Freie Universität zu Berlin

Masterthesis

Inference of Boolean Networks considering real-life time course Data

Nina Valery Kersten

Supervisors

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

Advisor

Phd. Robert Schwieger

November 20, 2018

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 Information derived from published and unpublished work of others has been acknowledged in the text and references are given in the list of sources.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

Abstract

Contents

List of Abbreviations

ATP	adenosin triphosphat
CSV	Comma Seperated Values
DREAM	Dialogue on Reverse Engineering Assessment and Methods
e.g.	exempli gratia
ELISA	Enzymatic Immunoassay
GRN	Gene regulatory network
IG	Interaction Graph
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

List of Figures

List of Tables

1 Introduction

The development and functioning of a cell and its organism is a product of a complex cellular machinery, where the interaction of genes, proteins, mRNA and many other substances induce a cascade of extracellular signals transducted by mechanisms of the cell membrane,reaching the nucleus of the cell, initiating a transcription process that controls the production and abundance of proteins. Proper functioning of these regulatory networks is essential to the survival and adapation of a cell. Malfunctioning has been identified as the cause of various diseases [2]. Recent advances in high-throughput techniques provide a big abundance of information about various biological interactions measured over a series of time. It is necessary to handle this big data poperly for significant analysis. Biological information can be considered at different levels (e.g. gene-gene interaction via mRNA, protein-protein interaction, metabolic interaction) described by a network with certain properties. Several strategies are known to infer a network from biological data like Baysian networks, Ordnianry Differential Equation (ODE) networks, Neural networks and Boolean networks [6].

1.1 Motivation

It is desirable to simplify a complex biological system such that the main important parts can be analyzed easily. For this reason the approach of inferring boolean networks is chosen in this work. The simplification starts by converting the continuous biological time-series measurements (e.g. expression value of a gene, concentration of a protein, abundance of metabolites) into discrete values. Therefore two discretization algorithms are applied: The Two cluster and the iterative k-means binarization. The components of a boolean network of a system are represented by nodes (vertices) and the interactions among the nodes are depicted by edges. The orientation of an edge can be directed such that an edge points from one node to another. This directed edge can be positive or negativ (sign),depending on the influence of one node (the regulator) to another node (the regultee). [6]

In this work three well known inference algorithms are applied (RevealL,Best-fit, Full-fit). [1] Boolean networks are inferred for two in silico data sets and a big real-life time course data set provided by a platform called Dream Challenge and contains concentration measurements

of $\sim$ 48 phosphoproteins of four cell lines of breast cancer tissue. To determine the predictive power of a model the prediction is compared to a gold standard data set. The aim of this work is to show the scalbility of a of boolean inference algorithms considering a big rela-life time course data set.
The first section is taking a glance on the different kinds of biological input data, the graphtheoretical background of boolean networks and its preprocessing. In the second section it is getting more detailed by giving an overview of different inference algorithms, describing the in silico data collection and the data collection of the Dream Challenge data set. Then the application of a pipeline including the binarization and inference algorithms is described, showing the which differences come along with the size and complexity of a network. Finally the results of the comparison of the prediction against the gold standard is analyzed and discussed.
In this section the intention of using different types of biological input data is explained and what potentially will be the occuring problems. Afterwards the discretization algorithms, which binarize the data of the biological settings are explained. Then the graphtheoretical background of a boolean network in terms of structural and dynamic manner is defined and explained. Inferred boolean network need to be assessed proving that the applied inference method is significant. This performance measurement is achieved by different scoring metrics described in the last part of this section.

1.2 Biological Background

Depending on the aim of a network inference the biological input data can be depicted by the interaction of proteins, genes or metabolic substances. The choice of the biological input decides about the information content and thus the structure of the input data for further network inference algorithms. Biological interactions can be observed at different levels of information integration of a cell (gene gene interaction, protein protein interaction and metabolic interaction). In general, a signal binds to a membran 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 (messanger RiboNucleotide Acid) results in a protein. The resulting proteins influence the cell's survival by its profileration, induction of cell differentiation and apoptosis.

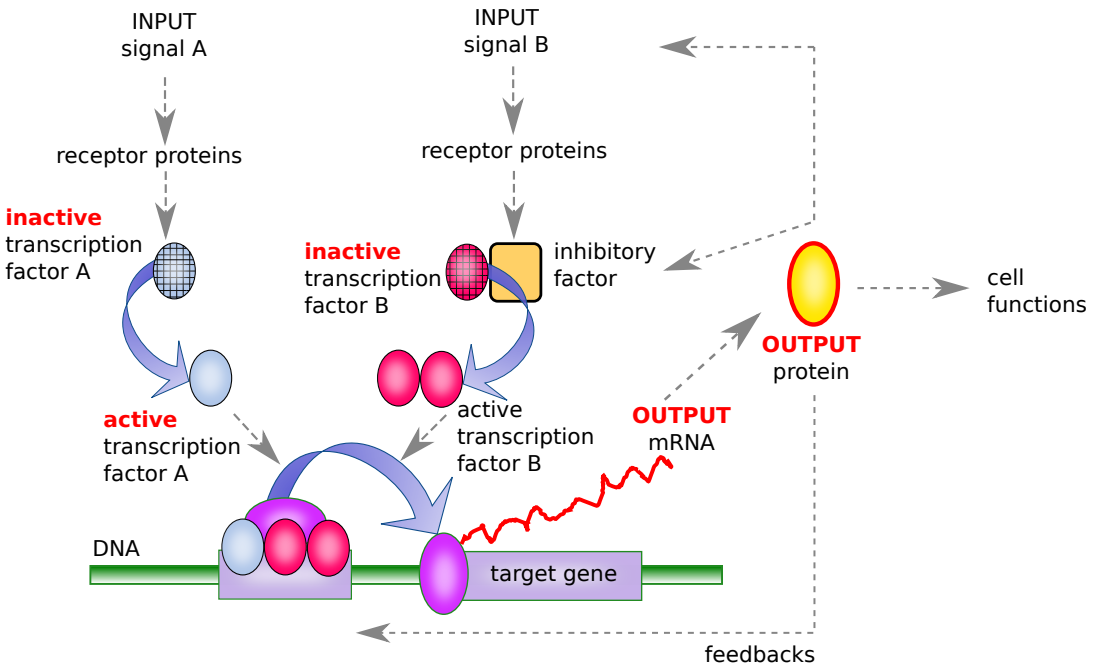

Figure 1.1: Transcriptional Signale Cascade This example shows two different input signals A and B (e.g hormon) bind to specific receptor protein. The complex of A activates the transcriptionfactor A that binds directly to the gene's cis-regulatory sequence inducing the expression of the target gene.The complex of B initiate a seperation of the inactive B (pink oval) from an ihibitory factor (yellow rectangle). Transcription factor B is then free to bind to the cis-regulatory sequence. Thus the expression level of this target gene is leveled by signal A and B. The mRNA output results in a protein poduct which can be neccessary for cell functions, play a role in the gene transcription process or inluence the signal cascade of the input signals[5].

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.[3] [**?**]

Gene Regulatory Networks

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

Metabolic networks

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

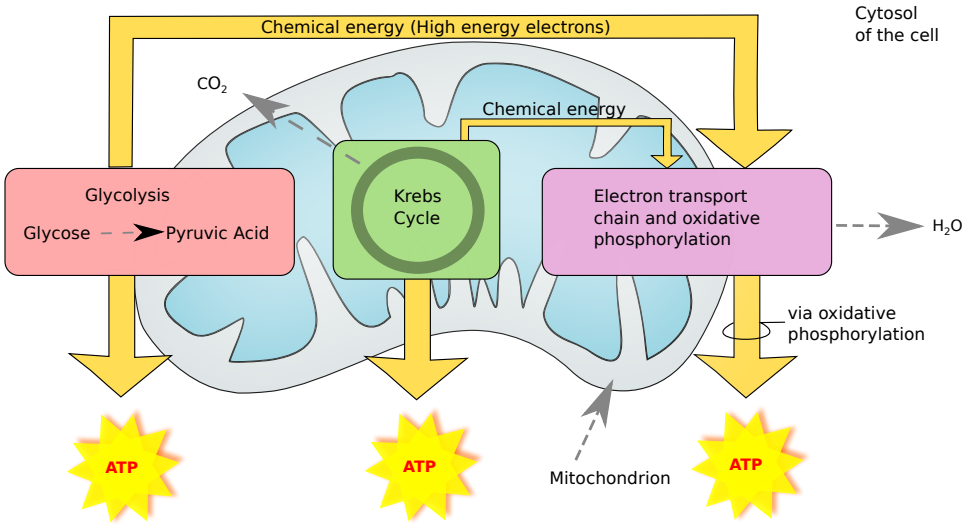

Figure 1.2: Metabolic Network of the cell respiration. Cell respiration in a mitochondrion, an essentiell compartment of the cell converting biochemnical energy from nutritiens into adenosin triphosphate (ATP) by releasing waste products of water H_2O and CO_2.

Protein-Protein-Interaction network

In contrast to the gene regulatory interaction network in a protein-protein interaction (PPI) network the proteins act directly among themselve. Thus the nodes in a network are the interacting proteins. Proteins interact by physical contacts(e.g. electrostatic forces) of high specificity. PPIs play a big role in electron transfer,signal transduction, transport across membranes and cell metabolism. The underlying assumption is that true interactions are likely to occur between proteins involved in the same biological process, proteins found in the same cell compartment, and proteins whose mRNA are coexpressed. The real-life data

set of the Dream8 Challenge 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 the biological context.

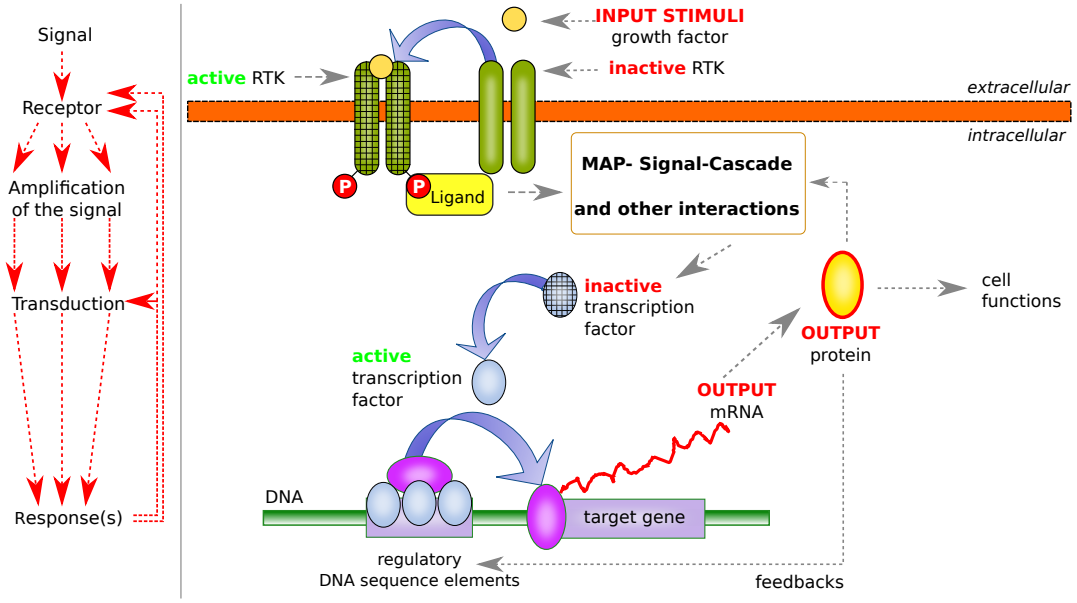

Figure 1.3: Transcriptional Signale Cascade An incoming stimuli (yellow circle) depicted as a growth factor binds to an inactive RTK (green), such that the RTK is activated. The RTK amplifies the signal and iniiates a signal transduction cascade, such that an inactive transcription fractor (blue oval) is activated, which binds to regulatory DNA sequence elements inducing the mRNA transcription of a target gene resulting in a new protein. The new protein can take place in cell functions or in the signal cascade. [**?**]

Referring to the general description of a transcriptional signal cascade (Figure 2.1) we state the receptor being an enzyme-associated receptor and the input signals are growth factors. A enzyme associated receptor has a polypetide chain integrated in the cell's membrane with a tyrosine kinase activity (Figure 2.3). Growthfactor receptors with a tyrosine kinase activity are called receptor tyrosine kinases (RTKs). These RTKs have the property of autophosphorylation (resp. they amplify their incoming signal). Binds a ligand to this receptor, first the receptor autophosphorylates and then phosphorylates the tyrosin residues of the ligand. By the phosphorylation of the receptor and several other ligands (rep. proteins) a phosphorylating cascade (e.g. signal transduction cascade) is induced. In this Mitogen activated protein phosphorylation cascade the MAP-kinase katalyzes the phosphorylation of effector proteins, such that inactive transcription factors are activated starting the transcription process of a target gene and finally resulting in a

protein. For the Dream8 Challenge growth factors were selected depicted as stimuli by pertubating the cell's signal transduction. Thus, different stimuli cause different signal transduction cascades. Depending on the incoming stimuli particular proteins appear and take place in the signal transduction cascade. The goal is figure out the protein- protein interaction in this phosphorylation cascade by inferring a boolean network based on the measurement of the proteins abundance considering different incoming growth factor (resp. stimuli) displayed in Table 2.1.

Notation	Name	Description
IGF1	Insulin like Growth Factor 1	Hormone, similar to the insulin function and structure
NRG1	Neuregulin 1	Membrane glycoprotein, mediating cell-cell signalling, critical role in growth and developement of the cell
HGF	Hepatocyte Growth Factor	Regulate cell growth, cell mortility and morphogenesis
FGF1	Fibroblast Growth Factor 1	Functions as a modifier of endothelial cell migration, proliferation and an angiogenic factor
Insulin	Insulin	Mutations in this gene are associated with type II diabetes and susceptibility to insulin resistance
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.
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.
Seerum	SRF	Member of the MADS box superfamily of transcription factors

Table 1.1: List of growth factors of Dream8 Challenge

The dysregulation of the genes of these growth factors have been linked to diseases such as cancer, schizophrenia, bipolar disorder and many more.

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. 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. 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 2.4).

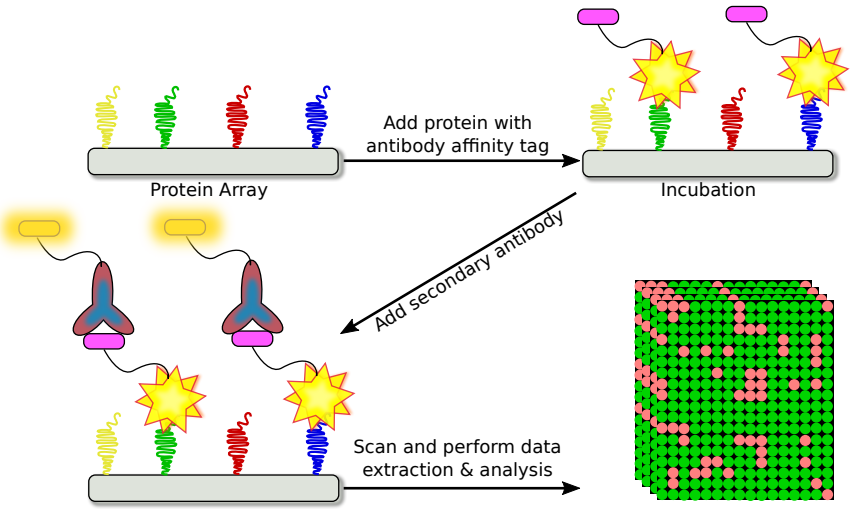

Figure 1.4: RPMA. Antibody binding and fluorometric detection. Proteins are tagged by a specific antibody. After incubation time the secondary antibody is added. Finally the abundance of proteins is determined by a fluorometric detection.

A detection reagent is put on each slide. Signal amplification and detection is done by an optical flatbed scanner if colormetric technique is used or by laser scanning. The resulting data is normalized, such that outliers are excluded from the data's structure.

The strength of RPMA is the high throughput, ulta-sensitive detection of proteins from extremly small numbers of input material which is not possible for western blotting and ELISA (Enzymatic Immunoassay). The small spot size on the microarray, ranging in diameter from 85 to 200 micrometres, enables the analysis of thousands of samples with the same antibody in one experiment. The high sensitivity of 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. A great improvement of RPMAs over traditional forward phase protein arrays is a reduction in the number of antibodies needed to detect a protein. The protein isn't detected directly which helps to preserve the proteins. Antibodies, especially phospho-specific reagents, often detect linear peptide sequences that may be masked due to the three-dimensional

conformation of the protein. This problem is overcome with RPMAs as the samples can be denatured, revealing any covered epitopes (part of a protein recognized by specific antibody).

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

1.3 Computational Background

After generating the biological data some preprocessing like normalisation and discretization has to be done. Therefore several strategies are known. Normalization is an essential step, because data can contain 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. In section 2.4 Data collection of Dream8 Challenge data a normalization method is described.

1.3.1 Binarization Algorithms

Before starting inferring a boolean network from real-life time course data the data has to be discretisized (resp. binarized) such that each continous value (e.g. concentration measurement) measured at a certain time point of a particular substance (e.g. gene, protein) becomes discrete and has either a value of 1 or 0. This simplified representation of measurements decides about the significance of an inferred network. Therefore the choice of an appropriate binarization algorithm is necessary. Here two well known algorithms are introduced, the two clusters k-means binarization algorithm and the iterative k-means binarization algorithm.
The input of an inference problem, of a binarization algorithm respectively, consists of 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. The binarization algorithms aim turning S into a binary trajectory B thus, a boolean network N can be inferred from B.

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 obervation'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).

$$\text{Two clusters k-means } = \begin{cases} 1 & \text{, if } S_i(t) \geq \mu(S) \\ 0 & \text{, if } S_i(t) \leq \mu(S) \end{cases} \tag{1.1}$$

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 interative k-means binarization method is introduced.

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 dijoint clusters $C^1_{S_i}, ...C^x_{S_i}$. In each Cluster all its values are replacd by the clusters mean $\mu(C^x_{s_i})$.

(2) In the next iteration step d is decremeted 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 1.1. Assume we have a time-series data with measurements for a gene A. Starting with a depth of $d = 3$ we have initally $k = 8$ clusters for each gene 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.

1.3.2 Graphtheoretical Background

In this section the knowledge of the discretisized biological data is put into a graphtheoretical context of a boolean network.

Definition 1.2. Undirected Graph and Directed Graph
In general, an undirected graph $G = (V, E)$ is defined as a set of vertices V describing the nodes of the system and a set of undirected edges $E = \{(i, j)|i, j \in V\}$ that define a relationship between node i and j. While in a ***directed graph*** *$G = (V, A)$ is an ordered pair, defined as a set of vertices V (nodes) and a set of directed edges A (arcs). A set of directed edges $A = \{(i, j)|i, j \in V\}$ describes the flow of information in network, where (i, j) describes the flow from i (tail)to j (head).*

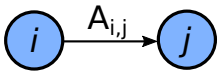

Figure 1.5: Directed Graph *G*.

Definition 1.3. Boolean Network
A boolean network is a directed graph $G(X, F)$, defined by a set of nodes X in a binary vector $X(t) = \{x_1(t), ..., x_n(t)\}$ representing state of a system at time t. Each element $x_i \in X$ corresponds to the state $x_i = 1$ or $x_i = 0$ of a species i. A set $F = \{f_1, ..., f_n\}$ of n transition functions (resp.: set of boolean transition functions $B = \{f_1, ...f_n\}$) contains a particular function for each x_i. Every transition function $f_i \in F$ is therefore a n-variable Boolean function $f : \mathbb{B}^n \leftarrow \mathbb{B}$ which we can represent by a Boolean expression over n input variables.
For every $f_i \in F$ s.t. $1 \leq i \leq n$,

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

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

A Boolean network model is a directed graph whose nodes represent the elements of a system, edges represent orientation of regulatory relationships between elements, and every node can have two possible initial states $x_i \in \{0, 1\}$ describing its activity [12-14]. The activity of a node means a qualtative rate a gene is being transcribed $x_i = 1$ or not $x_i = 0$, a transcription factor is active oder inactive, a protein's concentration is above or below a certain threshold (e.g. phosporylated or un-phosphorylated). Thus a network with n nodes will have 2^n possible states. As time passes the state of each node is determined by the

states of its neighbors, through a rule called a transition function. In a boolean context the transition function is depicted as a boolean function determining the future state of a node. This boolean function is represented by logical operators '&&', '||', '!' (resp. 'NOT', 'AND' and 'OR' ;resp. '$\wedge$' ,'$\vee$' , '$\neg$').
For instance a gene x_A could be influenced by another gene x_B positive and by a second gene x_C negative (2.2). Then the boolean algebra looks like this:

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

The application of the boolean function to a node's state returns a True or False state. Depending on the output of the boolean function, the state of the node either stays the same or changes.
A boolean network is abstracted to an interaction graph covering the structure of a network. Thus, an interaction graph captures the dependencies between the variables and not the dynamics of a system.

Definition 1.4. Interaction Graph
The interaction graph (IG) of a boolean network $G(X, F)$ is a directed graph $IG(X, \rightarrow)$ that consists of the node set X and the arc set $\rightarrow = \rightarrow_F \subseteq X \times X$ with $(x_i, x_j) \in \rightarrow$ iff f_{x_j} depends on x_i

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

Definition 1.5. *Sign* of an edge
The Sign of an edge is defined by $Sign(x_i \rightarrow x_j \subseteq \{+, -\})$.

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

The **in-degree** of a node describes the number of incoming edges determining the node's state. Hence the outdegree describes the number of outgoing edges of a node. The in-degree is a major aspect of representing the complexity in a system, thus of the inference problem and plays a major role in centrality analysis in a network (Example 1.7). Nodes with only

outcoing edges (in-degree= 0) are called sources, and nodes with only incoming edges (out-degree= 0) are sinks of the network. It is desireable to determine the nodes whos degree is the highest among other nodes, whose removal can break down the network into isolated clusters.

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 all the transition functions to their variables and updating their states. This computation leads from an Interaction Graph to a State Tansition Graph.

Definition 1.6. State Transition Graph
A State Transition Graph (STG), where $STG(X, \rightarrow)$ *is a directed graph with a set of nodes represented by a set of binary vectors* $F(t+1) = \{f_1(X(t+1)), ..., f_n(X(t+1))\}$ *representing the updated states of all variables after all of the functions in* F *have executed. The arcs denote possible transitions from one binary state vector to another.*

The State TransitionGraph is either updated *synchronously*, where each variable's state is updated simultaniously after all of the transition functions in F have executed or *asynchronously*, where the states are updated one at randomly choosing a transition function $f_i \in F$ and updating the state of x_i immediatly. In biological processes interaction of substances rarely happen at the same time, thus dynamical analysis is an important factor of detecting interacting substances.

The following example shows how an interaction Graph looks like, how the transition functions (resp.boolean rules) are derived from this graph and how the State Transition Graph is constructed for the synchronous and asynchronous case.

Example 1.7. The interaction Graph in Figure 1.6 shows a boolean network with three nodes $X = \{x_1, x_2, x_3\}$ (blue circle), where positve (resp. activating) edges (green) and negative (resp. inhibiting) edges (red) represent the interaction between the nodes. The in-degree of $x_1 = 2$, $x_2 = 1$ and $x_3 = 3$.

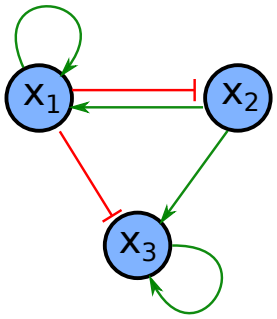

Figure 1.6: Interaction Graph

From the interaction graph the set of transition functions F (resp. boolean functions) can be derived (1.3) such that the state transition graph can be calculated.

$$f(x_{1,2,3}) = \begin{pmatrix} x_1 & \wedge & x_2 & \wedge & \neg x_3 \\ \neg x_1 & & & & \\ & & x_2 & \wedge & x_3 \end{pmatrix} \tag{1.3}$$

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 (1.4)-(1.11) below. This computation provide the new states for a synchronously updated state interaction graph displayed in the left graph in Figure 1.7.

$$x(t) = (0,0,0) \quad \rightarrow \quad f(x(t+1)) = (0,1,0) \tag{1.4}$$
$$x(t) = (0,0,1) \quad \rightarrow \quad f(x(t+1)) = (0,1,0) \tag{1.5}$$
$$x(t) = (0,1,0) \quad \rightarrow \quad f(x(t+1)) = (0,1,0) \tag{1.6}$$
$$x(t) = (1,0,0) \quad \rightarrow \quad f(x(t+1)) = (0,0,0) \tag{1.7}$$
$$x(t) = (1,1,0) \quad \rightarrow \quad f(x(t+1)) = (1,0,0) \tag{1.8}$$
$$x(t) = (1,0,1) \quad \rightarrow \quad f(x(t+1)) = (0,0,0) \tag{1.9}$$
$$x(t) = (0,1,1) \quad \rightarrow \quad f(x(t+1)) = (0,1,1) \tag{1.10}$$
$$x(t) = (1,1,1) \quad \rightarrow \quad f(x(t+1)) = (0,0,1) \tag{1.11}$$

The asynchronous updated state transition graph is computed by adding intermediate computation steps to the synchronus computation. Therefore the red highlighted computations (1.5),(1.9) and (1.11) are split by the amount of state changes. For instance in the computation step (1.5) $x(t) = (0,0,1)$ has two updates of states, x_2 changes and x_3 changes, too. As we know from the descriptin of asynchronous STG's in biological systems processes

happen uncommonly at the same time. Thus the initial state $x(t) = (0, 0, 1)$ provides two possible updates: $f(x(t+1)) = (0, 1, 1)$ and $f(x(t+1)) = (0, 0, 0)$. The same procedure is done with (1.9) and (1.11) and finally the asynchronous STG can be drawn, shown by the right graph in Figure 1.7.

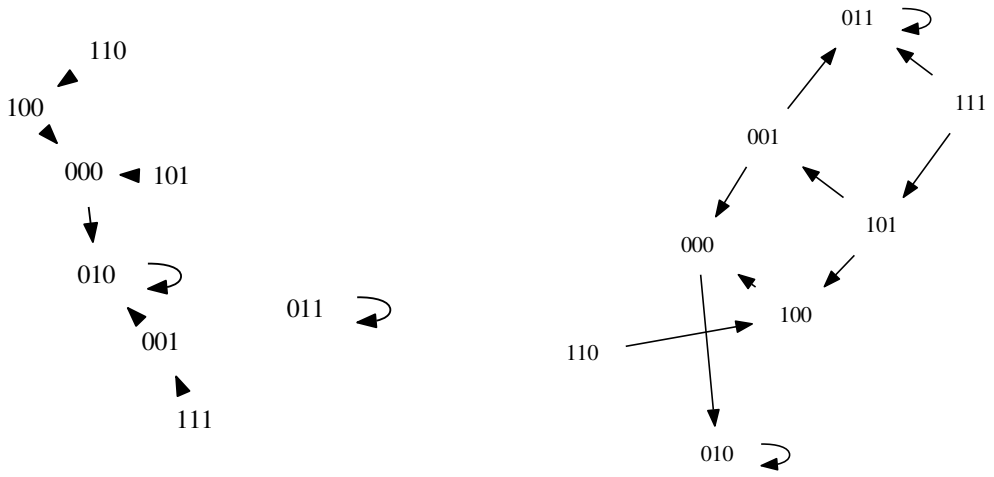

Figure 1.7: Synchronous and Asynchronous State Transition Graph. Left: Synchronous State Transition Graph; Right: Asynchronous State Transition Graph

2 Materials and Methods

A network inference problem is a classification problem in which edges in a network are classified as being true or false. In general, after selecting the parameter for inferring a network an experimental data set is divided up into two sets, a training and a test set. On the training set the network is inferred multiple times. The inferred network with the lowest error is tested by the test set. Experimental settings often contain batch effects from the laboratory and much of information (e.g. many nodes, pertubations by adding drugs), such that assessing an inference method by an *in silico* data set is useful. An *in silico* data set is a computational simulated data set containing no batch effects and with less nodes and pertubations representing a simplified version of the training data. Hence, by an *in silico* data set the parameters of inferring a network can be analyzed more easily. In this work an *in silico* data set is created from E.Coli for investigating the runtime and influence of the in degree of a network on the inference algorithms performance. Additionally a second *in silico* data set derived from the cell cycle network which is used to assess the performance of the inference algorithms relating to the cluster depth in binarization of the continous data and the number of sample points. The training data set is an experimental data set from the Dream8 Challenge. This challenge comes along with a test data set used as a gold standard with the same abundance of nodes as in the training set, but less sample points which makes it hard for the inference algorithms to perform. In this context a gold standard is a network being close to the structure of an 'unseen' network $G(X, A)$ the inference algorithms aims to infer. Thus the performance of an algorithms is achieved by testing an inferred network $G'(X', A')$ against a gold standard network $G(X, A)$. Often a gold standard network is a combination of prior knowledge from literature with previously inferred networks. In this work two gold standard networks are selected for the network evaluation in the Dream8 Challenge described in chapter 3.
The Figure 2.1 shows a raw structure of a pipeline for inferring and evaluating a network. Both *in silico* and experimental data are binarized, redundant transitions are removed and an inference algorithm is applied. The network with the lowest error is scored against a gold standard network.

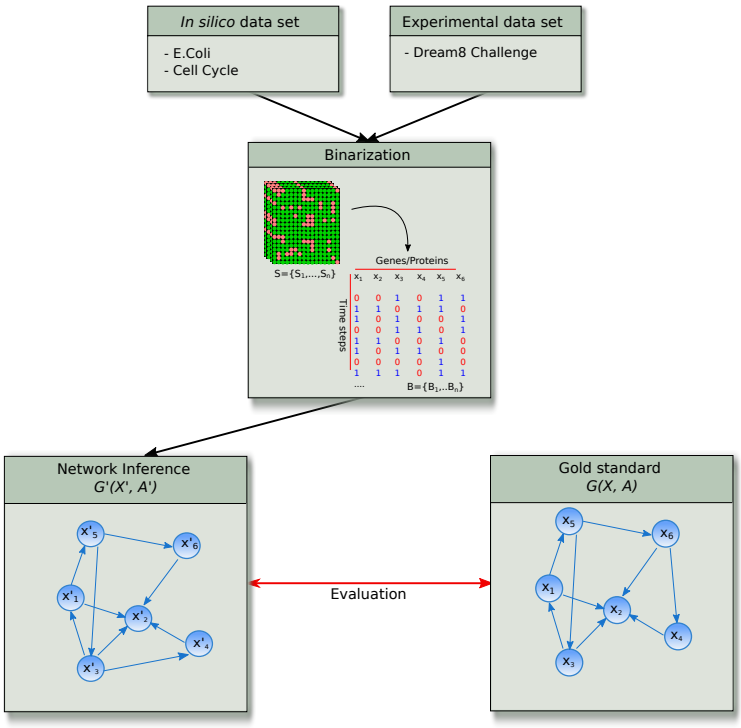

Figure 2.1: General Pipeline. Continous data sets S of *in silico* data sets and an experimental data set are discretisized to binary trajectories B. The inferred network G' is scored against a gold standard network G'.

2.1 Inferencealgorithms

Different approaches of inferring networks are known capturing several advantages and limitations. It is worhtwhile to find a good trade-off between simplicity, scalability, expressiveness and finding the abstraction level for a significant network. Many inference algorithms are known based on different computational models like the Bayesian approach , the ordinary diffential equation approach (ODE), the artificial neural networks)(ANN) and the Boolean model (BN).A Bayesian model is a graph-based model of joint multivariate probability distributions that captures properties of conditional independence between variables.

The system of differential equations (ODE's) creates networks in consideration with the kinetic properties of a biological system. An ODE is a powerful and flexible model to describe complex relations among components. But the higher the complexity of an unseen network is the more challenging it is to determine an appropriate set of equations which describe the network. Artificial Neural networks gather their knowledge by detecting patterns and relationships in data and learn through experience. An ANN is constructed by weighted

processing elements which constitute the neural layers and are organized in layers. Thus the behaviour of a ANN is determined by a transition function of each variable (neuron), by a learning rule and by the architecture itself. A big advantage, no previous knowledge is needed Boolean Models are simple Boolean Networks which are well known and an appropriate strategy of inferring the structure and the dynamics behaviour of complex data. A big advantage is that Boolean Network Models do not need any information about kintic parameters The relationships in a Boolean Network Model can be derived from a realtively small dataset. Furthermore a Boolean model could make qualitative predictions of large complex networks more feasible. For this reason the Boolean approach is chosen to show especially the scalability from a small *in silico* data set to a big expreimental data set.
Boolean Models are either probabilistic (PBN) or determinstic (DBN). In a PBN the next state of a node is determined by a transition function f_i selected with a certain probability from a set of transition function F. But this approch is limited due to the complexity of computational effort and the state-transitions and steady-state distributions . In a deterministic Boolean Model the 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. A Boolean Network Model learns from the binarized data set $B = \{B_1, ..., B_n\}$ a Boolean Network N by searching for a single node or a set of nodes describing the next state of another node.

In research a variety of Boolean Network Models have been published but their implementation efforts are rarely published, too. Participants of the Dream8 challenge submitted partly their implementations, bad documentated and with no Boolean approach. Therefore, the implementation of is used, providing three well known Boolean inference algorithms REVEAL, BESTFIT, FULLFIT and two binarization algorithms based on k-means clustering. The code is validated in such a way, that the experimental data of the Dream8 Challenge can be applied to , showing the algorithms scalability and is embedded in a pipeline for testing the algorithms performance for by evaluating the predicted networks N.

Redundancy Removal

Detecting the steady state in a Boolean network is necessary to indicate the significance of a transition. A steady state is defined as consecutive states, such that the state of a node does not change anymore (resp.:$X(t) = X(t+1)$). 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. Thus, except the last pair in the time-course data, each maximal consecutive sequence of states is removed, execpt one state. The remaining last pair in the time-course data set should indicate the true steady-state.

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.

Definition 2.1. Shannon-Entropy
The Shannon-Entropy is the probaility of observig a particular symbol of event $p(x)$, within a given sequence.

$$H = -\sum p(x) log p(x) \tag{2.1}$$

Example 2.2. 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 2.1: Table of states

$$H(x) = -p(0) * log[p(0)] - [1 - p(0)] * log[1 - p(0)] \tag{2.2}$$
$$H(x) = -0.4 log(0.4) - 0.6 log(0.6) \qquad = 0.97(40\%\ 0, 60\%\ 1) \tag{2.3}$$
$$H(y) = -0.5 log(0.5) - 0.5 log(0.5) \qquad = 1.00(50\%\ 0, 50\%\ 1) \tag{2.4}$$

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

$$H(x, y) = -\sum p(x, y) log p(x, y) \tag{2.5}$$

Example 2.4. The co-occurences of 1 and 0 in x and y are displayed in a quadratic matrix. The Joint Entropy $H(x, y)$ each combinatorial occurence of x with y is summed up (2.6).

y			
	1	3	2
	0	1	4
		0	1
		x	

$$H(x, y) = -0.1log(0.1) - 0.4log(0.4) - 0.3log(0.3) - 0.2log(0.2) = 1.85 \tag{2.6}$$

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

Definition 2.5. Mutual Information
The mutual information describes the rate of transmission.

$$M(X, Y) = H(X) + H(X, Y) - H(X, Y) \tag{2.7}$$

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

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

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

Example 2.6. 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 2.2 all initial possible combinatorial occurences of the nodes are represented in the left table. The right table shows the states of the nodes after one transition $(t + 1)$ for the node set $\{A', B', C'\}$.

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 2.2: 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'

Transition table "B":

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

Table 2.3

→

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 2.4

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 iteratio nof REVEAL stops here and the Boolean rule (resp. transition function) can be inferred (Table 2.5).

For further details on the calculation of the transition function f_B and f_C the reader is referred to the paper [1].
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

input	output
B	A
0	0
1	1

$\rightarrow$ $f_A = B$

Table 2.5

so on. Thus REVEAL is searching for the 'perfect' combination of nodes. Less computational sophisticated algorithms are BESTFIT and FULLFIT.

BESTFIT

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

$$T = \{X'(t) \in \{0,1\}^n : X_i(t+1) = 1\} \tag{2.9}$$

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

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

FULLFIT

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

function for each X_i. In practice this could be become infeasible.

Error Assessement with BooleanNet

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 was invented with the help of a Boolean simulator so called BooleanNet.
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 simultaniously updates all the states according to a synchronous simulation. Then the error of a Boolean network N with respect to B is defined by:

$$Error(N, B) = \frac{\sum_{1 \leq t \leq M}[(|B(t) - Y(t)|) * I_n]}{n * M} \tag{2.11}$$

The difference of B to the simulated Y in dependence on I_n a n-dimensional vector of all ones and M representing the number of binarized states in the reduced time-series. The lower the error the better the model fits the data. For this reason the model with the lowest error is selected for further analysis.

2.2 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 gold standard IG. The comparison is divided up into four possible classes displayed in the confusion matrix in Table 2.6. In 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, TP and FP denote the numbers of correctly and incorrectly predicted connections, respectively. And FN denotes the number of non-inferred connections of $G(V, A)$ in $G'(V', A')$, while TN denote the number of correct negative predictions.

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

Table 2.6: Confusion matrix

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

Definition 2.7. Precision

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

$$Precision = \frac{TP}{TP + FP} \tag{2.12}$$

Definition 2.8. Recall

Recall returns the proportion of actual positives identified correctly.

$$Recall = \frac{TP}{TP + FN} \tag{2.13}$$

Which means, *Recall* provides a percentage of inferred connections among the true connenction in $G(V, A)$.

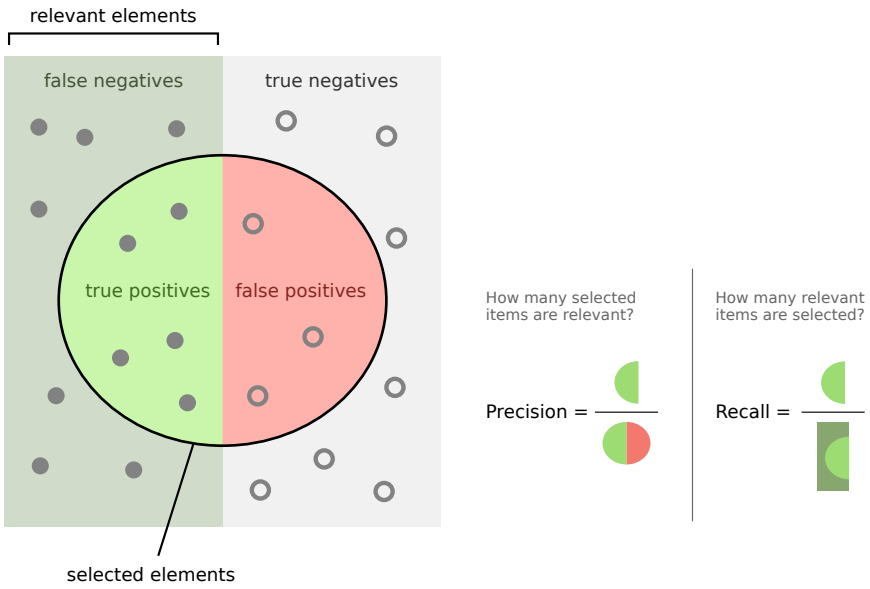

Figure 2.2: Precision and Recall

Definition 2.9. Accuracy

Accuracy is the percentage of correct predictions.

$$Accurancy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.14}$$

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

Definition 2.10. Balanced Accurancy (BACC)

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

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

Definition 2.11. Matthew Correlation Coefficient (MCC)

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

describes a total disagreement between prediction and observation.

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

Example 2.12. Given is a model that classified 100 tumors as malignant (the positive calss) or benign (the negative class):

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

Table 2.7: 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:

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

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

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

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

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

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

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

The *Accurary* (2.16) has a value of 0.91 which means 91% of the 100 total examples are predicted correctly. This result may look good at first sight, but this dataset is class-

imbalanced. In a class-imbalanced data set the labels of a binary classification problem have significantly different frequencies. For example, a disease data set in which 0.0001 of the examples have positve labels and 0.9999 have negative labels is a class-imbalanced problem. But a football game predictor in which 0.51 of example label one team winning and 0.49 label the other team winning is not a class-imbalanced problem.
Thus the significant disparity between the number of positive (here: $TP + TN = 91$) and negative labels (here: $FP + FN = 9$) falsifies the result. This observation is supported by the values of the $BACC$ and the MCC. The $BACC$ (2.21) has a value of 0.46, telling us that the prediction of the model is not that good as the *Accurary* shows and the MCC (2.22) has a value of 0.21 which is quite close to a value of 0. Thus the model predicted rather randomly than significantly. Furthermore the model has a *Precision* of 0.5, meaning when it predicts a tumor is malignant, it is correct 50% of the time. The *Recall* results in a value of 0.11, meaning the model correctly identifies 11% of all malignant tumors. In relation to this example it is worthwhile to identify most of the malignant tumors (high TP value) and get a low number of unidentified malignant tumors (low FN value).

2.3 Data collection: *In silico* data set

Before an inference algorithm and discretization method is applied to a real-life data set it is necessary to assess the performance. Several methods are known to generate an *in silico* data set, such that it was first tried to generate the *in silico* data set independant on a real life organism, by applying the Barabasi-Albert (BA) model [Quelle[22]] or generate multiple sets from one exmaple network.
The more sufficient method turned out to be creating an *in silico* data set by extracting subnetworks from the E.Coli network. E.Coli (Escherichia coli) is a well studied bacterium consisting of 1565 genes (resp. nodes) with 3758 interaction (resp. edges). The subnetworks extracted from E.Coli are generated by a tool, called GeneNetWeaver . A set of four networks with 10 to 14 nodes, each extended to 9 subnetworks, 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 $indegree \in \{1, ..., 9\}$. The range of 10 to 14 is selected due to the fact, that REVEAL is not performing by a system of 15 nodes and starting by 10 is for better comparison of the perfomance measurement to literature [PaperQuelle].

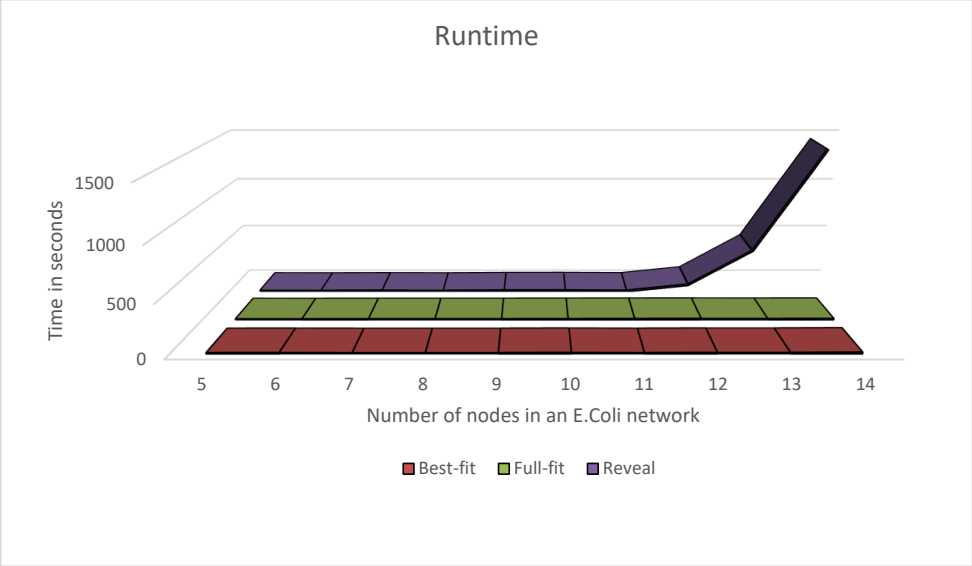

Figure 2.3: Starting by subnetworks of E.Coli of 5 nodes to a network of 14 nodes REVEAL is running out of time.

These networks help to figure out which algorithms performs the best by increasing the number of incoming links of a node. The number of incoming links represent the degree of complexity of the inference problem.
In addition a small real-life network of the mammalian cell cycle is used to asses the performance of the algorithms regarding the number of measurements and the clustering depth in the discretization step (Figure 2.4). 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.
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. This cell cycle network consist of 10 interacting transcriptional counterparts of genes with 35 edges and has a maximal in-degree of 5. Therefore this network is an appropriate network to test the algorithms with.

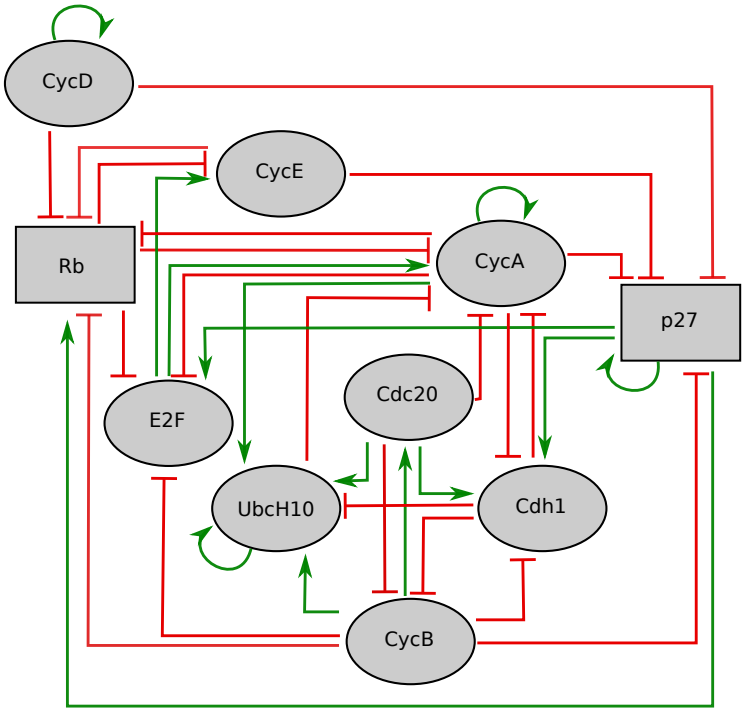

Figure 2.4: Cell Cycle. Logical regulatory directed graph for the mammalian cell cycle network. Each node represents a key regulatory element and each edge the interaction between them. Arrows describe activating activity (green) while blunt arrows describe inhibitory activity (red).

2.4 Data collection: Real-life course time data set

For a Boolean network inference of real-life time-course data, the data of a platform so-called Dialogue on Reverse Engineering Assessment and Methods (DREAM) - Challenge is used. The DREAM-Challenge is a non-profit, collaborative community effort consisting of contributors from across the research spectrum of questions in biology and medicine. This organization was built in 2006 and publishes crowdsourcing challenges with transparent sharing of data, thus everyone can participate the challenge. The DREAM-Challenge has partnered with Sage Bionetworks, which provide the infrastructure by Sage Bionetworks Synapse platform to get access to the open collaborative data analysis. Overall the DREAM-Challenge is a helpful instrument to get real-life data, comparing results and interact with other researchers all over the world, while contribute solutions to biological and medical questions.
The challenging question is to decide which Dream Challenge data set could be useful for inferring Boolean networks with the introduced algorithms. For inferring a Boolean network the desired data set should contain 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- Network Infernece 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.. knockout experiments, gene deletion experiments, applied drugs and enviromental 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 microarray data with less pertubation information, here the application of eight stimuli, but enough sample points by about $\sim$85 for in each data set. Therefore this Dream Challenge is selected.

2.4.1 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. Causal signaling networks contain causal edges which may represent direct effects or indirect effects that occur via unmeasured intermediate nodes. For example in Figure 3.3 the inhibition of a parent node A can change the abundance of the child node B described by a directed edge. If node A causally influences node B via

measured node C, the causal network should contain edges from A to C and from C to B,but not from A to B (top). However, if node C is not measured (and is not part of the network), the causal network should contain edges from A to B (bottom). In both cases the inhibition of A will lead to a change in node B.

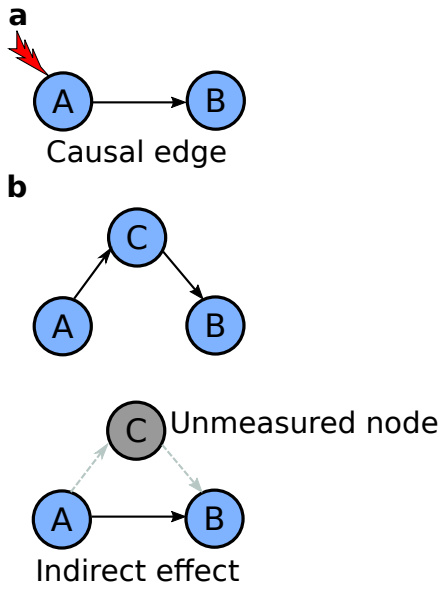

Figure 2.5: Causal edges. (a) Inhibition of a parent node A (red arrow) can change the abundance of child node B. (b) If node A influences B causually by measuring node C, then the causal network should contain edges from A to C and from C to B, but not from A to B.

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. 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. Knowledge about the *Sign* of an edge (i.e., wether activating or inhibitory) is neglected. In B the causal network is created and in the other two sub challenges the phosphoprotein time-course data is predicted under further pertubations and in the last challenge methods are developed to visualize these complex, multidimensional data sets. This work focuses on subchallenge SC1-A considering the inference of the interaction graph without taking the edge weight or *Sign* into account. Thus the scability of the three inference algorithms from a small network to a big one can

be assessed.

2.4.2 Data structure

The challenge spanned 32 different contexts, each definded 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 inhbit the ability of the target to catalyse phosphorylation of its substrates but not necessarily inhibit the phophorylation of the target itself. 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 abberations.

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 2.6).

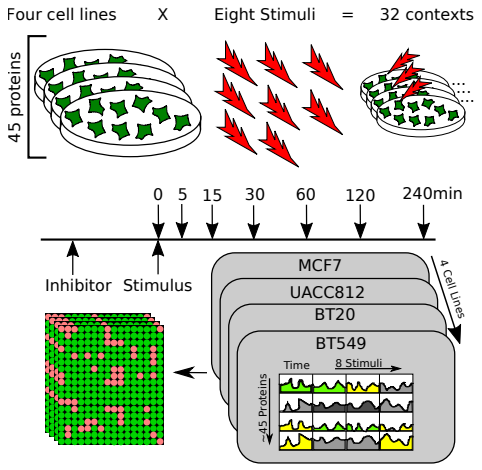

Figure 2.6: Data collection. Four cell lines (MCF7, UACC812, BT20, BT549) are treated with eight stimuli resulting in 32 contexts. Every experiment starts by adding an inhibitor followed by a stimlus. The concentration of 45phosphoprotein detecting antibodies is measured after 5,15,30,60,120 and 240 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. 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 32 contexts) is contained in a Comma Seperated Values (CSV) file format.

Normalization

The data is normalized 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 arraged 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'. 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.

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 2.8). 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 Anibody 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 timepoint of measurement followed by the concentration measured for each phosphoprotein detecting antibody.

			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 2.8: Table of CSV file

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 paramter 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 subnetworks and the cell cycle network continous data sets ($S = \{S_1, S_2, ..., S_n\}$) are generated with *odefy*, a MATLAB- and Octave-compatible toolbox for the automated tansformation of Boolean models into systems of ordinary differential equations [4] (Figure 3.3). 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 in the structure of the Dream8 Challenge input data (Table 2.8) 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.

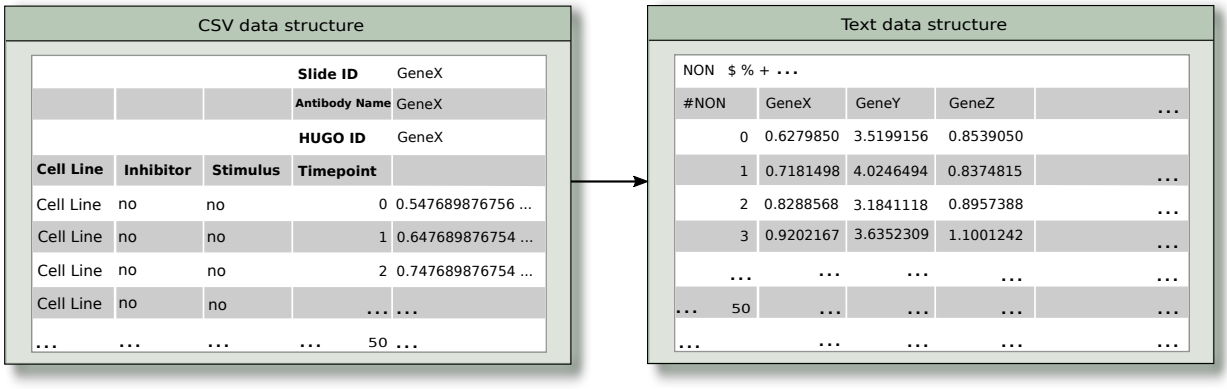

Figure 3.1: CSV to TXT: Anonymized single character represent the species and information about cell line,inhibitor and stimulus is neglected.

After discretizing continous 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, sucht that the inference algorithms runs i times until a network with an error ($Error(N, B)$) lower that the minimal error is achieved. It is worthwhile to get an error of 0, meaning the 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 processing.

Inference settings

For investigating the impact of the *in-degree* in a network on the algorithms performance, subnetworks 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 reliablity regarding the trade-off between simplicity and loss of information (e.g. oscillations).
For assessing the dependence of an inference algorithm to the number of sample points, continous data for the cell cycle is generated for $m \in \{50, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$. Starting by a number of 50 sample points is due to the fact, that with lower amount the inference algorithms can not run. Complex systems, like a cell cycle are oscillating, thus a large number of sample points are needed to achieve a 'good' Boolean network. And 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 $d = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. In Table 3.1 shows a summerized overview of these settings.

	E.coli	Cell cycle
# networks	45	1
# nodes	$\{10, 11, 12, 13, 14\}$	10
max. *in-degree*	$\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$	5
# sample points	100	$\{50, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$
cluster depth	3	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
Best-fit	✓	✓
Full-Fit	✓	✓
Reveal	✓	✓

Table 3.1: *In silico:* Setting Table

Prediction processing

The predicted Boolean networks are converted (from *bnet* format) to Interaction graphs (to a *sif* format) by *PyBoolNet* (Figure 3.2). Each interaction graph is scored against a 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 coeffiecient.

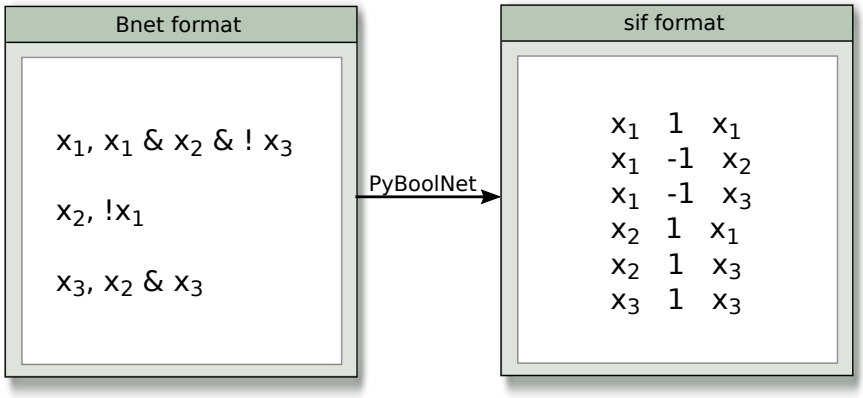

Figure 3.2: Boolean Network to Interaction Graph: The predicted Boolean network N is converted into an interaction graph IG.

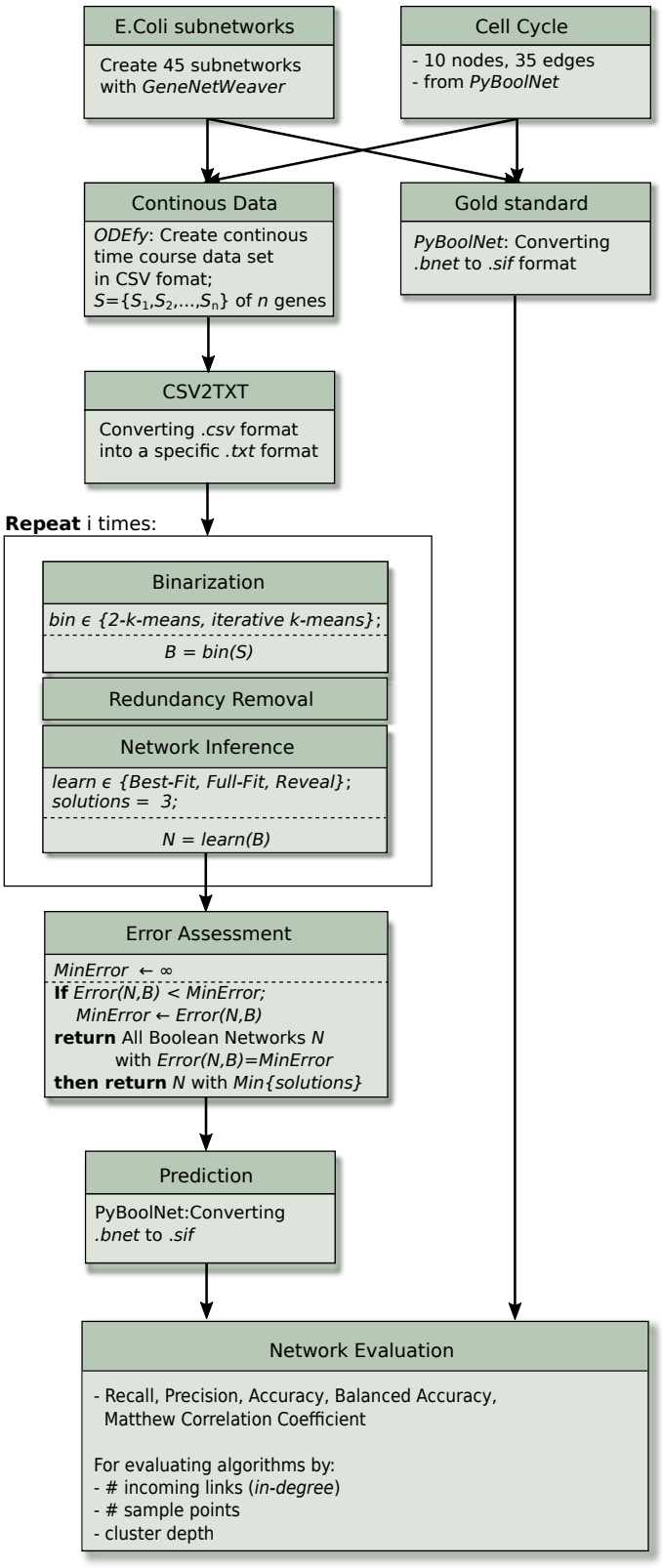

Figure 3.3: Pipeline *in silico*.

3.2 Results of the *in silico* data set

In-degree

Figure 3.4 shows 45 subnetworks of *E.coli* grouped into nine categories, each containing five subnetworks with n nodes; $n \in |V|$, where $|V| = \{10, 11, 12, 13, 14\}$. A category denote the number of incoming links (resp.*in-degree*) of each node in a network. The mean accuracy of *Best-Fit, Full-Fit and Reveal* shows that increasing the *in-degree* causes a decreasing of the accuracy.

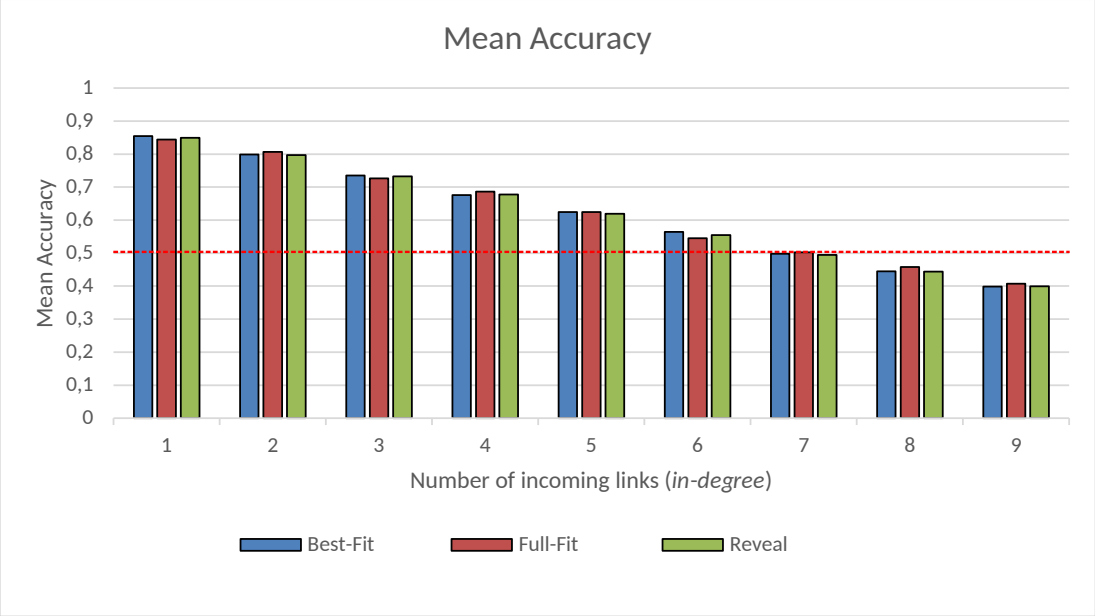

Figure 3.4: Average Structure Accuracy. *E.coli* subnetworks are grouped into nine groups definded by the *in-degree* of the nodes in a subnetwork. With increasing *in-degree* the average accuracy of all three inference algorithms decreases.

Thus the more nodes occure 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 covers the observation of [1], where the settings where slightly different. Instead of grouping sets of networks, they grouped the nodes of about 300 networks with different network sizes ($|V| = 10, 20$,...100). Here, no significant difference of the algorithms' performance can be observed.

Number of sample points

For the cell cycle continous data different sets with different number of sample points $m \in \{50, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$ are generated. The Figure 3.5 shows that changing the amount of sample point does not change significantly the performance of the algorithms.

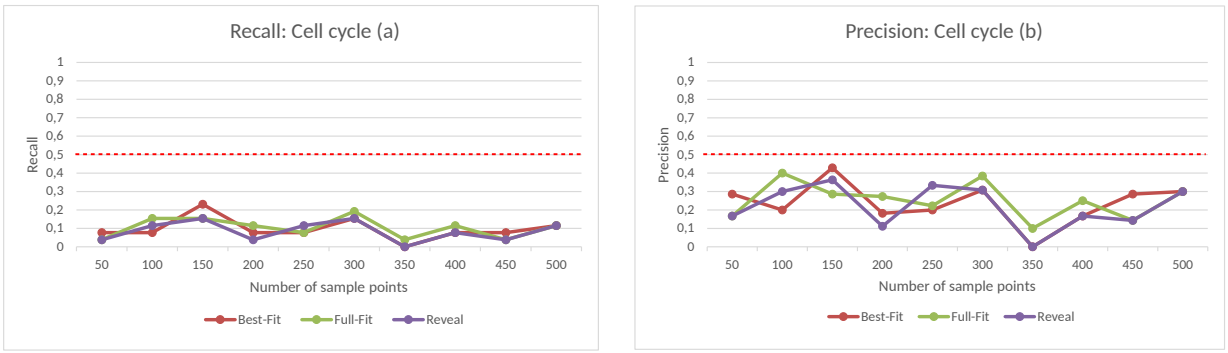

Figure 3.5: Number of sample points. Changing the number of sample points despite a few fluctuations does not change precision and recall of the inference algorithms.

Balanced accuracy shows the best that there is no significant dependence of the algorithms performance on the number of sample points (Figure 3.6). All inference algorithms range around a balanced accuracy value of $0,5$ showing that the algorithms perform almost at random.

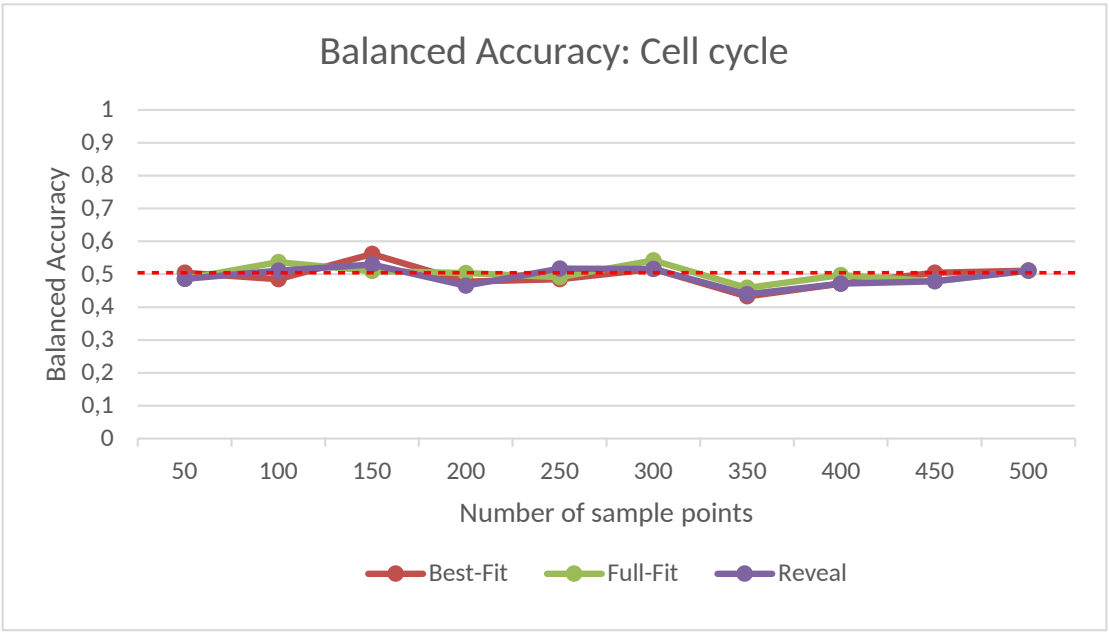

Figure 3.6: Balanced Accuracy: Number of sample points.

Cluster depth

For a clustering depth of 1 to 10 the three inference algorithms does not show any significant changes (Figure 3.7). As mentioned before, the iterative *k-means* binarization algorithms captures the oscillations in a system in contrast to the two cluster *k-means* algorithm. Especially regarding recall and balanced accuracy the performance reflects measured performance in investigating the influence of sample points.

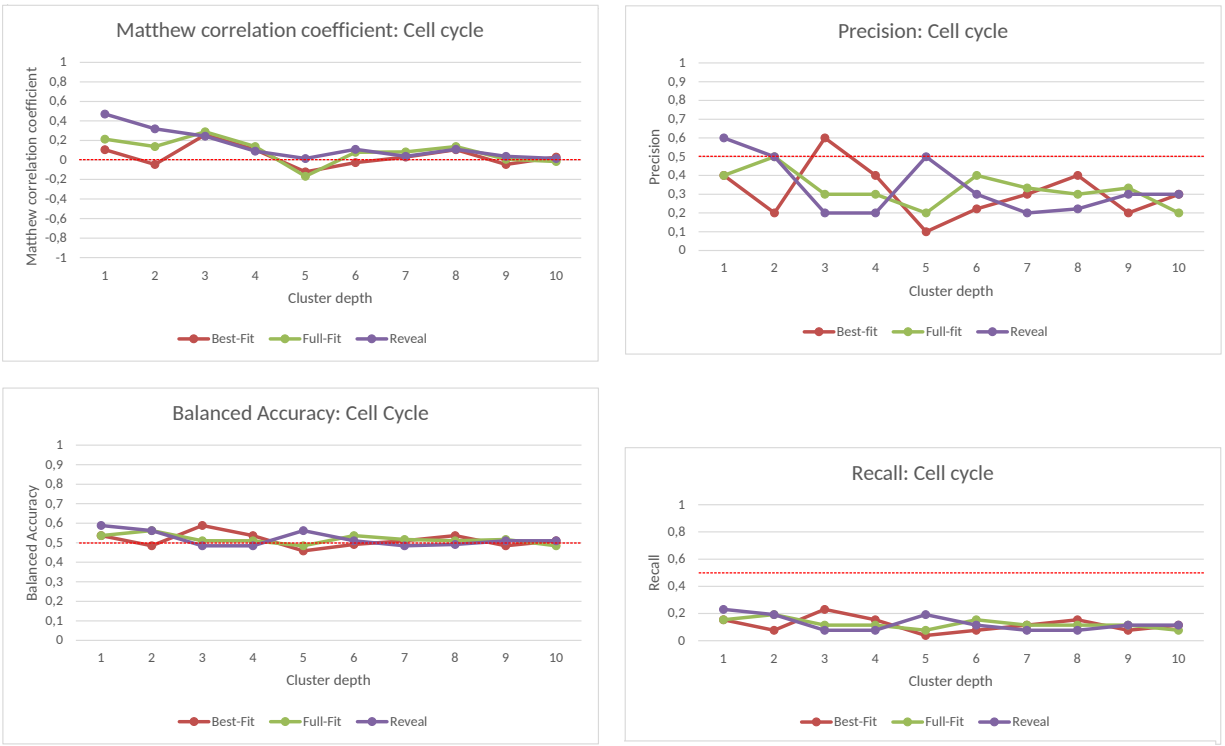

Figure 3.7: Performance considering cluster depth d

3.3 Pipeline of the Dream8 Challenge data set

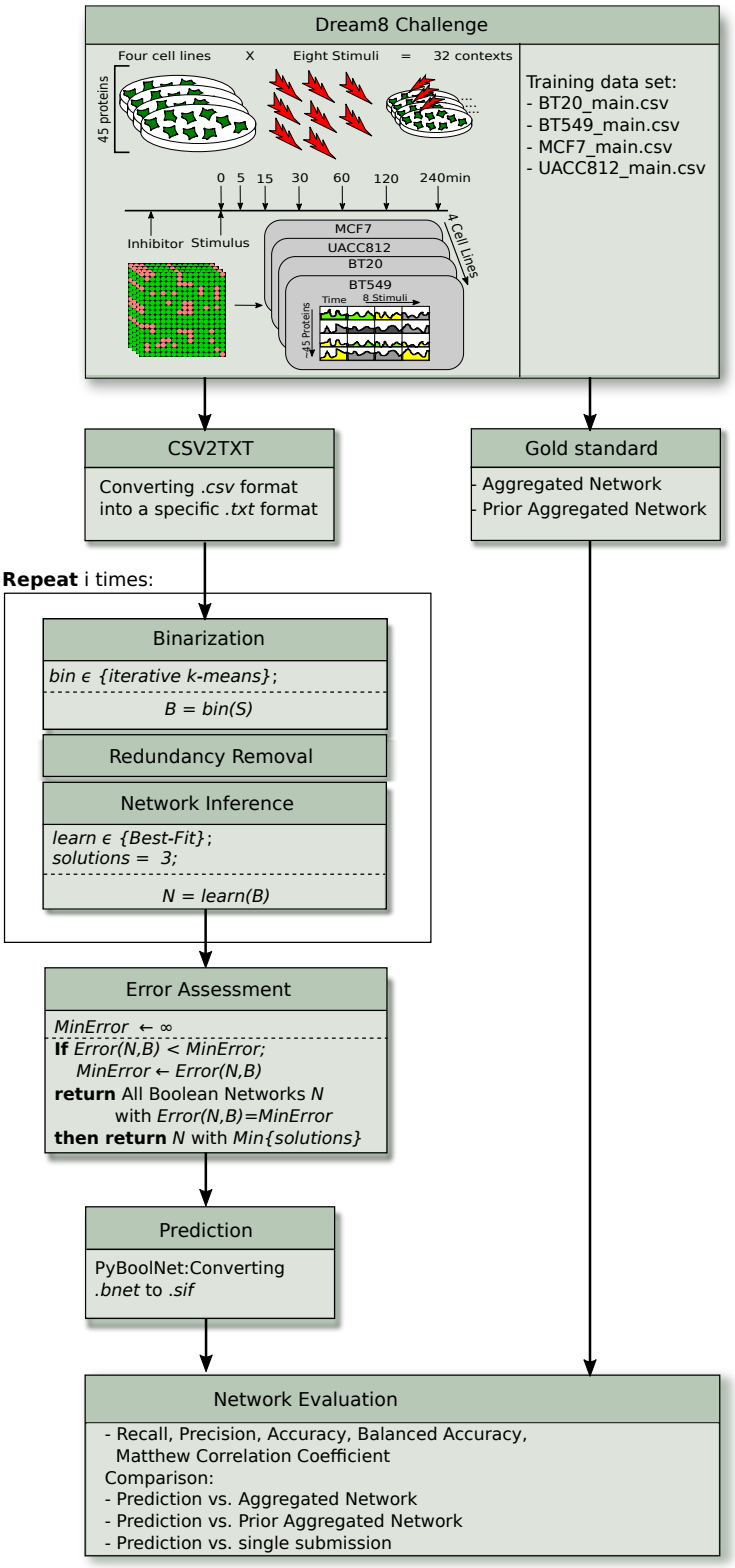

Figure 3.8: Pipeline Dream8 Challenge.

This section describes settings in the Dream8 Challenge pipeline which differ from the settings of the *in silico* pipeline. The 'main' training data set of the Dream8 Challenge is converted into a *txt* file format (Figure 2.8, Figure 3.1) by splitting each data set of a cell line into eight text files depending on the stimulus. Each of the resulting 32 text files contain about ∼85 sample points and about ∼45 antibody names (Figure 3.8).

Network inference

The higher the *in-degree* of the nodes and the amount of nodes in a network the higher is the complexity in a system and the higher is the computational cost of an inference algorithm. Due to computational limitations and a technical settings of 8 RAM and a Core i5 processor the application of Reveal to the Dream8 Challenge data set is an infeasible task (Figure 2.3). Furthermore, results of the *in silico* data set show that the choice of abundance of sample points (Figure 3.6) and cluster depth (Figure 3.7) has no significant influence on the performance. An increasing *in-degree* (Figure 3.4) shows an decreasing performance for all three algorithms, such that no algorithm can be excluded by this investigation. Thus, results of (Paper Quelle) regarding error assessment are taken into account, such that a recommanded combination of Best-Fit with the iterative *k-means* algorithm with $d = 3$ is applied to the Dream8 Challenge data set. This combination yielded the best performance, especially in complex systems.
Predicted Boolean networks (*bnet*) are converted into an Interaction graph (*sif*) (Figure 4.2) and scored against a gold standard (Aggregated Network, Prior Aggregated Network).

Gold standard selection in the Dream8 Challenge

Since gold standard networks are often based on literature, learning novel connections in a network is restricted. 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 by error assessment resulting in prior knowledge independent networks created with the training data set of the experimental data. It is useful to include prior knowledge, but this can come along with limitations, because it is difficult to truly mimic specific bilogical systems of interest.
Additionally an *in silico* data set covering main characteristics of the experimental setting, but without batch effects, is provided by the challenge. This *in silico* data set is a tool for assessing the performance of an algorithm in an idealized setting. Hence, the performance

of an algorithm excluding pertubational effects can be assessed. The *in silico* data set is generated from a nonlinear ordinary differential equations (ODE) model of the ERBB signaling pathway (ERBB:family of proteins containing four receptor tyrosine kinases, structually related to the epidermal grwoth factor receptor (EGFR)). Nevertheless pre-existing biological knowledge was included by several participants and seemed to be broadly benefical.

Evaluation in the Dream8 Challenge

Due to different strategies of evaluating predictions, e.g. choice of the scoring metric and implementation approach, it is hard to compare the resulting performance between the participants. For this reason the Dream8 Challenge provides a standard scoring tool 'DREAMTools' python package. This tool needs as input a *sif* and an *eda* file, where eda (electronic design automation) contains the confidence scores of edges in an Interaction Graph.
DREAMtools compares the confidence scores *eda* file of the prediction against the *eda* file of a gold standard by classifying the data by a threshold τ. 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 Positve 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.

Gold standard selection for the Prediction (Best-Fit)

This work makes use of an aggregated network for all 32 contexts and an aggregated prior network as 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. The aggregated prior network is a combination of 10 prior knowledge networks that participants used as part of their submission (Table 3.2). Predictions of all 74 participants of the Dream8 Challenge leaderboard and the

prediction in this thesis by Best-Fit are scored against the aggregated network and against the aggregated prior network, such that a new ranking reveals how Best-fit performs in relation to the participants' submission.

Network	# Edges	# Networks
Prediction (Best-Fit)	~ 140	1
Aggregated Network	~ 2200	66
Aggregated Prior Network	~ 1400	10

Table 3.2: Network structure

Evaluation for the Prediction (Best-Fit)

In contrast to the Dream8 Challenge the networks are scored by scoring metrics of recall, precision, accuracy, balanced accuracy and matthew correlation coefficient. 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 occurences 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.

3.4 Results of the Dream8 Challenge data set

Prediction versus Aggregated Network and Aggregated Prior Network

Scoring the prediction of the inference algorithm Best-Fit against the aggregated network and the aggregated prior network (Figure 3.9) shows that there is an imbalance in the classes. Especially regarding the accuracy of $\sim 15\%$ when the prediction is scored against the aggregated network. Taking the balanced accuracy improves the performane of up to $\sim 40\%$. The amount of edges differ in both networks a lot, such that the number of false negatives is tremendously high (Figure 3.2). The same imbalance is observed scoring the prediction against the aggregated prior network. Less edges of the aggregated prior network lead to a smaller improvement of the scoring values.
For emphasizing this observation the preiction is scored in Figure 3.10 instead against the aggregated prior network against a submission of the last participant (rank: 74.) of the Dream8 Challenge leaderboard.
For further analysis the accuracy is neglected, due to the imbalanced classes accuracy falsifies interpretation.

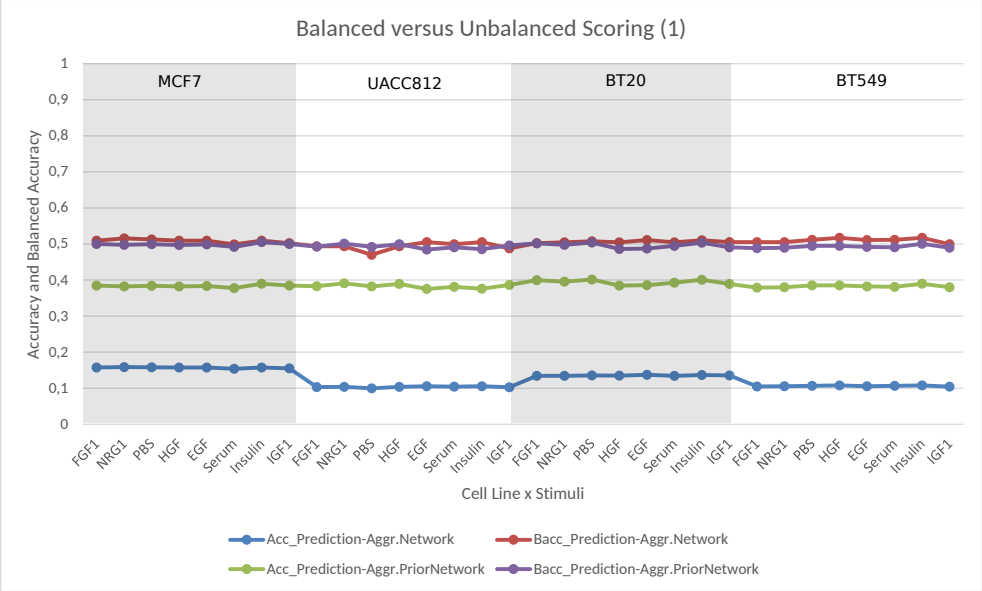

Figure 3.9: Imbalanced classes (1). Accuracy and Balanced Accuracy for the prediction scored against the aggregated network and aggregated prior network.

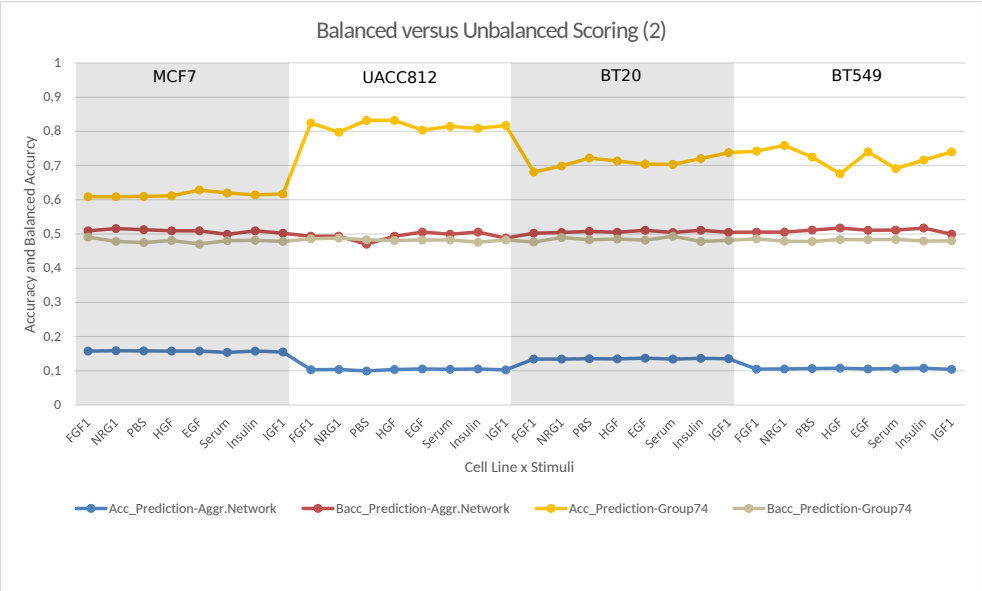

Figure 3.10: Imbalanced classes (2). Accuracy and Balanced Accuracy for the prediction scored against the aggregated network and aggregated prior network.

Figure 3.11 and Figure 3.12 show values of recall and precision for scoring the prediction against the aggregated network and the aggregated prior network. Recall ranges from $5,6\%$ to $7,7\%$ for the aggregated prior network and from $6,2\%$ to $7,5\%$ for the aggregated network. Both aggregated networks contain a lot more edges than the prediction, such that the number of true positive is much smaller than the number of false negatives. In contrast to precision, which is not taking false positives into account.

Hence, values for precision range for the aggregated network from $90,2\%$ to $97,8\%$ and for the aggregated prior network from $51,0\%$ to $66,6\%$. This is because the big size of the aggregated network covers most of the predicted edges, such that the number of false positives. This explaines why the precision of scoring against the aggregated prior network is worse.

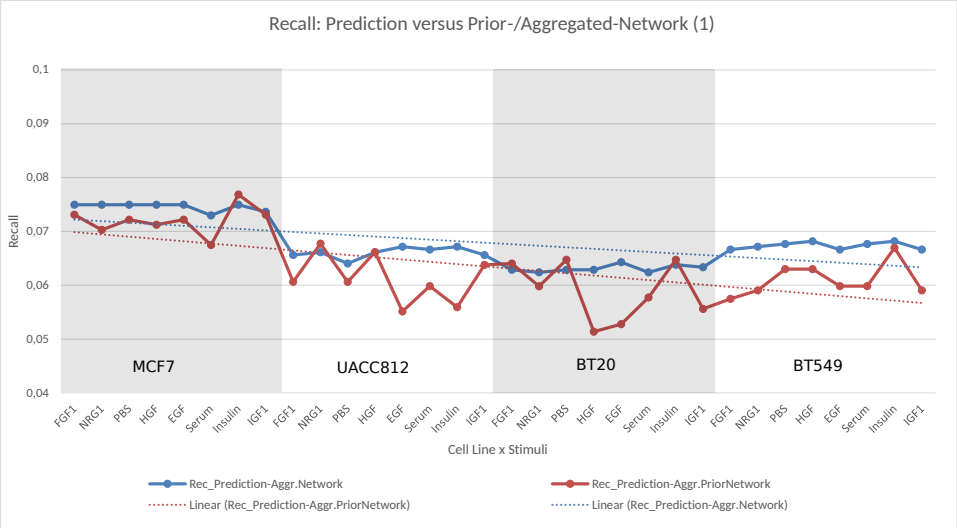

Figure 3.11: Recall: Prediction versus Aggregated/Prior Network

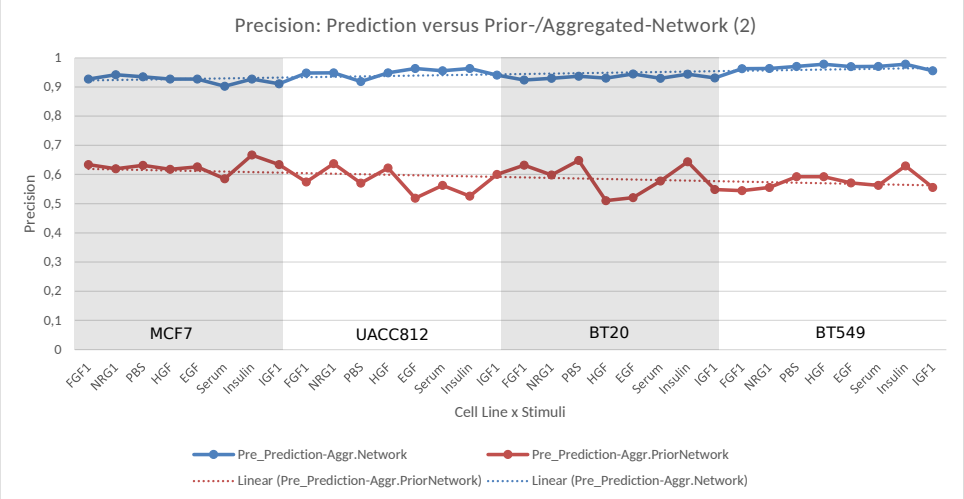

Figure 3.12: Precision: Prediction versus Aggregated/Prior Network

New Ranking: Aggregated Network and Aggregated Prior Network

In Figure 3.13-3.15 submitted networks of participants of the Dream8 Challenge are scored against the aggregated network and the aggregated prior network including the predicted network of Best-Fit. The networks are ranked by their value of balanced accuracy (BACC), Matthew correlation coefficient (MCC), precision and recall. This ranking results in a rank for the prediction depicted in Table 3.3 for each scoring case. It is noticed that scoring the prediction against the aggregated network yields a mean rank of ~ 38, which is much better than the mean rank of ~ 45 by scoring the prediction against the aggregated prior network.

The Matthew correlation coefficient and the balanced accuracy are both used to assess the performance when the classes are imbalanced. But the MCC yields a better ranking for the prediction in both cases, the aggregated network and the aggregated prior network than the balanced accuracy.

Scoring metric	**Type**	**Aggregated Network**	**Aggregated Prior Network**
MeanBACC	rank	43	49
MeanBACC	value	$\sim 0,0628$	$\sim 0,4950$
MeanMCC	rank	33	39
MeanMCC	value	$\sim 0,0097$	$\sim 0,0195$
MeanPrecision	rank	33	45
MeanPrecision	value	$\sim 0,9436$	$\sim 0,5909$
MeanRecall	rank	42	47
MeanRecall	value	$\sim 0,0677$	$\sim 0,0632$

Table 3.3: Ranking of the prediction

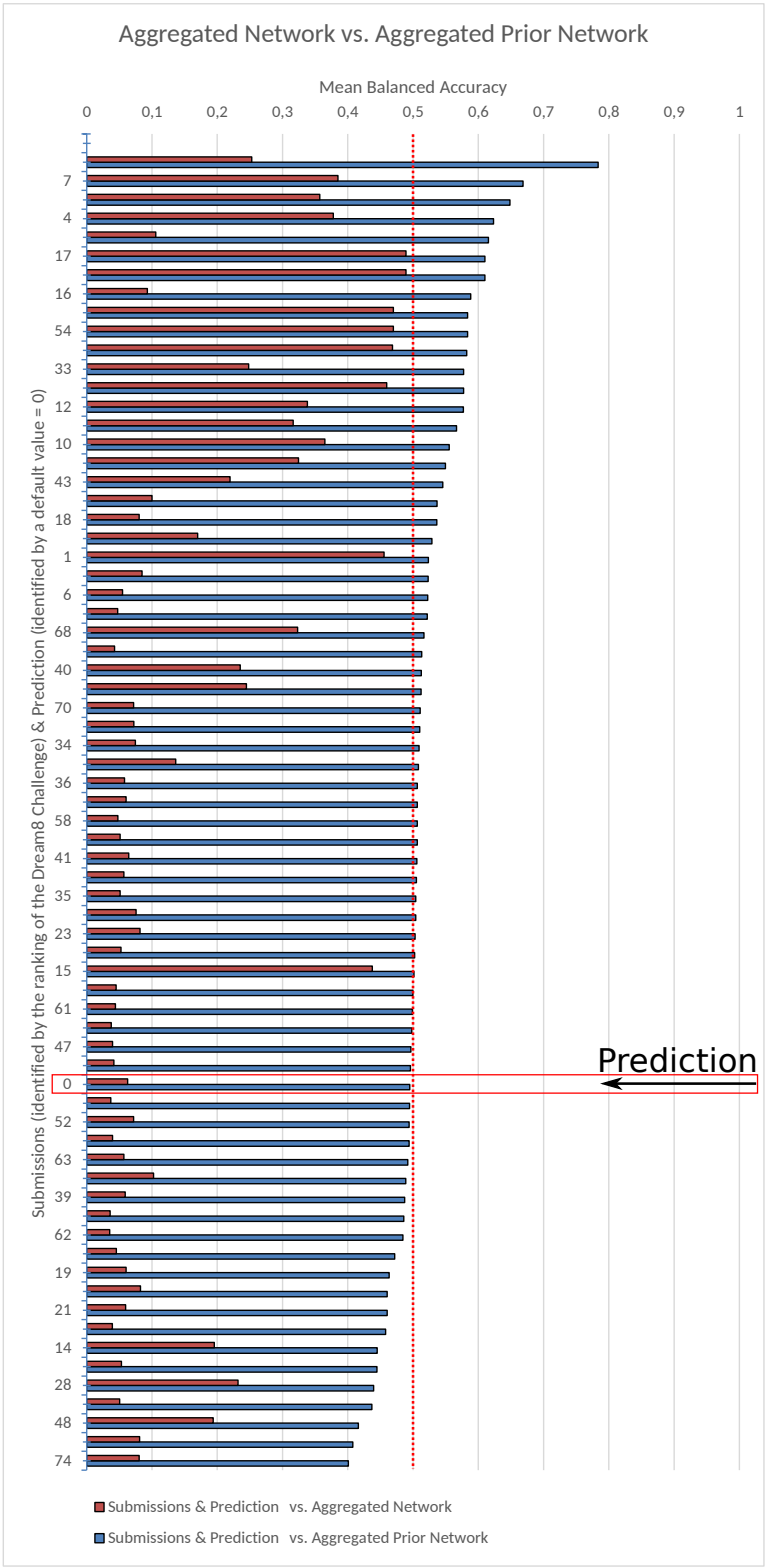

Figure 3.13: New Ranking (Balanced Accuracy): Aggr. Network and Aggr.Prior Network

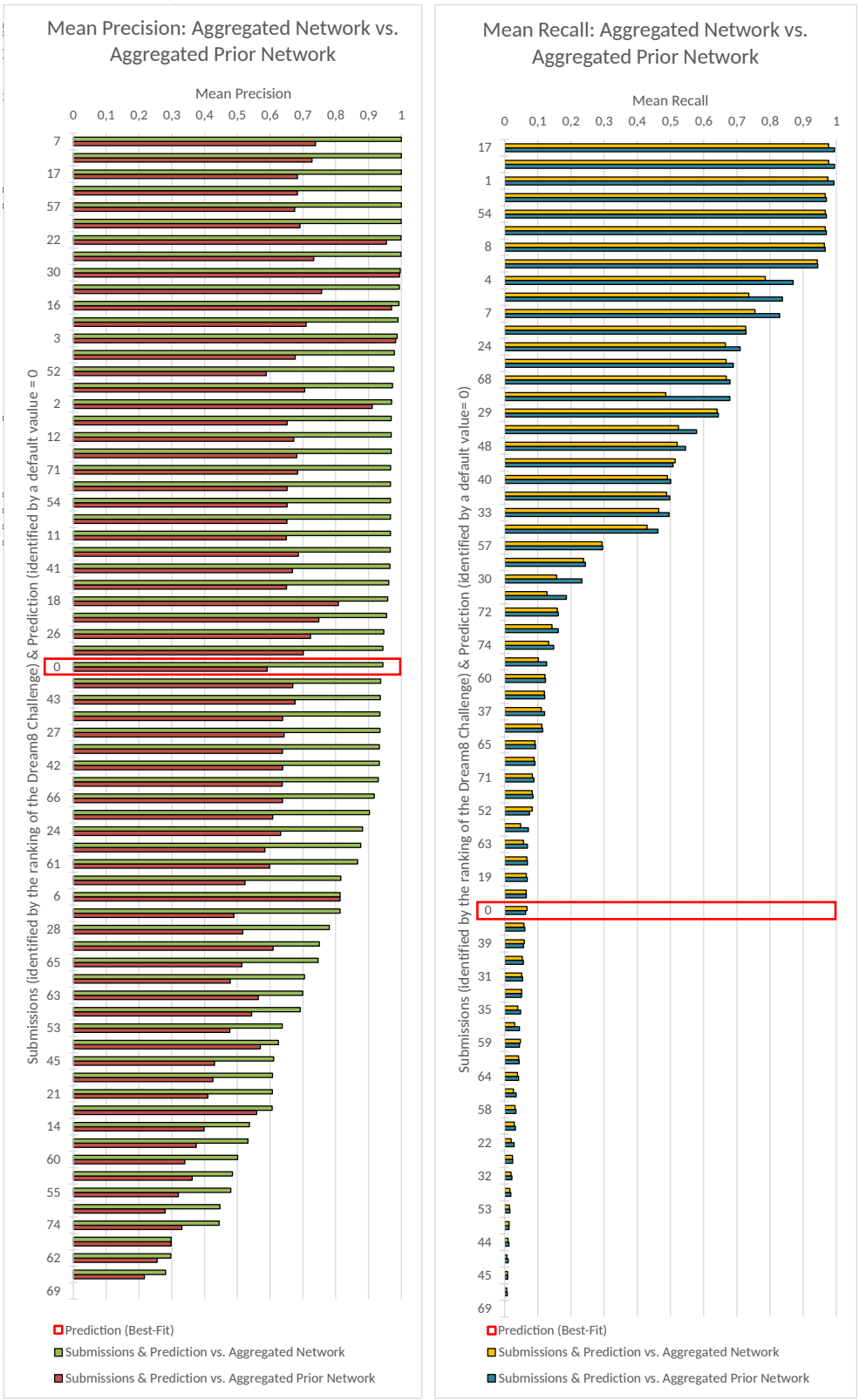

Figure 3.14: New Ranking (Precision and Recall): Aggr. Network and Aggr.Prior Network

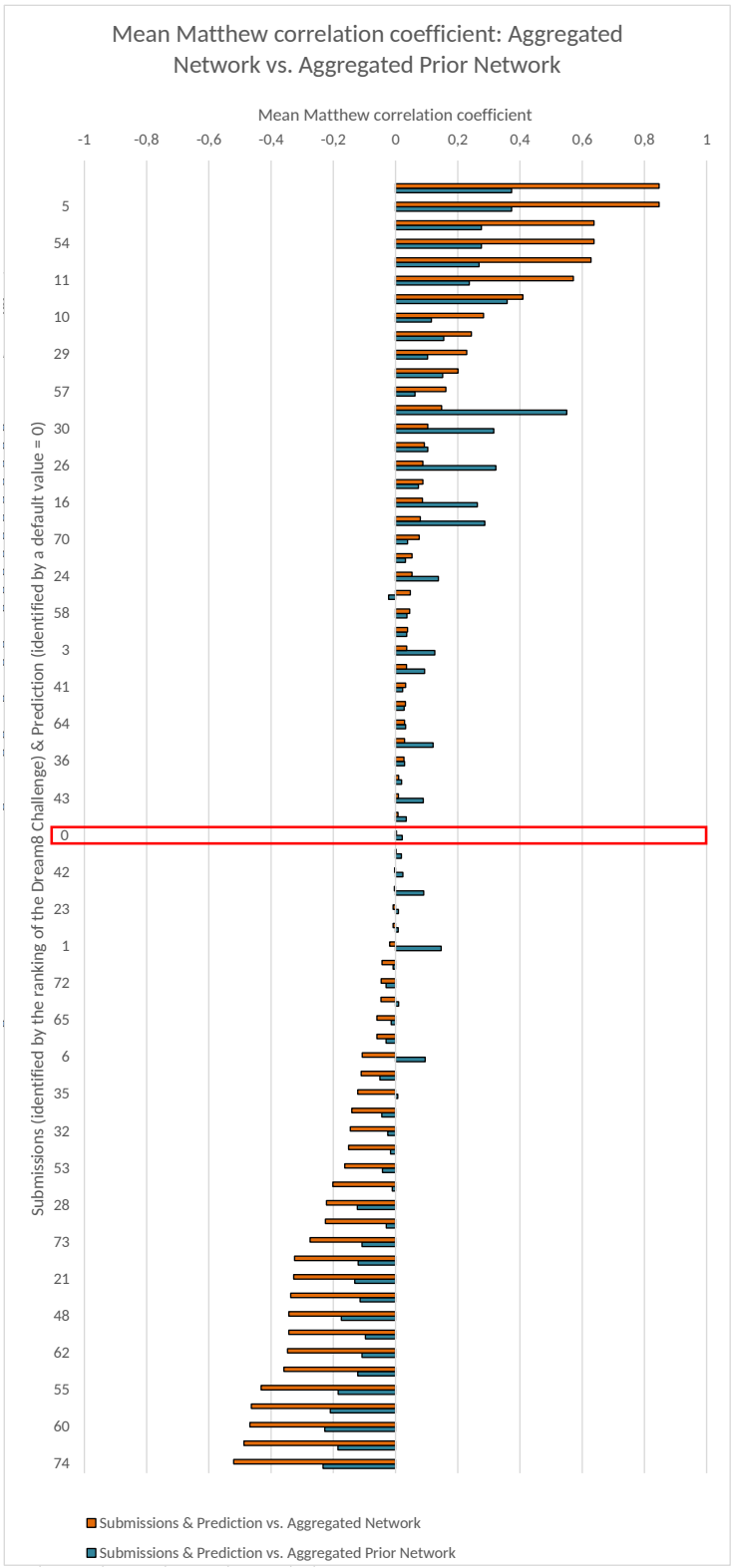

Figure 3.15: New Ranking (Matthew correlation coefficient): Aggr. Network and Aggr.Prior Network

4 Discussion

Bibliography

S. Barman and Y.-K. Kwon. A novel mutual information-based boolean network inference method from time-series gene expression data. *PLOS ONE*, 12(2):1–19, 02 2017. 10.1371/journal.pone.0171097. URL https://doi.org/10.1371/journal.pone.0171097.

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

H. A. Kestler, C. Wawra, B. Kracher, and M. Kuehl. Network modeling of signal transduction: establishing the global view. *BioEssays*, 30(11-12):1110–1125, 2008. ISSN 1521-1878. 10.1002/bies.20834. URL http://dx.doi.org/10.1002/bies.20834.

J. Krumsiek, S. Pölsterl, D. M. Wittmann, and F. J. Theis. Odefy - from discrete to continuous models. *BMC Bioinformatics*, 11(1):233, May 2010. ISSN 1471-2105. 10.1186/1471-2105-11-233. URL https://doi.org/10.1186/1471-2105-11-233.

D. of Energy Office of Science. Genomes to life program roadmap, April 2001. DOE/SC-0036.

A. Saadatpour and R. Albert. Boolean modeling of biological regulatory networks: A methodology tutorial. *Methods*, 62(1):3 – 12, 2013. ISSN 1046-2023. https://doi.org/10.1016/j.ymeth.2012.10.012. URL http://www.sciencedirect.com/science/article/pii/S1046202312002770. Modeling Gene Expression.