Universidade do Minho
Escola de Engenharia

David Saque Henriques

Network Inference for Logic-Based Ordinary Differential Equations

Janeiro de 2017

Network Inference for Logic-Based Ordinary Differential Equations

David Saque Henriques

Uminho | 2017

Universidade do Minho
Escola de Engenharia

David Saque Henriques

Network Inference for Logic-Based Ordinary Differential Equations

Dissertação de Doutoramento
Programa Doutoral em Informática
Departamento de Informática

Trabalho efetuado sob a orientação de:
Professor Miguel Rocha
Professor Julio R. Banga

Janeiro de 2017

DECLARAÇÃO DE INTEGRIDADE

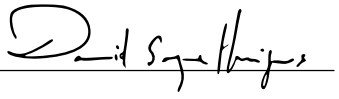
Declaro ter atuado com integridade na elaboração da presente Tese.

Confirmo que em todo o trabalho conducente à sua elaboração não recorri à prática de plágio ou qualquer forma de falsificação de resultados.

Mais declaro que tomei conhecimento integral do Código de Conduta Ética da Universidade do Minho.

Universidade do Minho, 10 de Janeiro de 2017.

Nome Completo: David Saque Henriques

Assinatura: 

Acknowledgements

I thank my family, friends and supervisors, all of them responsible for making this challenge a fun and productive experience.

I thank my family guidance and affection.

I thank my supervisors for giving me the freedom to think by myself while keeping me in the right track (otherwise I would never have submitted the thesis in due time).

Finally, I thank my friends for taking good care of me while living abroad.

The work presented in this thesis was funded by the EU FP7 project "NICHE" (Network for Integrated Cellular Homeostasis), Marie-Curie ITN Grant number 289384. Between March 2012 and March 2015 I was a Marie-Curie early stage researcher (ESR) at the IIM-CSIC in Vigo, Spain.

Abstract

Signaling is a highly dynamic and context specific process. When cells fail to interpret external stimuli from the environment or emitted by other cells the consequences can be disastrous. Mechanistic signaling models with predictive value have the potential to help developing new therapeutical strategies targeting molecules involved in signal transduction. However, the complexity of signaling networks, the nonlinear nature of these systems and several technological limitations regarding the ability to manipulate cells *in vitro* and measure post translational modifications experimentally, make the task of building quantitative models for signaling very difficult.

Many interactions in signaling pathways are known but, because they are not well characterized from the biochemical point of view, it is not straightforward to turn this information into a model. In this thesis, we present methods for reverse engineering mechanistic models combining data from cell-line perturbation experiments. Here, the model dynamics is described by means of logic-based ordinary differential equations, a recent formalism that through a set of reasonable assumptions describes regulatory mechanisms in a relatively simple, yet, dynamic and continuous manner. We formulate model selection and network inference as dynamic optimization problems, which are nonlinear non-convex and, thus very hard to solve.

Here, we formulate model selection as a mixed-integer dynamic optimization problem and solve it recurring to state of the art meta-heuristics for optimization and numerical methods for simulation. We apply the methods to several signaling case-studies and concluded the method scales up well. In addition, we develop a relaxation tailored for this problem that improves convergence in large problems.

The network inference problem is tackled with the help of mutual information and an ensemble approach. To compensate for the lack of prior knowledge, we build

data-driven networks based in mutual information. With the ensemble approach, we explore the landscape of possible models, providing more reliable predictions for trajectories and network inference. The method was applied to several *in silico* and experimental case studies including data from the HPN-DREAM Breast Cancer Network Inference challenge. We were able to generate predictions that were in some cases significantly better than those provided by the best performers.

To facilitate the implementation and redistribution of dynamic optimization problems in systems biology, such as those described above, we also develop a C library. This library is open-source and platform independent. The implementation and some applications of the library are discussed.

Building dynamic models of signaling with predictive power is possible despite of a number of well known pitfalls and limitations. The heavy computational cost of simulating ordinary differential equations models can be palliated by combining state of the art numerical methods with meta-heuristics and the power of cluster computing.

Resumo

A sinalização é um processo altamente dinâmico e que depende do contexto celular. Quando as células não estão aptas a interpretar estímulos ambientais ou emitidos por outras células, as consequências podem ser desastrosas. Os modelos quantitativos com valor preditivo têm o potencial para ajudar no desenvolvimento de novas estratégias terapêuticas. No entanto, a complexidade das redes de sinalização, a natureza não linear destes sistemas e diversas limitações tecnológicas na medição de modificações pós tradução tornam a tarefa de construir modelos para a sinalização muito difícil.

Muitas das interações entre proteínas nas vias de sinalização são conhecidas. No entanto muitas não estão bem caracterizadas do ponto de vista bioquímico e a transformação deste conhecimento qualitativo em modelos não é trivial. Nesta tese, apresentamos métodos para realizar engenharia reversa de modelos dinâmicos a partir de dados experimentais obtidos através da introdução de perturbações em culturas celulares. As dinâmicas são representadas através de equações diferenciais ordinárias.

Nesta tese, formulamos a seleção de modelos e a inferência de redes como problemas de otimização dinâmica. Estes problemas são não lineares e não convexos e portanto muito difíceis de resolver.

A seleção de modelos é formulada como um problema de otimização dinâmica inteira mista. Para resolver o problema, recorremos a meta-heurísticas. Este método foi aplicado a vários estudos de caso de sinalização e concluímos que o método se adapta bem a problemas com diferentes tamanhos. Para melhorar a convergência em problemas grandes desenvolvemos uma relaxação específica para esta formulação.

O problema de inferência de redes é abordado através da combinação de vários

modelos. Para compensar a falta de conhecimento prévio, construímos redes baseadas na informação mútua entre as variáveis do modelo. Com um conjunto de modelos conseguimos obter previsões mais robustas em relação aos modelos individuais. O método foi aplicado a vários estudos de caso *in silico* e experimentais, incluindo dados do HPN-DREAM Breast Cancer Network Inference Challenge. Fomos capazes de gerar previsões que em alguns casos, foram significativamente melhores que as dos vencedores do desafio.

Para facilitar a implementação e redistribuição de problemas de otimização dinâmica na área de biologia de sistemas, tais como os descritos anteriormente, desenvolvemos uma biblioteca em C. Esta biblioteca é distribuída em código aberto e independente de plataforma. A sua implementação e algumas aplicações são discutidas nesta tese.

A construção de modelos dinâmicos de sinalização com valor preditivo é possível, apesar de uma série de limitações bem descritas na literatura. O elevado custo computacional de simular modelos de equações diferenciais ordinárias pode ser atenuados através da combinação de métodos numéricos eficientes, utilização de meta-heurísticas e o poder de cálculo de supercomputadores.

Contents

List of Figures

List of Tables

Acronyms

ACOmi Ant-Colony for Mixed Integer.

AIC Akaike Information Criterion.

AMIGO Advanced Model Identification using Global Optimization.

ARACNE Algorithm for the Reconstruction of Accurate Cellular NEtworks.

ASP Answer Set Programming.

AUPR Area Under Precision Recall.

AUROC Area Under Receiving Operating Characteristic.

CLR Context Likelihood of Relatedness.

DDN Data-Driven Network.

DHC Dynamic Hill Climbing.

DREAM Dialogue for Reverse Engineering Assessments and Methods.

DREAMBT20 DREAM cell-line BT20.

DREAMBT549 DREAM cell-line BT549.

DREAMiS DREAM *in Silico*.

eSS enhanced Scatter Search.

FBA FBA.

FE Function Evaluation.

GRN Gene Regulatory Network.

hARACNE high-order Algorithm for the Reconstruction of Accurate Cellular NEtwork.

ILP Integer Linear Programming.

IVP Initial-Value Problem.

LP Linear Programming.

MAPK Mitogen-Activated Protein Kinase.

MAPKf Mitogen-Activated Protein Kinase full.

MAPKp Mitogen-Activated Protein Kinase partial.

MCMC Markov Chain Monte Carlo.

MEIGOR MEtaheuristics for bIoinformatics Global Optimization in R.

MEX Matlab EXecutable.

MI Mutual Information.

MI3 three-way Mutual Information.

MIDAS Minimum Information for Data Analysis in Systems biology.

MIDER Mutual Information Distance and Entropy Reduction.

MIDO Mixed-Integer Dynamic Optimization.

MIM Mutual Information Matrix.

MINET Mutual Information NETworks.

MINLP Mixed-Integer NonLinear Programming.

MIRIAM Minimal Information Required In the Annotation of Models.

MISQP Mixed-Integer Sequential Quadratic Programming.

MITS Mixed Integer Tabu Search.

MORE Mixed Optimization for Reverse Engineering.

MPeSS Multi-Phase enhanced Scatter Search.

MRNET Maximum Relevance minimum redundancy NETwork.

MRNETB Maximum Relevance minimum redundancy NETwork Backward.

NLP NonLinear Programming.

ODE Ordinary Differential Equation.

PKN Prior Knowledge Network.

PLA Profile Likelihood Analysis.

PSN Protein Signaling Network.

PTM Post-Translational Modification.

RHS Right-Hand Side.

RMSE Root Mean Squared Error.

ROC Receiving Operating Characteristic.

RPPA reverse phase protein array.

SBML Systems Biology Markup Language.

SELDOM enSEmbLe of Dynamic LOgic Models.

SSP Synthetic Signaling Pathway.

TDARACNE Time-Delay Algorithm for the Reconstruction of Accurate Cellular NEtworks.

Chapter 1

Introduction

1.1 Motivation

In complex organisms, signaling pathways play a critical role in the behavior of individual cells and ultimately in the organism as a whole. Cells adapt to the environmental conditions through the integration of signals released by other cells via endocrine or paracrine secretion or other environmental stimuli. The cell decisions to replicate, differentiate or die (apoptosis) are largely controlled by these signals [3].

Many of the interactions involved in signaling are commonly grouped in pathways. Pathways are typically depicted as sequences of steps where the information is relayed upon activation by an extracellular receptor promoting several downstream Post-Translational Modifications (PTMs), which will ultimately end by modifying gene expression or some other effector. These interactions are non static in the sense that the behavior of such pathways is known to be highly dependent on the cell type and context [79], which may change with time [96]. Additionally, many of these pathways interact with each other in ways that are often described as analog to a decision making process [64].

Signaling is a very dynamic and fast process specially if compared with gene expression. In order to build a mechanistic model, given a cell type or tissue, one should have data obtained from perturbation experiments, *i.e.* the system assumed to be homeostatic is perturbed with chemicals to which the cell may or not

react and the variations in the cell chemistry are recorded. When only the initial and final state are monitored, one is typically bound to assume that the system has shifted back to a different homeostatic state. The remark to this strategy is that such assumption might not be always true. Also important information is contained in the system dynamics acting at different time scales.

Many of the interactions in this network are known, some even quite well characterized from the biochemical point of view (some examples of this are the Mitogen-Activated Protein Kinase (MAPK) [114], NF$k\beta$ [75], JAK-STAT [120] signaling pathways). However, this network is most certainly incomplete [158] it is hypothesized that most *in vivo* phosphorylation sites have not yet been discovered.

There are at least three good reasons to infer a dynamic model of a signaling pathway. The first, and perhaps most obvious one, is to find novel interactions. The second is what we will refer throughout the rest of this thesis as the model selection task. Model selection can be defined as the process of using data to select (or exclude) a number of model features which are consistent with the current knowledge about a given system. This is particularly relevant in the case of cell lines, as different interactions will actively depend on the cell context. The third one is the usage of such a model to predict how the system will behave in new conditions that have not been tested before.

Despite enormous progress in high-throughput technologies and modeling efforts [6,158] the fact is model inference is not a solved problem. The ability to grow and perturb theses systems (individual cells, cell cultures or tissues) and quantify all the involved molecular states in a precise and well-resolved time and or space manner are important limiting factors.

All experimental data used in this work was obtained by measuring phosphorylation variations in several proteins, after perturbation of one or more cell lines using anti-body based methods (for a review on the different approaches to measure phosphoproteomic signals that can be used to model signaling networks we direct the reader to [158]). Phosphorylation is not the only relevant PTM for signaling regulation, however, because there are well established experimental methods to measure it, its prevalence and also because it affects other signaling subsystems [112] based in different PTMs it is often used to study pathways in a systematic manner.

Many years of basic molecular biology research have provided a reasonable picture of how many of these parts interact in a individual manner and this information is fairly accessible and well summarized in a number of data-bases [163]. Thus, an important part of the effort to develop mechanistic models of signaling transduction pathways focus in the combination of existing, yet highly context specific, knowledge with experimental data, modeling frameworks and reasonable assumptions (e.g. steady-state).

Although diseases like cancer are ultimately caused by mutations at the genome level, the end result is that abnormalities at the signaling level appear and cells fail to take decisions correctly. It is not strange that a large group of available targeted therapies are based on molecules that disrupt signaling, like kinase inhibitors or monoclonal antibodies which block growth factor receptors on the cell surface [119]. Understanding how this large number of parts is connected is important but not sufficient because mammalian signaling is highly dynamic and context specific. Thus mechanistic models are important to understand cell behavior, and to ultimately take part in the process of designing new drugs or treatment strategies (drug combinations, drug scheduling, etc.).

1.2 Problem Statement

In this thesis, all models used will be represented as Ordinary Differential Equations (ODEs). We assume that the experimental measurements used are from an average from many cells and that the ODEs can represent the behavior of signaling processes from cultures of a given cell-line with reasonable accuracy. Without prejudice that some of the methods described in this thesis can be extended (at least in part) to other applications, in this work we will focus mostly in handling high-throughput phosphoproteomics data from cell-lines of human signaling pathways.

We solve implicitly three classes of problems: *i*) parametric identification, *ii*) model selection and *iii*) network inference. What separates the different problems is the level of available prior knowledge. All three tasks will be handled in this thesis from an inverse problem point of view: given experimental data, we want to find a solution (or a set of solutions) that can explain the behavior of the

biological system subject to a number of assumptions/constraints (derived from prior knowledge). This can be described as:

$$\begin{aligned} &\underset{\theta}{\text{minimize}} \quad F(\tilde{x}, x) \\ &\text{subject to} \\ &\qquad x = \int_{t_0}^{t_f} \dot{x}(\theta, x)\ \mathrm{d}t, \end{aligned} \tag{1.1}$$

where F is a function of the experimental data (x) and the dynamic model output $\tilde{x}$) which quantifies model quality. The Right-Hand Side (RHS) equations ($\dot{x}$) are ODEs that depend on an set parameters (θ) and are integrated numerically.

For imposing the previously mentioned set of dynamic constraints, we will rely in the framework developed by Wittmann et al. [172], called logic-based ODE. Because signaling is a fast process, compared with gene expression, it is assumed that overall protein concentrations remain constant and the ODEs represent only changes in the PTMs, typically phosphorylation. The logic-based ODE model will be explained with greater detail in Section 2.1.1. The gold standard for building quantitative dynamic models in biology is the usage of mass action or some other associated enzyme kinetics (these are derived from mass action). However, this requires accurate knowledge on the biochemistry and is not compatible with the incomplete nature of qualitative networks available to derive our dynamic models.

An important aspect to keep in mind while working with cell-line data is that samples are in fact a lysate of many cells which can, for example, be at different states of the cell cycle and thus, these lysates might not entirely reflect the state of single cells and effects such as signal cancellation might exist [124]. Cell-lines are a convenient, yet not ideal, *in vitro* model for signaling related diseases.

In the parametric inference problem the model structure is assumed to be known and correct at least in the sense that the ODEs can explain with an error equal or inferior to the expected noise in the data.

In the context of this work, the inputs to this problem are a Boolean network[1], a data-set consisting of PTMs in proteins measured at different time points and

[1]In a Boolean network direction, sign (activation) and type of interaction/logic gate (AND, OR, etc.) are known.

perturbation experiments and a experimental description of the problem[2]. Here, the so-called perturbations should be understood as small-molecule inhibitors and ligands (e.g. hormones) and are introduced by means of small molecule inhibitors which are introduced in the model as control variables.

The numerical procedure to simulate these systems is often referred to as the Initial-Value Problem (IVP) and requires that the initial conditions are provided. However, it is often the case that due to experimental limitations some of the modeled proteins cannot be measured. Given this case, initial conditions can also be estimated.

When solving this problem, three very important aspects have to be kept in mind. The first is that the problem is nonlinear and non-convex [12]. There is no method in current literature that is able to solve problems of arbitrary size with guaranteed solutions in a reasonable amount of time. The second is that the problems are ill-posed due to so-called called practical[3] and structural limitations[4]. Finally, even if a solution can be uniquely determined, limitations due to the non-linearity of the problem arise, since often small changes in the data cause large variations in the estimated parameter values (this characteristic is typically referred to as ill-conditioning).

The model selection problem is more general than the parameter estimation. Generally, the goal is to use the experimental data to discriminate a set of hypothesis consistent with the available knowledge about a system [160]. As previously mentioned the behavior of pathways from each cell-line is highly dependent on the expression profiles. Thus, even if the possible interactions are well characterized the signals are relayed differently depending on the cell type.

The inputs to this problem are a Protein Signaling Network (PSN)[5], a dataset consisting of measurements from PTMs in proteins measured at different time points obtained upon multiple perturbations. In this problem, PSNs are an important source of information. However, these can not be used to generate a predictive

[2]The Boolean gates are not generally known from literature only sign and interaction.

[3]Unique solutions for the parameters cannot be located because the data does not contain the necessary information or is corrupted with to much noise.

[4]Unique solutions for the parameters cannot be located because of deficiencies in the model structure or poor choice of selected observed.

[5]A protein signaling network is a directed graph where edges are directed and the sign (activation) is known.

mechanistic model model in a straightforward manner [135].

Here, it is assumed that the network provided is correct in the sense that it should contain at least all the interactions necessary to explain the data. All potential pitfalls from parameter estimation problems apply here (if one assumes that the parameters are not known). Besides the potential lack of identifiability on the parameters, it is also possible that many model structures can explain the data equally or similarly well. Here, the rule of thumb is to apply Occam's Razor principle, *i.e.* choose the simplest solution possible that explains the data.

Finally the most general problem is to recover the network topology[6]. However in order to build a dynamic model from this graph we also need to specify the type of functional interaction and the parameters defining the quantitative behavior of the interactions. In this context the network inference can be seen as a sequence of model selection problems.

1.3 Objectives

In this thesis the main goal was to develop and apply methods for reverse engineering of signaling networks from experimental data. The focus was put in models developed using the formalism of logic-based ODEs, which represents systems in a dynamic manner in cases where the stoichiometry and underlying biochemical mechanism are unknown. To tackle this problem we identified a number of scientific/technical objectives:

1. To formulate the problem (reverse engineering) as a model selection one and solve it using an optimization framework in the form a non-linear programming or a Mixed-Integer NonLinear Programming (MINLP) problem.

2. Address cases where prior knowledge is incomplete[7] or unavailable.

3. Address the lack of identifiability and improve the predictive skills of the models.

[6] A direct graph that establishes causality

[7] Note that incomplete is meant here in a different sense that cell-type or context specific. A typical Prior Knowledge Network (PKN) is considered to be complete but part of the interactions might or not be present in the real biological system in a context specific manner.

4. Develop software tools to facilitate the usage of dynamic models with control engineering principles.

1.4 Thesis organization and outline

In the Introduction, we started by motivating and describing the problem of inferring signaling networks from phosphorylation measurements of cell-lines upon perturbation.

In Chapter 2, we will review the systems biology literature and show how similar problems have been addressed by other authors. It should become clear why our problem is in many aspects different from similar problems in the literature which explains why we applied different optimization strategies to solve these problems. Chapter 2 will be finished by a brief review of the tools used to solve nonlinear and MINLP problems.

In Chapter 3, we present an approach to reverse engineer logic-based ODE models from experimental data. We formulate the problem as MINLP. The methods are applied to two *in silico* and one experimental case study and we were able to scale up to a very reasonable problem size. A detailed analysis of performance is shown using a number of optimization tools. In addition to the already existing MINLP methods, we developed a relaxation[8] specific for this problem which improved convergence for larger problems.

In Chapter 4, we address the problem of inferring a dynamic model from experimental data in a purely data-driven manner. In this case, we were particularly interested in the predictive skill of the model. This can be seen as an extension to the work presented in Chapter 3. However, as no prior-knowledge was available to constrain the size of the problem, we had to think of alternative data-driven ways of doing this. The solution found was to use mutual information to derive an alternative to the PKNs which we call Data-Driven Networks (DDNs). These networks are much denser than necessary and, thus, we combine our method with model reduction techniques. Additionally, many different solutions gave similar results in terms of descriptive power. To improve predictions about untested experimental

[8]The relaxation consists in transforming the more complex problem into an approximated yet simpler to solve problem.

conditions, we build a so called ensemble model that consists in combining the predictions from individual models.

In Chapter 5 we describe a library to facilitate the implementation of dynamic optimization problems in C. This library was used as means to accelerate the implementation of three different problems; these are briefly discussed. The necessary inputs to work with this library, its structure and performance gains obtained by parallelizing certain tasks are also discussed.

In the last chapter, we present the conclusions reached during the development of this thesis and a critical perspective on the achieved results, limitations of the techniques and directions for future work.

Chapter 2

State-of-the-art

2.1 Logic models in Systems Biology

Systems Biology combines methods from biology with methods from mathematics, physics, computer science and engineering to describe and model biological systems [142] and arises from the need to summarize biological knowledge in a systematic and holistic way. The idea is to understand systems at a global way and not merely as the sum of the behaviour of their parts.

An important task in the systems biology field is the model building cycle. Models are useful to deliver quick and non expensive testable predictions, which are useful in several applications like developing new therapies [4] or optimizing mutant strains used to produce metabolites of industrial interest [129]. Perhaps an even more important feature of models is that they allow researchers to pose new questions and help in the reasoning process using computational tools, before performing laborious and often expensive procedures in a laboratory.

A corollary example of how these model formalisms can complement each other is the work by Covert and Palsson [32], where regulatory and signalling networks are combined with FBA (FBA). Here, Boolean logic is used for the regulatory layer, being combined with FBA used for simulating the central carbon metabolism and with ODEs for a detailed model of carbohydrate uptake control.

In 2012, such efforts culminated in the first whole-cell computational model on an organism describing the life cycle of the human pathogen *Mycoplasma gen-*

italium, in an effort that includes sub-models for all of its molecular components and their interactions. This tremendous achievement required the integration of 28 submodels and several model formalisms [83].

Despite these efforts, it is imperative to acknowledge that it is yet too soon to aim for an ODE-based whole cell model. However, interesting work by Kotte et. al. [89] was recently published showing a medium scale *E. coli* model (comparing to other state of the art ODE models in systems biology), which integrates aspects of regulation with the central carbon metabolism, hypothesizing that metabolic fluxes might actually have a much more important role in governing the cell homoeostasis than what was previously thought.

Although these works are of indisputable merit, one major concern after inspecting the model implementations is that those are typically built in a rather *ad-hoc* manner, which makes the implementation of such integrated approaches a very laborious task.

In the last years, there has been a growing interest in the application of logic formalisms to systems biology. A recent paper 'Logic modelling and the ridiculome under the rug' [20] points out the limitations of sheerly relying on omics data which is neither complete or fully correct and highlights the importance of using logic models to complement existing information available in these data-bases.

Logic models were first introduced by by Kauffman in 1969 to model gene regulatory networks [85]. Since then, diverse adaptations from the original formalism and methods to reverse engineer these models have been proposed. One example of early reverse engineering algorithms is the work by Akutsu et. al. which proposed a brute force approach which reverse engineers the Boolean function of only a few top k regulators in a node by node fashion [1]. Gradually, these methods evolved to accommodate continuous values (see [4], [21] and [90]) and to treat these networks in global manner (instead of fitting Boolean functions node by node) borrowing ideas from optimization and machine learning to avoid excessive model complexity [21] [135].

Saez-Rodriguez and colleagues have introduced methods for reverse engineering Boolean networks using sources of prior knowledge, such as PSNs [135], directed graphs that can be obtained from public repositories of manually curated networks, including KEGG, WikiPathways, Nature Pathway Interaction Database and Re-

actome [97]. The point here is that although these curated large-scale PSNs are useful in exploring complex biochemical pathways, they do not reveal how pathways respond to specific stimuli. Also, accumulation of molecular detail *per se* does not automatically yield an improved understanding of the ways in which signalling circuits process complementary and opposing inputs to control diverse physiological responses [135].

In this work, the authors used a reverse engineering approach, in which the original PSN was expanded to an hypergraph where all the possible logic gates were represented and then used a meta-heuristic optimization strategy (in this case, a genetic algorithm) to find which networks could best reproduce the data with the smallest number of hyperedges. The basis of this model formalism lies in the assumption that cells process information of certain stimuli by means of logic decisions.

The software CellNOptR [158] implements the Boolean logic and related formalisms and is designed to reverse engineer Boolean models, mainly in a protein signalling environment, given data from perturbation experiments. In a recent review by Macnamara et. al. [97], the different formalisms are explained in detail and examples are provided, being the main difference the way time is handled. Amongst these formalisms, the one which is most suited to handle time series in a precise manner are logic-based ODEs where the main idea is to transform the logic model into a continuous homologue in the form of ODEs.

Logic based-ODEs, in the form we will use in this thesis (i.e. multivariate polynomial interpolation), were first introduced by Wittman et.al accompanied with a software tool called Odefy [90]. Although, to the best of our knowledge, this is the most successful method for converting logic-models into ODEs, other authors have developed similar methods (see [103] and [41]), where the main disadvantage is that these formalisms are not able to represent all the types of logic gates.

The major advantage of using this formalism is that no information about the biochemistry (e.g. stoichiometry or type of kinetics) is needed. On one hand, we can use this formalism to represent the same type of mechanistic insight provided by Boolean logic models and, on the other, we get a model of differential equations which allows us to do accurate dynamic simulations for the state variables trajectories.

Nevertheless, there are several disadvantages when compared with other purely qualitative approaches like Boolean logic models. The multivariate polynomial interpolation method generates a large number of free parameters, which have to be estimated. Obtaining accurate estimates for these parameters is far from being a trivial task. As opposed to other kinetic models, there is no biochemical information about the parameter values. In previous work [65] (under the scope of a master thesis done by the author), we have addressed this problem by using optimization meta-heuristics, like scatter search, combined with local methods [48].

2.1.1 The Logic Based Ordinary Differential Equation Formalism

Boolean logic models describe the flow of information inside the cell by means of discrete states that can assume either the values 0 or 1. Each state i is, therefore, represented by a binary state that is systematically updated according to a Boolean function $B_i(x_{i1}, x_{i2}, ..., x_{iN})$, applied to another binary state serving as an input to the specified function.

This model formalism assumes cells process information by means of logic decisions, an approximation that is known to be accurate in some cases. For instance, if a specific protein is to be phosphorylated in two specific sites by different kinases, this can be modelled as a logic conjunction (AND gate). On the other hand, if two different kinases can bind to the same site activating the propagation of a certain downstream signal independently this can be regarded as a logic disjunction (OR gate). Furthermore, if a signal inhibits the propagation of another one, this can be depicted as a negation (NOT gate).

Every possible Boolean function can be represented by means of a truth table. Such tables represent the input/output relationship of specific Boolean functions. For instance, the AND, OR and NOT gates would be represented according to the Tables 2.1, 2.2 and 2.3. Additionally, to represent every possible truth table and, therefore, every possible Boolean function, only these three gates are necessary. A graphical representation for these gates is shown with the standard symbols in Figures 2.1(a) to 2.1(c) and in a hypergraph form in Figures 2.1(d) to 2.1(f).

To represent Boolean functions, it is common to make use of Boolean algebra.

x_1	x_2	Y
0	0	0
0	1	0
1	0	0
1	1	1

Table 2.1: Truth table for the AND gate

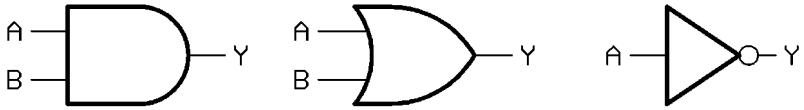
x_1	x_2	Y
0	0	0
0	1	1
1	0	1
1	1	1

Table 2.2: Truth table for the OR gate

x_1	Y
0	1
1	0

Table 2.3: Truth table for the NOT gate

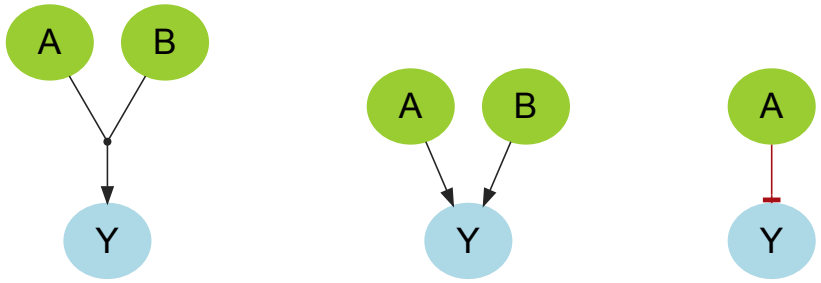
Boolean algebra is a form of symbolic logic that shows how logic gates operate [159]. The Boolean expression $\bar{A} \cdot B + C = Y$ would be read as: (A negated AND B) OR C equals the output Y.



(a) The AND gate represented as digital circuit.

(b) The OR gate represented as digital circuit.

(c) The NOT gate represented as digital circuit.



(d) A AND B activate Y. The AND gate is represent by an hyperdge.

(e) A OR B activate Y. The OR gate is represent by two edges.

(f) The NOT gate is represent by an edge with negative sign.

Figure 2.1: Representation of the different gates

Moreover, there are two canonical forms to represent truth tables with Boolean Algebra, the Sum of Products (SoP) and the Product of Sums (PoS). In SoP

Boolean algebra, expressions are represented by the so-called miniterms (products) composed by logic variables that can appear only once in each product and may be negated or not. The previously used example $\bar{A}\cdot B + C = Y$ is an example of an SoP.

To describe Boolean logic models, CellNOptR uses a graph structure (incidence matrix), where hyperedges represent miniterms (products) and simple edges describe sums. In biological terms, each miniterm and therefore each edge, describes a reaction where a set of conditions must be fulfilled to allow or block the propagation of a downstream signal. The incidence matrix records the target and direction of the reaction. Nevertheless, to mark the presence of negations, another matrix is required [88]. To illustrate this, the example $\bar{A}\cdot B + C = Y$ is represented in the Tables 2.4 and 2.5.

	Reaction 1	Reaction 2
A	-1	0
B	-1	0
C	0	-1
Y	1	1

Table 2.4: Hypergraph representation of $\bar{A}\cdot B + C = Y$. The hyperedge equivalent to reaction 1 goes from A (-1) and B (-1) to Y(1). The edge equivalent to reaction 2 goes from C to Y.

	Reaction 1	Reaction 2
A	1	0
B	0	0
C	0	0
Y	0	0

Table 2.5: Representation of logic negations in expression $\bar{A}\cdot B + C = Y$. The negation in reaction 1 ($\bar{A}$) is encoded with the 1 value.

The idea in logic-based ODE models is to convert each Boolean update function (B_i) into a continuous homologue ($\bar{B}_i$), where the species x_i is allowed to take continuous values between 0 and 1, $x_i \in [0, 1]$, and its temporal behaviour is described by:

$$\dot{\bar{x}}_i = \frac{1}{\tau}\cdot (\bar{B}_i(\bar{x}_{i1}, \bar{x}_{i2}, ..., \bar{x}_{ij}) - \bar{x}_i) \tag{2.1}$$

where τ_i can be interpreted as the life-time of the species x_i and species $\bar{x}_{ij}$ is a regulator of $\bar{x}_i$.

In order to achieve a continuous homologue, Krumsiek et al. [90] introduce HillCubes. These functions are based on multivariate polynomial interpolation

and incorporate Hill kinetics, which are known to provide a good generalized approximation of the synergistic dynamics of gene regulation .

To obtain HillCubes, a first transformation method is required to reach a continuous homologue from the Boolean update function. Tables 2.6 and 2.7 provide an example on how an AND and an OR gate, respectively, would be transformed into so-called BoolCubes, which are obtained by multi-linearly interpolating the Boolean update function:

$$\bar{B}_i^I(x_1, x_2, ..., x_N) = \sum_{x_1=0}^{1} \sum_{x_1=0}^{1} \sum_{x_N=0}^{1} \left[B_i(x_1, x_2, ..., x_N) \cdot \prod_{i=1}^{N} \left(x_i \bar{x}_i + [1 - x_i][1 - \bar{x}_i] \right) \right] \tag{2.2}$$

x_1	x_2	B_i	$\bar{B}_i^I = \sum$
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	$x_1 \cdot x_2$

Table 2.6: Multivariate polynomial interpolation of an AND gate

Although BooleCubes are accurate in the sense that B_i and $\bar{B}_i^I$ agree with the vertices of the unitary cube [90], they fail to represent the typical sigmoid shape switch-like behavior, often present in molecular interactions. The second transformation method is the introduction of Hill functions to achieve this goal:

$$f^H(\bar{x}_i) = \frac{{\bar{x}_i}^n}{\bar{x}_i^n + k^n} \tag{2.3}$$

x_1	x_2	B_i	$\bar{B}_i^I = \sum$
0	0	0	0
0	1	1	$x_1 \cdot (1 - x_2)$
1	0	1	$(1 - x_1) \cdot x_2$
1	1	1	$x_1 \cdot x_2$

Table 2.7: Multivariate polynomial interpolation of an OR gate

The coefficient n is a measure of cooperation of the interaction, since it determines the slope of the curve, while the parameter k sets the threshold where the activation is half maximal [172].

Since HillCubes never assume the value 1 (but instead approach it asymptotically), these are not accurate and, therefore, not perfect homologues. A simple solution for this is to normalize Hill functions to the unit interval:

$$f^{Hn}(\bar{x}_i) = \frac{f^H(\bar{x}_i)}{f^H(1)} \tag{2.4}$$

In Figure 2.2, it is possible to see how the normalized Hill functions vary according to the parameters n and k and the input value x. A further discussion about continuous homologues and the methodology to obtain logic-based ODE models can be found in [172].

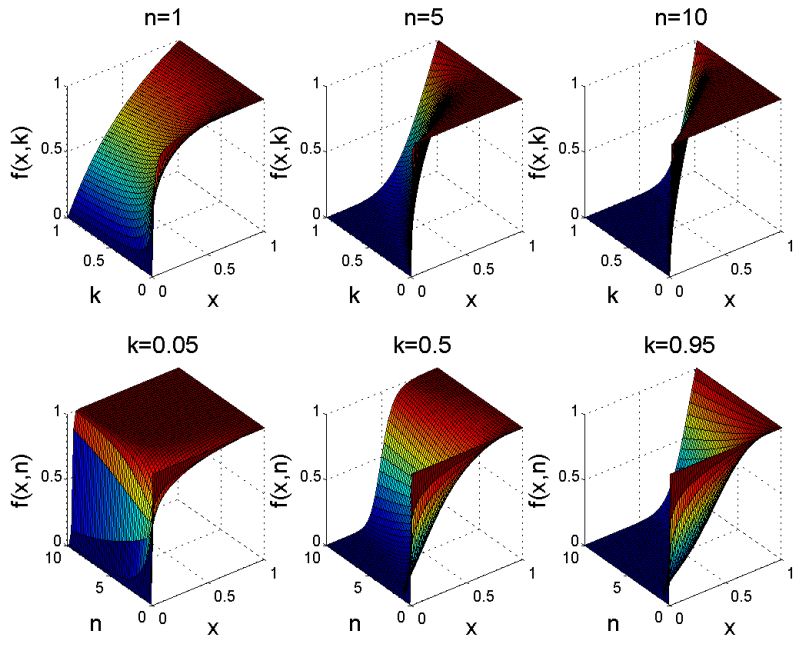


Figure 2.2: Normalized Hill function depending on the parameters

2.2 Methods for reverse engineering

One of the main objectives of this work is to find a suitable formulation that allows us to infer a Boolean network from a source of prior knowledge within the logic-based ODE framework. In a first stage, we focused in identifying the logic gates starting from a source of prior-knowledge which indicates explicitly which interactions exist. In a second advanced phase, we will expand the methods to be able to identify potential missing interactions or, as a more ambitious goal, to be able to fully reconstruct the networks from time-series data. In this section we offer an overview of the current perspective about these two problems in the field of computational systems biology.

2.2.1 Parameter estimation: the frequentist and the Bayesian point of view

Frequentist methods[1] for estimation of model parameters search for the parameter vector θ that minimizes the likelihood function[2]:

$$L(x|\theta) = \prod_{k=1}^{m} \prod_{l=1}^{d_k} \frac{1}{\sqrt{2\pi\sigma_{kl}^2}} \exp\left(-\frac{1}{2} \left(\frac{y_{kl} - y_k(t_l)}{\sigma_{kl}} \right)^2 \right), \tag{2.5}$$

where σ^2 is the estimated measurement error expected to be normally distributed for this version of the likelihood [13]. But what if there are several models or even parameter configurations which fit the data equally well? In the case of parameter estimation this is a common pitfall and has been well described in the literature [8] [10].

For parameter estimation, identifiability problems can be divided in two groups, structural and practical. Structural identifiability is a feature determined by the model structure and not the experimental data [31], being often derived from redundant parametrization. As an example consider the following differential equation:

[1] Also known as maximum likelihood estimation.

[2] In fact for computational reasons one typically minimizes the log-likelihood which shares the same minimizer.

$$\frac{dx}{dt} = k_1 \cdot k_2 \cdot y \tag{2.6}$$

where k_1 and k_2 appear only once in a given ODE model where $\frac{dx}{dt}$ describes the trajectory of x. Here it is obvious that k_1 and k_2 can assume any value since one can compensate the other. However this type of problems are usually non-trivial and typically require the application of methods performing symbolic manipulations. Chis et. al. highlight that there is no method amenable for every model [31]. Moreover, in order to solve structural identifiability problems modification, of the model structure is required (e.g. model reduction).

On the other hand, practical identifiability can be addressed with optimal experimental design. This type of lack of identifiability is originated by insufficient quality (e.g. noise) and/or quantity of data. To solve this problem, we first need a metric to quantify the lack of identifiability. Ideally, we want to obtain a confidence interval (or an approximation) for the accuracy of the parameter estimates. For this purpose, a widely used method is the Fisher Information Matrix (FIM) [13], [8] [10].

The Bayesian methods for inferring parameters are interesting in the sense that these allow (under certain conditions) a very precise way to model the propagation of uncertainty in the parameters to the model predictions. In Bayesian inference, the key idea is that the posterior distribution $P(\theta|y)$ is iteratively updated by sampling parameters vectors (θ) from a prior distribution ($P(\theta)$) and computing the likelihood function (equation 2.5):

$$P(\theta|y) = \frac{L(y|\theta)P(\theta)}{P(y)}, \tag{2.7}$$

where y is the experimental data and $P(y)$ is the marginal probability distribution which is typically treated as a normalizing constant factor since it is only possible to compute for very low dimensional problems [171]. If the problem is identifiable and one has narrow distributions for the parameter priors these methods should work well. However the existence of strong (*i.e.* constrained) meaningful priors for kinetic parameters in systems biology is rare. Additionally, specially in the absence of strong priors and considering a large number of parameters (*i.e.* the curse of dimensionality), the computational cost can rapidly become infeasible [126]. If the

problem is not structurally or practically identifiable there is a risk of applying great amount of computational effort with little change of convergence.

An interesting view by Raue et. al is that frequentist and Bayesian methods should work together [126]. More specifically they propose the use of a technique called Profile Likelihood Analysis (PLA) to first constrain the prior distribution before using the Markov Chain Monte Carlo (MCMC) methods which allows a better assessement of the uncertainty in the model predictions. PLA is typically used for practical identifiability analysis and (arguably) structural identifiability analysis [125] [150]. The idea in profile-likelihood is not too far from bootstraps or the jack-knife method. However, instead of perturbing the data, the parameters are fixed one by one in several different values and the rest of the parameters is re-optimized. By looking at the profile of the likelihood function it is possible to draw conclusions about the identifiability of the parameters.

More robust methods are Bootstraps or the Jack-knife method. The parametric bootstrap method works by repeating parameter estimation a large number of times introducing noise (equivalent to the assumed experimental error) in the simulated data[3]. Assuming that optimal or near optimal solutions are always achieved it is possible to derive a confidence interval for the parameter estimates [80]. Jack-knife works similarly but instead of perturbing the data with random noise a data-point is omitted for each new estimate. Although these methods are indeed more robust, the price to pay is that due to the need of repeating parameter estimation a large number of times the computational cost increases very rapidly.

2.2.2 Finding Logic Gates: a model selection problem

To find the logic gates which best describe the behaviour of a given network given known interactions, we will be interested in a formulation similar to what was used by Saez-Rodriguez et. al. [135] within a Boolean logic framework or by Morris et al. within the constrained fuzzy-logic formalism [111]. The idea here is that starting from a directed graph containing the interactions and their sign (activating or inhibitory), we can obtain an expanded hypergraph containing all the possible gates, where edges with two or more inputs (hyperedges) represent a

[3]This is equivalent of resampling the residuals.

logical conjunction (AND gate) and single edges represent a logical disjunction (OR gate). To calibrate such models (M), the authors formulated the inference problem as a binary multi-objective problem, where the first objective corresponded to how well the model described the experimental data and the second consisted in a complexity penalty to avoid overfitting:

$$F_{fitness}(M) + \alpha \cdot F_{complexity}(M) \tag{2.8}$$

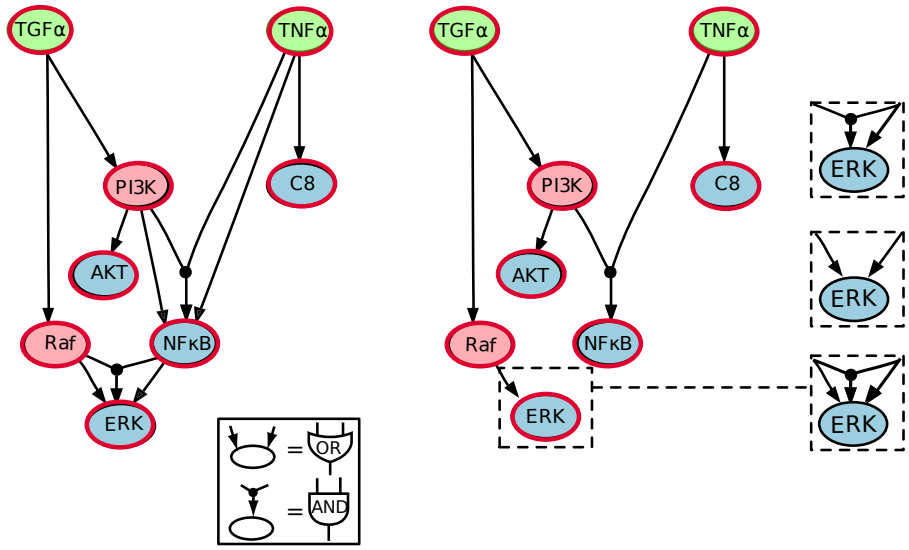
where $F_{fitness}(M)$ is the mean squared error and $\alpha \cdot F_{complexity}(M)$ is the product between a tunable parameter α and a function denoting the model complexity (AND gates and OR gates receive twice the penalty of a simple activating or inhibiting edge). Figure 2.3(a) shows a network containing all possible logic gates and figure 2.3(b) shows a network after calibrations.

This problem was solved by means of a genetic algorithm implemented in CellNOptR [158] and previously solved by other authors in more elegant, yet less accessible formulations, such as Integer Linear Programming (ILP) [109] or Answer Set Programming (ASP) [165].

Constrained fuzzy logic (also implemented in CellNOptR) searches for a network topology (or more precisely a family of networks) which can best represent the experimental data. Nevertheless, constrained fuzzy logic, uses normalized Hill functions in its transfer functions and must also search for a set of k parameters (the parameters n are fixed). To handle this, the authors discretize the k parameters into low, medium and high values, thus transforming it into a discrete problem. To solve this problem, Morris et. al. [111] used a discrete genetic algorithm.

Also, Mitsos et. al. proposed a non-linear programming formulation for estimating the parameters and a MINLP for calibrating both parameters and structure [108]. These formulations are particularly relevant for the problem under study here and will be further discussed in the following section.

Similarly to what is described in [135], covering the whole search space is, in most cases, infeasible since the it grows exponentially with the number of decision variables. Even if we could evaluate the whole search space of binary variables, we would still need to solve an NonLinear Programming (NLP) sub-problem for each set of decision variables. In previous work [65], we have addressed the parameter



(a) An expanded hypergraph containing all the possible gates, as seen [135].

(b) An optimized model , as seen in [135].

Figure 2.3: Illustration of the method from Saez-Rodriguez

estimation problem for this kind of problems. Specially for larger models, solving the parameter estimation problem is difficult due to its highly non-linear and non-convex nature and, consequently, the large number of evaluations needed. Furthermore, solving each IVP, tends to be quite expensive and, in the case of the parameter estimation problem, is usually the main bottleneck.

Rodriguez et al. have proposed a MINLP formulation based in the Akaike Information Criterion (AIC) to select between several competing hypothesis [131]. The AIC favors simpler solutions ($2K - 2L(y|\theta)$). The derivation of the AIC is relatively complex and we will not detail it here, however for further detail on theoretical aspects of the AIC we direct the reader to [25]. Besides the statistical sound arguments provided by the bias-variance framework[4], simpler solutions are typically easier to analyze and it is part of the scientific culture to assume the simplest solution is the most likely to be true [44].

[4]There are several other statistical/mathematical frameworks that support the Occam's Razor argument (see [44])

The results of AIC are reliable only under very particular assumptions (when parameter posteriors are unimodal and almost multivariate normal [87]) and the Bayesian framework is argued to be a much better tool for this purpose. Nevertheless, in contrast with the parameter inference problem, the computation of the marginal distribution $P(y)$ for the models (M) and parameter sets θ requires the approximate solution of the integral:

$$\int P(y|M_i, \theta_i) \cdot P(\theta_i|M_i) d\theta_i \ , \tag{2.9}$$

something which is only possible for very small problems (less than a few dozen of parameters[5]). For larger problems the uncertainty in the model trajectories can also be predicted by data-resampling methods (e.g. bootstrap) [87, 147, 171] with similar assumptions to those made in the Bayesian framework [87]. Note that while this is extremely computationally demanding, it scales up relatively well compared with thermodynamic integration procedures used to compute $P(y)$ [27].

Vyshemirsky & Girolami [171] point out that the high order differential equation models used in biological research can bring about complex nonlinear likelihoods rendering the results provided by AIC unreliable. However due to the existing limitations in computational power and for practical reasons, the AIC and the Bayesian Information Criterion are still part of the arsenal used to perform model reduction and selection of nonlinear biological models and are widely used [25, 87] despite of its known limitations.

2.2.3 The general network inference problem

The previous ideas can be used to some extent in the network inference problem (beyond finding the logic gates) in the sense that at least in principle it should be able to tell us which links appear not to be present according to the PKN. However, it cannot tell us anything on how to expand prior knowledge by means of experimental data. Additionally, what happens if there is no source of prior knowledge at all? Then, we would be faced with the more general problem of network inference which is discussed in detail for gene regulatory networks in the

[5] See recent studies with approximated Bayesian computations [162], [161], [171]

reviews [62], [17] and [39].

The Dialogue for Reverse Engineering Assessments and Methods (DREAM) challenge provides a framework where different research groups can test their algorithms for diverse reverse engineering problems relevant to systems biology in the form of a competition. In the third edition of the DREAM competition [121], a particular challenge of interest was the reverse engineering of an *in silico* network which is discussed in detail in [100]. The *in silico* data was generated by linear and non-linear (Hill type kinetics) dynamics for translation and transcription respectively.

The DREAM challenge organizers compared the performance of 29 methods, where most of these achieved predictions were not statistically more significant than random predictions. Interestingly, there were methods from all the most common types of inference algorithms, like information-theoretic, Bayesian and ODE based. Also, there seemed to be no correlation with the type of method used and the quality of the results implying that results were related with particularities in the implementation done by each team. The winning team combined an information-theoretic with non-linear differential equations to treat two types of data (steady-state and continuous) [174]. Also relevant here is that most methods (with the exception of the best performer) systematically failed to predict combined regulation, *i.e.* links with an in-degree greater than 1, thus making a strong point for the development of network inference methods based on logic-based ODEs.

An approach borrowing ideas from ODEs and Boolean logic is the Inferelator [21]. This method combined these with information theoretic scores and did very well in the DREAM 3 [98] and 4 [57]. This method encodes ODEs as:

$$\tau \frac{dy_i}{dt} = -y + g(\beta_{i1}x_{i1}, ..., \beta_{in}x_{in}) \tag{2.10}$$

where $\frac{dy}{dt}$ corresponds to the gene variation depending on its various regulators x_i. Each parameter x_{ij} has a corresponding weight parameter β_{ij} with a corresponding weight, and the parameter τ is related with the species half-life.

In its original version (1.0), Inferelator handles simple Boolean functions (AND, OR and XOR) with the min/max approach of only two regulators. For instance the equation:

$$g(\beta_{i1}x_{i1}) = \beta_1 x_1 + \beta_2 x_2 + \beta_3 min(x_1, x_2) \tag{2.11}$$

would correspond to an AND gate with $\beta = [1\ 0\ 0]$ and to an OR gate if $\beta = [-1\ 1\ 1]$. The ODE is approximated with finite differences:

$$\tau \frac{y_{m+1} - y_m}{\Delta t_m} + y_m = g(\beta_{i1}x_{i1}, ..., \beta_{in}x_{in}) \tag{2.12}$$

Despite some similarities with the logic-based ODE framework, this approach is in a strict sense closer to a supervised learning problem, like a neural network rather than typical applications evolving kinetic models or control engineering. In this framework, all regulators are measured and the model only has to be able to predict the next step instead of its whole trajectory. Also interestingly, for predictor selection, the authors used the L1 LASSO shrinkage criterion, where the shrinkage parameter is determined by cross validation to avoid overfitting. As long as we are able to keep reasonable computational cost, borrowing ideas from information-theoretic and machine learning might be helpful. Other similar methods apply L1 for inferring chemical reaction networks [9] or more recent ideas such as compressive sensing [115].

A closely related method to the objective of this work is the algorithm Mixed Optimization for Reverse Engineering (MORE) by Sambo et. al. [139]. Their authors propose a bi-level optimization where the discrete (binary) level communicates with the continuous (NLP) level and vice versa. For model representation, the authors use a structured formalism formally identical to dynamic recurrent neural networks:

$$\frac{dx}{dt} = \frac{k_{1i}}{1 + e^{-\left(\sum_{j=1,...,n} a_{ij}x_j + \sum_{l=1,...,n} b_{il}u_j\right)}} - k_{2i}x_i \tag{2.13}$$

producing a sigmoid which depends on other dynamic variables (x_{ij}) and external inputs (u_{lj}). Parameters a_{ij} and b_{ij} regulate the relative importance of each. Additionally, there is a linear degradation term depending on the parameter k_{i2}.

The authors point as a major advantage the use of numerical integration instead of trying to estimate the derivatives directly from temporal data, which amplifies noise (e.g. the Inferelator). However, it is also true that the introduction of many

non-linearities and parameters is likely to cause identifiability issues. Still, the bi-level optimization approach makes it easier to add common biological constraints such as sparsity and the introduction of prior knowledge. Nevertheless, when compared to the logic-based ODE, this modeling approach does not handle Boolean expressions of any type and, thus, the ability to gain mechanistic insight from the system seems rather limited.

It is important to highlight the importance of using biological constraints like the assumption of sparsity which can be introduced explicitly (e.g. maximum number of regulators) or implicitly (e.g. complexity penalties). For instance, Akutsu et. al. were able to use a brute force approach for reconstructing a Boolean network simply by assuming a small maximum number of regulators per gene [1] (note that here only input/output relations are considered and not the network as a whole).

2.3 Mixed Integer Programming and Relaxations

Both the model selection and the network inference problem can be handled as MINLP formulations. Here, we will discuss the state of the art methods to solve this type of problems putting more emphasis in meta-heuristic approaches and how to implement the constraints or penalties to solve the problem in an efficient manner.

An important disclaimer is that most of the methods have been developed to solve control engineering problems and are deterministic. Additionally, current technologies for solving this type of problems are not anywhere close to what has been done for Linear Programming (LP), NLP or integer programming [130]. These problems combine both the difficulties of solving non-linear, non-convex problems and those typical of combinatorial problems.

Bansal et al [18] discuss a set of strategies typically used to circumvent the use of Mixed-Integer Dynamic Optimization (MIDO) methods. A first approach is trying to transform the problem into a purely continuous NLP which is much easier to solve. For instance, the integer variable y which can assume the values of 0 and 1 could be represented in a smooth manner by:

$$y = \frac{1}{2}[tangh(\beta x) + 1] \tag{2.14}$$

where β is a large positive number and y will tend to 1 for $x > 0$.

This is, in practice, similar to what was implemented by Mitsos et. al. in [108].

In this sense, the choice of global stochastic optimization methods appears as a reasonable choice. Although these cannot offer guarantees about the optimality of the solution, if the problem is not pathologically ill-posed, stochastic methods are often able to locate its vicinity in reasonable computational times [128]. An additional feature of these methods is that usually these do not require a transformation of the original problem and we can treat them as a black-box.

Although most MINLP bibliography focuses on deterministic methods, these authors often end up discussing the previously mentioned benefits and drawbacks of stochastic methods. Two methods widely referred are Mixed Integer Tabu Search (MITS) [175] [53] and Ant-Colony for Mixed Integer (ACOmi) [26] [24]. The implementation details of these methods can be found elsewhere [49] [143]. It is worth highlighting that a key characteristic of both methods is that the stochastic algorithm is combined with a local solver called MISQP, a local method based on Sequential Quadratic Programming [113]considered to be the state of the art for this purpose.

A drawback pointed by [33] is that stochastic algorithms often have difficulties with highly constrained problems. However, this is in many cases caused by inefficient implementation of the constraints which often relies exclusively in the use of the so-called death penalty, where an extremely large fitness value is given to an infeasible solution.

A final note is that the access to computing clusters with many cores is nowadays something common amongst research groups (including the host groups of this work). Specially within the framework of stochastic (black-box) algorithms, it makes sense to use high performance computing methods to increase the portions of the search space we can explore. A method published recently [167] applies an iterative communication schema between different parallel optimization runs. In order to ensure proper exploration and exploitation, each optimization run has its own fine tuning parameters. An interesting feature of this approach is that

it appears to be highly extensible to most stochastic optimization methods with reasonable effort.

Chapter 3

Reverse engineering of logic-based differential equation models using a mixed-integer dynamic optimization approach

This chapter reproduces integrally the work accepted for publication in Bioinformatics in May 2015.

3.1 Introduction

In recent years, there has been a growing interest in the application of logic formalisms to systems biology, and in particular to model signal transduction [2,138]. The basis of this model formalism lies in the assumption that cells process information of certain stimuli approximately by logic circuits, and their simplicity, and their simplicitly makes them particularly amenable to model large networks and integration of pathway knowledge from databases and high-throughput experimental data [20].

Logic models were first introduced by [85] to model gene regulatory networks. Since then, diverse modifications from the original formalism were developed. In particular various extensions have been developed to accommodate continuous val-

ues(e.g. [4, 19, 21, 38, 103, 172]). Amongst these formalisms, logic-based ODEs are well suited to handle time series in a precise manner. The main idea is to transform the logic model into a continuous homologue in the form of ODEs. Since it is based on a logic circuit, this formalism does not require information about the biochemistry (e.g. stoichiometry or type of kinetics), and at the same time, since it provides a model of differential equations, we can accurately perform dynamic simulations for the state variables trajectories. Several methods have been proposed in the literature to transform Boolean logic model into ODE approximations [21, 103, 172]. CellNOpt, relies in multivariate polynomial interpolation introduced by [172].

Logic formalisms has been used to reverse engineer biochemical networks from data. One early example is the work by [1] which proposed a brute force approach that infers the Boolean function of a few top k regulators, node by node. Other methods treat these networks in a global manner (instead of fitting logic functions node by node) borrowing ideas from optimization and machine learning to avoid excessive model complexity [21, 135]. In [135] networks derived from of prior knowledge, from e.g. public repositories of manually curated networks, are expanded into an hypergraph, where all the possible logic gates are represented and optimization strategies are used to find which networks could best reproduce the data with the smallest number of hyperedges. This method is implemented in the software CellNOpt [157] for various logic formalisms and is designed to reverse engineer Boolean models, mainly in a protein signaling environment, given data from perturbation experiments.

Here, we present a mixed-integer global optimization approach for the problem of reverse engineering signalling and regulatory networks as logic-based ODEs from a source of prior-knowledge containing multiple possible regulation links and experimental data. In this work, we formulate the problem of identifying the logic gates as a simultaneous model selection and parameter identification problem. From the optimization point of view, this corresponds to a MIDO problem. Although MIDO problems are typically hard, we show here that solutions can be achieved for rather complex networks by applying certain global optimization meta-heuristics.

Only a few authors have considered the use of MINLP for reverse engineer-

ing purposes. [139] proposed the algorithm MORE, which consists in a bi-level optimization where the discrete (binary) level communicates with the continuous (NLP) level and vice versa. For model representation, a structured formalism, formally identical to dynamic recurrent neural networks, is used. [59] have presented a deterministic method for identification of regulatory structure and kinetic parameters in biochemical networks, transforming the MIDO problem into an approximated large-scale MINLP, which was then solved by a nonlinear branch and bound method. To avoid local minima the authors provided high quality initial solutions to the solver. These solutions were obtained by solving a set of relaxed problems from different starting points. Despite these advances, the major drawback of deterministic global methods is that the computational effort increases very rapidly with problem size. More recently, [134] have shown how to apply MINLP to perform simultaneous model discrimination and parameter estimation in dynamic models of cellular systems.

This paper is organized as follows: first, we present the formulation of the MIDO problem making use of logic-based dynamic models. Then we present a solution strategy based on global optimization metaheuristics. Next, the performance and capabilities of the new approach are illustrated with several reverse engineering case studies: a synthetic pathway of signaling regulation, a signal transduction pathway in bacterial homeostasis, and a signaling pathway in live cancer cells. Finally, the main conclusions are outlined.

3.2 Methods

3.2.1 Problem formulation

In order to find the logic gates which best describe the behavior of a given network, we will be interested in a formulation similar to what was used by [135] within a Boolean logic framework or [111] within the constrained fuzzy-logic formalism. The idea here is that starting from a directed graph containing only the interactions and their signs (activating or inhibitory) we can obtain an expanded hypergraph containing all the possible gates where edges with two or more inputs (an hyperedge) represent a logical conjunction (AND gate) and single edges

represent a logical disjunction (OR gate).

The problem can be formulated as the following:

$$
\begin{aligned}
\underset{n,k,\tau,w}{\text{minimize}} \quad & F(n,k,\tau,w) = \sum_{\epsilon=1}^{n_\epsilon} \sum_{o=1}^{n_o^\epsilon} \sum_{s=1}^{n_s^{\epsilon,o}} (\tilde{y}_s^{\epsilon,o} - y_s^{\epsilon,o})^2 \\
\text{subject to} \quad & \mathcal{E}_{sub} = \{e_i | w_i = 1\},\ i = 1, \ldots, n_{\text{hyperedges}} \\
& \mathcal{H}_{sub} = (V, \mathcal{E}_{sub}) \\
& \text{LB}_n \leq n \leq \text{UB}_n \\
& \text{LB}_k \leq k \leq \text{UB}_k \\
& \text{LB}_\tau \leq \tau \leq \text{UB}_\tau \\
& \dot{\bar{x}} = f(\mathcal{H}_{sub}, \bar{x}, n, k, \tau, t) \\
& \bar{x}(t_0) = \bar{x}_0 \\
& y = g(\mathcal{H}_{sub}, \bar{x}, n, k, \tau, t)
\end{aligned}
\tag{3.1}
$$

where $\mathcal{H}_{sub}$ is the subgraph containing only the hyperedges ($\mathcal{E}_{sub}$) , defined by the binary variables w. Additionally n, k and τ are the continuous parameters needed for the logic-based ODE approach. These parameters are limited by upper and lower bounds (e.g. LB_k). The model dynamics ($\dot{\bar{x}}$) are given by the function f. This set of differential equations varies according to the subgraph (and therefore also according to the integer variables vector w). Finally, the system of differential equations has to be solved to obtain the simulated data. The objective function is the squared difference between the simulated data (y) and the experimental data ($\tilde{y}$) and our goal is to minimize this value for every experiment (ϵ), observed species (o) and sampling point (s). The simulation data y is given by an observation function g of the model dynamics at time t.

3.2.2 Solving the mixed integer dynamic optimization problem

The problem considered in this work belongs to the category of network reverse engineering, where the objective is to simultaneously determine network topology and continuous mode parameters which explain a given set of data. The network

contains a series of possible regulatory mechanisms and our goal is to find the set that best describes the data. Our dynamic formulation, shown in the previous section, makes use of logic-based ODEs. Essentially, the binary variables define the structure of the system of ODEs describing the dynamic behaviour. Additionally, a set of continuous parameters modulating those dynamics need to be estimated. From the optimization point of view, this problem belongs to the class of MIDO.

In general, model calibration of a nonlinear dynamic model is a difficult task. Due to the nonlinear and constrained nature of the system dynamics, these problems are multi-modal (non-convex) [12,166]. The MIDO considered here augments the difficulties of solving non-linear, non-convex problems with those typical of combinatorial problems.

MIDO problems can be solved using deterministic or stochastic global optimization methods. Regarding deterministic methods, these offer guarantees of global optimality, and significant advances have been made recently(for example, [59]). However, these still suffer from the major drawback of deterministic global methods, *i.e.* computational effort increases extremely rapidly with problem size.

Stochastic algorithms for global optimization can not offer guarantees of global optimality, but usually converge to the vicinity of the global optimum in reasonable computation times, at least for small and medium scale problems. However, for larger problems their computational cost is very significant [110]. Hybrid approaches try to combine the best of the two worlds by combining global stochastic methods with efficient (local) deterministic optimization methods [16,133]. In this context, metaheuristics (*i.e.* guided heuristics) have been particularly successful, ensuring the proper solution of these problems by adopting a global optimization approach, while keeping the computational effort under reasonable values thanks to efficient local optimization solvers [132].

In this work, we have chosen a recent metaheuristic based on the combination of an enhanced Scatter Search (eSS) method as global solver [47] with a Mixed-Integer Sequential Quadratic Programming (MISQP) [50] local solver. eSS is an evolutionary algorithm for complex-process optimization that employs some elements of scatter search and path relinking. MISQP is a trust region sequential quadratic programming solver adapted to solve MINLP problems. In this code, instead of solving continuous quadratic programs, the solution is approximated by

a series of mixed-integer convex quadratic programming problems. In addition, MISQP accepts black-box problems and, thus, does not require the problem to be transformed into an algebraic form, a typical requirement of most MINLP methods. As shown below, we compared the performance of eSS with two other modern metaheuristics, ACOmi [143] and MITS [49]. For the class of problems considered here, we found that eSS consistently provided the best results.

3.2.3 A multi-phase scatter search with relaxed MINLPs

The MIDO problem formulated above is extremely challenging to solve. Although the initial results obtained with the eSS method [47] were promising, a second objective of this work was to improve the algorithm in terms of convergence speed while keeping robustness in order to ensure a good scale-up for realistic applications. For this purpose, we have devised a Multi-Phase enhanced Scatter Search (MPeSS) strategy which, in a first phase, computes intermediate solutions of relaxed MINLPs and, in a second phase, uses them as initial points for solving the original MINLP.

In order to reformulate a relaxed problem, we consider each hyperedge to be associated with a continuous weight instead of a binary variable. Each weight will appear as an additional term in its corresponding minterm from the truth table. When several weights affect a single minterm, then we can apply the multivariate polynomial interpolation of an OR gate. Table 3.1 and Figure 3.1 illustrate the problem formulation where variables $\bar{x_1}$ and $\bar{x_2}$ represent two different inputs: only $\bar{w_1}$ activates Y; only $\bar{w_2}$ activates Y; $\bar{w_1}$ and $\bar{w_2}$ are required to activate Y.

When solutions are of a binary nature this formulation holds exactly the same solution as the previously shown for the MINLP case. So far, this reformulation produces an over-parameterized problem which does meet the basic constraint that each hyperedge can only be present or not present. Thus, to enforce that solutions for w tend to be of a binary nature, we add a penalty. The objective function to be minimized becomes:

$$F_p = (\tilde{y_i} - y_i)^2 + P \tag{3.2}$$

$$P = \alpha \cdot \sum_{i=0}^{n_{int}} p_{w_i} \tag{3.3}$$

$$p_{w_i} = \begin{cases} w_i, & \text{if } w_i \leq 0.5 \\ 1 - w_i, & \text{if } w_i > 0.5, \end{cases} \tag{3.4}$$

where P_{w_i} is the penalty associated with the deviation of each w_i from the nearest binary value (0 or 1).

x_1	x_2	$\bar{B}^I(\bar{x_1}, \bar{x_2}) = ...$
0	0	$0 \cdot (1 - \bar{x_1}) \cdot (1 - \bar{x_2}) +$
0	1	$w_1 \cdot (1 - \bar{x_1}) \cdot \bar{x_2} +$
1	0	$w_2 \cdot \bar{x_1} \cdot (1 - \bar{x_2}) +$
1	1	$\mathrm{OR}(w_1, w_2, w_3) \cdot \bar{x_1} \cdot \bar{x_2}$

Table 3.1: Truth table with weights representing the presence of hyperedges in a continuous formulation for the graph shown in Figure 3.1. The multivariate polynomial interpolation of the OR gate is used to make a smooth approximation of a logical disjunction for the weights w_1, w_2 and w_3.

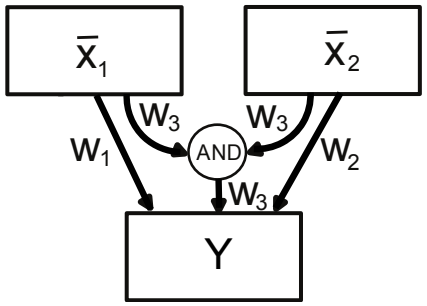


Figure 3.1: Diagram illustrating the association of the used weights (w) with each hyperedge. There are essentially four options in this example: if w_1 is equal to one $\bar{x_1}$ activates y. If w_2 is equal to one, $\bar{x_2}$ activates y. If w_3 is equal to one and both w_1 and w_2 are zero, $\bar{x_1}$ and $\bar{x_2}$ are required to activate y. If w_1, w_2 and w_3 are equal to zero y is never activated. OR gates are implicitly represented as simple edges.

The usage of this relaxed formulation to find MIDO solutions can be summarized as follows:

- In a first phase we solve the relaxed problem without any penalty to find a set of continuous parameters which are able to describe the data well.

- The solution found in the previous iteration is used to restart eSS with a given α penalty. Depending on the difficulty of the problem, this step might consist on only one iteration or multiple phases with increasing α. If α is increased too sharply, the penalty (P) will dominate over the goodness of fit and we risk guiding the metaheuristic towards uninteresting areas of the search space.

- In a final step, we apply eSS to solve the pure MINLP problem, where the best solution from the previous steps is used as an initial guess (rounding the previously relaxed variables).

3.2.4 Remarks on the tuning and performance assessment of metaheuristics

Meta-heuristics for global optimization are approximate stochastic methods which in general do not have proofs of convergence. Thus it is not possible to obtain an analytical prediction of the effort it will take to arrive to a solution of a certain quality. Similarly, it is not possible to ensure that the metaheuristic will arrive to near-global solutions in every run. A related problem is the tuning of the internal search parameters of the method. Although the eSS metaheuristic is mostly self-adapting in that sense, we still need to choose a stopping criterion.

Due to this lack of theoretical guarantees and the stochastic behaviour of these methods, one must resort to empirical tuning and performance assessments. We have performed this tuning and assessment based on repeated runs of the methods for each problem and the subsequent analysis of the convergence curves (objective function values versus number of function evaluations) and the distributions of the solutions found (see general discussion in [30]).

The analysis of these distributions for a number of trial runs allow us to choose the stopping criteria. In general, stopping criteria for metaheuristics are based on 3 metrics [55]: (i) after a fixed number (budget) of Function Evaluations (FEs) (or, similarly, computation time, or iterations) (ii) after a fixed number of iterations

without improvement in the cost function (iii) when the cost function arrives to a pre-set value-to-reach.

These criteria can be combined. In our study, we have chosen (i) because criteria (ii) can be reached with premature stagnation in local optima, and criteria (iii) requires a priori knowledge about the global solution. Criteria (i) is widely used [144] and is particularly useful when the evaluation of the cost function is computationally expensive (as in our study), since it also directly reflects practical limits on computation time.

3.3 Results

3.3.1 Case study 1: Synthetic Signaling Pathway

In order to illustrate the methodology we now turn to a published model used by [97]. This dynamic model is composed by 26 ODE and 86 continuous parameters. It was initially used to illustrate the capabilities and limitations of different formalisms related with logic-based models. Although this is a synthetic model, it was derived to be a plausible representation of a signaling transduction pathway. This model was used to generate pseudo-experimental data for 10 combinations of experimental perturbations of 2 ligands (TNFα and EGF) and two kinase inhibitors (for PI3K and RAF1). From a total of 26 dynamic states, 6 were observed (NFKB, P38, AP1, GSK3, RAF1 and ERK) and 5% of Gaussian noise was added to the data.

Following the methodology described in [135], we obtained an expanded version of this model containing every possible AND/OR logic gate given the initial graph structure. This so-called expansion procedure generated a nested model comprising 34 additional variables, one for each hyperedge (Figure 3.2).

The model and experimental setup were implemented using Advanced Model Identification using Global Optimization (AMIGO) [11]. The method of choice for the simulation of the IVP was CVODES [145].

As described previously, when using stochastic methods the recommended practice is to run each optimizer a number of times to assess their performance based on a distribution of results. This problem was solved in 30 runs by each method,

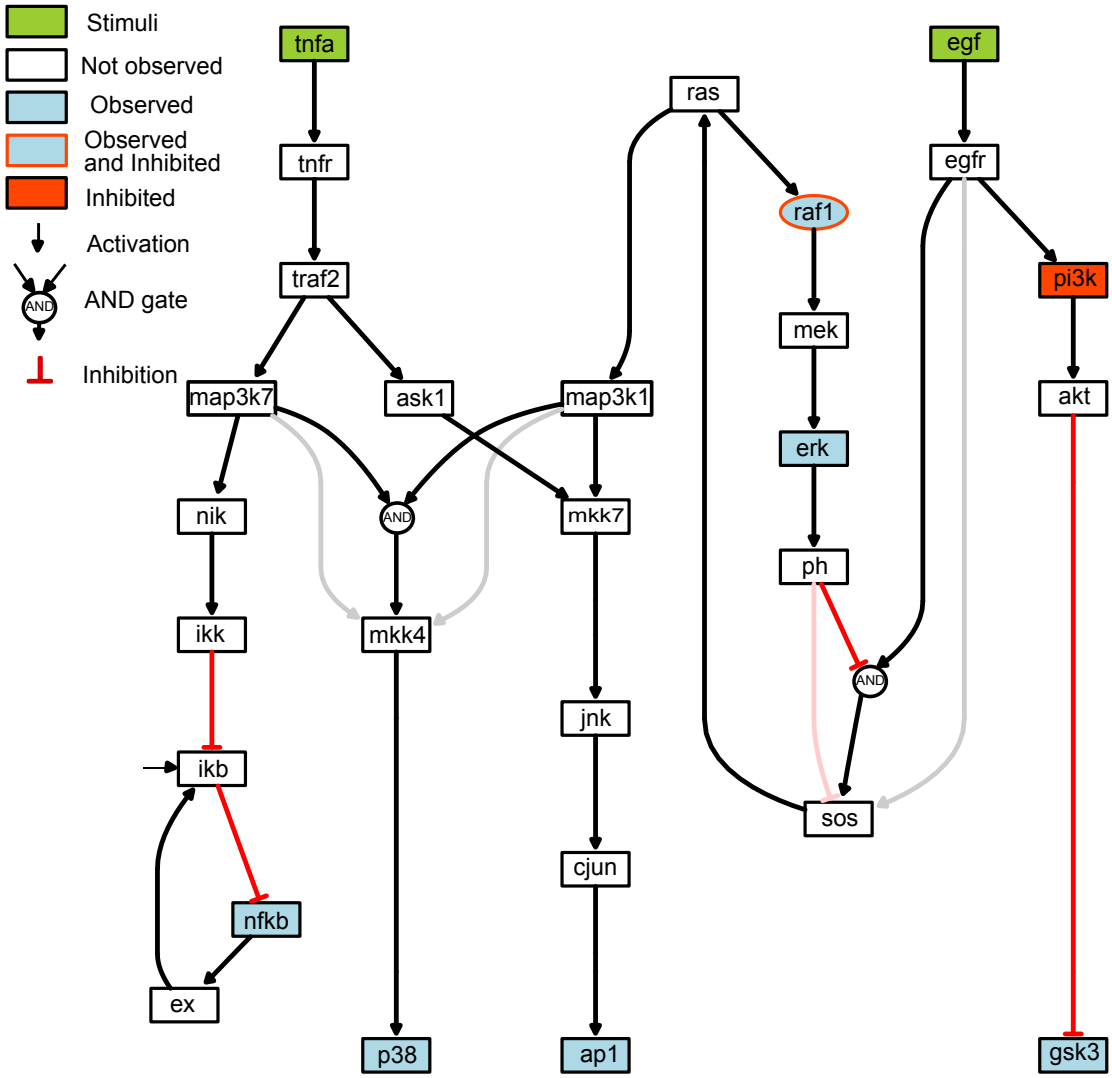


Figure 3.2: Case study 1 (synthetic signaling pathway): Hypergraph showing every possible logic gate consistent with the prior knowledge network. Strong red and dark hyperedges correspond to gates present in the original model used to generate the *in silico* data while gray and light red hyperedges show links not present in this model.

ACOmi, MITS, eSS and MPeSS, using a budget of $6 \cdot 10^4$ FEs. In the case of MPeSS this budget was equally distributed among three phases, with the first two using relaxations with $\alpha = 0$ and $\alpha = 6$, and with the third solving the original problem.

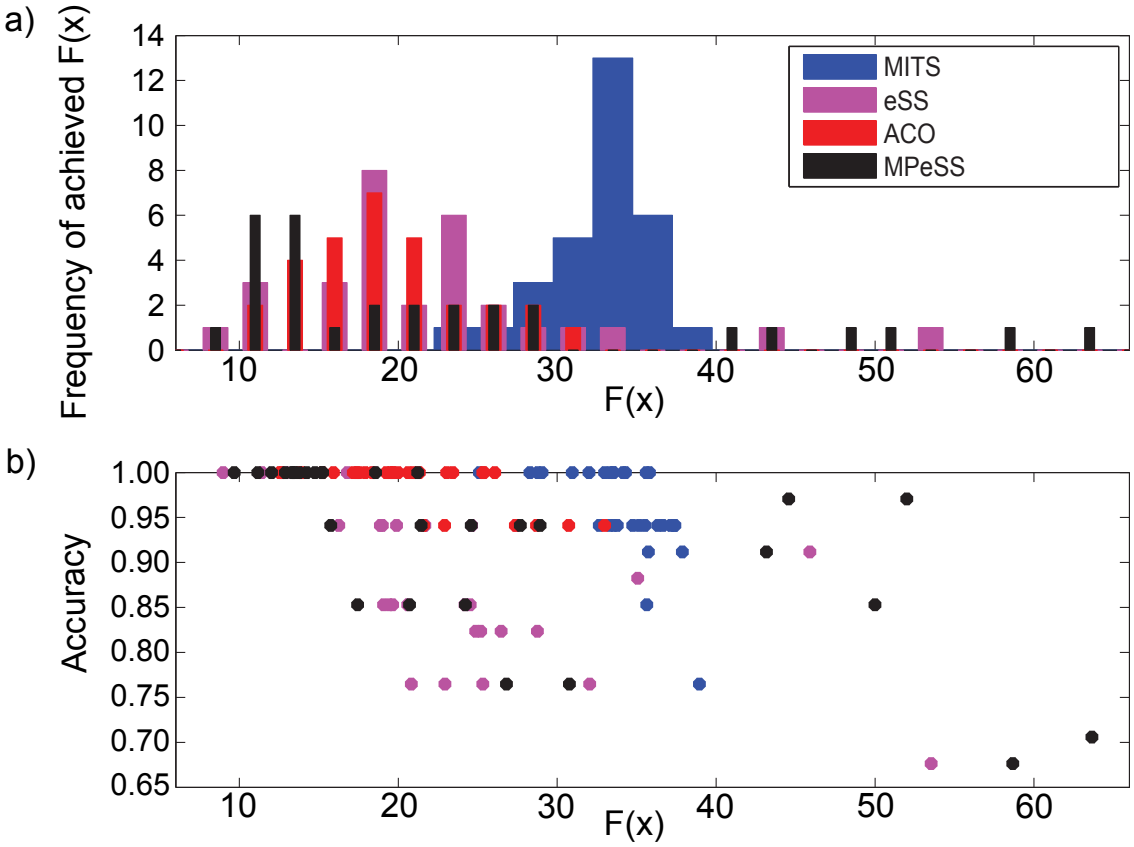


Figure 3.3: Case study 1 (synthetic signaling pathway): **(a)** Histogram of the final objective function achieved by each method across the multiple independent optimization runs. **(b)** The accuracy of the obtained solutions as a function of the objective function. Each dot describes the results of an independent optimization run.

Figure 3.3b represents the accuracy of the obtained solution as a function of the final objective function value achieved. Each dot describes the result of an independent optimisation run. Near-globally optimal solutions, with a final objective function value below a certain threshold, are always able recover the correct solution. The accuracy is computed as $(TP+TN)/(TP+TN+FP+FN)$, where TP is the number of true positive, TN the number of true negative, FP the number of false positive and FN the number of false negative hyper edges when compared with the correct solution (an accuracy of 1). Since the data has been generated *in silico* with known structure (see Figure 3.2) and parameters we can

compute the accuracy of the recovered model structures.

In Figure 3.3a, the histogram represents the distribution of final values achieved by each method, by combining both problem formulations (relaxed and MINLP), eSS is able to arrive to near-globally optimal values in approximately 47% of the runs. Additionally the time-course simulations (Figure 3.4) indicate a very good agreement with the pseudo-experimental data, which is also indicated by its low root mean square error (Root Mean Squared Error (RMSE)) of 0.099, defined as:

$$RMSE = \sqrt{\frac{\sum_{\epsilon=1}^{n_\epsilon} \sum_{o=1}^{n_o^\epsilon} \sum_{s=1}^{n_s^{\epsilon,o}} (\tilde{y}_s^{\epsilon,o} - y_s^{\epsilon,o})^2}{\sum_{\epsilon=1}^{n_\epsilon} \sum_{o=1}^{n_o^\epsilon} n_s^{\epsilon,o}}}. \tag{3.5}$$

Albeit no solver/configuration was able to recover the correct solution in every run, the MPeSS, where relaxed solutions are initially generated to help convergence, proved the most reliable. eSS was the second best method in terms of locating the vicinity of the optimal solution, although it was closely followed by ACOmi. MITS systematically failed to solve the problem for the considered FE budget. Convergence curves for the tested methods can be found in the supplementary materials (Figures S.2 and S.3);

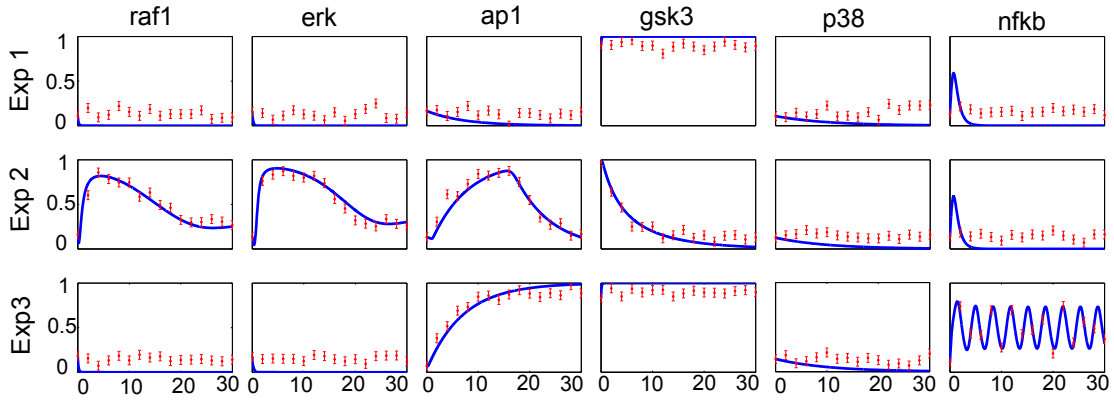


Figure 3.4: Case study 1 (synthetic signaling pathway): predicted versus observed time-series for the best solution found (experiments 1 to 3), showing a very good agreement of the simulation with the pseudo-experimental data used to calibrate the model.

3.3.2 Case Study 2: Application to the KdpD/KdpE two-component signal transduction pathway

In this section, we consider a model of K^+ regulation of the Kdpd/Kdpe two-component signal transduction pathway in *E. coli*. The main components of this system are the high-affinity K^+ transporter KdpFABC and two regulatory proteins, KdpD (sensor kinase) and KdpE (response regulator) [92]. The two proteins regulate the kdpFABC operon, which is activated in response to K^+ limiting conditions [63], restoring the intracellular K^+ concentration [81].

Recently, new experimental data has been generated using mutant strains with impaired K^+ properties and diverse K^+ stimulation conditions. Based on these data, [134] have postulated the possible existence of two new possible feedback loops and an alternative expression for a previous description of the stimuli counteraction responsible for restoring K^+ homeostasis. These new two feedback loops affected the translation and proteolysis of KdpFABC. Here, we write the differential equation describing the dynamics of KdpFABC as a logic-based ODE:

$$\frac{dKdpFABC}{dt} = \tag{3.6}$$

$$\begin{aligned}
&\Bigg(w_2 \cdot \Big[1 - f^{Hn}\Big(\frac{mRNA}{norm_{mRNA}}\Big)\Big] \cdot \Big[1 - f^{Hn}(KdpFABC)\Big] \\
&+ 0 \cdot \Big[1 - f^{Hn}\Big(\frac{mRNA}{norm_{mRNA}}\Big)\Big] \cdot f^{Hn}(KdpFABC) \\
&+ OR(w_1, w_2, w_3) \cdot f^{Hn}\Big(\frac{mRNA}{norm_{mRNA}}\Big) \cdot \Big[1 - f^{Hn}(KdpFABC)\Big] \\
&+ w_1 \cdot f^{Hn}\Big(\frac{mRNA}{norm_{mRNA}}\Big) \cdot f^{Hn}(KdpFABC) \\
&- KdpFABC \Bigg) \cdot \tau_{KdpFABC},
\end{aligned} \tag{3.7}$$

where $norm_{mRNA}$ is a parameter, used to scale mRNA to values between 0 and 1.

The expression for R3 controls the dephosphorylation of KdpEp:

$$\frac{dR_3}{dt} = \left[w_4 \cdot f^{Hn}(KdpFABC) - R_3\right] \cdot \tau_{R_3}, \tag{3.8}$$

where it is assumed that an the increase in the KdpFABC transporter will decrease internal K^+ concentration leading to an lower dephosporylation rate of KdpEp. More information about the model structure and context of this model can be found in the supplementary materials.

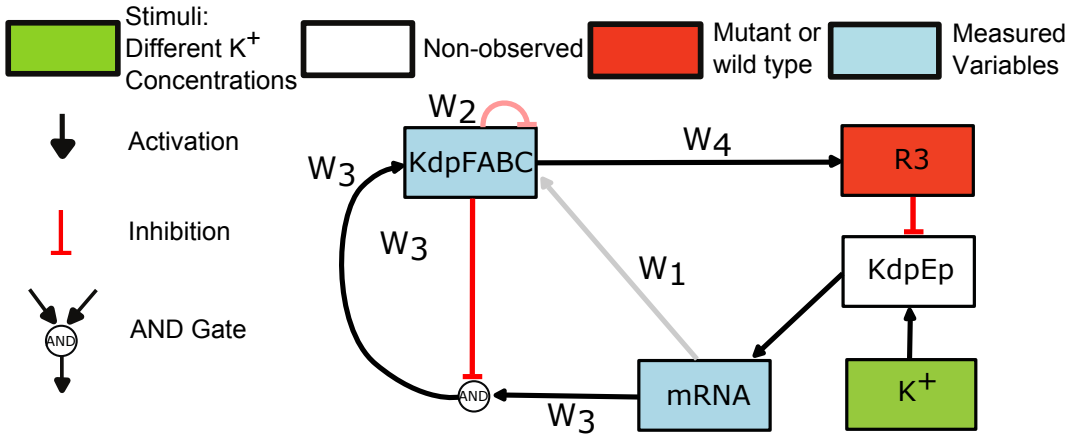


Figure 3.5: Case study 2 (*E. coli* homeostasis): The recovered model is depicted by strong red and dark hyperedges. Excluded hyperedges are represented in gray and light red.

To evaluate the ability of our method to describe and calibrate a model in a realistic scenario where multiple hypothesis are postulated, we used the model derived by Rodriguez-Fernandez and colleagues to generate pseudo-experimental data. We considered 10 different scenarios by varying the external concentration of K^+ and by considering a wild-type and a mutant strain. The mutant strain is modelled by removing the influence R_3 in the dephosphorylation of KdpEp. In the 10 experimental scenarios KdpFABC and mRNA were observed and perturbed with 5% of Gaussian noise.

We executed 30 optimization runs for each solver, eSS, ACOmi and MITS. The same budget of objective function evaluations was given to every run. In this case due to the smaller size of the problem we did not see any improvement by using MPeSS over eSS. The most robust method was clearly eSS (see Figures S.9 and S.10 in the supplementary materials). ACOmi was also able to solve the problem

in a few instances. MITS consistently failed to solve the problem for the allowed FE budget.

After redundant hypereges were filtered, all solutions showing a final objective function value below a given threshold (a total of 26) located the same solution. CellNOpt [157] was used to illustrate this solution (see Figure 3.5). In this problem 4 binary variables were considered; w_1, w_2 , w_3 and w_4. The hyperedges w_3 and w_4 were present in every of the top performing solutions while w_1 and w_2 were always absent.

When comparing the time course simulation of the best solution with the pseudo-experimental data we see that there is an excellent agreement between the two (normalised RMSE values of 0.0168 and 0.0191 for kdpFABC and mRNA, respectively).

3.3.3 Case Study 3: Signaling application to transformed liver hepatocytes

In this section, we explore the reverse engineering of a logic-based ODE model using liver cancer data (a subset of the data generated by [5]). It consists of phosphorylation measurements from an hepatocellular carcinoma cell line (HepG2) at 0, 30 and 180 minutes after perturbation.

To preprocess the network, we used CellNOptR, the R version of CellNOpt [157]. Basically, the network was compressed (see Figures S.16 and S.17 in the supplementary materials) to remove as many non-observable/non-controllable species. Subsequently, it was expanded to generate all possible hyperedges (AND gates) formed by a pair of inputs. The obtained full network (Figure S.18 in the supplementary materials) has a total of 109 hyperedges and 135 continuous parameters. To transform this model into a logic-based ODE model, we developed a parser that generates a C model file and Matlab scripts compatible with the AMIGO toolbox [11].

Although the data-set covers only three sampling time points it includes a large combination of 64 perturbations comprising 7 ligands stimulating inflammation and proliferation pathways as well as 7 small-molecule inhibitors blocking the activity of key kinases (see supplementary Figure S.15). To use logic-based ODE

models, all data should be in the $[0, 1]$ range and thus we simply normalised the data by rescaling it to this range . From the total 25 states present in the model, 16 corresponded to observed species. The initial conditions for the other 9 species are not known and were therefore estimated. In order not to increase the problem size and multi-modality unnecessarily estimated initials conditions were assumed the same for every of the 64-experiments.

The problem was solved in 20 independent instances by each solver: ACOmi, eSS and MPeSS. For this problem we considered a larger budget of $1.5 \cdot 10^5$ FEs. The budget for MPeSS was split int 6 phases. The first 5 with increasing values for α and a final round with eSS configured as MINLP solver.

ACOmi and eSS were occasionally able to find reasonable solutions. In contrast to previous cases, ACOmi found slightly better results (see Figure S.21 in the supplementary materials). However, the MPeSS strategy was again the winner, showing the best distribution of results (convergence curves are given in the supplementary materials, Figures S.19 and S.20).

In Figure 3.6 we show, for the best solutions (cost function under 65.0) the goodness of fit (cost function) obtained by each independent optimisation run as a function of the of the number of active variables, *i.e.* the number of binary variables plus the number of continuous parameters. Here we considered solutions in which the final objective function value is up to two times worse than best found. In general, one applies Occam's razor, *i.e.* we seek the simplest model which can explain the available data satisfactorily. The best model structure (see Figure 3.7) achieved a RMSE of 0.1211. Comparing with other solutions, it shows a good balance between goodness of fit and complexity (see Figure 3.6). Model structures for models B,C,D,E and F (Figures S.27 to S.31) along with goodness of fit measures (Figure S.25) are given in the supplementary materials.

Despite the uncertainty in the completeness of the PKN and the uncertainty in the experimental data, we are able to find relatively simple mechanistic models which explain the data. The agreement between the simulation and the experimental data is qualitatively and quantitatively good with the transient behaviour of phosphorylated proteins being well captured by the dynamic model depending on the different stimuli and inhibitors (trajectories available in the supplementary materials, Figures S.32 to S.35).

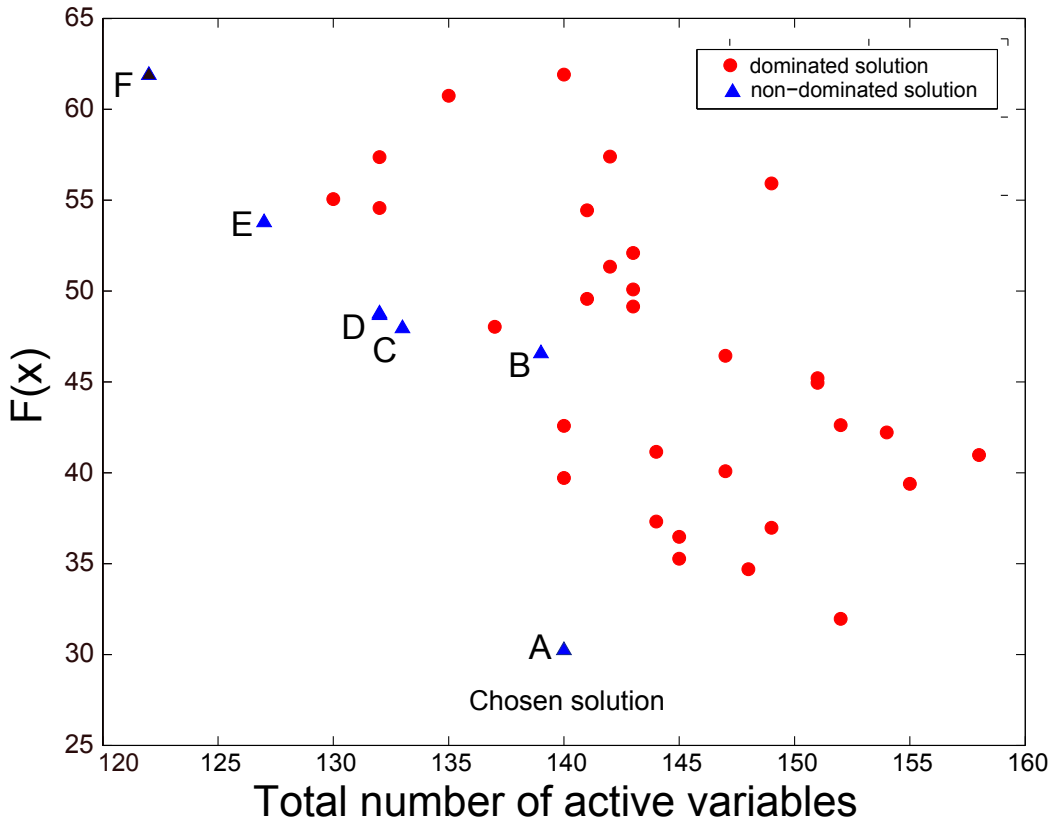


Figure 3.6: Case study 3 (HepG2): This figure shows the Pareto front for the trade-off between the goodness of fit obtained by each independent optimisation run and the number of active variables (number of active binary variables plus the number of active continuous parameters), which is a proxy for model complexity. The chosen solution shows a good balance between goodness of fit (RMSE of 0.121) and complexity.

3.4 Conclusion

In this contribution, we apply a mixed-integer global optimization approach to reverse engineer logic-based ODE models from time-course data. The problem is stated as simultaneously finding the binary variables that determine the model structure and its associated continuous parameters. Further, to improve computational efficiency, we present a relaxed non-linear programming reformulation of the problem that allows us to find good initial points for the MINLP problem.

With our approach, we are able to find a number of solutions which describe the

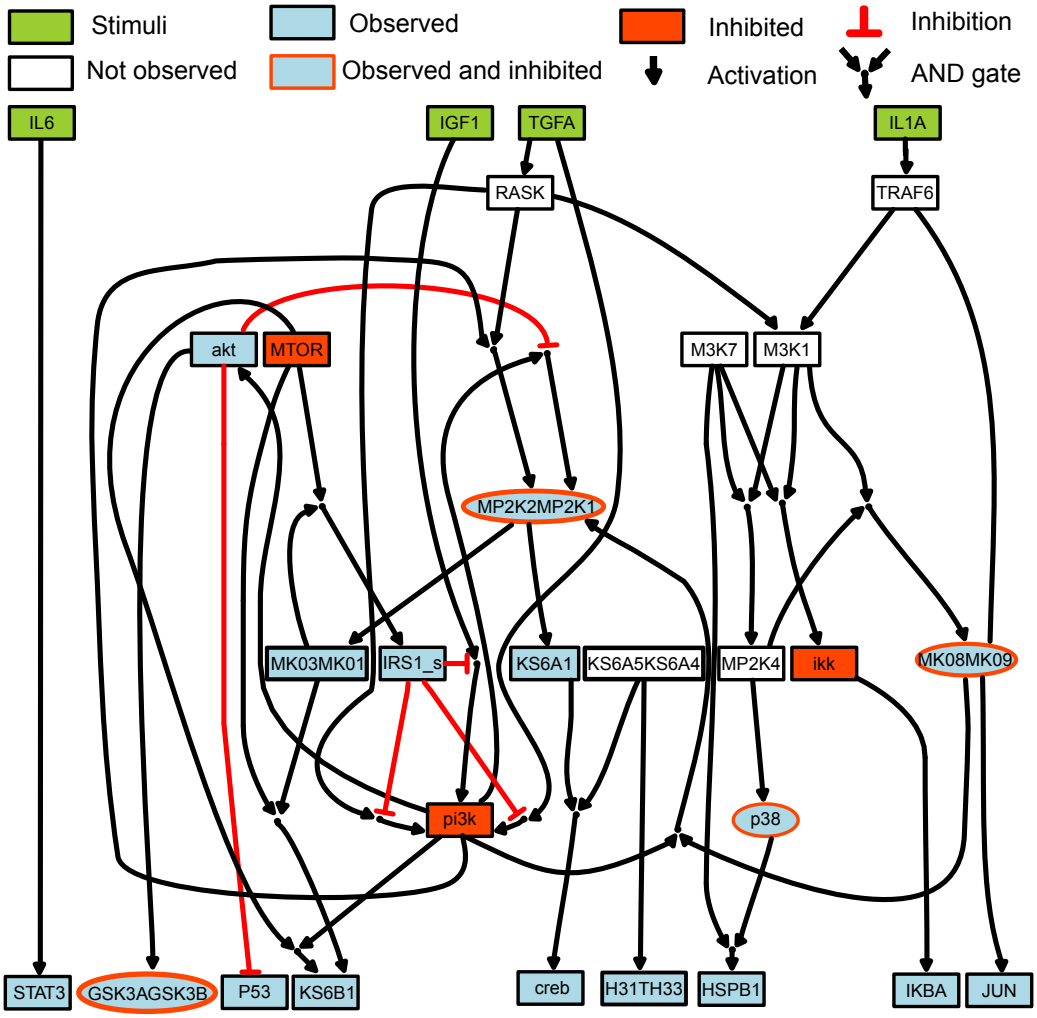


Figure 3.7: Case study 3 (HepG2): The network of solution A presents a good balance between goodness of fit and complexity (see Figure 3.6). The network was plotted with the CellNOptR software [157]. This solution has a squared error of 30.228 and RMSE value of 0.1211. Trajectories for all the states can be found in the supplementary materials in Figures S.23 to S.26.

data satisfactorily. It is important to highlight that the lack of unique solutions is common in reverse engineering problems. Even in the utopian case of large amounts of perfect data available, the reverse engineering of dynamic models can have non-unique solutions, and this is independent of the method used to recover them. For example, in the case of chemical reaction networks it has been shown that many network configurations can describe the same dynamical behavior [154].

Although the metaheuristic approach we present does not provide guarantees about the global optimality of the solutions, we show, by solving synthetic problems (case studies 1 and 2), that problems of realistic size can be successfully solved with a reasonable effort.

In the third case study, we apply the methods to a large signaling network given real experimental data from a liver cancer cell line (HepG2). Due to its size (109 binary variables and 135 continuous parameters) this is, from the optimization point of view, an extremely challenging problem and illustrates well the capability of the method regarding problems of realistic size. Here we did not recover unique solutions, as was expected due to the lack of structural identifiability typical of these problems: their underdetermined nature [148] and the corresponding indistinguishability and non-uniqueness [154]. Instead, we did find a family of solutions much simpler than the original superstructure containing all likely interactions, with a very good fit to the experimental data. This is illustrated by supplementary Figures S.18 (initial expanded superstructure) and S.23 (family of obtained solutions). This family of solutions has the potential to be exploited by approaches like ensemble modeling [91].

Although the obtained results are very encouraging, future work will focus on further improving the efficiency of the metaheuristic optimization methods by exploiting multi-method cooperation and high-performance computing (parallelization).

Chapter 4

SELDOM: enSEmbLe of Dynamic logic-based Models

This chapter reproduces integrally a work submitted for publication and currently under review process.

4.1 Introduction

Inferring the molecular circuits of the cell from experimental data is a fundamental question of systems biology. In particular, the identification of signaling and regulatory networks in healthy and diseased human cells is a powerful approach to unravel the mechanisms controlling biological homeostasis and their malfunctioning in diseases, and can lead to the development of novel therapies [86,137]. Given the complexity of these networks, these problems can only be addressed effectively combining experimental techniques with computational algorithms. Such network inference (or reverse engineering) efforts [166] have been largely developed for gene regulation [17, 39], and to a lesser extent for signal transduction [86]. Extensive work has been published on the inference of molecular circuits, either as a static network–that is, recovering only the topology of interactions– [17, 39, 169] or as dynamical system [21, 28]. It can be beneficial to tackle this network inference in conjunction with the prediction of data for new conditions, since a precise topology should help in the generation of high quality predictions, and the inability of

model topology to describe a given set of experiments suggests that the model is in some sense wrong or incomplete.

Signal transduction is a very dynamic process, and the identification and analysis of the underlying systems requires dynamical data of the status of its main players (proteins) upon perturbation with ligands and drugs. These experiments are relatively complex and expensive, and there is a trade-off between coverage and throughput [137], so that the problem is often ill-posed, leading to identifiability issues. The problem of handling parametric and structural uncertainty in dynamic models of biological systems has received great attention in systems biology and biotechnology [54,82,107,140]. Inference and identification methods can be used to find families of dynamic models compatible with the available data, but in general these models will still suffer from lack of identifiability in a certain degree [166].

Ensemble modeling can be used to improve the predictive capabilities of models, helping to overcome the fundamental difficulties associated with lack of structural and/or practical identifiability. The usage of ensemble methods is widespread in fields such as machine learning [42], bioinformatics [173], and weather forecasting, but not so much in computational systems biology, although it has been successfully applied in the context of regulatory [84, 164], metabolic [77, 155], and signaling [91] networks. Although there is no universally agreed explanation of the success of ensemble methods as classifiers in machine learning [127], it has been shown that they can improve generalization accuracy by decreasing variance [22], bias [141] or both [23], and the reasons for this are relatively well understood [42]. A common approach for building an ensemble is to train a number of so-called base learners in a supervised manner, using data re-sampling strategies. An example of the application of such methods in biology can be found in [73], where the inference of gene regulatory networks is formulated as a feature selection problem, and regression is performed using tree-based ensemble methods. This approach was recently extended to accommodate dynamics [74].

Ensembles of dynamic systems have been used for many years in weather forecasting. In that community, sets of simulations with different initial conditions (ensemble modeling) and/or models developed by different groups (multi-model ensemble) are combined to deliver improved forecasts [61, 156]. In the context of metabolism, Lee et al [94] have shown how to use ensembles to assess the robust-

ness of non-native engineered metabolic pathways. Using the ensemble generation method proposed in [77], a sampling scheme is used to generate representative sets of parameters/fluxes vectors, compatible with a known stoichiometric matrix. This approach is based on the fact that this problem is typically underdetermined, *i.e.* there are more reactions/fluxes than metabolites. Thus, model ensembles may be generated by considering all theoretically possible models, or a representative sample of it. The use of an ensemble composed by all models compatible with the data has been applied to gene regulatory [84] and signal transduction networks [60].

If the model structure is unknown, the ensemble generation needs to be completely data-driven. A common approach for inferring network structures from data is to use estimations of information-theoretic measures, such as entropy and mutual information. The central concept in information theory is entropy, a measure of the uncertainty of a random variable [146]. Mutual information, which can be obtained as a function of the entropies of two variables, measures the amount of information that one random variable provides about another. The mutual information between pairs of variables can be estimated from a data-set, and this can be used to determine the existence of interactions between variables, thus allowing the reverse engineering of network structure. For early examples of this approach, see e.g. the methods reviewed in [37, 52], which covers different modeling formalisms used in Gene Regulatory Network (GRN). The use of these techniques is not limited to GRNs; they can be applied to cellular networks in general [102]. Detailed comparisons of some of these methods can be found in several studies [7, 17, 71, 149].

De Smet et al [39] have studied the advantages and limitations of several network inference methods, stressing the strategies used to deal with underdetermination. For a recent review of information-theoretic methods, see [169]. Some state-of-the-art information-theoretic methods for network inference are Algorithm for the Reconstruction of Accurate Cellular NEtworks (ARACNE) [101], and its extensions Time-Delay Algorithm for the Reconstruction of Accurate Cellular NEtworks (TDARACNE) [177] and high-order Algorithm for the Reconstruction of Accurate Cellular NEtwork (hARACNE) [76], Context Likelihood of Relatedness (CLR) [51], Maximum Relevance minimum redundancy NETwork (MRNET) [106],

three-way Mutual Information (MI3) [95] and Mutual Information Distance and Entropy Reduction (MIDER) [170], to name a few. All of them are based on estimating some information-theoretic quantity from the data and applying some criterion for determining the existence of links between pairs of variables. While the details vary from one method to another, it is difficult to single out a clearly "best" method. Instead, it has become clear in recent years that every method has its weaknesses and strengths, and their performance is highly problem-dependent; hence, the best option is often to apply "wisdom of crowds" methods, akin to the ensemble approach described above, as suggested by the results of recent DREAM challenges [99, 122]. In this spirit, recent software tools aim at facilitating the combined use of several methods [72].

Here, we present enSEmbLe of Dynamic LOgic Models (SELDOM), a method developed with the double goal of inferring network topologies, *i.e.* finding the set of causal interactions between a number of biological entities, and of generating high quality predictions about the behaviour of the system under untested experimental perturbations (also known as out-of-sample cross-validation). SELDOM makes no *a priori* assumptions about the model structure, and hence follows a completely data-driven approach to infer networks using mutual information. At the core of SELDOM is the assumption that the information contained in the available data will not be enough to successfully reconstruct a unique network. Instead, it will be generally possible to find many models that provide a reasonable description of the data, each having its own individual bias. Hence SELDOM infers a number of plausible networks, and uses them to generate an ensemble of logic-based dynamic models, which are trained with experimental data and undergo a model reduction procedure in order to mitigate overfitting. Finally, the simulations of the different models are combined into a single ensemble prediction, which is better than the ones produced by individual models.

The remaining of this paper is organised as follows. First, the Methods section provides a step by step description of the procedure followed by SELDOM. Then a number of experimental and *in silico* case studies of signaling pathways of different sizes and complexity are presented. In the Results and Discussion section the performance of SELDOM is tested on these case studies and benchmarked against other methods. We finish by presenting some conclusions and guidelines for future

work.

4.2 Methods

The SELDOM workflow, outlined in Figure 4.1, combines elements from information theory, ensemble modelling, parametric dynamic model identification, logic-based modeling and model reduction. The final objective is to provide high quality predictions of dynamic behavior even for untested experimental conditions. The method starts from time-course continuous experimental data ($\tilde{y}$) and uses DDNs as intermediate scaffolds. The workflow can be roughly divided into the following 5 steps:

- Dense DDN inference using Mutual Information (MI) from experimental data $\tilde{y}$: build an adjacency (dense DDN) matrix based on the mutual information of all pairs of measured variables.
- Sampling of DDNs: sample $n_{\mathcal{M}}$ DDNs based on the MI.
- Independent model training: parametric identification of a set of ODE models based on the DDNs.
- Independent model reduction: iterative model reduction procedure of the individual models via a greedy heuristic.
- Ensemble prediction: build ensemble of models to obtain predictions for state trajectories under untested experimental conditions.

The term network topology is defined here as a directed graph G. A directed graph (digraph) is a graph where all the edges are directed. The term node or vertex refers to a biological entity such as a protein, protein activity, gene, etc. A directed edge (interaction) starting from node v_i and pointing to v_j implies that the behavior of node v_i interferes with the behavior of node v_j. In this case, v_i is said to be adjacent to v_j. The in-degree of node v_i ($deg^-(v_i)$) is the number of edges pointing to v_i. The directed graph G is composed by the ordered pair $G(V(G), E(G))$, where $V(G)$ is the set of n vertices and $E(G)$ is the set of m edges.

The input to the SELDOM algorithm is an experimental data-set formatted as a Minimum Information for Data Analysis in Systems biology (MIDAS) file [136] and the maximum in-degree ($deg^-(v_i)$) allowed for each node in the networks sampled. The MIDAS file should specify for each experiment the observed signals, the observation times and the treatments/perturbations applied. Two types of perturbations are currently supported: inhibitors and stimuli. These are typical in most experimental studies of signaling pathways, where inhibitors are e.g. small molecules blocking kinase function, and stimuli are upstream ligands (e.g. hormones) whose initial concentration can be manipulated.

4.2.1 Mutual Information

The mutual information $\mathrm{MI}(\tilde{y}_i, \tilde{y}_j)$ between two random variables $\tilde{y}_i$ and $\tilde{y}_j$ is a measure of the amount of information that one random variable contains about another. It can also be considered as the reduction in the uncertainty of one variable due to the knowledge of another. It is defined as follows:

$$\mathrm{MI}(\tilde{y}_i, \tilde{y}_j) = \sum_{\epsilon=1}^{n_\epsilon} \sum_{s=1}^{n_s^\epsilon} p(\tilde{y}_i^{\epsilon,s}, \tilde{y}_j^{\epsilon,s}) log \, \frac{p(\tilde{y}_i^{\epsilon,s}, \tilde{y}_j^{\epsilon,s})}{p(\tilde{y}_i^{\epsilon,s}) p(\tilde{y}_j^{\epsilon,s})} \tag{4.1}$$

where $\tilde{y}_i$ and $\tilde{y}_j$ are discrete random vectors with probability mass functions $p(\tilde{y}_i)$ and $p(\tilde{y}_j)$, and log is usually the logarithm to the base 2, although the natural logarithm may also be used.

Since mutual information is a general measure of dependency between variables, it can be used for inferring interaction networks: the stronger the interaction between two network nodes, the larger their mutual information. If the probability distributions $p(\tilde{y}_i)$ and $p(\tilde{y}_j)$ are known, $\mathrm{MI}(\tilde{y}_i, \tilde{y}_j)$ can be derived analytically. In network inference applications, however, this is not possible, so the mutual information must be estimated from data, a task for which several techniques have been developed [151].

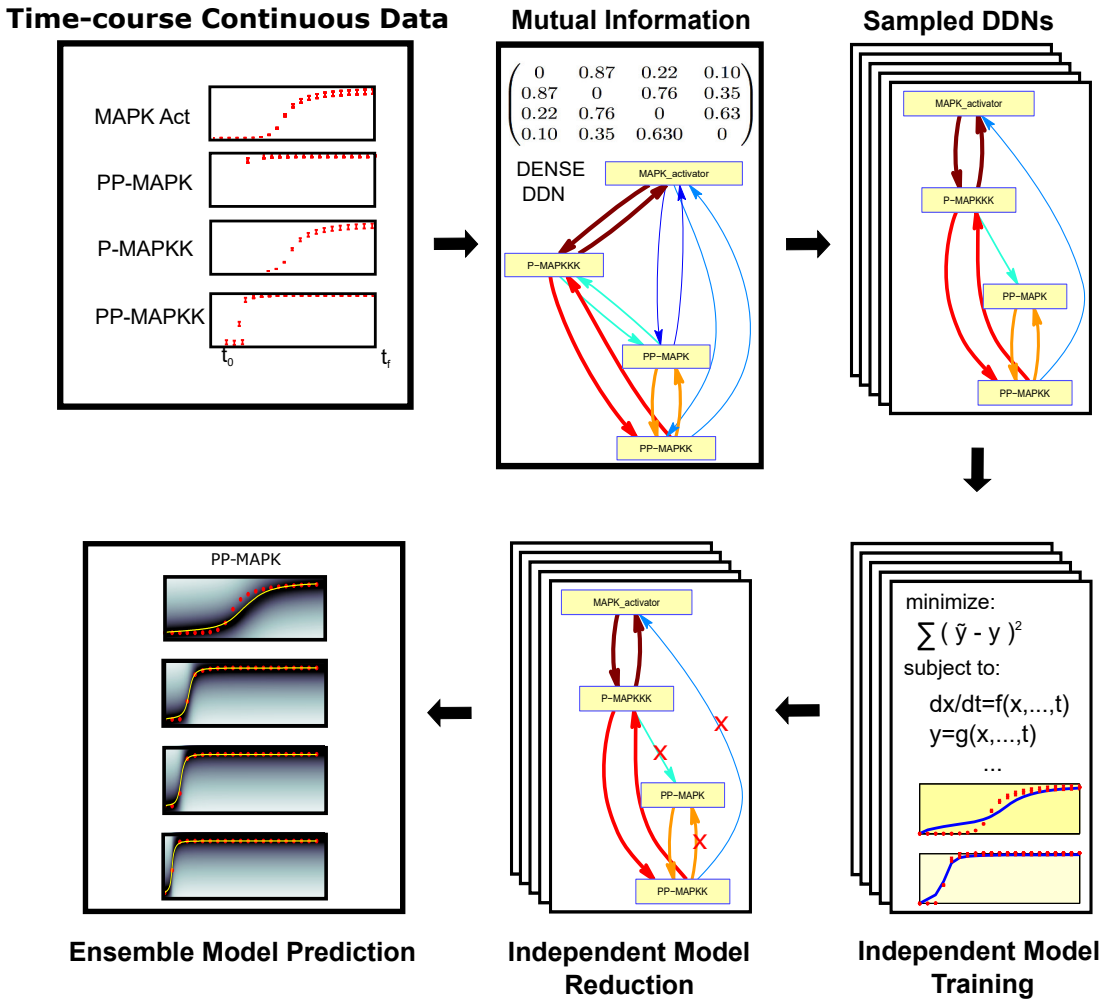


Figure 4.1: **SELDOM workflow.** The experimental data is used to build an adjacency (a dense DDN) matrix based on the mutual information of all pairs of variables. Through a simple sampling scheme, and limiting the maximum in-degree for each node, a set of more sparse DDNs are generated. Each individual DDN is then used as an scaffold for independent model training and model reduction problems. The resulting models are used to form an ensemble which is able to produce predictions for state trajectories under untested experimental conditions.

4.2.2 Sampling Data-Driven Networks

Whatever the approach used to estimate the MI, estimation leads to errors, due to factors such as limited measurements or noisy data. Therefore, it is often the case that MI is over-estimated, which results in false positives. Network inference

methods usually adopt strategies to detect and discard false positives. For example, ARACNE uses the data processing inequality, which states that, for interactions of the type $X \rightarrow Y \rightarrow Z$, it always holds that $\mathrm{MI}(X,Y) \geq \mathrm{MI}(X,Z)$. Thus, by removing the edge with the smallest value of a triplet, ARACNE avoids inferring spurious interactions such as $X \rightarrow Z$. However, this in turn may lead to false negatives.

In the present work, we are interested in building DDNs that are as dense as possible, in the sense that these should ideally contain all the real interactions, this leads to containing some false positives too (the issue of the false positives will be handled in the independent model reduction step). However, the subsequent dynamic optimization formulation used to train the models benefits from limiting the number of interactions (i.e. the number of decision variables grows very rapidly with the in-degree).

To find each DDN, we build an adjacency matrix using the array $\mathrm{MI}(\tilde{y}_i, \tilde{y}_j)$. Each column j represents the edges starting from v_i and pointing to v_j. From this vector we iteratively select as many edges as the maximum in-degree (a predefined parameter of the method). In each selection step, an edge is chosen with a probability proportional to $\mathrm{MI}(\tilde{y}_i, \tilde{y}_j)$. This process is repeated for every node.

4.2.3 Independent Model Training

The DDNs obtained in the previous step represent a set of possible directed interactions. In order to obtain an objective function for model calibraton (parametric identification) a mathematical description of the model dynamics must be chosen. Here, we rely in multivariate polynomial [90,172] interpolation as it is particularly well-suited to represent signaling pathways.

This technique was initially developed with the aim of facilitating the transformation of Boolean models into ODE based time-course descriptions and is able to describe a wide range of behaviours. A multivariate polynomial is able to represent any type of combinatorial interaction (OR, AND, XOR, etc).

For each edge $e_{ij} \in E(G)$, a function $H_{\phi_{ij}}$ describes the type of nonlinearity that governs the relation between an upstream regulator $x_k(t)$ and the behaviour of a downstream variable x_i described by $\dot{x}_i$. Nevertheless, we have chosen to use the

normalized Hill function because, apart from being able to describe other simpler behaviours (such as Michaelis Menten type kinetics), it is also able to represent the switch-like behaviour seen in many molecular interactions [172].

This framework is very general and requires very few assumptions about the system under study. This comes at the cost of a large number of parameters that need to be estimated. Formally, we describe the parametric identification problem (parameter estimation in dynamic models) as:

$$
\begin{aligned}
\underset{n,k,\tau,w}{\text{minimize}} \quad & F = \sum_{\epsilon=1}^{n_\epsilon} \sum_{o=1}^{n_o^\epsilon} \sum_{s=1}^{n_s^{\epsilon,o}} (\tilde{y}_s^{\epsilon,o} - y_s^{\epsilon,o})^2 \\
\text{subject to} \quad & \\
& N_i = deg^-(v_i) \\
& \phi_i = \{j \mid e_{ij} = 1\},\ i = 1, \ldots, n,\ j = 1, \ldots, n \\
& H_{\phi_{ik}} = \frac{\bar{x}^{n_{\phi_{ik}}}}{\bar{x}^{n_{\phi_{ik}}} + k^{n_{\phi_{ik}}}} \cdot (1 + k^{n_{\phi_{ik}}}) \\
& \bar{B}_i = \sum_{x_{i1}=0}^{1} \ldots \sum_{x_{iN_i}=0}^{1} \left[w_{x_{i1},\ldots,x_{iN_i}} \cdot \prod_{k=1}^{N_i} \left(x_{ik} H_{\phi_{ik}} + [1 - x_{ik}][1 - H_{\phi_{ik}}] \right) \right] \\
& \dot{\bar{x}}_i = (\bar{B}_i - \bar{x}_i) \cdot \frac{1}{\tau_i} \\
& \bar{x}_i(t_0) = \bar{x}_{i0} \\
& y = g(\bar{x}, n, k, \tau, t) \\
& 0 \leq w_i \leq 1 \\
& \text{LB}_n \leq n \leq \text{UB}_n \\
& \text{LB}_k \leq k \leq \text{UB}_k \\
& \text{LB}_\tau \leq \tau \leq \text{UB}_\tau,
\end{aligned}
\tag{4.2}
$$

where w, n, k and τ are the continuous parameters needed for training the dynamic model. These parameters are limited by upper and lower bounds (e.g. LB_k). The model dynamics ($\dot{\bar{x}}$) are given by the function f. This set of differential equations varies according to the network derived from the mutual information. Finally, the system of differential equations has to be solved to obtain the simulated data. The objective function is the squared difference between the model predictions (y) and

the experimental data ($\tilde{y}$). The goal is to minimize this cost function for every experiment (ϵ), observed species (o) and sampling point (s). The model prediction y (obtained by simulation) is a discrete data set given by an observation function g of the model dynamics at time t.

The variables w define the model structure. We highlight that this representation can reproduce several behaviours of interest (see Table 4.1). For example, if we consider that a signaling state in the model is controlled by two regulators, an AND type behaviour would be defined by setting $w_{i,1,1}$ to 1 and the other w's ($w_{i,0,0}$,$w_{i,0,1}$ and $w_{i,1,0}$) to 0. On the other hand, the OR gate can be represented by setting $w_{i,1,0}$,$w_{i,0,1}$ to 1 and $w_{i,1,1}$. and $w_{i,0,0}$ to 0. By linear combinations of these terms it is possible to obtain any of the 16 gates that can be composed by two inputs.

Recently we have shown [66] how to train a more constrained version of this problem using MINLP. Here, due to its size, the problem is first relaxed into a NLP problem. The corresponding parameter estimation problem is non-convex, so we use the scatter search global optimization method [46] as implemented in the MEtaheuristics for bIoinformatics Global Optimization in R (MEIGOR) toolbox [45].

Several studies that have considered simultaneous network inference and parameter estimation have chosen discretization methods for the solution of the IVP [21, 28]. This has some advantages regarding the computational tractability, but forces the $\dot{x}$ values to be estimated directly from noisy measurements, which is specially challenging when samples are sparse in time. Here, to avoid this problem, the IVP is solved with the CVODE solver from the SUNDIALS package [68].

4.2.4 Independent Model Reduction

Model reduction is a critical step in SELDOM. The underlying rationale is twofold: (i) we are interested in reducing the network to keep only interactions that are strictly necessary to explain the data (feature selection); (ii) following Occam's razor principle, it is expected that the ideal model in terms of generalization is the one with just the right level complexity [43].

Here, we have chosen a simple heuristic that has proved very effective. This heuristic is partially inspired by the work of Sunnaker et al [152], where a search tree starting from the most complex model is used to find the complete set of all the simplest models by iteratively deleting parameters. In contrast, here we use a greedy heuristic which does not guarantee that the simplest model is found. Nevertheless, this helps to maintain diversity in the solutions and guarantees that spurious edges are not considered. Furthermore, it drastically reduces the computational time needed to find the simplest solution. The iterative model procedure is described in Algorithm 1. At each step (edge), the constraint $H_{\phi_{ik}}$ is set to 0 (see Table 4.1) and the model is trained with a local search using Dynamic Hill Climbing (DHC). To avoid potential bias caused by the model structure, edges are deleted in a random order.

x_1	x_2	$\bar{B}_i = ...$	$\bar{B}_i^* = ...$
0	0	$w_{i,0,0} \cdot (1 - H_{\phi_{i1}}) \cdot (1 - H_{\phi_{i2}}) + ...$	$w_{i,0,0} \cdot [1 - H_{\phi_{i1}}] \cdot 1 + ...$
0	1	$w_{i,0,1} \cdot (1 - H_{\phi_{i1}}) \cdot H_{\phi_{i2}} + ...$	$0 + ...$
1	0	$w_{i,1,0} \cdot H_{\phi_{i1}} \cdot (1 - H_{\phi_{i2}}) + ...$	$w_{i,1,0} \cdot H_{\phi_{i1}} \cdot 1 + ...$
1	1	$w_{i,1,1} \cdot H_{\phi_{i1}} \cdot H_{\phi_{i2}}$	0

Table 4.1: The function multivariate polynomial interpolation $\bar{B}_i$ is simplified by setting $H_{\phi_{ij}}$ to 0 which results in function $\bar{B}_i^*$. In practice this is the equivalent of removing the edge e_{ij}. The remaining parameters are then estimated starting from the best known solution. If the new simpler solution is better from the AIC point-of-view, it is accepted and the heuristic proceeds on trying to reduce the model further.

To decide about the new simplified model, we use the AIC, which for the purpose of model comparison is defined as:

$$AIC = 2K + n \cdot ln(F), \tag{4.3}$$

where K is the number of active parameters. The theoretical foundations for this simplified version of the AIC can be found in [25].

Data: Time-course continuous data $\tilde{y}$, a graph $G_a(V, E)$ and the optimal parameters (n, k, τ, w)

Result: A simplified graph $G_a(V, E)*$

for *each* $e_{\phi_{ik}} \in G_a$ **do**

$$\underset{n^*,k^*,\tau^*,w^*}{\text{minimize}} \; F = \sum_{\epsilon=1}^{n_\epsilon} \sum_{o=1}^{n_o^\epsilon} \sum_{s=1}^{n_s^{\epsilon,o}} (\tilde{y}_s^{\epsilon,o} - y_s^{\epsilon,o})^2$$

subject to

$H_{\phi_{ik}} = 0$

...

if $\text{AIC}(n^*, k^*, \tau^*, w^*) < \text{AIC}(n, k, \tau, w)$ **then**

$E_a \leftarrow E_a \setminus e_{\phi_{ik}}$

$\{n, k, \tau, w\} \leftarrow \{n^*, k^*, \tau^*, w^*\}$

end

end

Algorithm 1: Greedy heuristic used to reduce the model. At each step of the model reduction the new simpler solution is tested against the previous more complex one using the AIC.

4.2.5 Ensemble Model Prediction

To generate ensemble predictions for the trajectories of state x_i, SELDOM uses the median value of x_i across all models for a given experiment i_{exp} and sampling time t_s. This is the simplest way to combine a multi-model ensemble projection. More elaborate schemes for optimally combining individual model outputs exist. Gneiting et al. [56] point out that such statistical tools should be used to obtain the full potential of a multi-model ensemble. However, the selection of such weights requires a metric describing the model performance under novel untested conditions (*i.e.* forecasting), and finding such metric is a non trivial task. For example, in the context of weather forecasting, Tebaldi et al [156] point out that, in the absence of a metric to quantify model performance for future projections, the usage of simple average is a valid and widely used option that is likely to improve best guess projections due to error cancellation from different models.

4.2.6 Implementation

SELDOM has been implemented mainly as an R package (together with calls to C solvers) and can be installed and run in large heterogeneous clusters and supercomputers. The model training and model reduction are embarrassingly parallel tasks which are automated using shell scripts and a standard queue management system. In addition to the parallelization layer at the level of individual model training and reduction, the simulation of each experiment is implemented as parallel individual threads using openMP [34] exploiting a multi-core processors.

The dynamic optimization problem associated to model trained is solved as a master NLP with an inner IVP. The NLPs are solved using the R package MEIGOR [45], with the evaluation of the objective function performed in C code. The solutions of IVPs are obtained by using the CVODE solver [68].

The experimental data is provided using the MIDAS file format, and it is imported and managed using CellNOptR [158]. The SELDOM code is open source and it is distributed as is (with minimal documentation), along with the scripts needed to reproduce all the results and figures. The main code uses R version 2.15, while Intel compilers were used for the solvers implemented as C/C++ or Fortran codes.

4.2.7 Case studies

To assess the performance of SELDOM, we have chosen a number of *in silico* and experimental problems in the reconstruction of signaling networks. Table 4.2 shows a compact description of some basic properties of these case studies along with a more convenient short name for the purpose of result reporting.

For each case study, two data-sets were derived, one for inference and the second one for performance analysis. We highlight that training and performance assessment data-sets are not just two realizations of the same experimental designs; they were obtained by applying different perturbations, such as different initial conditions or the introduction of inhibitors either experimentally or *in silico*.

Table 4.2: An overview of the characteristics of all case studies approached in this work. The most relevant factors are the number of observed variables, the number of experiments considered for training, the number of experiments considered for prediction and the different maximum in-degrees tested in each case study.

case study	Short name	Reference	Data	N_{obs}	N Train	N Prediction	$deg^-(v_i)$
1a	MAPKp	[70]	*in silico*	4	10	10	$A = 1, B = 2, C = 3$
1b	MAPKf	[70]	*in silico*	13	10	10	$A = 3, B = 4, C = 5$
2	SSP	[96]	*in silico*	13	10	36	$A = 3, B = 4, C = 5$
3	DREAMiS	[29]	*in silico*	2	20	128	$A = 3, B = 4, C = 5$
4	DREAMBT20	[67]	Experimental	54	29	8	$A = 3, B = 4, C = 5$
5	DREAMBT549	[67]	Experimental	52	24	7	$A = 3, B = 4, C = 5$

4.2.7.1 Case studies 1a and 1b: MAPK signaling pathway

Huang et al. [70] developed a model explaining the particular structure of the MAPKs. This is a highly conserved motif that appears in several signaling cascades (ERK, p38, JNK) [78] composed by 3 kinases. Essentially, Huang et al [70] explain how this arrangement of three kinases sequentially phosphorylated in different sites allows that a graded stimuli is relayed in a ultrasensitive switch-like manner.

To create this benchmark, the model shown in Figure 4.2 was used to generate artificial data with no noise. The full system is composed by 12 ODEs. Based in this system, we have derived two case studies, one fully observed (MAPKf) and the second partially observed (MAPKp). The fully observed system is essentially the same as used in [170], while in the partially observed case only one phosporylation state per kinase was considered (MAPK-PP, MAPKK-PP and MAPKKK).

We highlight that the model representation used in SELDOM is particularly suitable to represent such compact descriptions of signaling mechanisms due to the usage of Hill functions. Additionally, looking at partially observed systems is well in line with experimental practice as state-of-the-art methods for studying signaling pathways are typically targeted to particular states (e.g. phosphorylation) of the proteins (e.g. kinases) involved in the signaling pathways.

Both the data-sets used for training and predictions are composed by 10 different experiments, each with different initial conditions and without added noise. The data used for MAPKp case study is a sub-set of the MAPKf data-set.

4.2.7.2 Case study 2: A synthetic signaling network

Resorting to logic-based ODEs, MacNamara et al [96] derived a synthetic model representative of a typical signaling pathway. The goal was to illustrate the benefits and limitations of different simulations for signaling pathways. This model includes three MAPK systems (p38, ERK and JNK1) and two upstream ligand receptors for EGF and TNFα. Apart from different on/off combinations of EGF and TNFα, the model simulations can be perturbed by inhibiting PI3K and RAF.

The training data-set is composed by 10 experiments with different combinations of ligands (EGF and TNFα on and off) and the inhibitors for RAF and PI3K.

Figure 4.2: **MAPK signaling network.** The model by Huang et al. [70] was used to generate pseudo-experimental data for two sub-problems. The first (MAPKp) partially observed (MAPK-PP, MAPKK-PP and MAPKKK), and the second fully observed MAPKf.

The data-set used to assess performance was generated using the synthetic signaling model with the same combinations of EGF and TNFα, but changing the inhibitors. Instead of inhibiting PI3K and RAF, we generate new experiments by considering all other states observed with exception of EGF and TNFα. The final outcome is a validation data-set with 36 experiments.

Both data-sets (training and validation) were partially observed (11 out of 26 variables) and Gaussian noise (with standard deviation $\sigma = 0.05$ and 0 mean) was added. In this case study the inhibitors are implemented as:

$$\dot{\bar{x}}_{\text{inh},i} = (\bar{B}_i - \bar{x}_i) \cdot \frac{1}{\tau_i} \cdot (1 - \text{inh}_i), \tag{4.4}$$

where inh_i is chosen as 0.9.

4.2.7.3 Case study 3: HPN-DREAM breast cancer network inference, *in silico* sub-challenge

This is an *in silico* problem developed by the HPN-DREAM consortium. It is a synthetic problem that replicated the reverse phase protein array (RPPA) experimental technique for studying signalling pathways with multiple perturbations as

realistically as possible. These perturbations often consist in manipulating ligand concentrations and adding small molecule inhibitors. To achieve this, the authors used a large dynamic model of ErbB signaling pathways [29]. The model was partially observed (17 variables) and perturbed with a noise model aimed at reproducing the RRPA experimental technique as accurately as possible. In addition to these 17 variables, 3 dummy variables consisting of noise were included to make the challenge even more difficult. All names in the model were replaced by aliases (eg. AB1, AB2, etc).

The training data-set is composed of 20 experiments obtained by considering different combinations of 2 ligands (off, low and high) and 2 small molecule inhibitors. The data-set used for performance assessment is composed by 128 experiments considering the inhibition of the other 15 observed states not considered in the generation of the training set and different combinations of ligand concentrations (off, low and high).

Regarding the implementation of the inhibitors, we followed the same strategy used in Synthetic Signaling Pathway (SSP) case-study where these are implemented under the assumption that an inhibitor inh_i of state x_i directly affects the concentration of x_i. Such assumption is based on the challenge design and made following the instructions of the challenge developers.

4.2.7.4 Case studies 4a and 4b: HPN-DREAM breast cancer network inference

One of the richest data-sets of this type was recently made publicly available in the context of the DREAM challenges (www.dreamchallenges.org). DREAM challenges provide a forum to crowdsource fundamental problems in systems biology and medicine, such as the inference of signaling networks [67, 122], in the form of collaborative competitions. This data-set comprised time-series acquired under eight extracellular stimuli, under four different kinase inhibitors and a control, in four breast cancer cell lines [67].

The HPN-DREAM breast cancer was composed of two sub-challenges. In the experimental sub-challenge the participants were asked to make predictions for 44 observed phosphoproteins, although the complete data-set was larger. As opposed

to the *in silico* sub-challenge, the participants were encouraged to use all the prior knowledge they could use and the experimental protocol along with the real names of the measured quantities, used reagents, inhibitors, etc.

Using different combinations of inhibitors and ligands (on and off), the authors have generated a data-set comprising 29 experiments. An additional data-set generated with the help of a fourth inhibitor was kept unknown to the participants, which were asked to deliver predictions for several possible inhibitors.

Here, it is assumed that the inhibitors affect mostly the downstream activity of a given kinase. However, it is unknown how it actually influences the kinase concentration or the ability to measure it the mutual information used find DDN variants is computed here as:

$$\mathrm{MIM}_{\mathrm{inh}} = \max\Big(\mathrm{MIM}\big(\tilde{y}\cdot(1-\mathrm{inh}_i)\big), \mathrm{MIM}\big(\tilde{y}\big)\Big) \tag{4.5}$$

where inh_i is a vector of the same size as $\tilde{y}$, filled with 0.9 when the inhibition is applied and with 0 otherwise. Regarding the implementation of the dynamic behaviour, this is performed by modifying $H_{\mathrm{inh},\phi_{ik}}$ of an inhibited species x_k to:

$$H_{\mathrm{inh},\phi_{ik}} = \frac{\bar{x}^{n_{\phi_{ik}}}\cdot(1-\mathrm{inh}_k)}{\bar{x}^{n_{\phi_{ik}}}\cdot(1-\mathrm{inh}_k)+k^{n_{\phi_{ik}}}}\cdot(1+k^{n_{\phi_{ik}}}) \tag{4.6}$$

Due to the computational cost of the approach we have considered only two cell-lines BT20 and BT549.

4.3 Results and discussion

4.3.1 Numerical experiments and method benchmarking

In this section, we describe the numerical experiments carried to show the validity of our ensemble based approach. Besides particular considerations in the data preprocessing or additional constraints added to the DO problem which depend on the prior knowledge existent about the case study at hand, SELDOM has two tuning parameters: the ensemble size and the maximum in-degree allowed in the training process. Thus, besides showing how the method performs and illustrating the process we also wanted to show that the method is relatively robust to the

choice of these parameters and provide guidelines for the choice of such parameters in future applications.

For each case study we have chosen 3 in-degrees (A, B and C) which are shown in Table 4.2 and we have chosen a fairly large ensemble size of 100 models. We remark that, while the choice of the ensemble size was arbitrary, the method is robust with respect to this parameter and performs similarly well with smaller sizes, as shown in Figure 4.6.

To assess performance in terms of training and predictive skills of the model, we use the RMSE:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{\epsilon=1}^{n_\epsilon}\sum_{o=1}^{n_o^\epsilon}\sum_{s=1}^{n_s^{\epsilon,o}}[\tilde{y}_s^{\epsilon,o} - y_s^{\epsilon,o}]^2}{\sum_{\epsilon=1}^{n_\epsilon}\sum_{o=1}^{n_o^\epsilon} n_s^{\epsilon,o}}} \tag{4.7}$$

To assess performance in terms of network topology inference, we have chosen the AUPR curve, where precision (P) and recall are defined as (R):

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4.8}$$

and

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{4.9}$$

where TP and FP correspond to the number of true and false positives, respectively and FN corresponds to the number of false negatives.

Other valid metrics exist, such as the Area Under Receiving Operating Characteristic (AUROC). The Receiving Operating Characteristic (ROC) plots the recall, R, as a function of the false positive rate, FPR, which is defined as

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{4.10}$$

.

However, it has been argued that ROC curves can paint an excessively optimistic picture of an algorithm's performance [36], because a method can have low

precision (i.e. large FP/TP ratio) and still output a seemingly good ROC. Hence we have chosen to use the AUPR measure instead.

4.3.2 Predicting trajectories for new experimental perturbations

To assess the performance of SELDOM, we have run the analysis described in the previous section to all case studies. In most cases the ensemble behaved better than the model with lowest RMSE training value. This effect is particularly evident in the DREAM *in Silico* (DREAMiS) case-study and is illustrated with the help of figure 4.3. Additionally, in a number of case-studies (DREAMiS, DREAM cell-line BT20 (DREAMBT20), DREAM cell-line BT549 (DREAMBT549)) there is little correlation between the training RMSE and the predition RMSE provided that the models are reasonably well trained.

In Figure 4.4, we show the overall picture regarding the predictive skills. Two strategies were considered for the generation of predictions: the best individual model and SELDOM. The RMSE values were normalized by problem and plotted as an heatmap. Additionally, for DREAMiS, DREAMBT20 and DREAMBT549, we added the prediction RMSE values for the top performing participants in the corresponding DREAM challenge. The greatest gain of using an ensemble approach as shown here is in robustness. The effect of the model reduction was relatively small (yet not neglectable) in terms of RMSE for prediction.

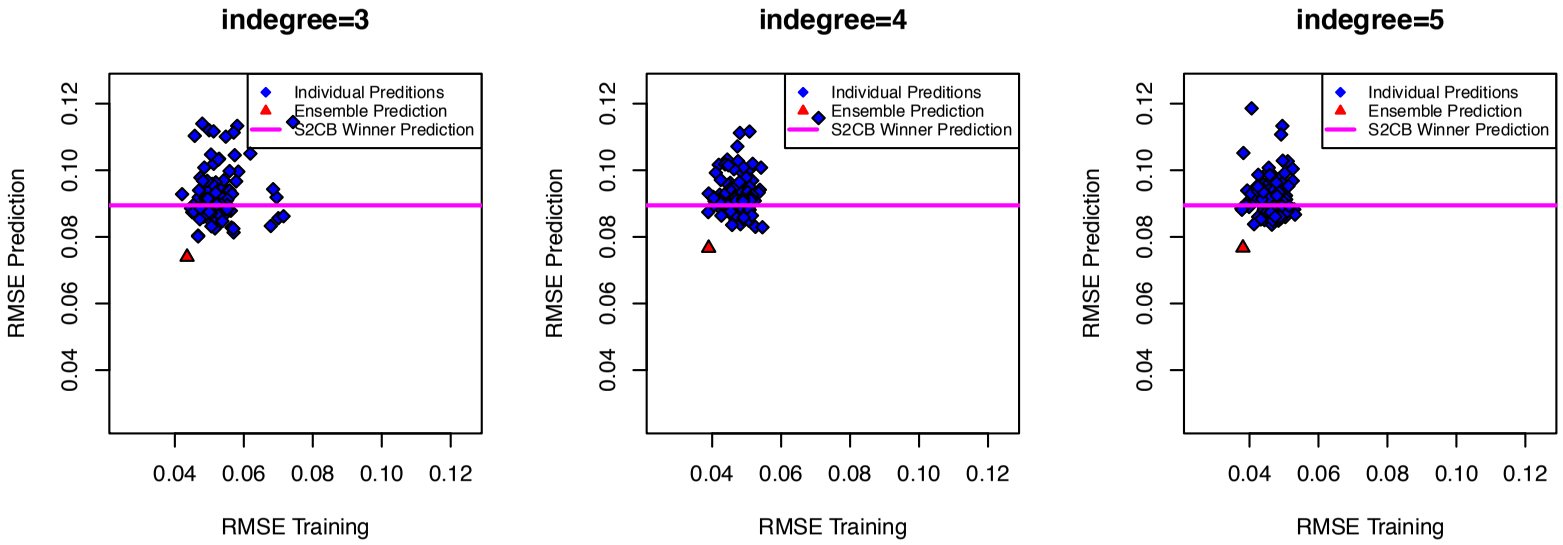


Figure 4.3: The prediction RMSE is plotted here against the training RMSE for each individual model (blue) and the ensemble (red). Additionally the magenta line shows the RMSE prediction value for the top performer of the time-course prediction in the DREAMiS case-study. Although this was not exclusive to the DREAMiS case study there is very little correlation between the training RMSE and prediction RMSE.

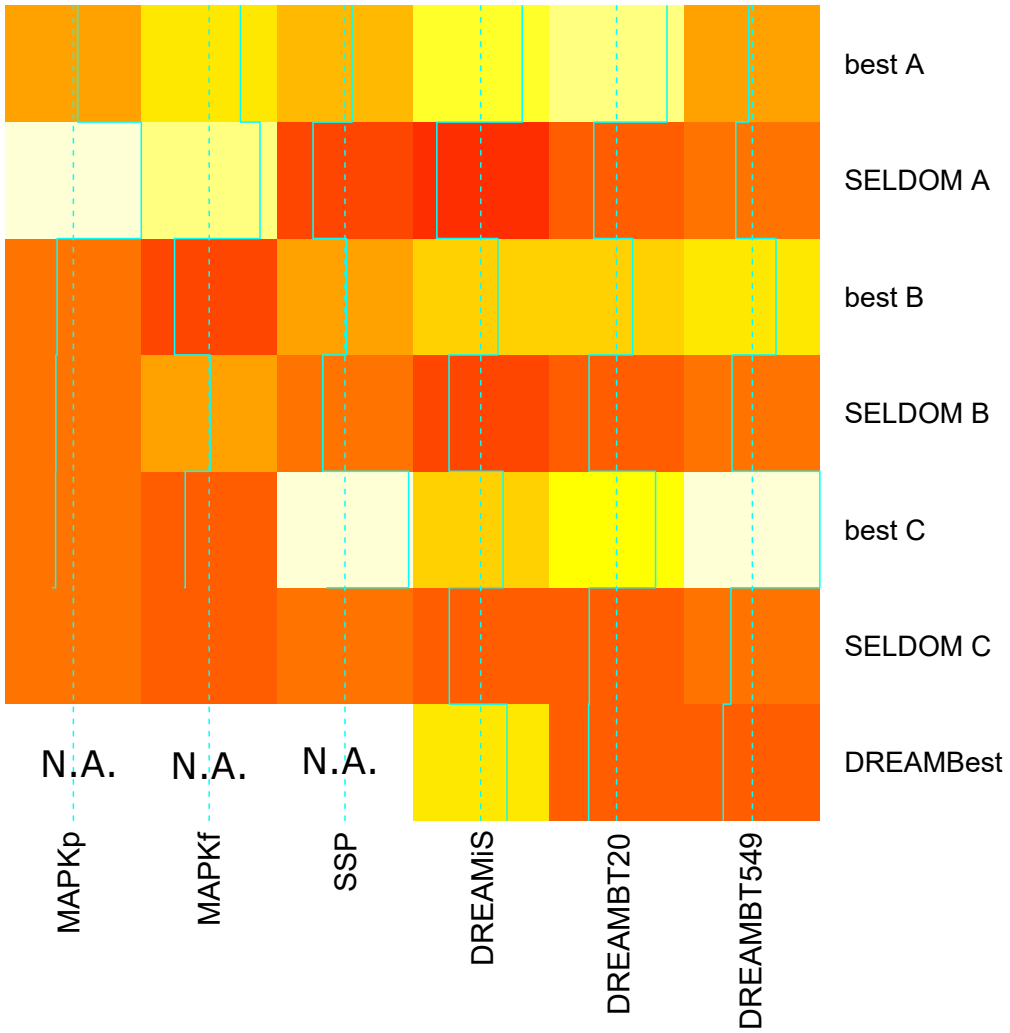


Figure 4.4: **The prediction RMSE values were normalized by case-study and are shown here as an heatmap.** The case studies and methods/method variants are ordered by similarity using hierarchical clustering. SELDOM B and SELDOM C were clearly the most robust strategies doing very well in all problems.

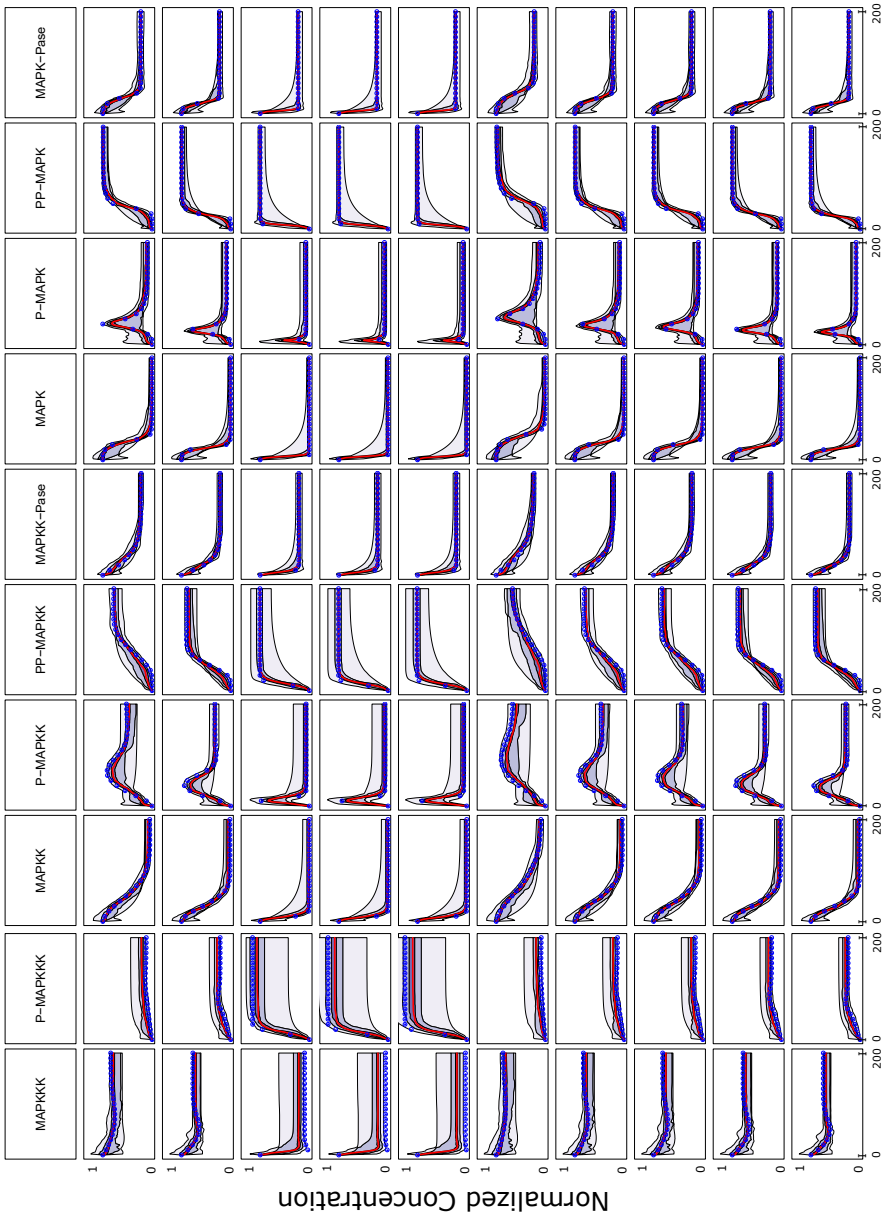


Figure 4.5: **Time course predictions for the MAPKf case study**. The median in red is surrounded by the predicted non-symmetric 20% ,60% and 95% confidence intervals.

The choice of ensemble size parameter affects the predictive skill of the ensemble and the computational resources needed to solve the problem. To verify if this choice was an appropriate one we plotted the average prediction RMSE as a function of the number of models $n_{\mathcal{M}}$ used to generate the ensemble. The average RMSE was computed by sampling multiple models from the family of 100 models available to compute the trajectories. This is shown in Figure 4.6 for the DREAMiS case-study. With the exception of the combination MAPKp/SELDOM A the outcome for all case-studies is that SELDOM would have done similarly well with a smaller number of models and the prediction RMSE *versus* $n_{\mathcal{M}}$ always converged asymptotically. The mediocre results MAPKp/SELDOM A appear to be the result the of a poor choice for the maximum in-degree parameter (A=1) which is too small.

Figure 4.6: **Ensemble predictive skill depending on ensemble size (case study DREAMiS).** This curve was computed by bootstrapping multiple $n_{\mathcal{M}}$ models from the available 100 models, *i.e.* we sampled multiple realizations of the individual predictions for the same ensemble size and computed the average value. These curves converge asymptotically and show that the chosen ensemble size parameter is adequate. Equivalent predictions could have been obtained with smaller ensemble sizes.

4.3.2.1 Ensemble for network inference

To assess the performance of SELDOM for the network topology inference problem, we compared SELDOM with a number methods implemented in the Mutual Information NETworks (MINET) package [105]: MRNET [106], Maximum Relevance minimum redundancy NETwork Backward (MRNETB) [104], CLR [51] and ARACNE [51]. This comparison is particularly pertinent in this case as the estimation of the mutual information is done using the same method and parameterization. However, these methods are not designed to recover directed networks. To surmount this limitation, we have introduced the comparison with two other methods for directed networks, TDARACNE [101] and MIDER [170].

In Figure 4.7, we show the overall results regarding the ability of SELDOM and other network inference methods to reverse engineer the known synthetic networks associated with the models used to generate the data. Comparing with static inference methods, SELDOM behaved consistently well in terms of providing networks with high AUPR score. The sparsest case of SELDOM (A) provided the most interesting results and the network found is comparable to the best solution found by the winning team of the *in silico* sub-challenge.

Without the independent model reduction step, the results were mediocre regarding the inference of the network topology. The independent model reduction is fundamental for the performance of SELDOM as a method for network inference and the information contained in the dynamics can help discard spurious links.

4.4 Conclusions

In this paper we have presented an ensemble method for the generation of dynamic predictions and inference of signaling networks. The method handles the indeterminacy of the problem by generating, in a data-driven way, an ensemble of dynamic models combining methods from information theory, global optimization and model reduction. When making predictions about untested experimental conditions, the ensemble approach was the most robust and most of the times the best option comparing with the individual model predictions. Regarding the network inference problem, the ensemble approach did systematically well in all of the *in*

silico cases considered in this work. This suggests that exploiting the information contained in the dynamics, as SELDOM does, helps the network inference problem allowing to disregard spurious interactions.

The proposed pipeline is flexible and can be adapted in principle to any signaling or gene regulation dataset obtained upon perturbation, even if prior knowledge is not available. At the same time, it is also able to to incorporate prior knowledge about the problem, for instance in the form of constraints (e.g. the small-molecule inhibitors used in the DREAMBT20 and DREAMBT549 case studies). We have tackled the indeterminacy of the problem by generating a family of solutions, although other strategies, based in data-re-sampling methods and supervised learning (similarly to what has been recently proposed by Huynh-Thu et al. [74]), might work well too. A systematic comparison of ensemble generation methods either based in problem structure or data re-sampling techniques should be considered in further work.

All the relevant software used here is available as open source, including the scripts with the implementations of the problems considered. Date files use the MIDAS [136] format.

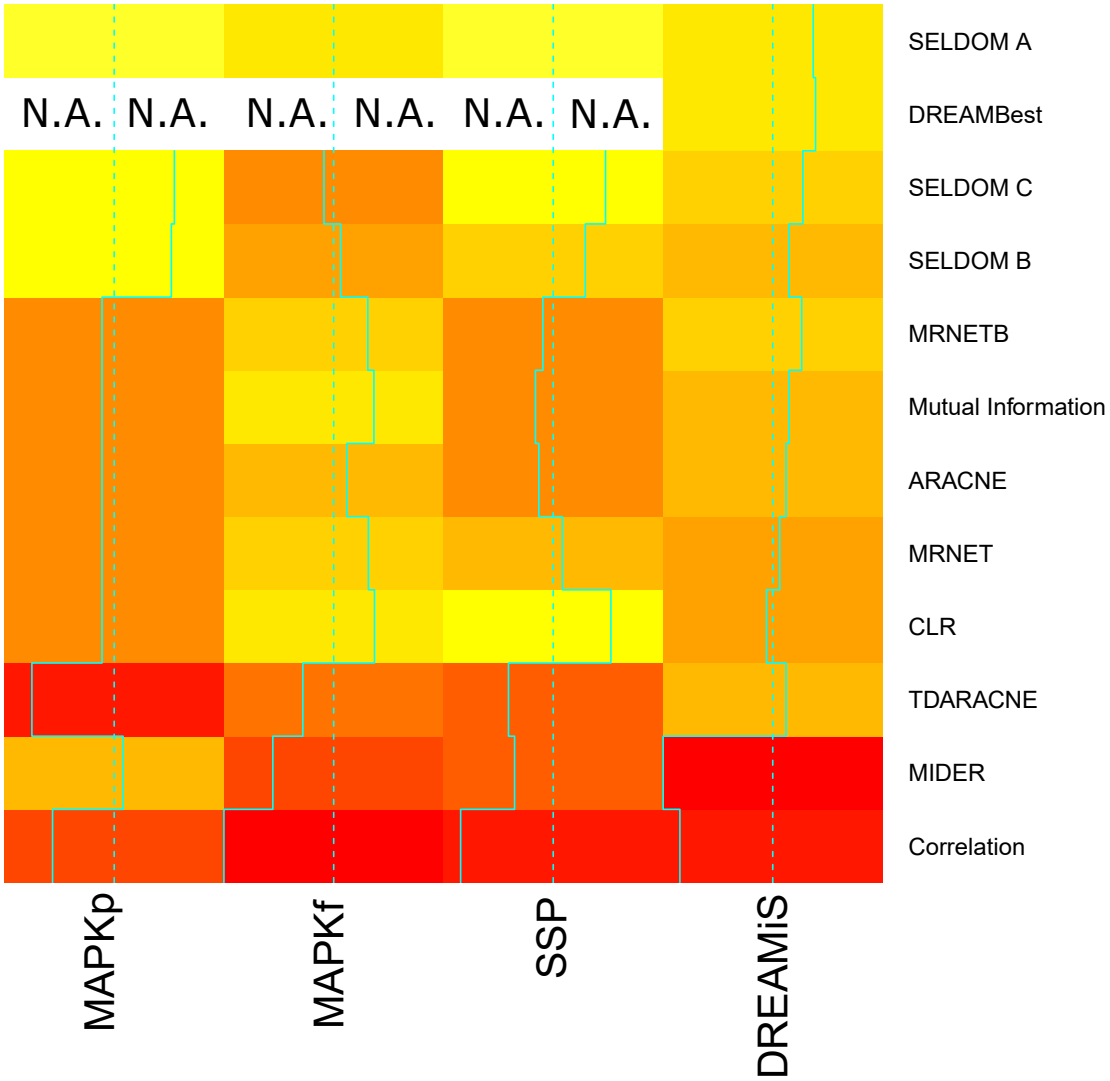


Figure 4.7: **Heatmap with AUPR scores for different methods and case studies.** The AUPR values were normalized by case-study and are shown here as an heatmap. The sparsest version of SELDOM (A) did consistently well in all the case studies. SELDOM B and C did an average job with MAPKf but provided good solutions for all other case-studies. The comparisons are only provided for *in silico* problems with known solution. Additionally, the solution for the top performing team in the DREAM challenge is only available for DREAMiS.

Chapter 5

libAMIGO: A generic library for defining dynamic optimization problems in C

libAMIGO is a C library for implementing and sharing dynamic optimization problems in systems biology. The library was developed because of the need to share problems with other research groups and colleagues in a platform independent and license free manner. The inputs and outputs are well specified and its organization follows a similar structure to that of the AMIGO [11] software in Matlab. Two interfaces are provided, one for R and another for Matlab, which can be used as starting point to build interfaces for other languages.

5.1 Problem Definition

The type of problems considered by AMIGO [11] can be described in a simplified manner as:

$$\begin{aligned} \underset{\theta,u,x_0}{\text{minimize}} \quad & F(y,...) \\ \text{subject to} \quad & \dot{x} = f(x,u,w,\theta) \\ & x(t_0) = x_0 \\ & y = g(x,t,\theta,u) \end{aligned} \tag{5.1}$$

where $\dot{x}$ describes a dynamic system of ODE and g is an observation of x which is obtained by integrating $\dot{x}$ between t_0 (initial time) and t_f (final time). The integration is interrupted at given time points to observe the system (t_s) via an observation function g or to introduce discontinuities in the system (t_u) through control variables (u). Apart from the ODEs, the system behavior is controlled by a number of continuous (θ) or integer/binary (w) parameters, initial conditions x_0 and control variables (u), may or not be partially known *a priory*, and the formulated optimization problems typically tries to find the values for these variables which minimize some criterion defined by F.

The problems considered may be of distinct nature. For example, in a parameter estimation problem, F is typically the squared difference between the simulation and some experimentally observed value, scaled by a weight typically related with the confidence deposited in the accuracy of the observations, for every experiment (ϵ), observed species (o) and sampling point (s):

$$F = \sum_{\epsilon=1}^{n_\epsilon} \sum_{o=1}^{n_o^\epsilon} \sum_{s=1}^{n_s^{\epsilon,o}} \frac{(\tilde{y}_s^{\epsilon,o} - y_s^{\epsilon,o})^2}{\sigma_s^{\epsilon,o}}, \tag{5.2}$$

and our goal is to find θ such that this value is minimized. To provide a general implementation for equation 5.2, we need to manage data from several experiments including initial conditions, observation errors experimental data and implement simulation breaks when perturbations/controls are added to the system (*i.e.* handle discontinuities). In addition most deterministic optimization solvers will require that a gradient is provided, and this is also implemented in libAMIGO. The gradient can be roughly approximated by finite differences [15] or computed more reliably by numerically solving the parameter sensitivity equations, which can be done efficiently e.g. CVODES [145]. The parametric sensitivities($S_i(t)$) are given by the solution of $\dot{S}_i$:

$$\dot{S}_i = \frac{\partial f}{\partial y} S_i + \frac{\partial f}{\partial p_i}, \tag{5.3}$$

where $\frac{\partial f}{\partial y}$ are the partial derivatives of the RHS equations with respect to the model states, and $\frac{\partial f}{\partial p_i}$ are the partial derivatives of the RHS equations with respect to the model parameters we want to estimate.

The library can also be used to implement other more general dynamic optimization problems. An example would be the case where u (sometimes also the values for t_u) should make the system behave optimally in some sense. Such formulations are commonly applied for experimental design, *i.e.* find the most informative experiment possible or in industrial applications, e.g. maximize the production of a given compound [14].

5.2 Implementation

The implementation of libAMIGO was guided by three main objectives: portability, efficiency, scalability. Regarding efficiency, the implementation of libAMIGO circles around CVODES [145] from the SUNDIALS suite. CVODES is a solver for the solution of dynamic systems of ODE, that also enables sensitivity analysis. This tool is implemented in C and is currently the state of the art for this purpose, being actively maintained and improved.

The advantage of using libAMIGO instead of CVODES alone is the reduction in the time needed to implement a problem. We establish well defined inputs for the dynamic optimization problem, perform memory management, implement a number of common objectives and simulation tasks (e.g. sensitivity analysis), and provide two interfaces, one in Matlab (AMIGO2) and the other in R (SELDOM). These interfaces can be extended to other scripting environments with C interfacing capabilities, such as Python or Julia. Memory allocation and deallocation are managed by the library being the programmer responsible for populating the memory.

Regarding portability, the code can be easily compiled for Linux, Windows and MacOS. Additionally, we have chosen to use only components written in open-source C or Fortran that can be compiled with GNU compilers. Nevertheless, proprietary compilers (e.g. Intel compilers) can also be used.

The code is organized around two main C structures: the AMIGO_Model and the AMIGO_problem, which illustrated in Figure 5.1. Each AMIGO_model contains all the information about a given experiment: the initial conditions, experimental data and results. AMIGO_Problem stores the pointers to all AMIGO_models and the lower and upper bounds for all global parameters (which affect all ex-

periments), local local parameters (affect only some experiments, e.g. estimated initial conditions), and other information needed for the optimization problems. All memory allocation and deallocation is dynamic and handled by libAMIGO by means of simple commands.

Regarding scalability, the implementation of libAMIGO is thread safe and built to make use of MPI [58] and openMP [35]. Parallelism using openMP is implemented by default and is achieved by parallelizing the loop that simulates all experiments. However, other smarter ways of implementing this task can be achieved. For example, in [116] libAMIGO was used in conjuction with openMP [35] to parallelize a more coarse grained loop. This could be easily achieved by creating multiple copies of the data structures allowing parallelization in a shared memory environment.

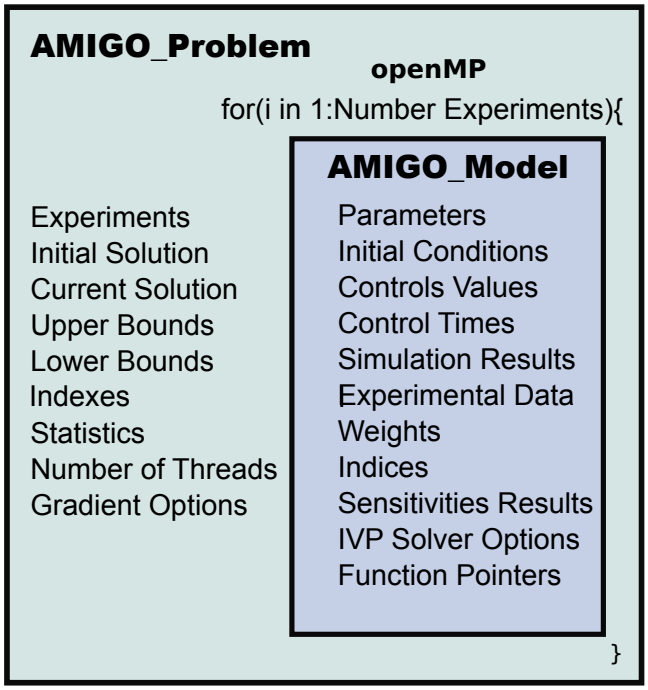


Figure 5.1: Data structures used in libAMIGO. All the information necessary for the simulation of an experiment is contained in the AMIGO_model which is completely independent from other experiments. This makes usage of openMP [35] to parallelize certain loops trivial. The information needed to interpret the overall results from all experiments is gathered in AMIGO_Problem.

5.3 Applications

5.3.1 AMIGO2

AMIGO2 is a large update to the first version of AMIGO [11] tool box. This toolbox gathers a large collection of numerical methods for simulation and optimization of systems biology problems, namely: identifiability analysis, optimal experimental design and parameter estimation. The last three tasks are formulated as NLP optimization problems. The libAMIGO was developed for AMIGO2.

The main goal of having a library independent of Matlab was that the problems could easily be exported in a platform independent manner. For example, libAMIGO implements the log-likelihood, which allows the parameter estimation problem to be solved with a C or Fortran optimization solvers. In [116–118] libAMIGO was used to run problems exported by AMIGO [11] in Matlab.

However, libAMIGO can also be used inside of AMIGO. Implementing the whole dynamic optimization problem in C gives us easy access to NLP solvers available in C or Fortran which avoids the overhead of constantly using callbacks from C to Matlab. Additionally embarrassingly parallel tasks can be easily parallelized without the need of proprietary software by means of openMP [35] and MPI [58].

The structure of the interface built between Matlab, AMIGO and libAMIGO is depicted in Figure 5.2. The experimental data is processed and RHS equations in C are generated by AMIGO. Three so-called execution modes that use libAMIGO are allowed in AMIGO2: costMex, fullMex and fullC. The costMex mode evaluates the cost function. The fullMex version can be used to run local searches with a nonlinear least square estimator (NL2SOL [40]) without the need of Matlab callbacks. Finally, the fullC is meant to be run without the usage of Matlab. While using the costMex and fullMex, the RHS file is compiled within MALTAB using the Matlab EXecutable (MEX) engine along with a C interface designed for this purpose. On the other hand, while using the fullC execution mode, a RHS file is generated, an illustrative main file and instructions to compile with GNU compilers are provided.

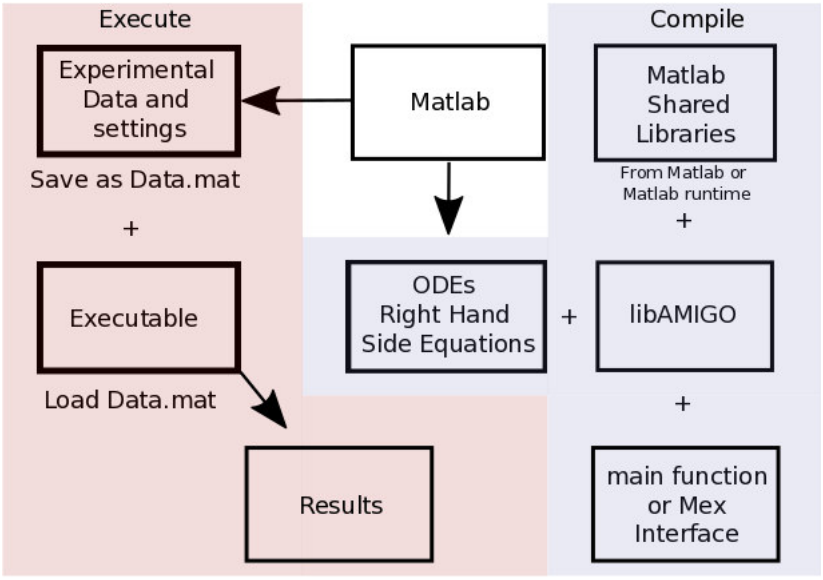


Figure 5.2: The sructure of the interface built for libAMIGO. Matlab in conjunction with AMIGO2 is used to generate the RHS equation and necessary data structures. The MEX interface written in C is compiled along with Matlab shared libraries and the RHS equations. It is possible to call the interface from C programs as long as the data has been saved *a priori* in a Matlab data file.

5.3.2 BioPreDyn-Bench

From the optimization point of view, the parametric identification is an interesting and open problem. Due to its non-convexity, even relatively small problems can not be solved with guarantees of optimality in a reasonable amount of time. Due to the lack of a standardized definition for the full problem (model and experimental design), it is hard and error-prone to implement such dynamic optimization problems solely from literature. More relevantly, this typically requires some field specific knowledge and collaboration with groups with know-how in optimization is often hindered because problems can not be easily shared.

The Systems Biology Markup Language (SBML) was successful in providing a standardized format for defining models of systems biology. Not only it works well, but is also widely accepted as the *de facto* standard by the community. However, systems biology problems as the ones considered in this work, often require more complex considerations involving the experimental design. Efforts in this direction have been made. An example is the standard Minimal Information Required in

the Annotation of Models (MIRIAM) [93]. Despite of these the efforts, to the best of our knowledge, no tools for simulation supporting MIRIAM have been made available.

The BioPreDyn-Bench [168] is a suite of bechmark problems for dynamic modeling in systems biology. This suite is composed of 6 problems (problems B1 to B6) which try to capture different problem sizes and biological aspects (metabolism, signaling and gene regulation). Formally, the problems are described in great detail and distributed in several formats: Matlab, AMIGO [11], Copasi [69] and C. The C implementations are based in libAMIGO. A main C function is provided along with examples on how to use for different purposes like simulating, computing the cost function or adapting a specific cost function. For the purpose of benchmarking optimization algorithms, the user only needs to write a small program as optimization driver, therefore allowing the use of any optimization code which can be interfaced with C.

Here, the option of using openMP to accelerate problems with multiple experiments/simulations is also available. Particularly, problems B2 and B5 fall in this category. To assess the performance gains by using openMP, we compiled libAMIGO with GNU compilers and ran the problem in a Linux cluster node. This node is composed of two octa-core Intel Xeon E5-2660 CPUs. The number of cores used to simulate the different experiments was incremented until the total number of experiments, and the time needed to evaluate the cost function was recorded. Figures 5.3 and 5.4 show the obtained speedup as a function of the used number of cores. Despite being below the theoretical optimal, the usage of openMP results in a significant speedup . Using 3 cores in problem B2 resulted in a 2.5 fold reduction in the time needed to perform the computations while using 5 cores resulted in a 4.5 fold reduction in problem B5.

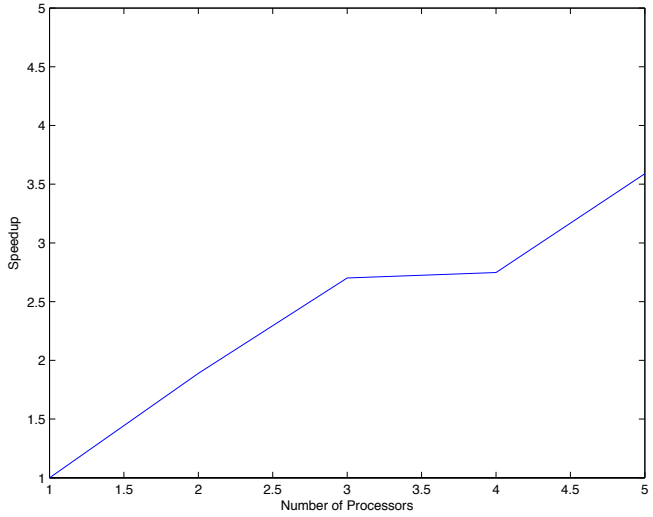


Figure 5.3: The speedup gained by using openMP in problem B2. The speedup in the time spent in the computation is shown as a function of the number of used cores.The perfomance gain is below optimal but quite reasonable when 3 cores are used. Performance decreases rapidly when more than 3 cores are used.

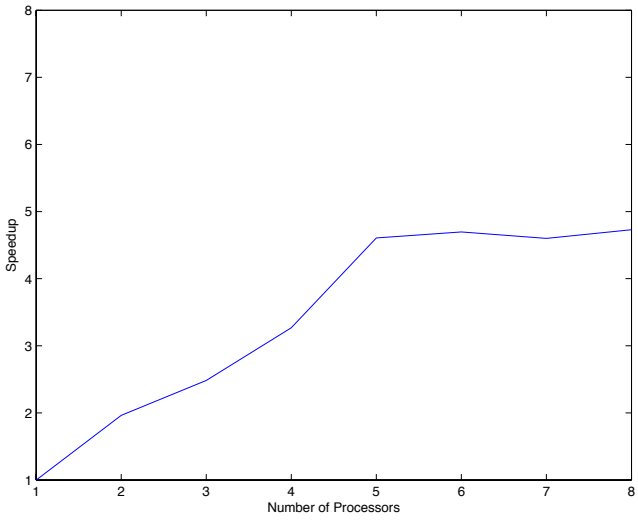


Figure 5.4: The speedup gained by using openMP in problem B5. The speedup in the time spent in the computation is shown as a function of the number of used cores. The perfomance gain is below optimal but quite reasonable when 5 cores are used. Performance decreases rapidly when more than 5 cores are used.

5.3.3 Exploiting cluster computing using SELDOM

The methodological framework of SELDOM is discussed in detail in Chapter 4. However, in this section we provide more details on the implementation and some

computational aspects of SELDOM and its relation with libAMIGO.

Briefly, SELDOM performs inference of dynamic models/networks by using an ensemble approach. Instead of training a single model, in SELDOM, we train many models at the same time and then combine the results at the end. The individual training of each model in the ensemble is an embarrassingly parallel task as no communication between them is necessary.

The shift from Matlab to R was motivated by the lack of Matlab licenses at the computer cluster accessible to perform the computations shown here. However, as libAMIGO was already implemented the cost of writing an interface in R to libAMIGO was relatively small. Additionally, SELDOM uses a number of preprocessing features from other softwares already implemented in R, such as CellNOptR [157].

SELDOM was built as an R package containing the libAMIGO source code. libAMIGO computes the cost the function and the built interface is responsible for populating the memory. The implementation is independent of the model size and all memory is dynamically allocated. Basically the RHS consists of a series of loops that interpret the vector illustrated in Table 5.1. This means problem size can be changed dynamically without any intermediate compilation.

Each independent model training was launched as a batch job using the LSF grid system [176]. To reduce further the time to obtain the solutions, we activated openMP in the evaluation of different experiments. The results shown in Chapter 4 were computed this such approach.

To assess the speedup obtained by openMP we launched R in a node with 32 cores composed of 4 Intel® Xeon® Processor E5-2670. We computed the speedup achieved while computing the dynamic trajectories for the whole ensemble model (100 trajectories for the DREAMiS case study). This is illustrated in Figure 5.5. We increased the number of used cores/threads used by openMP [35] until 10 and monitored the obtained speedups. Until the threshold of 8 cores the improvement is close to optimal. After that, performance drops quickly. Comparing with the results shown in the previous section, there is a huge improvement which is likely due to the usage of a more recent processor, the Intel compilers and other problem dependent characteristics.

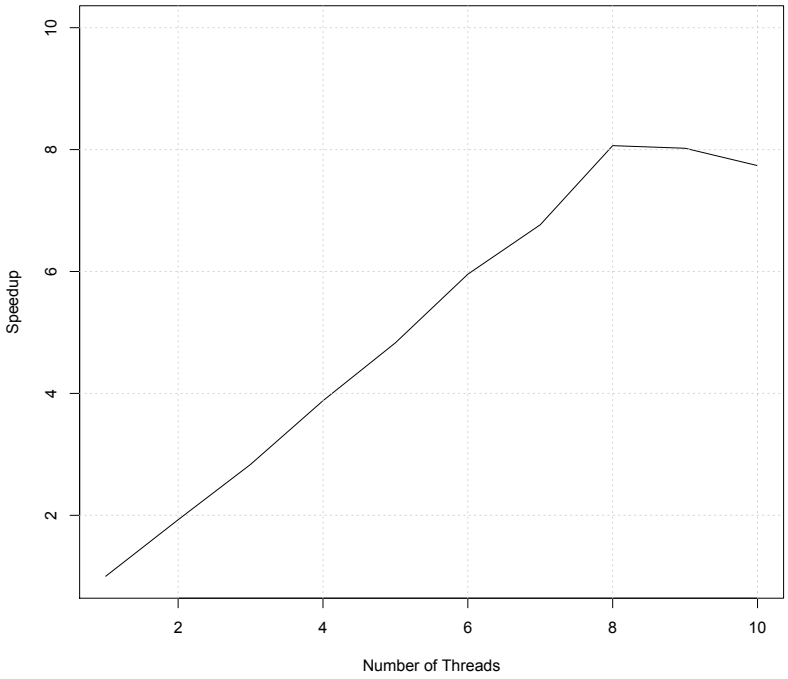


Figure 5.5: The speedup in the time spent in the computation is shown as a function of the number of used cores for the DREAMiS case-study while using SELDOM. The speedup gained by using openMP [35] is almost linear until 8 cores. With more than 8 cores there is a significant loss of of performance.

Number of inputs	Input 1	Input 2	n_1	k_1	n_2	k_2	$w_{0,0}$	$w_{0,1}$	$w_{1,0}$	$w_{1,1}$	τ_1
2	5	8	3	0.5	4	0.6	0	0	0	1	0.5

Table 5.1: Each model in the ensemble is encoded as a vector which is interpreted by the RHS function.

Chapter 6

Conclusions

6.1 Summary of the work and main contributions

In this thesis, we have developed methods for reverse engineering of signaling networks from experimental data. Because the detailed molecular mechanisms (e.g. reactions) behind the functioning of these systems are poorly characterized from the biochemical point of view it is hard to build dynamic models for these systems in a straightforward manner. Despite of this, certain pathways are relatively well characterized from a qualitative point of view and graphs describing interactions between different proteins and the flow of information can be obtained from databases, literature mining or derived manually by experts on a particular biological question.

In the work described in Chapter 3, we assume a PSN is available. This network is a directed graph with a known sign (activation or inhibition). This network is then used to build a nested model where several mechanistic/qualitative hypotheses are encoded and are associated with binary decision variables. On the other hand, the parameters that describe the interactions between model states quantitatively are not known and have to be estimated. Exploring the whole space of models is unfeasible for the size of problems considered here. Thus, we formulated the problem as a MINLP problem and used metaheuristic methods combined with deterministic local solvers to find solutions that are able to describe the experimental data well. In *in silico* case studies we could recover the correct model

structure in a reliable manner. In a different case study with data from a hepatocellular carcinoma cell line (HepG2), we were able to locate a number of high quality solutions.

Finding solutions within the MINLP framework is more manageable than full search space exploration. However, MINLPs are very hard to solve, even using metaheuristics. We compared the behavior of several algorithms and found that the convergence specially for the HepG2 case study was relatively low. To improve convergence, we developed a relaxation (the integrality constraints in the decision variables were dropped) tailored to this problem. The problem was then formulated as a sequence of NLP problems and the final solutions polished with the MINLP solvers.

In Chapter 4, we extend the work of Chapter 3 by assuming no prior knowledge PSN is available. We rapidly found that the method on chapter 3 could not be applied if we assumed a fully connected graph. Thus, we derived a so-called DDN composed of mutual information scores between pairs of variables and imposed a constraint on the maximum number of input connections (indegree) from each node in the graph. We were well aware, both from the theoretical perspective [154] and from the practical of the HepG2 case-study results in Chapter 3, that several networks would be able to explain the network behavior equally well. Thus, we decided to explore the landscape of possible model structures by developing a sampling procedure based in the mutual information scores. Each graph was then used to build a nested model similar to those from Chapter 3 and trained individually.

In this case, we were specially concerned with the predictive skills of the models obtained, more particularly we were interested in making predictions about untested experimental conditions. Therefore, excessive model complexity was addressed by applying a simple, yet effective, iterative model reduction procedure based in the AIC. Indeed, we found many models explained the data similarly well, but it was hard to select a model with high predictive skill. The combination of all trained models into a an ensemble model proved to be the most robust choice for predicting the trajectories for new experiments and for predicting the network structure. Our results were assessed using a number of *in silico* case studies derived by us, and another set of *in silico* and experimental case studies from the

HPN-DREAM breast cancer network inference challenge.

Ensemble methods have been widely used in weather forecasting and for machine learning applications. In a review of ensemble methods applied to systems biology, Swigon [153] describes the usage of ensemble methods using the Bayesian framework. Although the Bayesian point of view is a nice framework to represent the problem, the application of Bayesian inference to the dynamic models derived for SSP, DREAMiS, DREAMBT20 and DREAMBT549 would have been unfeasible. With a relatively small number of models we were able to generate predictions similar (DREAMBT20 and DREAMBT549) or significantly better (DREAMiS) than those of the best performers of the DREAM challenge.

Finally, in Chapter 5 we describe the implementation of the libAMIGO. This library was used to implement the methods used in the studies shown in Chapters 3 and 4. This library is built around the CVODES solver and was developed to facilitate the implementation and sharing of systems biology models. This library arose from the need of being able to share and implement our dynamic optimization problems in a license free and platform independent manner. Some practical applications of libAMIGO are described. Because the library is implemented in C, we could explore parallelization using libraries such as openMP. Very significant speedups could be obtained for some problems with moderate implementation efforts.

The inference of signaling pathways can be handled using a dynamic optimization framework. Reliability of the models needs to assessed *a posteriori* due expected ill-posedness and ill-conditioning. The problems at hand are nonlinear and non-covex. The problem, as stated here, is composed by several computationally demanding tasks, specially in terms of computation time. However, by applying state of the art numerical methods, heuristics, meta-heuristics, relaxations and the usage of multicore and cluster computing, significant gains can be obtained.

6.2 Further work

In chapter 4 we derived an ensemble approach to tackled the indeterminacy of the problem by generating a family of solutions. Recently Huynh-Thu et al. [74] have proposed an ensemble approach to build dynamic models based in data-re-

sampling methods and supervised learning. A possible reformulation of SELDOM could benefit form this type of strategies. A systematic comparison of ensemble generation methods either based in problem structure or data re-sampling techniques (e.g. boosting method) should be considered in further work.

Due to limitations in the available computational power we formulated the training of the SELDOM models as an NLP problem. Considering binary decision variables could improve the interpretability of the models and perhaps even improve their predictive power. This was not tried and should be also considered in further work.

The DREAM-HPN Breast Cancer challenge provides data for 4 cancer cell-lines. It is expected that differences at the genome level cause cell-lines to behave differently at the signaling level. However, these cell-lines certainly share some common features. Maybe a formulation taking into the account the resemblance in terms of model structure and parameters of the different cell-lines can help us build models with improved predictive power or highlight the mechanisms by which different cell-lines behave differently.

During this work we have used and developed a number of *in silico* and experimental case-studies. Often developing or integrating the case-studies was a very laborious task. The BioPreDyn-Bench [168] presents a series of well documented benchmark problems implemented in a systematic manner facilitating its integration in existing pipelines for parameter estimation. A similar collection of problems for reverse engineering of dynamic models and network inference could be built by implementing and documenting the problems considered here in a standard format like MIDAS [136].

Bibliography

[1] T. Akutsu, S. Miyano, and S. Kuhara. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. *Pac Symp Biocomput*, 5:17–28, 1999.

[2] Réka Albert and Juilee Thakar. Boolean modeling: a logic-based dynamic approach for understanding signaling and regulatory networks and for making useful predictions. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 6(5):353–369, July 2014.

[3] Bruce Alberts, Dennis Bray, Julian Lewis, Martin Raff, Keith Roberts, James D Watson, and AV Grimstone. Molecular biology of the cell (3rd edn). *Trends in Biochemical Sciences*, 20(5):210–210, 1995.

[4] Bree B Aldridge, Julio Saez-Rodriguez, Jeremy L Muhlich, Peter K Sorger, and Douglas A Lauffenburger. Fuzzy logic analysis of kinase pathway crosstalk in TNF/EGF/Insulin-induced signaling. *PLoS Comput. Biol.*, 5:e1000340, 2009.

[5] Leonidas G Alexopoulos, Julio Saez-Rodriguez, Benjamin D Cosgrove, Douglas A Lauffenburger, and Peter K Sorger. Networks inferred from biochemical data reveal profound differences in toll-like receptor and inflammatory signaling between normal and transformed hepatocytes. *Mol. Cell Proteomics*, 9:1849–1865, 2010.

[6] Leonidas G. Alexopoulos, Julio Saez-Rodriguez, and Christopher W. Espelin. *High-Throughput Protein-Based Technologies and Computational Models for*

Drug Development, Efficacy, and Toxicity, pages 29–52. "John Wiley and Sons, Inc.", 2008.

[7] Goekmen Altay and Frank Emmert-Streib. Revealing differences in gene network inference algorithms on the network level by ensemble methods. *Bioinformatics*, 26(14):1738–1744, 2010.

[8] Maksat Ashyraliyev, Yves Fomekong-Nanfack, Jaap a Kaandorp, and Joke G Blom. Systems biology: parameter estimation for biochemical models. *The FEBS journal*, 276(4):886–902, March 2009.

[9] Elias August and Antonis Papachristodoulou. Efficient, sparse biological network determination. *BMC Systems Biology*, 3(1):25, 2009.

[10] Eva Balsa-Canto, Antonio a Alonso, and Julio R Banga. An iterative identification procedure for dynamic modeling of biochemical networks. *BMC systems biology*, 4:11, January 2010.

[11] Eva Balsa-Canto and Julio R Banga. AMIGO, a toolbox for advanced model identification in systems biology using global optimization. *Bioinformatics*, 27:2311–2313, 2011.

[12] Julio R Banga. Optimization in computational systems biology. *BMC Syst. Biol.*, 2:47, 2008.

[13] Julio R Banga and Eva Balsa-Canto. Parameter estimation and optimal experimental design. *Essays in biochemistry*, 45:195–210, 2008.

[14] Julio R Banga, Eva Balsa-Canto, Carmen G Moles, and Antonio A Alonso. Dynamic optimization of bioreactors: a review. *PROCEEDINGS-INDIAN NATIONAL SCIENCE ACADEMY PART A*, 69(3/4):257–266, 2003.

[15] Julio R Banga, Eva Balsa-Canto, Carmen G Moles, and Antonio A Alonso. Dynamic optimization of bioprocesses: Efficient and robust numerical strategies. *Journal of Biotechnology*, 117(4):407–419, 2005.

[16] Julio R Banga, Carmen G Moles, and Antonio A Alonso. Global optimization of bioprocesses using stochastic and hybrid methods. In *Frontiers in global optimization*, pages 45–70. Springer, 2004.

[17] Mukesh Bansal, Vincenzo Belcastro, Alberto Ambesi-Impiombato, and Diego di Bernardo. How to infer gene networks from expression profiles. *Mol. Syst. Biol.*, 3(1):78, 2007.

[18] Vikrant Bansal, Vassilis Sakizlis, Roderick Ross, John D. Perkins, and Efstratios N. Pistikopoulos. New algorithms for mixed-integer dynamic optimization. *Computers & Chemical Engineering*, 27(5):647–668, May 2003.

[19] Marti Bernardo-Faura, Stefan Massen, Christine S Falk, Nathan R Brady, and Roland Eils. Data-derived modeling characterizes plasticity of mapk signaling in melanoma. *PLoS computational biology*, 10(9):e1003795, 2014.

[20] Michael Blinov and Ion Moraru. Logic modeling and the ridiculome under the rug. *BMC Biol.*, 10:92, 2012.

[21] Richard Bonneau, David J. Reiss, Paul Shannon, Marc Facciotti, Leroy Hood, Nitin S. Baliga, and Vesteinn Thorsson. The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biol.*, 7(5):1–16, 2006.

[22] L Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, 1996.

[23] L Breiman. Arcing classifiers. *Ann. Stat.*, 26(3):801–824, 1998.

[24] Samuel Burer and Adam N Letchford. Non-Convex Mixed-Integer Nonlinear Programming : A Survey. (February):1–30, 2012.

[25] Kenneth P Burnham and David R Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Science & Business Media, 2003.

[26] Michael R. Bussieck, Stefan Vigerske, James J. Cochran, Louis A. Cox, Pinar Keskinocak, Jeffrey P. Kharoufeh, and J. Cole Smith. MINLP Solver

Software. In *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Inc., 2010.

[27] Ben Calderhead and Mark Girolami. Estimating bayes factors via thermodynamic integration and population mcmc. *Computational Statistics & Data Analysis*, 53(12):4028–4045, 2009.

[28] Young Hwan Chang, Joe W. Gray, and Claire J. Tomlin. Exact reconstruction of gene regulatory networks using compressive sensing. *BMC Bioinform.*, 15(1):1–22, 2014.

[29] William W. Chen, Birgit Schoeberl, Paul J. Jasper, Mario Niepel, Ulrik B. Nielsen, Douglas A. Lauffenburger, and Peter K. Sorger. Input-output behavior of ErbB signaling pathways as revealed by a mass action model trained against dynamic data. *Mol. Syst. Biol.*, 5(1):239, 2009.

[30] Marco Chiarandini, Luís Paquete, Mike Preuss, and Enda Ridge. Experiments on metaheuristics: Methodological overview and open issues. Technical Report DMF-2007-03-003, The Danish Mathematical Society, Denmark, 2007.

[31] Oana-Teodora Chis, Julio R Banga, and Eva Balsa-Canto. Structural identifiability of systems biology models: a critical comparison of methods. *PloS one*, 6(11):e27755, January 2011.

[32] Markus W Covert, Nan Xiao, Tiffany J Chen, and Jonathan R Karr. Integrating metabolic, transcriptional regulatory and signal transduction models in Escherichia coli. *Bioinformatics (Oxford, England)*, 24(18):2044–50, September 2008.

[33] Aspen Custom. Global Methods for Dynamic Optimization and Mixed-Integer Dynamic Optimization. pages 8373–8392, 2006.

[34] L Dagum and R Menon. OpenMP: An industry standard API for shared-memory programming. *IEEE Comput. Sci. Eng.*, 5(1):46–55, 1998.

[35] Leonardo Dagum and Rameshm Enon. Openmp: an industry standard api for shared-memory programming. *Computational Science & Engineering, IEEE*, 5(1):46–55, 1998.

[36] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[37] H De Jong. Modeling and simulation of genetic regulatory systems: A literature review. *J. Comp. Biol.*, 9(1):67–103, 2002.

[38] Hidde de Jong. Modeling and simulation of genetic regulatory systems: a literature review. *Journal of computational biology : a journal of computational molecular cell biology*, 9(1):67–103, 2002.

[39] Riet De Smet and Kathleen Marchal. Advantages and limitations of current network inference methods. *Nature Rev. Microbiol.*, 8(10):717–729, 2010.

[40] John E Dennis Jr, David M Gay, and Roy E Welsch. Algorithm 573: Nl2sol—an adaptive nonlinear least-squares algorithm [e4]. *ACM Transactions on Mathematical Software (TOMS)*, 7(3):369–383, 1981.

[41] Alessandro Di Cara, Abhishek Garg, Giovanni De Micheli, Ioannis Xenarios, and Luis Mendoza. Dynamic simulation of regulatory networks using SQUAD. *BMC bioinformatics*, 8:462, January 2007.

[42] Thomas G. Dietterich. *Multiple Classifier Systems: First International Workshop*, chapter Ensemble Methods in Machine Learning, pages 1–15. Springer, Berlin, Heidelberg, 2000.

[43] P Domingos. The role of occam's razor in knowledge discovery. *Data Min. Knowl. Discov.*, 3(4):409–425, 1999.

[44] Pedro Domingos. The role of occam's razor in knowledge discovery. *Data mining and knowledge discovery*, 3(4):409–425, 1999.

[45] Jose A. Egea, David Henriques, Thomas Cokelaer, Alejandro F. Villaverde, Aidan MacNamara, Diana-Patricia Danciu, Julio R. Banga, and Julio Saez-Rodriguez. MEIGO: an open-source software suite based on metaheuristics for global optimization in systems biology and bioinformatics. *BMC Bioinform.*, 15(1):1–9, 2014.

[46] Jose A. Egea, Rafael Marti, and Julio R. Banga. An evolutionary method for complex-process optimization. *Computers & Operations Research*, 37(2):315–324, 2010.

[47] Jose A Egea, Rafael Martí, and Julio R Banga. An evolutionary method for complex-process optimization. *Comput. Oper. Res.*, 37:315–324, 2010.

[48] Jose A Egea, María Rodríguez-fernández, and Julio R Banga. Scatter Search for Chemical and Bio-Process Optimization.

[49] Oliver Exler, Luis T Antelo, Jose A Egea, Antonio A Alonso, and Julio R Banga. A tabu search-based algorithm for mixed-integer nonlinear problems and its application to integrated process and control system design. *Comput. Chem. Eng.*, 32:1877–1891, 2008.

[50] Oliver Exler, Thomas Lehmann, and Klaus Schittkowski. A comparative study of sqp-type algorithms for nonlinear and nonconvex mixed-integer optimization. *Math. Program. Comput*, pages 383–412, 2012.

[51] 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 Biol.*, 5(1):54–66, 2007.

[52] Jose P. Faria, Ross Overbeek, Fangfang Xia, Miguel Rocha, Isabel Rocha, and Christopher S. Henry. Genome-scale bacterial transcriptional regulatory networks: reconstruction and integrated analysis with metabolic models. *Brief. Bioinform.*, 15(4):592–611, 2014.

[53] C. a. Floudas and C. E. Gounaris. A review of recent advances in global optimization. *Journal of Global Optimization*, 45(1):3–38, August 2008.

[54] Liesbet Geris and David Gomez-Cabrero. *Uncertainty in Biology: A Computational Modeling Approach*, chapter An Introduction to Uncertainty in the Development of Computational Models of Biological Processes, pages 3–11. Springer International Publishing, Cham, 2016.

[55] Fred Glover and Gary A Kochenberger. *Handbook of metaheuristics*. Springer, 2003.

[56] T. Gneiting and A.E. Raftery. Weather forecasting with ensemble methods. *Science*, 310(5746):248–249, 2005.

[57] Alex Greenfield, Aviv Madar, Harry Ostrer, and Richard Bonneau. DREAM4: Combining genetic and dynamic information to identify biological networks and dynamical models. *PloS one*, 5(10):e13397, January 2010.

[58] William Gropp, Ewing Lusk, Nathan Doss, and Anthony Skjellum. A high-performance, portable implementation of the mpi message passing interface standard. *Parallel computing*, 22(6):789–828, 1996.

[59] Gonzalo Guillén-Gosálbez, Antoni Miró, Rui Alves, Albert Sorribas, and Laureano Jiménez. Identification of regulatory structure and kinetic parameters of biochemical networks via mixed-integer dynamic optimization. *BMC Syst. Biol.*, 7:113, 2013.

[60] Carito Guziolowski, Santiago Videla, Federica Eduati, Sven Thiele, Thomas Cokelaer, Anne Siegel, and Julio Saez-Rodriguez. Exhaustively characterizing feasible logic models of a signaling network using Answer Set Programming. *Bioinformatics*, 29(18):2320–2326, 2013.

[61] R Hagedorn, FJ Doblas-Reyes, and TN Palmer. The rationale behind the success of multi-model ensembles in seasonal forecasting - I. Basic concept. *Tellus A*, 57(3):219–233, 2005.

[62] Michael Hecker, Sandro Lambeck, Susanne Toepfer, Eugene van Someren, and Reinhard Guthke. Gene regulatory network inference: data integration in dynamic models-a review. *Bio Systems*, 96(1):86–103, April 2009.

[63] Ralf Heermann and Kirsten Jung. The complexity of the simple two-component system KdpD/KdpE in escherichia coli. *FEMS Microbiol. Lett*, 304:97–106, 2010.

[64] Tomáš Helikar, John Konvalina, Jack Heidel, and Jim A Rogers. Emergent decision-making in biological signal transduction networks. *Proceedings of the National Academy of Sciences*, 105(6):1913–1918, 2008.

[65] David Henriques. Calibration of Ordinary Differential Equation Models. Master's thesis, Engineering School, 2011.

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

[67] Steven M Hill, Laura M Heiser, Thomas Cokelaer, Michael Unger, Nicole K Nesser, Daniel E Carlin, Yang Zhang, Artem Sokolov, Evan O Paull, Chris K Wong, et al. Inferring causal molecular networks: empirical assessment through a community-based effort. *Nat. Methods*, 2016.

[68] AC Hindmarsh, PN Brown, KE Grant, SL Lee, R Serban, DE Shumaker, and CS Woodward. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Software*, 31(3):363–396, 2005.

[69] Stefan Hoops, Sven Sahle, Ralph Gauges, Christine Lee, Jürgen Pahle, Natalia Simus, Mudita Singhal, Liang Xu, Pedro Mendes, and Ursula Kummer. Copasi—a complex pathway simulator. *Bioinformatics*, 22(24):3067–3074, 2006.

[70] CYF Huang and JE Ferrell. Ultrasensitivity in the mitogen-activated protein kinase cascade. *Proc. Natl. Acad. Sci. USA*, 93(19):10078–10083, 1996.

[71] Daniel Hurley, Hiromitsu Araki, Yoshinori Tamada, Ben Dunmore, Deborah Sanders, Sally Humphreys, Muna Affara, Seiya Imoto, Kaori Yasuda, Yuki Tomiyasu, Kosuke Tashiro, Christopher Savoie, Vicky Cho, Stephen Smith, Satoru Kuhara, Satoru Miyano, D. Stephen Charnock-Jones, Edmund J. Crampin, and Cristin G. Print. Gene network inference and visualization tools for biologists: application to new human transcriptome datasets. *Nucleic Acids Res.*, 40(6):2377–2398, 2012.

[72] Daniel G. Hurley, Joseph Cursons, Yi Kan Wang, David M. Budden, Cristin G. Print, and Edmund J. Crampin. NAIL, a software toolset for inferring, analyzing and visualizing regulatory networks. *Bioinformatics*, 31(2):277–278, 2015.

[73] Van Anh Huynh-Thu, Alexandre Irrthum, Louis Wehenkel, and Pierre Geurts. Inferring Regulatory Networks from Expression Data Using Tree-Based Methods. *PLoS ONE*, 5(9):1–10, 2010.

[74] Van Anh Huynh-Thu and Guido Sanguinetti. Combining tree-based and dynamical systems for the inference of gene regulatory networks. *Bioinformatics*, 31(10):1614–1622, 2015.

[75] AEC Ihekwaba, DS Broomhead, RL Grimley, N Benson, and DB Kell. Sensitivity analysis of parameters controlling oscillatory signalling in the nf-kb pathway: the roles of ikk and ikba. *Syst Biol*, 1:93–103, 2004.

[76] In Sock Jang, Adam Margolin, and Andrea Califano. hARACNe: improving the accuracy of regulatory model reverse engineering via higher-order data processing inequality tests. *Interface Focus*, 3(4):20130011, 2013.

[77] Gengjie Jia, Gregory Stephanopoulos, and Rudiyanto Gunawan. Ensemble kinetic modeling of metabolic networks from dynamic metabolic profiles. *Metabolites*, 2(4):891–912, 2012.

[78] GL Johnson and R Lapadat. Mitogen-activated protein kinase pathways mediated by ERK, JNK, and p38 protein kinases. *Science*, 298(5600):1911–1912, 2002.

[79] Claus Jørgensen and Rune Linding. Simplistic pathways or complex networks? *Current opinion in genetics and development*, 20(1):15–22, 2010.

[80] M Joshi, a Seidel-Morgenstern, and a Kremling. Exploiting the bootstrap method for quantifying parameter confidence intervals in dynamical systems. *Metabolic engineering*, 8(5):447–55, September 2006.

[81] Kirsten Jung, Luitpold Fried, Stefan Behr, and Ralf Heermann. Histidine kinases and response regulators in networks. *Curr. Opin. Microbiol.*, 15:118–124, 2012.

[82] Hans-Michael Kaltenbach, Sotiris Dimopoulos, and Joerg Stelling. Systems analysis of cellular networks under uncertainty. *FEBS Lett.*, 583(24):3923–3930, 2009.

[83] Jonathan R Karr, Jayodita C Sanghvi, Derek N Macklin, Miriam V Gutschow, Jared M Jacobs, Benjamin Bolival, Nacyra Assad-Garcia, John I Glass, and Markus W Covert. A whole-cell computational model predicts phenotype from genotype., July 2012.

[84] S Kauffman. A proposal for using the ensemble approach to understand genetic regulatory networks. *J. Theor. Biol.*, 230(4):581–590, 2004.

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

[86] Boris Kholodenko, Michael B Yaffe, and Walter Kolch. Computational Approaches for Analyzing Information Flow in Biological Networks. *Sci. Signal.*, 5(220):re1, 2012.

[87] Paul Kirk, Thomas Thorne, and Michael PH Stumpf. Model selection in systems and synthetic biology. *Current opinion in biotechnology*, 24(4):767–774, 2013.

[88] Steffen Klamt, Julio Saez-rodriguez, and Ernst D Gilles. Structural and functional analysis of cellular networks with CellNetAnalyzer. 13:1–13, 2007.

[89] Oliver Kotte, Judith B Zaugg, and Matthias Heinemann. Bacterial adaptation through distributed sensing of metabolic fluxes. *Molecular systems biology*, 6(355):355, January 2010.

[90] Jan Krumsiek, Sebastian Poelsterl, Dominik M. Wittmann, and Fabian J. Theis. Odefy - From discrete to continuous models. *BMC Bioinform.*, 3(1):1–21, 2009.

[91] Lars Kuepfer, Matthias Peter, Uwe Sauer, and Joerg Stelling. Ensemble modeling for analysis of cell signaling dynamics. *Nat. Biotechnol.*, 25(9):1001–1006, 2007.

[92] Vera Laermann, Emina Ćudić, Kerstin Kipschull, Petra Zimmann, and Karlheinz Altendorf. The sensor kinase KdpD of *Escherichia coli* senses external K+. *Mol. Microbiol.*, 88:1194—-1204, 2013.

[93] Nicolas Le Novère, Andrew Finney, Michael Hucka, Upinder S Bhalla, Fabien Campagne, Julio Collado-Vides, Edmund J Crampin, Matt Halstead, Edda Klipp, Pedro Mendes, et al. Minimum information requested in the annotation of biochemical models (miriam). *Nature biotechnology*, 23(12):1509–1515, 2005.

[94] Yun Lee, Jimmy G. Lafontaine Rivera, and James C. Liao. Ensemble Modeling for Robustness Analysis in engineering non-native metabolic pathways. *Metab. Eng.*, 25:63–71, 2014.

[95] Weijun Luo, Kurt D. Hankenson, and Peter J. Woolf. Learning transcriptional regulatory networks from high throughput gene expression data using continuous three-way mutual information. *BMC Bioinform.*, 9(1):467, 2008.

[96] Aidan MacNamara, David Henriques, and Julio Saez-Rodriguez. Modeling signaling networks with different formalisms: A preview. 1021:89–105, 2013.

[97] Aidan MacNamara, Camille Terfve, David Henriques, Beatriz Peñalver Bernabé, and Julio Saez-Rodriguez. State–time spectrum of signal transduction logic models. *Phys. Biol.*, 9:045003, 2012.

[98] Aviv Madar, Alex Greenfield, Eric Vanden-Eijnden, and Richard Bonneau. DREAM3: network inference using dynamic context likelihood of relatedness and the inferelator. *PloS one*, 5(3):e9803, January 2010.

[99] Daniel Marbach, James C. Costello, Robert Kueffner, Nicole M. Vega, Robert J. Prill, Diogo M. Camacho, Kyle R. Allison, Manolis Kellis, James J. Collins, Gustavo Stolovitzky, and DREAM5 Consortium. Wisdom of crowds for robust gene network inference. *Nat Methods*, 9(8):796–804, 2012.

[100] Daniel Marbach, Robert J Prill, Thomas Schaffter, Claudio Mattiussi, Dario Floreano, and Gustavo Stolovitzky. Revealing strengths and weaknesses of methods for gene network inference. *Proceedings of the National Academy of Sciences of the United States of America*, 107(14):6286–91, April 2010.

[101] AA Margolin, I Nemenman, K Basso, C Wiggins, G Stolovitzky, R Dalla Favera, and A Califano. ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7(1):1–15, 2006.

[102] Florian Markowetz and Rainer Spang. Inferring cellular networks - a review. *BMC Bioinform.*, 8(6):1–17, 2007.

[103] Luis Mendoza and Ioannis Xenarios. A method for the generation of standardized qualitative dynamical systems of regulatory networks. *Theor Biol Med Model*, 3:13, 2006.

[104] Patrick Meyer, Daniel Marbach, Sushmita Roy, and Manolis Kellis. Information-theoretic inference of gene networks using backward elimination. In *BioComp'10, International Conference on Bioinformatics and Computational Biology*, pages 700–705, 2010.

[105] Patrick E. Meyer, Frederic Lafitte, and Gianluca Bontempi. minet: A R/Bioconductor Package for Inferring Large Transcriptional Networks Using Mutual Information. *BMC Bioinform.*, 9(1):461, 2008.

[106] P.E. Meyer, K. Kontos, F. Lafitte, and G. Bontempi. Information-theoretic inference of large transcriptional regulatory networks. *EURASIP J Bioinform Syst Biol*, 2007(1):1–9, 2007.

[107] Ljubiša Mišković and Vassily Hatzimanikatis. Modeling of uncertainties in biochemical reactions. *Biotechnology and bioengineering*, 108(2):413–423, 2011.

[108] Alexander Mitsos, Ioannis N Melas, Melody K Morris, Julio Saez-Rodriguez, Douglas a Lauffenburger, and Leonidas G Alexopoulos. Non Linear Programming (NLP) Formulation for Quantitative Modeling of Protein Signal Transduction Pathways. *PloS one*, 7(11):e50085, January 2012.

[109] Alexander Mitsos, Ioannis N Melas, Paraskeuas Siminelakis, Aikaterini D Chairakaki, Julio Saez-Rodriguez, and Leonidas G Alexopoulos. Identifying drug effects via pathway alterations using an integer linear programming optimization formulation on phosphoproteomic data. *PLoS computational biology*, 5(12):e1000591, December 2009.

[110] Carmen G Moles, Pedro Mendes, and Julio R Banga. Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome Res.*, 13(11):2467–2474, 2003.

[111] Melody K Morris, Julio Saez-Rodriguez, David C Clarke, Peter K Sorger, and Douglas A Lauffenburger. Training signaling pathway maps to biochemical data with constrained fuzzy logic: quantitative analysis of liver cell responses to inflammatory stimuli. *PLoS Comput. Biol.*, 7:e1001099, 2011.

[112] Jesper V Olsen, Blagoy Blagoev, Florian Gnad, Boris Macek, Chanchal Kumar, Peter Mortensen, and Matthias Mann. Global, in vivo, and site-specific phosphorylation dynamics in signaling networks. *Cell*, 127(3):635–648, 2006.

[113] Nonconvex Mixed-integer Optimization, Oliver Exler, Thomas Lehmann, and Klaus Schittkowski. A Comparative Study of SQP-Type Algorithms for Nonlinear and Nonconvex Mixed-Integer Optimization 1 Oliver Exler 2 , Thomas Lehmann 3 , Klaus Schittkowski 2. (4600003917):1–32, 2012.

[114] Richard J Orton, Oliver E Sturm, Vladislav Vyshemirsky, Muffy Calder, David R Gilbert, and Walter Kolch. Computational modelling of the receptor-tyrosine-kinase-activated mapk pathway. *Biochemical Journal*, 392(2):249–261, 2005.

[115] Wei Pan, Ye Yuan, Joaquim Goncalves, and Guy-Bart Stan. Reconstruction of arbitrary biochemical reaction networks: A compressive sensing approach. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 2334–2339. IEEE, 2012.

[116] David R Penas, Patricia Gonźalez, Jośe A Egea, Julio R Banga, and Ramón Doallo. Parallel metaheuristics in computational biology: An asynchronous cooperative enhanced scatter search method. *Procedia Computer Science*, 51:630–639, 2015.

[117] DR Penas, JR Banga, P González, and R Doallo. Enhanced parallel differential evolution algorithm for problems in computational systems biology. *Applied Soft Computing*, 33:86–99, 2015.

[118] DR Penas, Julio R Banga, P González, and R Doallo. A parallel differential evolution algorithm for parameter estimation in dynamic models of biological systems. In *8th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB 2014)*, pages 173–181. Springer, 2014.

[119] Curt Peterson. Drug therapy of cancer. *European Journal of Clinical Pharmacology*, 67(5):437–447, 2011.

[120] Andrea C. Pfeifer, Jens Timmer, and Ursula Klingmüller. Systems biology of jak/stat signalling. *Essays In Biochemistry*, 45:109–120, 2008.

[121] Robert J Prill, Daniel Marbach, Julio Saez-Rodriguez, Peter K Sorger, Leonidas G Alexopoulos, Xiaowei Xue, Neil D Clarke, Gregoire Altan-Bonnet, and Gustavo Stolovitzky. Towards a rigorous assessment of systems biology models: the DREAM3 challenges. *PloS one*, 5(2):e9202, January 2010.

[122] Robert J. Prill, Julio Saez-Rodriguez, Leonidas G. Alexopoulos, Peter K. Sorger, and Gustavo Stolovitzky. Crowdsourcing Network Inference: The DREAM Predictive Signaling Network Challenge. *Sci. Signal.*, 4(189):mr7, 2011.

[123] Mixed Integer Non-linear Programming and Global Optimization. Generalized disjunctive programming: a framework for formulation and alternative algorithms for minlp optimization. pages 1–26.

[124] Jonathan M. Raser and Erin K. O'Shea. Noise in gene expression: Origins, consequences, and control. *Science*, 309(5743):2010–2013, 2005.

[125] a Raue, C Kreutz, T Maiwald, J Bachmann, M Schilling, U Klingmüller, and J Timmer. Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics (Oxford, England)*, 25(15):1923–9, August 2009.

[126] Andreas Raue, Clemens Kreutz, Fabian Joachim Theis, and Jens Timmer. Joining forces of Bayesian and frequentist methodology: a study for inference in the presence of non-identifiability. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 371(1984):20110544, February 2013.

[127] M Re and G Valentini. Ensemble methods: A review. In *Advances in Machine Learning and Data Mining for Astronomy*, pages 563–594. Chapman & Hall, 2010.

[128] Luis Miguel Rios and Nikolaos V. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, July 2012.

[129] Isabel Rocha, Paulo Maia, Pedro Evangelista, Paulo Vilaça, Simão Soares, José P Pinto, Jens Nielsen, Kiran R Patil, Eugénio C Ferreira, and Miguel Rocha. OptFlux: an open-source software platform for in silico metabolic engineering. *BMC systems biology*, 4:45, January 2010.

[130] Guillaume Rochart, Eric Monfroy, and Narendra Jussien. MINLP Problems and Explanation-based Constraint Programming.

[131] Maria Analia Rodriguez and Aldo Vecchietti. Inventory and delivery optimization under seasonal demand in the supply chain. *Computers & Chemical Engineering*, 34(10):1705–1718, 2010.

[132] Maria Rodriguez-Fernandez, Jose A Egea, and Julio R Banga. Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems. *BMC bioinformatics*, 7:483, 2006.

[133] Maria Rodriguez-Fernandez, Pedro Mendes, and Julio R Banga. A hybrid approach for efficient and robust parameter estimation in biochemical pathways. *Biosystems*, 83(2):248–265, 2006.

[134] Maria Rodriguez-Fernandez, Markus Rehberg, Andreas Kremling, and Julio R Banga. Simultaneous model discrimination and parameter estimation in dynamic models of cellular systems. *BMC Syst Biol*, 7:76, 2013.

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

[136] Julio Saez-Rodriguez, Arthur Goldsipe, Jeremy Muhlich, Leonidas G. Alexopoulos, Bjorn Millard, Douglas A. Lauffenburger, and Peter K. Sorger. Flexible informatics for linking experimental data to mathematical models via DataRail. *Bioinformatics*, 24(6):840–847, 2008.

[137] Julio Saez-Rodriguez, Aidan MacNamara, and Simon Cook. Modeling Signaling Networks to Advance New Cancer Therapies. *Annu. Rev. Biomed. Eng.*, 17(1):143–163, 2015.

[138] Regina Samaga and Steffen Klamt. Modeling approaches for qualitative and semi-quantitative analysis of cellular signaling networks. *Cell Commun Signal*, 11:43, 2013.

[139] Francesco Sambo, Marco Antonio Montes de Oca, Barbara Di Camillo, Gianna Toffolo, and Thomas Stutzle. More: Mixed optimization for reverse engineering – an application to modeling biological networks response via sparse systems of nonlinear differential equations. *IEEE/ACM Trans Comput Biol Bioinform*, 9:1459–1471, 2012.

[140] Schaber, J. and Liebermeister, W. and Klipp, E. Nested uncertainties in biochemical models. *IET Syst Biol*, 3(1):1–9, 2009.

[141] RE Schapire, Y Freund, P Bartlett, and WS Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Ann. Stat.*, 26(5):1651–1686, 1998.

[142] Thomas Schlitt and Alvis Brazma. Current approaches to gene regulatory network modelling. *BMC bioinformatics*, 8 Suppl 6:S9, January 2007.

[143] Martin Schlüter, Jose A Egea, and Julio R Banga. Extended ant colony optimization for non-convex mixed integer nonlinear programming. *Comput. Oper. Res.*, 36:2217–2229, 2009.

[144] Fabio Schoen. Stochastic global optimization: Stopping rules. *Encyclopedia of Optimization*, pages 3743–3746, 2009.

[145] Radu Serban and Alan C Hindmarsh. CVODES: An ode solver with sensitivity analysis capabilities. Technical report, Technical Report UCRL-JP-200039, Lawrence Livermore National Laboratory, 2003.

[146] C.E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948.

[147] Jun Shao. Bootstrap model selection. *Journal of the American Statistical Association*, 91(434):655–665, 1996.

[148] Caroline Siegenthaler and Rudiyanto Gunawan. Assessment of network inference methods: How to cope with an underdetermined problem. *PloS one*, 9:e90481, 2014.

[149] N. Soranzo, G. Bianconi, and C. Altafini. Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks: Synthetic versus real data. *Bioinformatics*, 23(13):1640–1647, 2007.

[150] Bernhard Steiert, Andreas Raue, Jens Timmer, and Clemens Kreutz. Experimental design for parameter estimation of gene regulatory networks. *PloS one*, 7(7):e40052, January 2012.

[151] R Steuer, J Kurths, CO Daub, J Weise, and J Selbig. The mutual information: Detecting and evaluating dependencies between variables. *Bioinformatics*, 18(suppl 2):S231–S240, 2002.

[152] Mikael Sunnaker, Elias Zamora-Sillero, Reinhard Dechant, Christina Ludwig, Alberto Giovanni Busetto, Andreas Wagner, and Joerg Stelling. Automatic Generation of Predictive Dynamic Models Reveals Nuclear Phosphorylation as the Key Msn2 Control Mechanism. *Sci. Signal.*, 6(277):ra41, 2013.

[153] David Swigon. Ensemble modeling of biological systems. *Mathematics and Life Sciences. Walter de Gruyter*, pages 19–42, 2012.

[154] Gábor Szederkényi, Julio R Banga, and Antonio A Alonso. Inference of complex biological networks: distinguishability issues and optimization-based solutions. *BMC Syst Biol*, 5:177, 2011.

[155] Yikun Tan and James C. Liao. Metabolic ensemble modeling for strain engineers. *Biotechnol. J.*, 7(3, SI):343–353, 2012.

[156] Claudia Tebaldi and Reto Knutti. The use of the multi-model ensemble in probabilistic climate projections. *Phil. Trans. R. Soc. A*, 365(1857):2053–2075, 2007.

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

[158] Camille Terfve and Julio Saez-Rodriguez. *Advances in Systems Biology*, chapter Modeling Signaling Networks Using High-throughput Phosphoproteomics, pages 19–57. Springer New York, New York, NY, 2012.

[159] Roger L. Thokheim. *Digital Principles*. McGraw-Hill, third edit edition, 1994.

[160] Tina Toni and Michael P. H. Stumpf. Simulation-based model selection for dynamical systems in systems and population biology. *Bioinformatics*, 26(1):104–110, 2010.

[161] Tina Toni and Michael PH Stumpf. Parameter inference and model selection in signaling pathway models. *Computational Biology*, pages 283–295, 2010.

[162] Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael PH Stumpf. Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202, 2009.

[163] D Turei. Benchmark of literature curated signaling pathway resources. *Manuscript submitted for publication*, 2016.

[164] S. M. Minhaz Ud-Dean and Rudiyanto Gunawan. Ensemble Inference and Inferability of Gene Regulatory Networks. *PLoS ONE*, 9(8):e103812, 2014.

[165] Santiago Videla, Carito Guziolowski, Federica Eduati, Sven Thiele, Niels Grabe, Julio Saez-rodriguez, and Anne Siegel. Revisiting the Training of Logic Models of Protein Signaling Networks with a Formal.

[166] Alejandro F. Villaverde and Julio R. Banga. Reverse engineering and identification in systems biology: strategies, perspectives and challenges. *J. R. Soc. Interface*, 11(91):20130505, 2014.

[167] Alejandro F Villaverde, Jose A Egea, and Julio R Banga. A cooperative strategy for parameter estimation in large scale systems biology models. 2012.

[168] Alejandro F Villaverde, David Henriques, Kieran Smallbone, Sophia Bongard, Joachim Schmid, Damjan Cicin-Sain, Anton Crombach, Julio Saez-Rodriguez, Klaus Mauch, Eva Balsa-Canto, et al. Biopredyn-bench: a suite of benchmark problems for dynamic modelling in systems biology. *BMC systems biology*, 9(1):1, 2015.

[169] Alejandro F Villaverde, John Ross, and Julio R Banga. Reverse engineering cellular networks with information theoretic methods. *Cells*, 2(2):306–329, 2013.

[170] Alejandro F. Villaverde, John Ross, Federico Moran, and Julio R. Banga. MIDER: Network Inference with Mutual Information Distance and Entropy Reduction. *PLoS ONE*, 9(5):e96732, 2014.

[171] Vladislav Vyshemirsky and Mark A Girolami. Bayesian ranking of biochemical system models. *Bioinformatics*, 24(6):833–839, 2008.

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

[173] Pengyi Yang, Yee Hwa Yang, Bing B. Zhou, and Albert Y. Zomaya. A Review of Ensemble Methods in Bioinformatics. *Curr Bioinform*, 5(4):296–308, 2010.

[174] Kevin Y Yip, Roger P Alexander, Koon-Kiu Yan, and Mark Gerstein. Improved reconstruction of in silico gene regulatory networks by integrating knockout and perturbation data. *PloS one*, 5(1):e8121, January 2010.

[175] Zhihong Yuan and Bingzhen Chen. State-of-the-Art and Progress in the Optimization-based Simultaneous Design and Control for Chemical Processes. 58(6), 2012.

[176] Songnian Zhou. Lsf: Load sharing in large heterogeneous distributed systems. In *I Workshop on Cluster Computing*, 1992.

[177] 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(1):1–15, 2010.

Appendices

Appendix A

Supplementary Materials

Additional File 1 - S1 File

Supplementary materials for chapter 3.

Additional File 2 - S2 File

Case studies, used scripts and results for chapter 4.
https://drive.google.com/file/d/0B2Kwf3dJqHSOcmYyeGhNdTF4Q3c

Additional File 3 - S1 Fig

Relationship between the training RMSE and the prediction RMSE for the MAPKp problem. The prediction RMSE is plotted here against the training RMSE for each individual model (blue) and the ensemble (red).

Additional File 4 - S2 Fig

AUPR curves for different algorithms applied to MAPKp problem.

Additional File 5 - S3 Fig

Ensemble predictive skill depending on ensemble size (case study MAPKp). This curve was computed by bootstrapping multiple $n_{\mathcal{M}}$ models from the available

100 models, *i.e.* we sampled multiple realizations of the individual predictions for the same ensemble size and computed the average value. These curves converge asymptotically and show that the chosen ensemble size parameter is adequate. Equivalent predictions could have been obtained with smaller ensemble sizes.

Additional File 6 - S4 Fig

Time course trajectories for the training data (MAPKp case study). The median in red is surrounded by the predicted non-symmetric 20% ,60% and 95% confidence intervals.

Additional File 7 - S5 Fig

Time course predictions for the MAPKp case study. The median in red is surrounded by the predicted non-symmetric 20% ,60% and 95% confidence intervals.

Additional File 8 - S6 Fig

Relationship between the training RMSE and the prediction RMSE for the MAPKf problem. The prediction RMSE is plotted here against the training RMSE for each individual model (blue) and the ensemble (red).

Additional File 9 - S7 Fig

AUPR curves for different algorithms applied to MAPKf problem.

Additional File 10 - S8 Fig

Ensemble predictive skill depending on ensemble size (case study MAPKf). The effect of the ensemble size $n_{\mathcal{M}}$ in the prediction RMSE value. This curve was computed by bootstrapping multiple $n_{\mathcal{M}}$ models from the available 100 models, *i.e.* we sampled multiple realizations of the individual predictions for the same

ensemble size and computed the average value. These curves converge asymptotically and show that the chosen ensemble size parameter is adequate. Equivalent predictions could have been obtained with smaller ensemble sizes.

Additional File 11 - S9 Fig

Time course trajectories for the training data (MAPKf case study). The median in red is surrounded by the predicted non-symmetric 20% ,60% and 95% confidence intervals.

Additional File 12 - S10 Fig

Relationship between the training RMSE and the prediction RMSE for the SSP problem. The prediction RMSE is plotted here against the training RMSE for each individual model (blue) and the ensemble (red).

Additional File 13 - S11 Fig

AUPR curves for different algorithms applied to SSP problem.

Additional File 14 - S12 Fig

Ensemble predictive skill depending on ensemble size (case study SSP). The effect of the ensemble size $n_{\mathcal{M}}$ in the prediction RMSE value. This curve was computed by bootstrapping multiple $n_{\mathcal{M}}$ models from the available 100 models, *i.e.* we sampled multiple realizations of the individual predictions for the same ensemble size and computed the average value. These curves converge asymptotically and show that the chosen ensemble size parameter is adequate. Equivalent predictions could have been obtained with smaller ensemble sizes.

Additional File 15 - S13 Fig

Time course predictions for the SSP case study. The median in red is surrounded by the predicted non-symmetric 20% ,60% and 95% confidence intervals.

Additional File 16 - S14 Fig

Time course trajectories for the training data (SSP case study). The median in red is surrounded by the predicted non-symmetric 20

Additional File 17 - S15 Fig

AUPR curves for different algorithms applied to DREAMiS problem.

Additional File 18 - S16 Fig

Time course trajectories for the training data (DREAMBT20 case study). The median in red is surrounded by the predicted non-symmetric 20

Additional File 19 - S17 Fig

Time course predictions for the DREAMiS case study. The median in red is surrounded by the predicted non-symmetric 20% ,60% and 95% confidence intervals.

Additional File 20 - S18 Fig

Relationship between the training RMSE and the prediction RMSE for the DREAMBT20 problem. The prediction RMSE is plotted here against the training RMSE for each individual model (blue) and the ensemble (red).

Additional File 21 - S19 Fig

Ensemble predictive skill depending on ensemble size (case study DREAMBT20). The effect of the ensemble size $n_{\mathcal{M}}$ in the prediction RMSE value. This curve was computed by bootstrapping multiple $n_{\mathcal{M}}$ models from the available 100 models, *i.e.* we sampled multiple realizations of the individual predictions for the same ensemble size and computed the average value. These curves converge asymptotically and show that the chosen ensemble size parameter is adequate. Equivalent predictions could have been obtained with smaller ensemble sizes.

Additional File 22 - S20 Fig

Time course predictions for the DREAMBT20 case study. The median in red is surrounded by the predicted non-symmetric 20%, 60% and 95% confidence intervals.

Additional File 23 - S21 Fig

Time course trajectories for the training data (DREAMBT20 case study). The median in red is surrounded by the predicted non-symmetric 20% ,60% and 95% confidence intervals.

Additional File 24 - S22 Fig

Relationship between the training RMSE and the prediction RMSE for the DREAMBT549 problem. The prediction RMSE is plotted here against the training RMSE for each individual model (blue) and the ensemble (red).

Additional File 25 - S23 Fig

Ensemble predictive skill depending on ensemble size (case study DREAMBT549). The effect of the ensemble size $n_{\mathcal{M}}$ in the prediction RMSE value. This curve was computed by bootstrapping multiple $n_{\mathcal{M}}$ models from the available 100 models, *i.e.* we sampled multiple realizations of the individual predictions for the same ensemble size and computed the average value. These curves converge asymptotically and show that the chosen ensemble size parameter is adequate. Equivalent predictions could have been obtained with smaller ensemble sizes.

Additional File 26 - S24 Fig

Time course predictions for the DREAMBT549 case study. The median in red is surrounded by the predicted non-symmetric 20% , 60% and 95% confidence intervals.

Additional File 27 - S25 Fig

Time course trajectories for the training data (DREAMBT549 case study). The median in red is surrounded by the predicted non-symmetric 20% ,60% and 95% confidence intervals.