR (area under the precision recall curve) metric is used for imbalanced classes in the confusion matrix taking precision and recall in relationship by shifting τ.