Thesis for the Degree of Licentiate of Engineering

A model identification algorithm for cell signalling pathways

Peter Gennemark

Department of Computing Science
Chalmers University of Technology
and Göteborg University
SE-412 96 Göteborg, Sweden

Göteborg, Sweden 2002

A model identification algorithm for cell signalling pathways
Peter Gennemark

Technical report no. 9L
Department of Computing Science
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Göteborg, Sweden 2002

A model identification algorithm for cell signalling pathways

Peter Gennemark
Department of Computing Science
Chalmers University of Technology and Göteborg University

Abstract

The dynamic behaviour of cell signalling pathways is usually studied by differential equation models. In order to build such models we have classified common biochemical reactions into different types that are used as structural building blocks. To compare data from different experiments we have also classified experiments into different categories.

Usually, models are manually inferred from experimental data. As the main result of this thesis we present a model identification algorithm that automatically identifies both the structure and the parameters of a model from experimental data, provided that this data is sufficiently extensive. The algorithm is a carefully designed heuristic algorithm that is efficient for pathways of realistic size.

Presently, artificial, but biologically plausible, models and simulated data from these models have been used to test the algorithm. The algorithm can potentially handle real biological experiments: the number of measurement points can be reduced to acceptable levels and the algorithm can handle noisy data.

As a secondary result of this thesis we present a prototype software tool, where data simulation and model identification are integrated into a single virtual laboratory environment.

Keywords: model identification, signalling pathways, biological modelling, parameter estimation.

Preface

The thesis

This thesis is written within a joint project between Computing Science, Chalmers University of Technology and Cell and Molecular Biology, Göteborg University. The project started in the year 2000 in order to establish mathematical models and software tools for the modelling, simulation, visualisation and analysis of signalling pathways in general.

Acknowledgements

I would like to thank:

My supervisor, Docent Dag Wedelin (Computing science, Chalmers University of Technology), who has structured my work and helped me to logically attack the problems and, at the same time, introduced me to the computing science research field. Dag has also given me well formulated and well motivated feedback on this manuscript numerous times.

The bioinformatics coordinator at Chalmers, Professor Olle Nerman (Mathematical statistics, Chalmers University of Technology), who has given me valuable advices concerning modelling in general and stochastic simulation in particular. Olle has also given me the opportunity to go to Humboldt University, Berlin, on two occasions.

My pair PhD-project colleague, Bodil Nordlander (Cell and Molecular Biology, Göteborg University), who has patiently explained and discussed the biology of signalling pathways and the experimental issues concerning the pathways. Bodil has also tested the software tool and proposed many valuable improvements.

My co-supervisor, Professor Stefan Hohmann (Cell and Molecular Biology, Göteborg University), who has explained the details of the HOG pathway via discussions and extensive written reviews. In particular, Stefan has helped me to understand the role of the HOG pathway from a biophysical perspective.

My co-supervisor, Associate Professor Per Sunnerhagen (Cell and Molecular Biology, Göteborg University), who has put valuable questions concerning the algorithm and who has also explained and discussed the HOG pathway. Furthermore, Per has helped me to categorize experiments used in signalling pathway research.

Dr. Edda Klipp (Theoretical Biophysics, Humboldt University, Berlin, now at Max-Planck Institute for Molecular Genetics, Berlin), who has hosted me in Berlin and there introduced me to biological modelling. In particular, Edda has discussed modelling of the HOG pathway with me.

All colleagues and friends at Matematiskt Centrum and Lundberg laboratory for giving me a nice and pleasant work environment.

Tesse, for always being there supporting me.

Peter Gennemark, September 2002

Contents

Chapter 1

Introduction

Signalling pathways are found in all cells and are involved in a complex network of information transfer inside the cell. The structure of a pathway can be described by a graph, where the substances (vertices) are connected by interactions (edges). A directed edge indicates that a given substance affects another substance, see figure 1.1 for an example. This gives a useful overview of the pathway, but it is not a complete description, since the strength of the interactions, the speed of the reactions, the concentrations of the substances etc. are not described. Despite this fact, this is the level of detail at which biologists traditionally model the pathways. In part this is due to lack of quantitative experimental data and the difficulty to manually infer the model from such data.

To create more powerful descriptions of signalling pathways, mathematical models can be considered. One of several basic motives for creating a more complete model is to simulate the system. In order to be simulated, a model must contain both the structure and the parameters, such as rate constants. A sufficiently exact model may then be able to predict data from the corresponding real system.

Ideally, it would be possible not only to simulate experimental data from a model, but also to automatically identify a model from experimental data. Those two issues, data simulation and model identification, complement each other as illustrated in figure 1.2, and close a loop between model and data.

The main result of this thesis is a model identification algorithm, that reconstructs both the structure and the parameters of a model from experimental data. The output of the algorithm is the model that best fits the data. The algorithm simultaneously takes advantage of all experimental data in a set of experiments. This is especially important when experiments are not directly comparable, which is usually the case in reality. For example, ex-

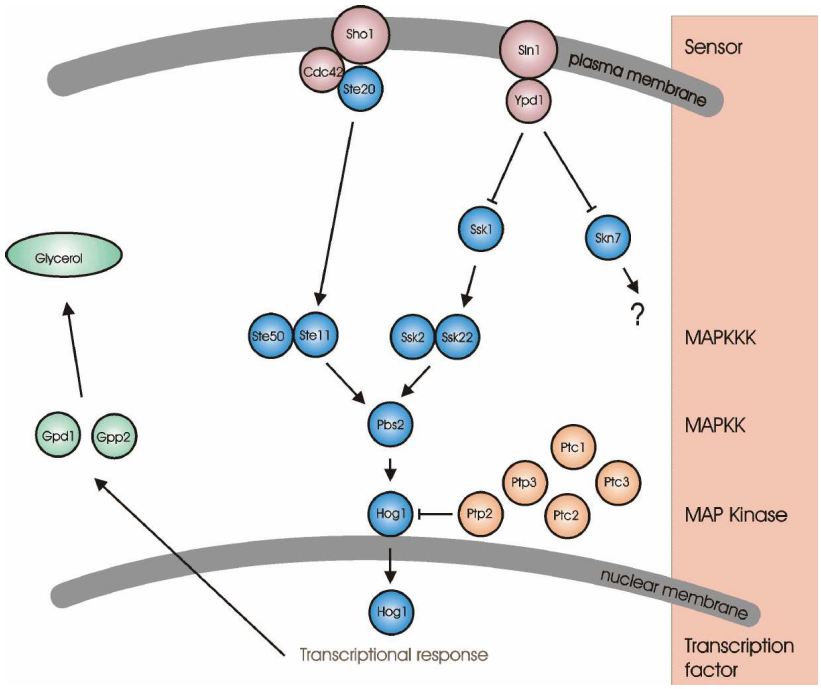


Figure 1.1: *A simplified overview of the main components in the High Osmolarity Glycerol (HOG) signalling pathway in yeast. The details of the pathway are covered in section 2.1.*

periments might have different input stimuli and genetic background (genes can be deleted and the corresponding protein has zero concentration).

Presently, only artificial, but biologically plausible, models and experimental data from these models have been used. There are several advantages of using artificial data. It is much easier to improve and test the details of the algorithm. This is mainly because experimental technical obstacles are removed and because it takes short time to simulate an experiment. It is also easier to develop the methodology of the work process. A future goal is to apply the algorithm to real experimental data. This is further discussed in chapter 7.

In order to evaluate the performance of the algorithm, simulated data have been produced and processed in different ways. As a base case, data have been simulated deterministically. In an attempt to resemble real experimental data, a stochastic simulation method has been employed and the data have also been exposed to measurement noise.

As a secondary result of this thesis, the functionalities presented in figure 1.2 have been implemented in a prototype software tool. The model identification algorithm reconstructs the model structure and parameters based

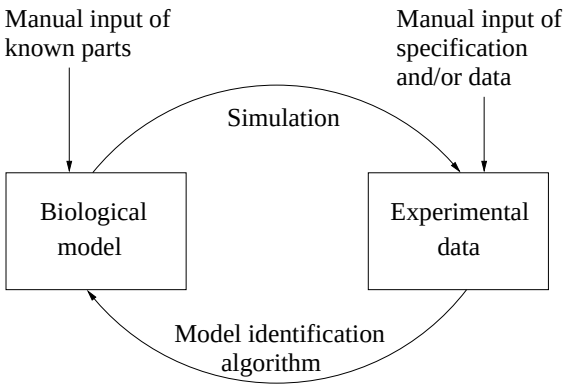


Figure 1.2: *The desired relationship between a biological model and experimental data involves two main functionalities: data simulation and model identification.*

on experimental data. In order to decrease the complexity of the analysis, known parts of the biological model can be added manually. Experimental data can be added manually or simulated from a model. In the latter case, the experiments are specified by manual input.

The intention of the prototype software tool is to show the principle of the model identification algorithm, as well as its crucial role in an integrated environment. Several benefits arise if a well functioning integrated environment can be established. For example, in experimental planning, the tool could be used to propose the best experimental strategy.

The main target group of this thesis is bioinformaticians. In order to let people without previous knowledge of biology read the thesis, an introduction to the biology of signalling pathways and to biological modelling is included.

1.1 Related work

To infer a model from experimental data is a fundamental question considered in many scientific disciplines. In biology, the rapid development of large-scale experimental techniques, such as microarrays, has highlighted the demand for proper model identification algorithms. A general article covering reverse engineering of biological complexity is written by Csete and Doyle [1].

Other efforts in this direction have been made by Koza et al. [2], who have used genetic programming to reconstruct networks of chemical reactions from observed time domain data. Both the structure of the networks and the rates of each reaction within the network for two models, the phos-

pholipid cycle and the synthesis and degradation of ketone bodies, were reconstructed. The phospholipid cycle is the larger of the two models. It is composed of four enzymatic reactions, similar to reactions that will be used in this work. The difference is that Koza et al. are modelling metabolic pathways and therefore need reaction types having several substrates and/or products. The concentration of each enzymes as a function of time was considered known and served as input to the model. For instance, an enzyme could have a linear increase in concentration over time. Data was artificially produced from the model and taken from one out of six metabolites. The strength of this method is that output data is not needed from all of the metabolites. The drawback is the high demand of computational power and it is also unclear how the method can handle noisy data. Therefore, there is a need for more efficient ways of reconstructing biological models.

A related area is reconstruction of gene regulatory networks, where the effects of genes on the transcription of other genes are considered. This approach is mainly focused on large-scale systems. Morohashi and Kitano have applied genetic algorithms in order to identify gene regulatory networks from time series data [3]. Liang et al. [4] have created a reverse engineering algorithm (REVEAL) for reconstructing genetic networks. In this approach, genes are idealised as being either on or off.

The principle of the algorithm presented in this thesis is to determine the structure incrementally. This approach is taken from Wedelin [5], who reconstructs the statistical interaction structure and parameters in multidimensional binary samples.

Chapter 2

Background

2.1 Signalling pathways

Signalling pathways are the means by which cells communicate with their environment and with each other. They sense changes in the environment outside the cell or inside the cell. In general, a protein or a complex of proteins located in the cell membrane (transmembrane proteins) works as a sensor. A cascade of proteins in the cell transmits the signal and finally initiate a transcriptional response, that is, genes are expressed. The translational response involves protein synthesis of the expressed genes. See figure 2.1 for an overview. Typically, a signalling pathway consists of several proteins activating/deactivating each other.

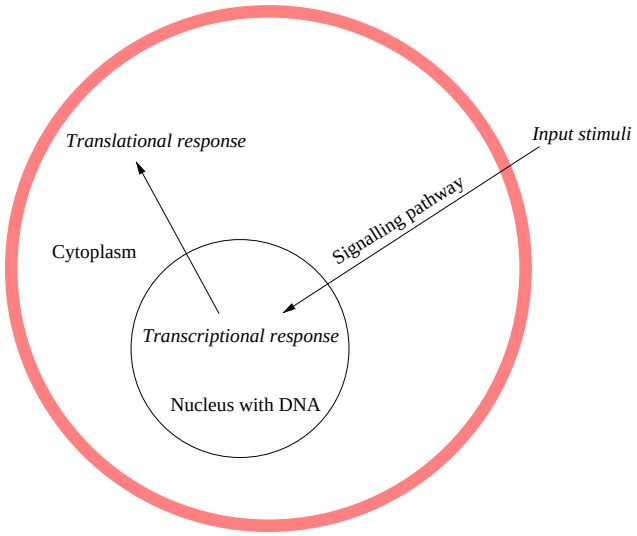


Figure 2.1: *The information flow of a signalling pathway in a yeast cell.*

The principles of a signalling pathway might best be understood by studying an example. Therefore, we focus on the High Osmolarity Glycerol (HOG) pathway [6, 7] of *Saccharomyces cerevisiae*. The HOG pathway is activated by external osmotic stress (an increase in extracellular osmotic pressure due to e.g. increased salt concentration). Like any other cell, the yeast cell has to adjust to altered osmotic pressure to maintain a turgor pressure[1] that is needed for growth and morphogenesis[2], and a relative internal water concentration for optimal efficiency of biochemical reactions. When the solute concentration of the extracellular medium increases, water flows out of the cell and consequently turgor pressure and cell volume drop. One response, followed by the rapid activation of the HOG pathway by osmotic shock, is increased glycerol production. Glycerol works as an osmolyte and drives water into the cell to regain volume and turgor. With that, essential intracellular processes are re-established.

A simplified overview of the main components in the HOG signalling pathway is shown in figure 1.1 in the Introduction. The response is mediated by two independent upstream branches that converge on the protein Pbs2, leading to the activation of Hog1. One branch is dependent on the Sho1 transmembrane protein [8, 9, 10]. Sho1 is not the actual sensor, but plays a crucial role in the pathway. The sensor is not yet discovered. In the other branch, the transmembrane protein Sln1 works as an osmosensor [11]. Two independent pathways carry the signal down to Pbs2, which is phosphorylated (activated) by Ssk2 and Ssk22 [12, 13] and associated to the complex Ste11·Ste50 [14]. Furthermore Pbs2 phosphorylates Hog1, which upon activation is entering the nucleus [13, 15] and, in turn, activates several transcription factors[3]. Feedback reactions are believed to take place on several levels in the pathway and a general de-phosphatase (de-activation) activity is also present. The genes *GPD1* and *GPP2* are involved in the metabolism of glycerol, and they are both strongly up-regulated by an osmotic shock.

The HOG pathway in yeast is an example of a signalling pathway containing a mitogen activated protein kinase[4] (MAPK) module, see figure 2.2. A MAPK module consists of three protein kinases: a MAPK kinase kinase that activates a MAPK kinase, which, in turn, activates a MAPK enzyme [16]. Specific phosphorylation and activation of enzymes in the MAPK module transmits the signal down the cascade, resulting in phosphorylation of

[1] Turgor pressure - hydrostatic pressure that develops within a walled cell, such as a yeast cell, when the osmotic pressure of the cell contents is greater than the osmotic pressure of the surrounding fluid.

[2] Morphogenesis - the evolutionary or embryological development of the physical form of an organism.

[3] Transcription factor - any protein other than RNA Polymerase that is required for transcription.

[4] Kinase - an enzyme that transfers a phosphate group from another molecule to the substrate.

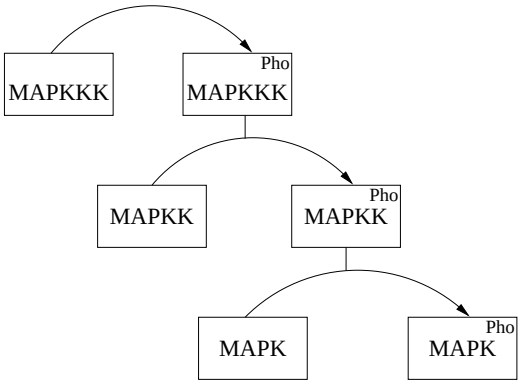


Figure 2.2: *The activation cascade of a MAPK pathway. All boxes represent proteins. Each protein exists in two states, one inactive and one active (denoted pho for phosphorylated). In this case MAPK kinase kinase is activated by a sensor transmembrane protein and the activated MAPK affects transcription factors in the cell nucleus.*

many proteins with regulatory functions throughout the cell, including other protein kinases, gene transcription factors and other enzymes.

Proteins that are able to bind several (different) other proteins, are called scaffold proteins. They might facilitate signal transduction by forming multi-molecular complexes that can be rapidly activated by an incoming signal. In the HOG pathway, Pbs2 is believed to act as a scaffold protein [16, 17]. In many cases, scaffold proteins are necessary for full activation of a signalling pathway [18, 19].

To analyse the different events in the HOG signalling pathway, genetics and molecular biology are used in numerous ways. Cells are exposed to high osmolarity medium and the response to the hyperosmotic stress is analysed. The phosphorylation (activation) state of Hog1 is measured to elucidate the kinetics and the duration of the response. mRNA expression patterns of a few genes, dependent on activated Hog1, are also studied. In order to understand the physiological response to the stress, the rate of glycerol production and intracellular levels of glycerol are measured.

2.2 Mathematical modelling of biological systems

In a biological system, a substance X can have several states, $X_1, X_2, ..X_n$. Different states usually correspond to different levels of activity within the system. From now on, the short notation $X_i(t)$ will be used (instead of the ordinary $[X_i(t)]$) for denoting the concentration of X_i at time t. The

total concentration of all states of X, X_{tot}, can be assumed constant during short time periods (minutes). The assumption being that production and degradation are both zero (or that their sum is zero).

Signal transmission in biological systems occurs mostly through two mechanisms [20]: (1) protein-protein interactions (two substances bind to each other) and (2) enzymatic reactions such as protein phosphorylation and dephosphorylation. The Michaelis-Menten model combines those two mechanisms and accounts for the kinetic properties of many enzymes [21]. As an example, we consider this model more in detail, since it illustrates fundamental principles of biological modelling.

A substance state X_1 is turned into another state X_2 by an enzyme E according to the following reaction

$$E + X_1 \underset{k_2}{\overset{k_1}{\longleftrightarrow}} EX_1 \overset{k_3}{\longrightarrow} E + X_2$$

where EX_1 = transition state complex, k_1 = reaction constant of $E+X_1 \longrightarrow EX_1$, k_2 = reaction constant of $EX_1 \longrightarrow E+X_1$ and k_3 = reaction constant of $EX_1 \longrightarrow E + X_2$. It is assumed that the reaction $E + X_2 \longrightarrow EX_1$ does not occur. An implicit assumption is that $X_1 >> E$. This assumption is usually valid for metabolic systems, but may not be valid for signalling pathways.

We want to obtain an expression for the rate of product formation in the variables X_1, E and rate constants. Initially, we have

$$\frac{d}{dt}X_2(t) = k_3 EX_1(t). \tag{2.1}$$

The concentration of E can be expressed as

$$E(t) = E_{tot} - EX_1(t). \tag{2.2}$$

A relationship between E, X_1 and EX_1 can be identified. First note that the rate of formation of EX_1 equals $k_1 E\, X_1$ and that the rate of breakdown of EX_1 equals $(k_2 + k_3)EX_1$. At catalytic steady-state we obtain

$$EX_1(t) = \frac{E(t)X_1(t)}{K_M} \tag{2.3}$$

where $K_M = \frac{k_2+k_3}{k_1}$. Substitute equation 2.2 into equation 2.3

$$EX_1(t) = \frac{(E_{tot} - EX_1(t))X_1(t)}{K_M}. \tag{2.4}$$

We rearrange and solve for EX_1

$$EX_1(t) = \frac{E_{tot}X_1(t)}{X_1(t) + K_M}. \tag{2.5}$$

Finally, equation 2.5 is substituted into equation 2.1

$$\frac{d}{dt}X_2(t) = \frac{k_3 E_{tot}X_1(t)}{X_1(t) + K_M}. \tag{2.6}$$

Equation (2.6) gives the sought expression: the product formation in terms of X_1, E and rate constants. By assuming $K_M \geq X_1$ in equation 2.6, a linear approximation is obtained. We want to point out that there are other ways of modelling the enzymatic reaction considered above. There also exist other kinds of reactions in a cell, which must be considered in order to model cellular systems. One example of this could be reactions having several substrates and/or products. We would also like to emphasize that higher order derivatives are usually not considered in this kind of modelling.

By combining a set of substances with reactions (like the reaction presented above), a full differential equation model of biological system can be created. For instance, several models of MAPK pathways can be found in the literature. Huang and Ferrell [22] developed a model to describe MAPK activation in *Xenopus oocytes*. Within a large model of second messenger cascades in neurons, Bhalla and Iyengar [20] also consider the MAPK module. Another model, described by Asthagiri and Laufenburger [23], illustrates adaptation of a MAPK cascade. Other references covering biological modelling of signalling pathways are found in [16, 18, 19, 24, 25, 26, 27, 28].

2.3 Simulation of biological models

Systems of differential equations are often difficult to solve analytically, but can be simulated by numerical methods. The simplest method is *Euler's method*, which will be used within the scope of this thesis. The formula for the method is

$$X(t + \Delta t) = X(t) + \Delta t X'(t) \tag{2.7}$$

This procedure is repeated for all substances and for the desired number of iterations (time). We note that the formula is asymmetrical: it advances the solution through an interval Δt, but uses derivative information only at the beginning of that interval. Several better integration methods exist, but the basic principle for them is the same as in *Euler's method*.

In signalling pathways, the number of molecules of each substance is only in the order of 1000 per cell. For that reason, it may be useful to consider each molecule individually. In that case, we shift from continuous models represented by differential equations whose variables are concentrations, to discrete models, represented by stochastic processes whose variables are numbers of molecules. In the real world, the concentrations undergo stochastic fluctuations. When the concentrations are low, as they might be in signalling pathways, the fluctuations should not be neglected. In order to simulate such systems in a more realistic way, stochastic simulation can be applied.

A reaction based on differential equations (like the Michaelis-Menten reaction), can easily be adapted to the discrete case. Instead of considering X as concentration of a substance, we let it reflect the number of molecules of that substance. In the differential equation model, the reaction constants are called macroscopic or deterministic rate constants. In the discrete model, we instead consider mesoscopic rate constants, which are related to, but not identical to, macroscopic rate constants [29]. When converting from macroscopic to mesoscopic rate constants we must take into account that the number of molecules are absolute values and not concentrations. There are standard methods to perform stochastic simulation on biological models.

2.4 Literature data

An ordinary signalling pathway includes a number of reactions and thereby a number of parameters. It is difficult to experimentally measure concentrations and values of parameters, but there are some values given in the literature. The origin of those values are usually *in vitro*[5] experiments and it is not obvious that the corresponding parameter values *in vivo*[6] are the same. In table 2.4 values of total concentration of MAPK:s are presented. The values are collected from the literature [19, 20, 22, 26, 27, 30]. The differences of the values in table 2.4 have two main origins: (1) the values are low and difficult to measure, and (2) different cell types and different MAPK pathways have been studied.

[5]Latin, literally "in glass." Refers to tests or reactions taking place outside a living organism, on a microscope slide, in a test tube, etc.

[6]Latin, literally "in life." Refers to tests or reactions taking place in a living organism.

	Ref. [19]	Ref. [20]	Ref. [22]	Ref. [26]	Ref. [27]	Ref. [30]
Protein	μM	μM	μM	μM	μM	μM
MAPKKK	0.3		< 0.015[1]		0.1	
MAPKK	0.2	0.18	> 0.24[2]	> 0.6[6]	0.3	< 0.035[4]
MAPK	0.4	0.36	0.24[3]	> 0.3[7]	0.3	0.1[5]

Table 2.1: *Total concentration given in the literature of different MAPK:s. Notes: 1. The MAPKKK Mos modelled between 0.6nM-0.015μM 2. The MAPKK Mek-1 modelled between 0.24-6μM 3. The MAPK p42 modelled between 0.24-6μM 4. Ste7p 5. Kss1p and Fus3p 6. MAPKK modelled between 0.6-1.3μM 7.The MAPK p4/p442 modelled between 0.3-2.8μM.*

It is even harder two find estimates of the reaction parameters. For Michaelis-Menten reactions, parameter k is proposed to be 0.01-0.1 s^{-1} [19, 27], parameter d proposed to be 0.05-0.8 s^{-1} [19] and parameter a proposed to be 0.5-20 $\mu M^{-1}s^{-1}$ [19]. Values of $K_M = \frac{d+k}{a}$ are also presented in the literature and range from 0.01 to 1.5 μM [22, 27]. An assumption that $d = 4 * k$ is also mentioned [20].

We conclude that it is difficult to experimentally measure concentrations and parameters of signalling pathways, but that the order of their magnitude can be estimated from the literature.

Chapter 3

Modelling of signalling pathways

By studying the HOG pathway, and analysing the kinetic equations for that specific pathway, we have identified several different general reaction types. By combining such building blocks, also other pathways (in yeast and other cell types) are possible to model. For the purposes of this work, four reaction types were selected. They include a sensor reaction, a non-catalysed reaction and two different catalysed reactions. This collection is large enough to let us build interesting and non-trivial test models and was therefore selected at this stage. However, we wish to point out that the four reaction types are not sufficient to fully model the HOG signalling pathway, why other reactions must also be considered in the future.

It is assumed that other reactants (ATP, water etc.) are present at constant concentration and so can be included in the rate constants. Similar assumptions can be found for instance in reference [22].

3.1 Reaction types

Sensor reaction (reaction 1) is used when a physical effect (pe) affects one substance X to change state from X_1 to X_2.

$$X_1 \xrightarrow{pe} X_2$$

As an example, the physical effect might be osmotic stress, which means increased salt concentration around the cell. The magnitude of the physical

effect (e.g. salt concentration) over time is given by the function $f(t)$. The rate of formation of the substance according to this reaction is given by

$$\frac{d}{dt}X_1(t) = -\frac{d}{dt}X_2(t) = -k_{pe}X_1(t)f(t). \tag{3.1}$$

where k_{pe} is a parameter for the effect of pe on the reaction. The simplest form of $f(t)$ is a step function being high after a given stimulation time point, that is

$$f(t) = \begin{cases} l_1, & t \geq t_s \\ l_2, & \text{otherwise} \end{cases} \tag{3.2}$$

where l_1 and l_2 are constants and t_s is the stimulation time point.

Specifically, we define $\text{step}(t_s)$ to be a step function with $l_1 = 0$ and $l_2 = 1$ according to

$$\text{step}(t_s) = \begin{cases} 1, & t \geq t_s \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

Furthermore, we define $\text{stairs}(t_1, t_2)$ to be a double step function according to

$$\text{stairs}(t_1, t_2) = \begin{cases} 1, & t \geq t_2 \\ 0.5, & t_1 \leq t < t_2 \\ 0, & \text{otherwise} \end{cases} \tag{3.4}$$

The two functions, step and stairs, will be used as examples when testing the model identification algorithm.

Non-catalysed reaction (reaction 2) is used for the spontaneous transition between two states, X_1 and X_2.

$$X_1 \xrightarrow{k} X_2$$

where k is the reaction constant. The rate of formation of the substance according to this reaction is given by

$$\frac{d}{dt}X_1(t) = -\frac{d}{dt}X_2(t) = -kX_1(t). \tag{3.5}$$

Catalysed reaction (reaction 3) is used for a catalysed transition between two states, X_1 and X_2.

$$X_1 + E \overset{k}{\longrightarrow} X_2 + E$$

where k is the reaction constant and E is a substance working as catalyst (E=Enzyme). The rates of formation of the substances according to this reaction are given by

$$\frac{d}{dt}X_1(t) = -\frac{d}{dt}X_2(t) = -kX_1(t)E(t). \tag{3.6}$$

The enzyme is not affected by this reaction.

Catalysed reaction of the Michaelis-Menten type (reaction 4) is used for a catalysed transition between two states, X_1 and X_2. Thus, reaction 4 is the non-linear alternative to reaction 3.

$$X_1 + E \overset{k, K_M}{\longrightarrow} X_2 + E$$

where k is the reaction constant, K_M is the Michaelis-Menten constant and E is a substance working as catalyst (E=Enzyme). The rates of formation of the substances according to this reaction are given by

$$\frac{d}{dt}X_1(t) = -\frac{d}{dt}X_2(t) = -\frac{kE(t)X_1(t).}{X_1(t) + K_M} \tag{3.7}$$

The enzyme is not affected by this reaction.

A simplified model of a signalling pathway can be constructed by defining a set of substances, their different states and a set of reactions of type 1-4. In general, a model is defined by a structure and a set of parameters. The structure is composed of substances with reactions between them. Examples are presented in the next section.

3.2 Test models

We present two artificial, but biologically plausible models of signalling pathways. Those will serve as test models when evaluating the algorithm. They also exemplify the way of combining several reactions to a model of a biological system. The structure of the test models are similar to the structure of a MAPK signalling pathway. However, a specific model of the HOG pathway has presently not been considered. Instead, the main effort has been to develop the model identification algorithm in order to close the loop between model and data. The application of the HOG pathway on the model identification algorithm is discussed in chapter 7.

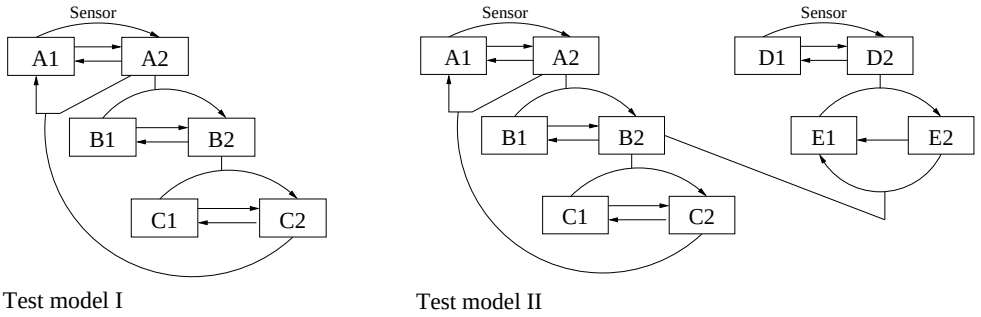


Figure 3.1: *Test model I and II. Curved reactions with "Sensor" label correspond to sensor reactions (type 1), straight-line reactions correspond to non-catalysed reactions (type 2) and curved reactions without label correspond to catalysed reactions (type 3 or 4).*

Test models I and II are presented in figure 3.1 and detailed information about the reactions are listed in table 3.1. Test model I is composed of three substances (A,B and C) and ten reactions. The substance states A_1 and A_2 are two different states of the substance A. In the same way, B_1 and B_2 are two states of substance B and C_1 and C_2 of substance C. A is a sensor activated by an external physical effect. Non-catalysed reactions occur between all states. There are also three catalysed reactions: A_2 catalyses the transition of B_1 to B_2, B_2 catalyses the transition of C_1 to C_2 and C_2 catalyses the transition of A_2 to A_1. Test model II is an extended version of Test model I. It is composed of five substances (A, B, C, D and E) and 16 reactions. D is a sensor just like A. D_2 catalyses the transition of E_1 to E_2. B_2 catalyses the transition of E_2 to E_1. Note that there is no reaction of type 2 from E_1 to E_2.

All substances in Test models I and II have a total concentration of 1 (arbitrary unit).

From the defined set of reactions, the system of differential equations for

Type	Substances	Parameter
1	$A1 \longrightarrow A2$	$k_{pe} = 0.04$
2	$A1 \longrightarrow A2$	$k_1 = 0.02$
2	$A2 \longrightarrow A1$	$k_2 = 0.02$
2	$B1 \longrightarrow B2$	$k_3 = 0.02$
2	$B2 \longrightarrow B1$	$k_4 = 0.06$
2	$C1 \longrightarrow C2$	$k_5 = 0.02$
2	$C2 \longrightarrow C1$	$k_6 = 0.06$
3	$B1 \longrightarrow B2\ (A2)$	$k_7 = 0.10$
3	$C1 \longrightarrow C2\ (B2)$	$k_8 = 0.06$
3	$A2 \longrightarrow A1\ (C2)$	$k_9 = 0.20$

Type	Substances	Parameter
1	$A1 \longrightarrow A2$	$k_{pe\_1} = 0.04$
1	$D1 \longrightarrow D2$	$k_{pe\_2} = 0.08$
2	$A1 \longrightarrow A2$	$k_1 = 0.02$
2	$A2 \longrightarrow A1$	$k_2 = 0.02$
2	$B1 \longrightarrow B2$	$k_3 = 0.02$
2	$B2 \longrightarrow B1$	$k_4 = 0.06$
2	$C1 \longrightarrow C2$	$k_5 = 0.02$
2	$C2 \longrightarrow C1$	$k_6 = 0.06$
2	$D1 \longrightarrow D2$	$k_7 = 0.04$
2	$D2 \longrightarrow D1$	$k_8 = 0.08$
2	$E2 \longrightarrow E1$	$k_9 = 0.06$
3	$B1 \longrightarrow B2\ (A2)$	$k_{10} = 0.10$
4	$C1 \longrightarrow C2\ (B2)$	$k_{11} = 0.06$ $K_M = 0.2$
3	$A2 \longrightarrow A1\ (C2)$	$k_{12} = 0.20$
3	$E1 \longrightarrow E2\ (D2)$	$k_{13} = 0.08$
3	$E2 \longrightarrow E1\ (B2)$	$k_{14} = 0.14$

Table 3.1: *Reactions in Test models I (left) and II (right)*

Test model I can be obtained as

$$\frac{d}{dt}A_2(t) = k_{pe}A_1(t)f(t) + k_1A_1(t) - k_2A_2(t) - k_9A_2(t)C_2(t) \tag{3.8}$$

$$\frac{d}{dt}B_2(t) = k_3B_1(t) - k_4B_2(t) + k_7B_1(t)A_2(t) \tag{3.9}$$

$$\frac{d}{dt}C_2(t) = k_5C_1(t) - k_6C_2(t) + k_8C_1(t)B_2(t). \tag{3.10}$$

The differential equations for A_1, B_1 and C_1 are not needed to integrate the system, since only two states of each substance exist ($A_1(t) = A_{tot} - A_2(t)$ etc.). The system of differential equations for Test model II can be derived in the same way and are left out here.

Chapter 4

Specification and simulation of experiments

In order to analyse models of biological systems, we must consider experimental data. There are a lot of different laboratory techniques in this field of biology and for that reason it is important to find a way of specifying experiments. In the first part of this chapter we focus on these questions. We end this chapter with simulation of experimental data and some other data related issues.

4.1 Specification of experiments

We define an *experiment* to be a measurement of a single variable from a model or biological system over time. For example, the variable may be the concentration of a substance in a given state. As mentioned in chapter 1, experiments measure models or systems that may have different genetic background and input stimuli, but there are also other attributes that have to be specified in order to fully describe an experiment. In table 4.1, we propose a template of the information needed. In this thesis we mainly consider four of the attributes from the table: genetic background, physical effect, measured variable and time series data. All other attributes are considered constant. They play a role in real experiments, but are hard to introduce in a model. Thus, at this stage we do not include them.

Based on the attributes in table 4.1, we define an *experiment category* to be a set of experiments that have the same genomic background, physical effects, experimental technique, species, strain, experimental set-up, cell state and time series start and stop time. Consequently, the only difference between experiments in a category is the measured variable and the time series data

(the unit could differ, but that is not considered in this work). Thus, an experiment category is a set of experiments that measure the same system. The use of different experiment categories is very common, when studying biological systems. For instance, by deleting a particular gene affecting a feedback loop, it is possible to cut off the loop in order to better understand the system. Grouping into experiment categories is important for the model identification algorithm, since all experiments in a category can be generated in a single simulation.

In table 4.2 we specify the experiments for Test models I and II that are used in this work. Two physical effects are used: step(20) and stairs$(20, 50)$ (equations 3.3 and 3.4 in section 3.1). In Test model I there are three experiment categories, namely [wild-type, step(20)], [Gene deletion B, step(20)] and [Gene deletion C, step(20)]. In Test model II there are five experiment categories, namely [wild-type, step(20)], [wild-type, stairs$(20, 50)$], [Gene deletion B, step(20)], [Gene deletion C, step(20)] and [Gene deletion D, step(20)].

Attribute	**Explanation and/or examples**	**Consideration in this thesis**
Genomic background	Wild-type, gene deletion, functional mutant or over expression.	Wild-type and gene deletion.
Physical effects	Time-dependent input to the experiment. Note that there might be several physical effects belonging to the same experiment. Each effect must define a variable (e.g. temperature or external osmotic pressure), a unit and a function of time.	The functions Step and stairs, defined in section 3.1.
Measured variable	A substance in a given state, reaction parameter or physical parameter (volume for instance) that is measured. The variable must exist in the model.	The concentration or number of molecules of a substance state.
Unit	Relative or absolute (e.g. Molar and number of molecules).	Absolute values assumed.
Time series data	The experiment may also consider location scale (the location is in its most general form x,y,z-coordinates, but may be simplified to different compartments in the cell). However, each location may be viewed as one experiment and then only time series data need to be considered.	Time series data for 8-201 data points.
Experimental technique	E.g. northern blot, western blot, protein phosphorylation and microarray.	Constant.
Species	E.g. *S. cerevisiae*.	Constant.
Strain	E.g. S288C.	Constant.
Experimental set-up	E.g. size of cultivation wells, stirring, cell medium, batch/chemostate.	Constant.
Cell state	Lag phase, exponential phase or stationary phase. Time-dependent if the experiment is run over long time.	Constant.

Table 4.1: *Experimental attribute template. Additional minor attributes might be included as well: experimentalist, date of experiment, references and comments. Those attributes need no further explanation.*

Measured variable	Genomic background	Physical effect
A2	Wild-type	step(20)
B2	Wild-type	step(20)
C2	Wild-type	step(20)
A2	Gene del. B	step(20)
C2	Gene del. B	step(20)
A2	Gene del. C	step(20)
B2	Gene del. C	step(20)

Measured variable	Genomic background	Physical effect
A2	Wild-type	step(20)
B2	Wild-type	step(20)
C2	Wild-type	step(20)
D2	Wild-type	step(20)
E2	Wild-type	step(20)
A2 ... E2	Wild-type	stairs(20, 50)
A2, C2 ... E2	Gene del. B	step(20)
A2,B2,D2,E2	Gene del. C	step(20)
A2 ... C2, E2	Gene del. D	step(20)

Table 4.2: *Specification of experimental data for Test models I (left) and II (right). The physical effects step*(20) *and stairs(20,50) are explained in section 3.1. Time series data are specified to go between 0 and 100 (arbitrary unit). The experiments are divided into different categories. Experiments within the same category belong to the same box in the table. Note that the four last categories of Test model II are condensed to one row each in the table.*

4.2 Simulation of experiments

In order to simulate a particular experiment, all attributes of the experiment and the parameters of the model must be specified. For example, assume experiments of the category [Gene deletion B, step(20)] are to be simulated from Test model I. Thus, the system of differential equations 3.8-3.10 is simulated. The initial concentrations are taken from steady state, but since B is deleted, its initial concentration is set to zero. Thus, the concentrations of B_1 and B_2 will remain zero for the whole simulation. The physical effect function (step(20)) specified by the experiment category is used and the final result is simulated time series data for all substance states.

Deterministically simulated experimental time series data were produced from the two test models by integrating the system of differential equations with *Euler's method*. An overview of the different experiments produced for Test model I and II is shown in table 4.2. Time series data goes from 0 to 100 (arbitrary unit) with a step-size of 0.5, giving rise to 201 measurement points. In order to create a smaller set of measurement points, we sample from the 201 measurement points. Plots of the experiments are found in Appendix A.

The same experiments were stochastically simulated using the *Direct method* [31] that is briefly presented below.

In a given state, the number of molecules of each substance state is known. The algorithm calculates probabilistically, which reaction occurs next and when it occurs. For each reaction a probability (propensity) is computed by multiplying the rate constant of the reaction with the concentration of its substrates. Then a random number is used to perform a selection according to the relative probabilities of all reactions, and a second random number determines the execution time used for this reaction. The execution time is taken from an exponential distribution, where the parameter is the sum of all propensities. The chosen reaction is executed. For example, assume the reaction $X_1 \longrightarrow X_2$ catalysed by E is chosen. Then X_1 is decreased by one molecule and X_2 increased by one molecule. The algorithm is summarized as

1. Initialise (set initial numbers of molecules and set $time = 0$).
2. Calculate the propensity function A_i for all reactions i.
3. Choose one reaction according to the relative propensities.
4. Choose Δt from the distribution $Exp(\sum_i A_i)$.
5. Update number of molecules to reflect execution of the reaction.
6. Set $time = time + \Delta t$.
7. Go to step 2.

As mentioned in section 2.3, reactions represented as differential equations can easily be adapted to the discrete case. The volume was set to one and the total number of molecules of each substance was set to 1000. In order to change from macroscopic to the mesoscopic scale, the parameters in the catalysed reactions are scaled to new values. In reaction type 3, the parameter k is divided by 1000 and in reaction type 4, the K_M is multiplied by 1000. Since a real experiment usually is not a single-cell experiment, several cells were simulated and the average value was considered in some test cases.

Noise from different sources in the measurement process disturbs a real biological experiment. In this work all sources is treated as one, called measurement noise. The variance of the measurement noise at a measurement point t_i is assumed to be

$$var(t_i) = c * e(t_i) \tag{4.1}$$

where c is a constant and $e(t_i)$ is the experimental value at time t_i. Normal distribution is assumed. The different simulations are presented in table 4.3 and plots of the experimental data are found in Appendix B.

Simulation	**Number of cells**	**Measurement noise constant**
1	1	0
2	50	0
3	50	0.2
4	50	0.5
5	50	1.0

Table 4.3: *Stochastic simulations of Test Models I and II.*

4.3 Interpolation of experimental data

In the model identification algorithm it is necessary to estimate concentrations and derivatives of concentrations at arbitrary time points, within the time range of an experiment. The most basic approach is to use linear interpolation. For the derivative, it is natural to use the forward difference

$$\frac{d}{dt}\widehat{X}(t) = \frac{X(t_{j+1}) - X(t_j)}{t_{j+1} - t_j} \tag{4.2}$$

for an estimation on the interval between t_j and t_{j+1}.

The above methods are rough estimates. In order to improve the estimation we use cubic spline interpolation [32], which is a standard method in numerical analysis. The method is built on the same principle as the linear interpolation, but a cubic polynomial is used instead of the linear.

4.4 Model ambiguity of experimental data

It can happen that two different biological models create the same experimental data. We illustrate this point by an example.

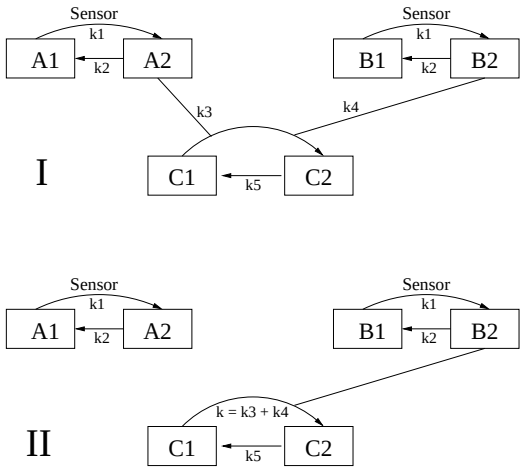


Figure 4.1: *Experimental data of models I and II are identical. Curved reactions with "Sensor" label correspond to sensor reactions (type 1), straight-line reactions correspond to non-catalysed reactions (type 2) and curved reactions without label correspond to catalysed reactions (type 3).*

Consider the biological models presented in figure 4.1. In model I, two sensors (A and B) both activate substance C, while only sensor B activate C in model II. As indicated in the figure, the rate parameter (k) of the catalysed reaction from $C1$ to $C2$ in model II is the sum of the corresponding rate parameters ($k3$ and $k4$) in model I. All other rate parameters are the same in both models. Furthermore, reactions on A and B share the same parameters ($k1$ and $k2$), that is, the activation kinetics of the two sensors are identical. If we consider a wild-type experiment, the two models will produce the same experimental data for all the substances. That is, the data does not unambiguously derive from one biological model.

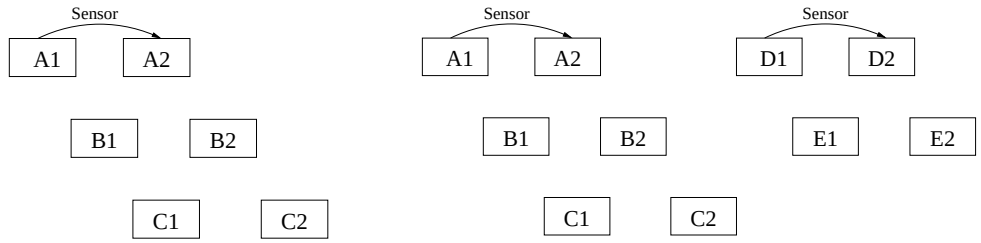
It is important to note that if we add another experiment category where one of the sensors is deleted, the set of all data will unambiguously derive from either model I or model II. This technique was successfully used by

Maeda, Takekawa and Saito [11] when revealing the basic structure of the HOG signalling pathway.

Chapter 5

The model identification algorithm

In this chapter the algorithm for reconstructing signalling pathways from experimental data is presented. The input to the algorithm is an initial structure and a set of experiments. The initial structure contains all substances, the sensor reactions (type 1) and any number of other reactions of the model. It corresponds to the established knowledge of the system. In this thesis we consider only the worst-case examples, where the initial structures lack all reactions of type 2-4, see figure 5.1. All parameters are assumed unknown.



Initial structure of Test model I Initial structure of Test model II

Figure 5.1: *Initial structure of Test model I and II. Curved reactions with "Sensor" label correspond to sensor reactions (type 1).*

The output of the algorithm is the best structure found, its parameters and a measure of the error of that model. The goal of the algorithm is to find a model that minimizes a function, representing the error between the model and the experimental data. The error of the model for a single experiment is calculated by summing the square of the difference between simulated data from the model and the experimental data for each data point. The total

error of the model is calculated by summing the errors for all experiments. Thus, the objective of the algorithm is expressed as

$$\min \sum_{e \in E} \sum_{t_i \in e} \left(e(t_i) - e_{sim}(t_i)\right)^2 \tag{5.1}$$

where E is the set of all experiments, $e(t_i)$ is the measured concentration in experiment e at time point t_i and $e_{sim}(t_i)$ is the simulated value of e at time t_i using the model. The search for a better structure ends when a certain termination criterion is satisfied.

The present version of the algorithm has the following data requirements:

- It is only possible to have two states of each substance.
- Experimental data points are given with correct units (non-normalized).
- The total concentration X_{tot} of each substance is known.
- Experimental data for at least one state of every substance must be given. This is usually not the case in reality. In section 5.8, we demonstrate that this restriction can probably be relaxed in the future.

The requirements are analysed further in the discussion of chapter 7.

5.1 Top level algorithm

To explain the algorithm we first consider only one experiment category as input. For example, for Test model I only the experiments of category [wild-type, step(20)] are present, i.e. we consider the three experiments where A_2, B_2 and C_2 are measured in a wild-type genetic background and with a step function as input.

The main principle of the algorithm is a heuristic that reconstructs the model structure incrementally. A best structure and a best set of parameters are always maintained. In a pre-processing step, all possible non-catalysed reactions (type 2) are added to the initial model and the error is calculated. Then, every possible catalysed reaction (type 3 and type 4) is temporarily added to the model one by one. For each reaction that is tried, the error of the resulting model is calculated. The best reaction is added and the process is repeated until a termination criterion is fulfilled. At the end of each iteration, reactions of type 2-4 that have a small rate parameter are removed.

The evaluation of a particular model structure can be divided into three steps: parameter estimation, simulation and error calculation. First the parameters are estimated, then the model is simulated and finally the error is calculated by equation 5.1. When simulating, a deterministic method is always used. The initial concentration is taken from the first experimental data point of the substance.

The algorithm in pseudo-code is presented below.

INPUT:
S - initial structure
E - set of experiments

OUTPUT:
S - structure of estimated model
P - parameters in estimated model

// PRE-PROCESSING
$R2$:= allPossibleReactionsType2(S)
$R3$:= allPossibleReactionsType3(S)
$R4$:= allPossibleReactionsType4(S)
$R3UR4 := R3 \cup R4$
$S := S \cup R2$
P:=estimateParameters(S, E)
E_{sim}:=simulate(S, P, E)
ϵ_{min}:=calculateError(E_{sim}, E)

// TEST CATALYSED REACTIONS
LOOP
    FOR ALL $testReaction \in R3UR4$ DO
        $S := S + testReaction$
        P_{local}:=estimateParameters(S, E)
        E_{sim}:=simulate(S, P_{local}, E)
        $\epsilon^{testReaction}$:=calculateError(E_{sim}, E)
        $S := S - testReaction$
    END
    r :=bestReaction$(R3UR4, \epsilon)$
    IF (terminationCriterion$(\epsilon_{min}, \epsilon^r, P, r)$) THEN
        BREAK
    ELSE BEGIN
        $S := S + r$
        removeReactionsWithSmallRates(S,P)
        P:=estimateParameters(S, E)
        E_{sim}:=simulate(S, P, E)

ϵ_{min}:=calculateError(E_{sim}, E)
END
END

RETURN S, P, ϵ_{min}

The parameter estimation and the termination criterion are covered more in detail in the following sections.

5.2 Parameter estimation

We continue to consider only one experiment category as input. In order to obtain a low error, we want to find the best parameters for a particular structure. For any experiment, the substance concentration for a couple of measurement points are given. The derivative of the concentration can be estimated. With these data, the set of differential equations corresponding to the structure is reduced to an overdetermined set of equations in the unknown parameters. Each measurement point gives one equation. The overdetermined system of equations is solved with the least-square method if it is linear. If a catalysed reaction of type 4 is involved in the equation it becomes non-linear, and Marquardt's method [32, 33] is used instead. In practice, every differential equation is considered separately, and the differential equations are handled in turn. The parameter estimation is now described by an example.

Consider Test model I and the differential equation of A_2, see equation 3.8. Each term on the right hand side corresponds to one reaction. The parameters to estimate are k_{pe}, k_1, k_2 and k_{12}. By estimating $\frac{d}{dt}A_2(t)$ and all concentrations on the right hand side from experimental data, equation 3.8 gives us a linear equation. Each data point in the experiment where A_2 is measured gives one such equation. The notation $\widehat{X_i}(t)$ will denote a concentration estimation of substance X_i given data from the considered experiment category. The full system can be written

$$Mk = b \tag{5.2}$$

where

$$M = \begin{pmatrix} \widehat{A_1}(t_1)f(t_1) & \widehat{A_1}(t_1) & -\widehat{A_2}(t_1) & -\widehat{A_2}(t_1)\widehat{C_2}(t_1) \\ \widehat{A_1}(t_2)f(t_2) & \widehat{A_1}(t_2) & -\widehat{A_2}(t_2) & -\widehat{A_2}(t_2)\widehat{C_2}(t_2) \\ \vdots & \vdots & \vdots & \vdots \\ \widehat{A_1}(t_n)f(t_n) & \widehat{A_1}(t_n) & -\widehat{A_2}(t_n) & -\widehat{A_2}(t_n)\widehat{C_2}(t_n) \end{pmatrix}, \tag{5.3}$$

$$k = \begin{pmatrix} k_{pe} \\ k_1 \\ k_2 \\ k_9 \end{pmatrix} \tag{5.4}$$

and

$$b = \begin{pmatrix} \frac{d}{dt}\widehat{A_2}(t_1) \\ \vdots \\ \frac{d}{dt}\widehat{A_2}(t_n) \end{pmatrix}. \tag{5.5}$$

In equation 5.3, t_1 and t_n refer to the first and last experimental time point respectively. The system of equations is overdetermined and is solved by the least-square method, which minimizes the Euclidean norm between Mk and b [34], that is

$$\min_k \parallel b - Mk \parallel_2 . \tag{5.6}$$

If the column vectors are linearly independent ($M^T M$ positive definite), the solution to the least-square problem is obtained from the linear system

$$M^T Mk = M^T b. \tag{5.7}$$

It is important to note that the minimization function 5.6 coincides with the original minimization function 5.1 *only if the model structure is correct*. This is because experimental data are used in order to estimate the concentrations and the derivatives. Thus, the best parameters in terms of the original minimization function are obtained only in this case. This is an algorithmic short-cut in order to speed up the algorithm and it works because we have a complete data set. This is also why we need a subsequent simulation step in the algorithm to determine the true error of the current model.

If there is a catalysed reaction (type 4) in the differential equation, the ordinary least-square method will not do. Instead we employ Marquardt's method for least-squares estimation of non-linear parameters [32, 33]. Marquardt's method works well in practice and has become a standard for non-linear least-squares. Briefly, the method varies smoothly between two methods, the inverse-Hessian method and the steepest descent method. The latter method is used far from the minimum, switching continuously to the former as the minimum is approached. The method is not covered more thoroughly here.

5.3 Termination criterion and thresholds

The search for new reactions of type 3 and 4 is terminated when:

$$\epsilon^r > \epsilon_{min} * \beta \qquad \text{OR} \qquad k_r < \delta_i \tag{5.8}$$

where ϵ^r is the lowest error found when testing new catalysed reactions, ϵ_{min} is the presently best (lowest) error, $\beta \leq 1$, k_r is the rate constant of the reaction proposed to be added to the model, $\delta_i > 0$ and $i \in 3, 4$. The constants, β and δ_i are specified by the user of the algorithm. δ_3 and δ_4 are used when the added reaction is of type 3 or 4, respectively. Thus, the search ends when either the decrease of the error is too small or when the reaction to add has to small rate constant.

In a final step of the loop in the algorithm, reactions of type 2-4 might be removed from the model (removeReactionsWithSmallRates(S,P) in the pseudo-code), the criterion being

$$k < \delta_i \tag{5.9}$$

where k belongs to a reaction of type $i \in 2, 3, 4$ and, as previously, $\delta_i > 0$.

In general, a model with a complex structure is more likely to have low error, because the parameter space is large and the model can be fine-tuned to fit experimental data. Presently, the complexity of the model is not explicitly considered in the minimization function and is only implicitly considered in the termination criterion, which is necessary to avoid overfitting. There are several ways to punish high complexity, but this is a complex issue and will be considered in the future.

5.4 Extension to several experiment categories

We now generalize the algorithm to handle several experiment categories. Again, consider Test model I, but let all experiment categories presented in table 4.2 be included. The parameter estimation and the top level algorithm are both affected by the change. In the parameter estimation, each differential equation is still considered separately, but the experiments from all experiment categories are merged and considered simultaneously. In the main algorithm, simulation is then performed for each experimental category. The simulation itself and the error calculation are not affected.

As an example, consider Test model I and the differential equation of A_2, see equation 3.8. In order to take all experiments where A_2 is measured into

account, all such experiments are merged in matrix M (equation 5.3). The experiments are [A_2, wild-type, step(20)], [A_2, Gene deletion B, step(20)] and [A_2, Gene deletion C, step(20)]. Each data point in each experiment where A_2 is measured gives one row in M. The number of columns in the matrix is not affected, since the number of unknown parameters is the same. The number of rows corresponds to the total number of experimental measurement points of A_2 in all experiment categories. As before, the system of equations is overdetermined and is solved by the least-square method. The same approach holds for the non-linear case with Marquardt's method.

5.5 Methods for increasing the speed

The short-cut of not minimizing the original error function 5.1 significantly reduces the computational time of the algorithm. This simplification also gives us the opportunity to further increase the speed of the algorithm. We make one observation:

When adding (testing) a catalysed reaction (type 3 or 4) affecting substance X, only parameters in the differential equation for X need to be re-estimated.

All other parameters are unaffected by the change of the model. From this follows that only the differential equation of X needs to be simulated and that only experiments measuring X need to have their errors re-calculated. As an example, consider Test model I (section 3.2). Assume we want to add the reaction $A_1 \longrightarrow A_2$ catalysed by B_1 to the model. Only the parameters in the differential equation of A_2 (equation 3.8) need to be considered. All other parameters remain the same. Furthermore, only the differential equation of A_2 must be simulated, and consequently, only those experiments measuring A_2 must have their errors re-calculated.

The basic procedure of estimating parameters remains the same. The difference is that each differential equation is not considered when re-calculating the new set of parameters. Only the differential equation for the substance that is changing state is considered.

The simulation is affected too: instead of simulating the full set of differential equations, we only simulate one differential equation. As before, the initial concentration is taken from the first experimental data point of the substance. Concentrations of other substances occurring in the differential equation are estimated from experimental data. For example, assume that time series data for the substance A_2 are simulated given Test model I and the experimental attributes of experiment e. Thus, the differential equation

3.8 is simulated. The initial concentration value of A_2 is taken from experimental data, while all other data points are simulated. Concentrations of other substances (C_2 in this case) occurring in the differential equation are estimated from experimental data, they are *not* simulated. As before, the parameters k_1, k_2 and k_{12} must have been estimated in advance. The result is simulated time series data for substance A_2. Since only parts of the model is simulated and the other parts estimated from data, the result may not be the same as if the whole model was simulated. Again, we note that we depend on a complete data set in order to use this short-cut.

In the top level algorithm, the error of each individual experiment (denoted ϵ_e) must be monitored. The abbreviation *cat* is used for *category*. The algorithm in pseudo-code is given below.

INPUT:
S - initial structure
E - set of experiments

OUTPUT:
S - structure of estimated model
P - parameters in estimated model

// PRE-PROCESSING
$R2$:= allPossibleReactionsType2(S)
$R3$:= allPossibleReactionsType3(S)
$R4$:= allPossibleReactionsType4(S)
$R3UR4 := R3 \cup R4$
$S := S \cup R2$

P:=estimateParameters(S, E)
FOR ALL $cat \in E$ DO
    E^{cat}_{sim}:=simulate(S, P, E, cat)
END
FOR ALL $e \in E$ DO
    ϵ^e:=calculateError(E_{sim}, e)
END
$\epsilon_{min} := \sum_{all\ e \in E} \epsilon^e$
LOOP
    FOR ALL $e \in E$ DO
        ϵ^e_{old}:=ϵ^e
    END
    FOR ALL $testReaction \in R3UR4$ DO
        FOR ALL $e \in E$ DO
            $\epsilon^{e,\ testReaction}$:=$\epsilon^e_{old}$

```
            END
            S := S + testReaction
            s:=substanceChangingState(testReaction)
            Ed:={e ∈ E | e.measured_variable ∈ s}
            P^testReaction:=estimateParametersSingle(S, P, E, s)
            FOR ALL e ∈ Ed DO
                  e_sim:=simulateSingle(S, P^testReaction, s, E, e)
                  ε^(e, testReaction):=calculateErrorSingle(e_sim, e)
            END
            ε_E^testReaction := Σ_(all e∈E) ε^(e, testReaction)
            S := S − testReaction
      END
      r :=bestReaction(R3UR4, ε_E)
      IF (terminationCriterion(ε_min, ε_E^r, P, r)) THEN
            BREAK
      ELSE BEGIN
            S := S + r
            removeReactionsWithSmallRates(S,P)
            P:=estimateParameters(S, E)
            FOR ALL cat ∈ E DO
                  E_sim^cat:=simulate(S, P, E, cat)
            END
            FOR ALL e ∈ E DO
                  ε^e:=calculateError(E_sim, e)
            END
            ε_min := Σ_(all e∈E) ε^e
      END
END
RETURN S, P, ε_min
```

There are other possible short-cuts. We note that reactions of type 3 and 4 are similar in the sense that they are both catalysed reactions. If a low error is obtained by adding a particular reaction of type 3, the corresponding reaction of type 4 will probably also give a low error when added, and vice versa. Since the non-linear parameter estimation demands more computational time, we first test the reaction of type 3. If the error of that model is sufficiently bad, no test of the corresponding reaction of type 4 occurs. We formulate the following rule

```
IF (ε^r3 > γ * ε_min) THEN
      skip test of corresponding r4
```

where $\gamma > 1$ is a constant. The above code-fragment can easily be included

in the main loop of the top-level algorithm. This short-cut has been used in this thesis with $\gamma = 1.2$.

5.6 Computational time of the algorithm

The computational time of the algorithm is difficult to exactly formulate, since the choice of parameter estimation method depends on the specific model. Without reactions of type 4 the least-square method is applied, otherwise the computational much more expensive Marquardt's method is used. In this section we consider a base case where reactions of type 4 are not included at all. This simplification can partly be justified by the quite sparse use of Marquardt's method when the short-cut of skipping some tests of reactions of type 4 is employed (section 5.5).

We consider the computational time as a function of the variables presented in table 5.1.

Variable	**Description**
n_s	Number of *substances* in the model.
n_{ec}	Number of *experiment categories*.
n_{dp}	Total number of experimental *data points*, measuring a particular substance.
Δt	*Step size* in simulation.
t_{sim}	*Simulation time*.

Table 5.1: *Variables used in calculation of computational time. For simplification, we assume that n_{dp} is equal for all substances and that t_{sim} is equal for all experiments.*

The time complexity of the algorithm, T_{alg}, can be expressed as

$$T_{alg} = N_{loops} N_{tests} \left(T_{pe} + T_{sim} + T_{err} \right) \tag{5.10}$$

where N_{loops} is the number of loops in the algorithm, N_{tests} is the number of reaction tests within one loop and T_{pe}, T_{sim} and T_{err} are the time complexity for one parameter estimation, one simulation and one error calculation, respectively.

It is difficult to estimate N_{loops}, since it is dependent on the iterative behaviour of the algorithm. In particular, N_{loops} is strongly affected by the termination criterion. A typical value for N_{loops} involves the variable n_{cr}

which is the number of *catalysed reactions* that are not included in the initial structure but belong to the correct structure of the model. Assuming that we find the correct structure we obtain

$$N_{loops} = n_{cr} + 1 \tag{5.11}$$

which can be motivated by an example: In Test model I the number of loops is ideally four, three loops for identifying each of the three catalysed reactions and one loop for reaching the termination criterion.

If all possible reactions are added to the model, N_{loops} is dramatically increased: one loop for each possible catalysed reaction is required. Since every substance reacts in two directions ($X_1 \longrightarrow X_2$ and $X_2 \longrightarrow X_1$) and the enzyme can be any other substance, each existing in two different states, $(2(n_s - 1))$, we obtain the function

$$N_{loops} = 4n_s(n_s - 1) \in O(n_s^2). \tag{5.12}$$

There is also a possibility that the algorithm shows a cyclic behaviour. In that case, N_{loops} may potentially go to infinity, given the current termination criterion.

One cycle of the loop contains tests of all possible catalysed reactions. The same reasoning as for equation 5.12 gives us

$$N_{tests} = 4n_s(n_s - 1) \in O(n_s^2). \tag{5.13}$$

For each reaction that is tested, the resulting model is subjected to parameter estimation, simulation and error calculation. Before analysing them in turn, we define n_r to be the number of *reactions* affecting a particular substance. We obtain an upper bound for n_r by observing that n_r equals $4(n_s - 1)$ reactions of type 1 and 3 respectively (compare to equation 5.13) and two reactions of type 2 in worst case. Thus,

$$n_r = 8(n_s - 1) + 2 \in O(n_s). \tag{5.14}$$

The parameter estimation is performed by the least square method, where the matrix is of size $n_{dp} \times n_r$ (equation 5.3). The method runs in polynomial time, since it requires $n_r^2 n_{dp} - n_r^3/3$ multiplications and a similar number of additions (QR factorisation) [35]. Substituting n_r for n_s according to equation 5.14, an upper bound for T_{pe} is obtained as

$$T_{pe} \in O(n_s^2 n_{dp}). \tag{5.15}$$

One simulation is performed for each experiment category and the running time of each simulation depends on Δt, t_{sim} and n_r. Substituting n_r for n_s, T_{sim} is obtained as

$$T_{sim} \in O(n_{ec}\frac{t_{sim}}{\Delta t}n_s). \tag{5.16}$$

The time complexity of the error calculation is linear in time w.r.t. n_{dp}, giving

$$T_{err} \in O(n_{dp}). \tag{5.17}$$

Inserting equation 5.15, 5.16 and 5.17 into equation 5.10, we obtain the time complexity of the algorithm as

$$T_{alg} = N_{loops}N_{tests}\left(O(n_s^2 n_{dp}) + O(n_{ec}\frac{t_{sim}}{\Delta t}n_s) + O(n_{dp})\right). \tag{5.18}$$

Most computational time of the algorithm is spent evaluating different reactions added to the model. Test runs indicate that T_{err} always can be neglected in comparison to T_{pe} and T_{sim}. However, the relationship between T_{pe} and T_{sim} is not straightforward. For large n_{dp}, $T_{pe} > T_{sim}$, while for small n_{dp}, $T_{pe} < T_{sim}$. For example, given Test model I and experiments with 201 data points, the parameter estimation takes about 4 times longer time as the simulation. For 8 or 16 data points per experiment the simulation takes about 9 times longer time as the parameter estimation. Thus, for small n_{dp}, an approximation to equation 5.18 can be obtained as

$$T_{alg} \approx N_{loops}N_{tests}O(n_{ec}\frac{t_{sim}}{\Delta t}n_s) \tag{5.19}$$

and for large n_{dp}, a similar approximation is obtained as

$$T_{alg} \approx N_{loops}N_{tests}O(n_s^2 n_{dp}). \tag{5.20}$$

We would also like to emphasize that for non-linear models usually $T_{pe} \gg T_{sim}$.

Based on the analysis of the computational time above, we can give a rough estimate of the difference in running time between Test model I and II. In those cases we assume that N_{loops} equals its typical value according to equation 5.11. This actually turns out to be true for our test cases. For Test model I, $n_{cr} = 3$, $n_s = 3$ and $n_{ec} = 3$, and for Test model II, $n_{cr} = 5$, $n_s = 5$ and $n_{ec} = 5$. n_{dp} is proportional to n_{ec}, since the number of data

points in each experiment is constant. Using equation 5.19, the difference in computational time would approximately be a factor of 13 when the number of data points per experiment is 8 or 16. Using equation 5.20 and considering 201 data points per experiment, the same factor would approximately be 21. The running time of the algorithm on Test models I and II are given in the results section (5.7).

Since we use an heuristic approach and the increase of computational effort is typically polynomial w.r.t. number of substances and amount of experimental data, we argue that significantly larger models than Test model II are possible to identify with reasonable computational effort using this or a similar algorithm.

5.7 Test results

The algorithm has been implemented in Java as a part of the integrated environment (chapter 6). A linear algebra package for Java, JAMA [36], was used for basic linear algebra manipulations. As mentioned before, *Euler's method* has been used for simulations. A more accurate method, the *fifth order Runge-Kutta Method with adaptive step-size* [32] has also been employed, both to produce experimental data and to run the simulations in the algorithm. However, for our present purposes the choice of integration method did not give an evident effect of the performance of the algorithm. For that reason, only *Euler's method* is used to produce the test results. The performance of the algorithm is presented in terms of test runs of Test models I and II. All tests were run on a Sun Enterprise 450, Dual UltraSparc 300 MHz, 512 MB RAM.

Results with deterministically simulated data

We first consider experimental data simulated deterministically and without any noise (see section 4.2). In order to test the algorithm under best possible conditions, all (201) simulated data points of each experiment served as input. In this case the trivial linear interpolation was used, instead of the cubic spline interpolation. This is because the data is simulated using *Euler's method* and therefore the forward difference (equation 4.2) is the exact one. The algorithm was able to correctly reconstruct both the structure and the parameters of Test model I and II from the initial structures (figure 5.1) and the experimental data given.

In order to test the algorithm under more realistic conditions, the number of experimental data points per experiment was reduced. In this case, the cubic spline interpolation was used. For both Test model I and II the number of

data points per experiment could be reduced down to eight before the correct structure was not found any more. See table 5.2 and 5.3 for detailed results.

The running time of the algorithm is also presented in the tables. The running times of Test model I and II differ by a factor of 16, 10, and 15 for the three different test runs with different number of data points. Those factors are reasonable considering the theoretical calculation in section 5.6, where the factors were roughly calculated to 21, 13 and 13 respectively.

Type	**Substances**	**Correct parameter**	**Estimated parameter n=201**[1]	**Estimated parameter n=16**	**Estimated parameter n=8**
1	$A_1 \longrightarrow A_2$	0.04	0.040	0.026	0.019
2	$A_1 \longrightarrow A_2$	0.02	0.020	0.013	0.011
2	$A_2 \longrightarrow A_1$	0.02	0.020	0.012	0.0088
2	$B_1 \longrightarrow B_2$	0.02	0.020	0.020	0.017
2	$B_2 \longrightarrow B_1$	0.06	0.060	0.061	0.053
2	$C_1 \longrightarrow C_2$	0.02	0.020	0.020	0.019
2	$C_2 \longrightarrow C_1$	0.06	0.060	0.059	0.058
3	$B_1 \longrightarrow B_2$ (A_2)	0.10	0.10	0.10	0.089
3	$C_1 \longrightarrow C_2$ (B_2)	0.06	0.060	0.059	0.058
3	$A_2 \longrightarrow A_1$ (C_2)	0.20	0.20	0.13	0.10
Running time (s)			16	4.4	1.9

Table 5.2: *Results from reconstruction of Test model I, n is the number of data points per experiment. Parameters* $\delta_2 = 0.002$, $\delta_3 = 0.001$, $\delta_4 = 0.001$ *and* $\beta = 0.85$. *1) Linear interpolation.*

Type	**Substances**	**Correct parameter**	**Estimated parameter n=201**[1]	**Estimated parameter n=16**	**Estimated parameter n=8**
1	$A_1 \longrightarrow A_2$	0.04	0.040	0.022	0.037
1	$D_1 \longrightarrow D_2$	0.08	0.080	0.036	0.092
2	$A_1 \longrightarrow A_2$	0.02	0.020	0.0092	0.023
2	$A_2 \longrightarrow A_1$	0.02	0.020	0.0088	0.020
2	$B_1 \longrightarrow B_2$	0.02	0.020	0.019	0.014
2	$B_2 \longrightarrow B_1$	0.06	0.060	0.056	0.041
2	$C_1 \longrightarrow C_2$	0.02	0.020	0.018	0.016
2	$C_2 \longrightarrow C_1$	0.06	0.060	0.056	0.050
2	$D_1 \longrightarrow D_2$	0.04	0.040	0.015	0.054
2	$D_2 \longrightarrow D_1$	0.08	0.080	0.032	0.098
2	$E_2 \longrightarrow E_1$	0.06	0.060	0.056	0.037
3	$B_1 \longrightarrow B_2$ (A_2)	0.10	0.10	0.094	0.068
4	$C_1 \longrightarrow C_2$ (B_2)	k=0.06 K_M=0.20	k=0.060 K_M=0.20	k=0.066 K_M=0.34	k=0.071 K_M=0.50
3	$A_2 \longrightarrow A_1$ (C_2)	0.20	0.20	0.10	0.20
3	$E_1 \longrightarrow E_2$ (D_2)	0.08	0.080	0.075	0.051
3	$E_2 \longrightarrow E_1$ (B_2)	0.14	0.140	0.13	0.089
Running time (s)			260	44	28

Table 5.3: *Results from reconstruction of Test model II, n is the number of data points per experiment. Parameters* $\delta_2 = 0.002$, $\delta_3 = 0.001$, $\delta_4 = 0.001$ *and* $\beta = 0.85$. *1) Linear interpolation.*

Results with stochastically simulated data with added noise

We now consider data simulated by the stochastic method and with measurement noise added. Again, we refer to Appendix B where plots of the data are shown.

To get an idea about to what extent data are disturbed, we first ran the parameter estimation by itself given the correct structures of Test models I and II. Note that we are not running the model identification algorithm. The results are presented in Appendix C, table C.1 and C.2. They show that parameters estimated from stochastically simulated data differ from the original parameters, but that the difference gets smaller, with data that were averaged over several simulations, which is to be expected. Data with a higher level of added noise, naturally, make the result worse.

We now consider model identification from stochastically simulated data. Note that it is only the data that are simulated in a stochastic manner, the simulations within the algorithm are still deterministic. The results obtained for Test models I and II are summarized in table 5.4 and 5.5. Both the structure of Test model I and II were identified using data averaged from several stochastic simulations. The models were almost fully identified using data with moderate levels of added noise. For some of the non-identified reactions, the corresponding reaction of type 4 was found instead of the correct reaction. Thus, the principal structure of the pathway, but not the correct kinetic behaviour was identified.

The reason why the correct structure is not found in some cases is because of the noisy data. The structure found gives an error (according to our current error function 5.1) that is lower than the error of the correct structure. We note that adjustment of the error function may improve the ability of the algorithm to find the correct structure. We also note that the reactions that are not found by the algorithm generally have small rate constants.

The running times of the algorithm are similar to those presented in table 5.2 and 5.3.

Type	**Substances**	**Correct parameter**	**cells=50 c=0**	**cells=50 c=0.2**	**cells=50 c=0.5**	**cells=50 c=1.0**
1	$A_1 \longrightarrow A_2$	0.04	0.027	0.025	0.021	0.032
2	$A_1 \longrightarrow A_2$	0.02	0.012	0.011	0.0093	0.020
2	$A_2 \longrightarrow A_1$	0.02	0.012	0.011	0.0086	0.017
2	$B_1 \longrightarrow B_2$	0.02	0.018	0.031	0.036	0.045
2	$B_2 \longrightarrow B_1$	0.06	0.054	0.080	0.088	0.016
2	$C_1 \longrightarrow C_2$	0.02	0.012	0.019	0.029	-
2	$C_2 \longrightarrow C_1$	0.06	0.036	0.058	0.090	-
3	$B_1 \longrightarrow B_2$ (A_2)	0.10E-3	0.093E-3	-	-	-
3	$C_1 \longrightarrow C_2$ (B_2)	0.06E-3	0.038E-3	-	-	-
3	$A_2 \longrightarrow A_1$ (C_2)	0.20E-3	0.13E-3	0.12E-3	0.10E-3	-
4	$B_1 \longrightarrow B_2$ (A_2)	-	-	k=0.13	k=0.17	-
		-	-	K_M=580	K_M=810	-
4	$C_1 \longrightarrow C_2$ (B_2)	-	-	k=0.062	k=0.13	-
		-	-	K_M=480	K_M=810	-
4	$A_2 \longrightarrow A_1$ (C_2)	-	-	-	-	k=0.25
		-	-	-	-	K_M=990
4	$B_2 \longrightarrow B_1$ (A_1)	-	-	-	-	k=0.041
		-	-	-	-	K_M=290

Table 5.4: *Typical results from model identification of Test model I given stochastic data. cells = number of cells (simulations) from which the average value is calculated. c = measurement noise constant (see equation 4.1). The symbol - indicates that a reaction is not present in the structure. The four last reactions are not included in the correct structure. The number of experimental data points per experiment is 25 in all runs. Parameters* $\delta_2 = 0.002$, $\delta_3 = 0.001E-3$, $\delta_4 = 0.001E-3$ *and* $\beta = 0.9$.

Type	**Substances**	**Correct parameter**	**cells=50 c=0**	**cells=50 c=0.2**	**cells=50 c=0.5**	**cells=50 c=1.0**
1	$A_1 \longrightarrow A_2$	0.04	0.028	0.027	0.021	0.021
1	$D_1 \longrightarrow D_2$	0.08	0.050	0.049	0.045	0.039
2	$A_1 \longrightarrow A_2$	0.02	0.014	0.013	0.011	0.021
2	$A_2 \longrightarrow A_1$	0.02	0.013	0.013	0.0093	0.014
2	$B_1 \longrightarrow B_2$	0.02	0.020	0.027	0.040	0.0060
2	$B_2 \longrightarrow B_1$	0.06	0.061	0.065	0.10	0.016
2	$C_1 \longrightarrow C_2$	0.02	0.018	0.014	0.040	-
2	$C_2 \longrightarrow C_1$	0.06	0.054	0.042	0.12	0.0018
2	$D_1 \longrightarrow D_2$	0.04	0.025	0.025	0.023	0.021
2	$D_2 \longrightarrow D_1$	0.08	0.049	0.048	0.044	0.038
2	$E_2 \longrightarrow E_1$	0.06	0.060	0.034	-	-
3	$B_1 \longrightarrow B_2$ (A_2)	0.10E-3	0.10E-3	-	-	0.026E-3
4	$C_1 \longrightarrow C_2$ (B_2)	k=0.06 K_M=200	k=0.064 K_M=340	k=0.047 K_M=270	k=0.19 K_M=630	- -
3	$A_2 \longrightarrow A_1$ (C_2)	0.20E-3	0.14E-3	0.13E-3	0.11E-3	-
3	$E_1 \longrightarrow E_2$ (D_2)	0.08E-3	0.080E-3	0.048E-3	-	-
3	$E_2 \longrightarrow E_1$ (B_2)	0.14E-3	0.14E-3	0.083E-3	-	-
4	$B_1 \longrightarrow B_2$ (A_2)	- -	- -	k=0.12 K_M=780	k=0.22 K_M=1000	- -
4	$A_2 \longrightarrow A_1$ (C_2)	- -	- -	- -	- -	k=0.095 K_M=320
3	$C_1 \longrightarrow C_2$ (B_2)	-	-	-	-	0.0053E-3
3	$E_1 \longrightarrow E_2$ (D_1)	-	-	-	-	0.0048E-3
3	$E_2 \longrightarrow E_1$ (B_1)	-	-	-	-	0.0064E-3

Table 5.5: *Typical results from model identification of Test model II given stochastic data. cells = number of cells (simulations) from which the average value is calculated. c = measurement noise constant (see equation 4.1). The symbol - indicates that a reaction is not present in the structure. The five last reactions are not included in the correct structure. The number of experimental data points per experiment is 25 in all runs. Parameters* $\delta_2 = 0.002$, $\delta_3 = 0.001E-3$, $\delta_4 = 0.001E-3$ *and* $\beta = 0.9$.

5.8 Extension to handle an incomplete dataset

The parameter estimation presented in section 5.2 will not do for an incomplete experimental dataset, where data from at least one substance is missing. In an first attempt to show the feasibility of methods of this kind when some of the data is missing we apply a more general method, *Powell's method* [32]. It minimizes the error function by searching the full parameter space for a given model structure. In general, *Powell's method* is used to find a parameter set that minimizes a function, for which the gradient can not be calculated. The search starts at a point P in the N-dimensional parameter space, and proceeds from there in some vector direction. In order to calculate the length of the step, a line minimization sub-algorithm is called. The method consists of sequences of such line minimizations. At each step, the next direction to try is chosen. This is done by testing several (N) directions and calculate the best possible direction (by an heuristic function of the test results). For a more thoroughly description of *Powell's method*, we refer to [32].

This approach is more accurate than the former parameter estimation method (section 5.2), because the correct error function is minimized. However, the drawback is a dramatically extended computational time. The algorithm makes several function evaluations. In order to evaluate the error function, the model must be simulated and the error calculated. Thus, most of the computational time is spent on simulation and error calculation. The principle of the model identification algorithm is not affected by the change of parameter estimation method.

An incomplete dataset was created by removing the experiments [B_2, wild-type, step(20)] and [B_2, Gene deletion C, step(20)] from the set of experiments belonging to Test model I. The algorithm successfully identified the parameters given the correct structure.

It was also possible to reconstruct the structure of Test model I from the reduced dataset. In this case, the steady-state concentrations of B_1 and B_2 were assumed known. Furthermore, the input structure presented in figure 5.1 was slightly modified. The reaction $C_1 \longrightarrow C_2$ (B_2) was added. Without this change substance B would have no connection to the other substances in the model. We would also like to emphasize that it is impossible to predict both structure and parameters of a model, if a substance that misses data has no structural connection to other substances. The running time was about 8 hours. Test model II was not tested with an incomplete dataset, since the present algorithm is not fast enough.

To summarize, we have demonstrated that it is possible to run the model identification algorithm with a reduced data set. The computational time is

dramatically increased, but this was not our main focus at this stage. It is probably possible to significantly reduce the computational time of similar algorithms in the future.

Chapter 6

The prototype software tool

The prototype software tool realizes the integrated data simulation and model identification environment presented in figure 1.2 in chapter 1. It is possible to work with models of biological systems and experimental data within the same application. These two components are combined by the possibility to run simulations and use the model identification algorithm to go backwards from experiment to model. The software tool is not built for a specific biological type of system, but is intended to be as general as possible.

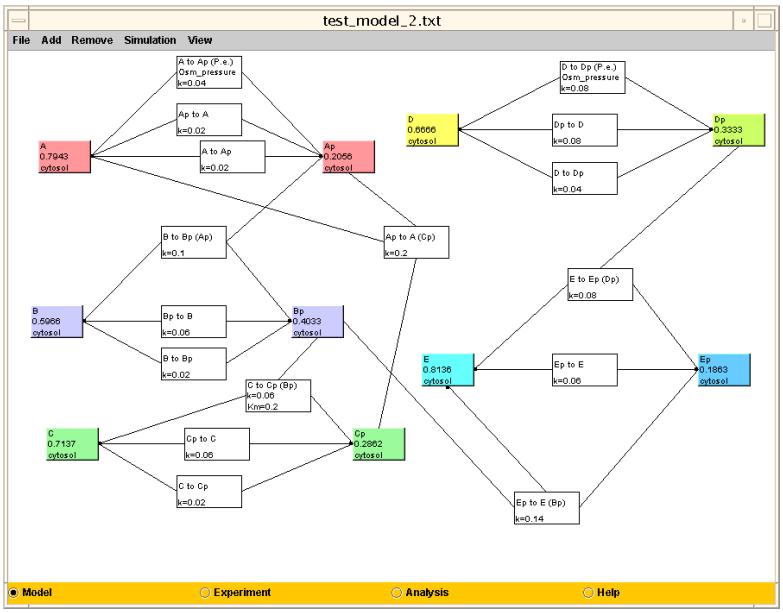


Figure 6.1: *Snapshot of the model panel.*

There are two main panels within the application: the model panel and the

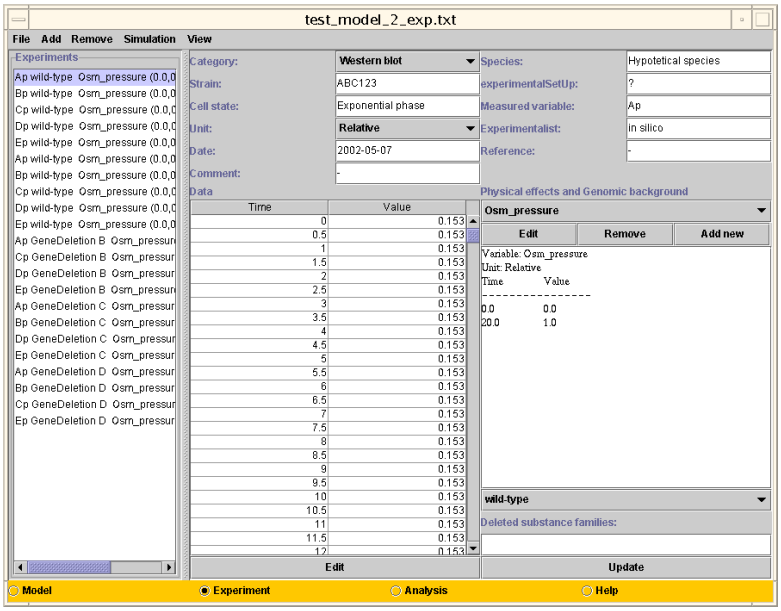


Figure 6.2: *Snapshot of the experiment panel.*

experiment panel. In the model panel, a model can be built by defining a set of substances and by connecting them with reactions. In figure 6.1, Test model II is shown. All substances and reactions are represented as boxes in the graphical user interface. When a substance box or a reaction box is double-clicked, a dialog for setting the attributes (parameters etc.) appears. To create and remove objects, *Add* and *Remove* in the main menu are used. In the experiment panel, experiments are specified and visualised. In figure 6.2, the specification of the experiments for Test model II is shown. All attributes of an experiment are easily set within the application.

From the model panel, it is possible to simulate the model. Two simulation algorithms are implemented, *Euler's method* and *fifth order Runge-Kutta Method with adaptive step-size*. In figure 6.3 a plot frame of the experimental data of Test model II is shown. The plot frame shows up at the end of a simulation. The analysis algorithm is started and monitored from an analysis panel, which is also shown in figure 6.3.

The main target group of the software tool are biologists and bioinformaticians. In the future development the usability of the software system is of great importance. That involves improvements of the graphical user interface, but also to carefully decide which mathematical and algorithmic details, that should be presented for the user and which should be hidden. To fully make use of the expertise of the biologists, they should be forced to translate their knowledge into mathematical expressions valuable for a

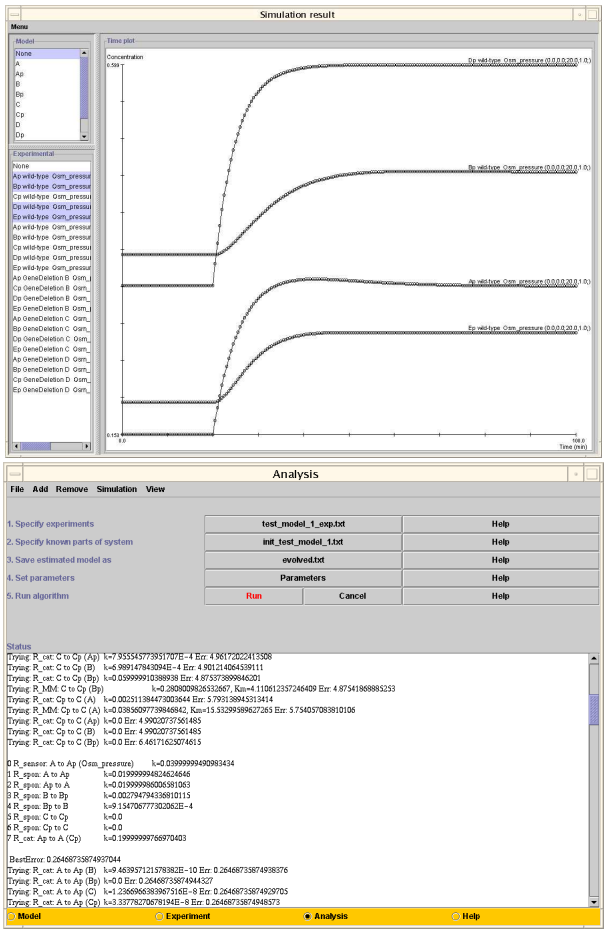


Figure 6.3: *Snapshot from the software tool showing a plot frame (upper) and the control frame for the analysis algorithm (lower).*

model. Partly, that can be done by letting the software tool ask relevant question in a non-mathematical language.

Another important issue in the future development of the software is to improve the educational use of the program, for biologists and bioinformaticians, but also for mathematicians and computer scientists. A software tool could help people from these disciplines to learn more about the other subjects. The tool could also facilitate communication between these groups when exchanging ideas.

The prototype software tool is implemented in Java and is thus portable between different operating systems.

Chapter 7

Discussion

The main result of this thesis is an algorithm for reconstructing signalling pathways from time series data. The algorithm reconstructs both the structure and the parameters of two test models given deterministically simulated data. The algorithm takes advantage of data from several different experiment categories at the same time. It is possible to include non-linear reactions w.r.t. the parameters by applying a non-linear parameter estimation algorithm.

The test results indicate that the algorithm can potentially handle biologically realistic situations. First of all, the number of measurement points can be reduced to acceptable levels. Secondly, the algorithm can handle data that are simulated stochastically and that have measurement noise added to them. Finally, we have demonstrated that it is possible to use an incomplete dataset in order to identify a model. We would like to emphasize that the worst-case model identification scenarios have been tested, since only a basic initial structure has been assumed. In reality, parts of the structure are usually known.

The main effort in the development of the model identification algorithm has been to increase its speed, both in order to make it attractive to users and to enable us to study its behaviour conveniently. The running time of the algorithm is considerably lower than other model identification algorithms in the literature.

7.1 Modelling of signalling pathways

In this work only four different types of reactions are used. As mentioned in chapter 3, this is a too small set to fully model a signalling pathway. However, it is straightforward to include additional reaction types. For

instance, a two-substrate and two-product reaction could be added. It is also possible to have different reaction types with the same variables, something that was demonstrated by the two catalysed reactions (reaction types 3 and 4).

It is not only the available number of reaction types that limits the possibility to create realistic models. Also the response of the pathway must be taken into consideration in order to properly model the HOG pathway. The pathway stimulates glycerol production in order to increase the intracellular turgor pressure, and a model must probably include parts of the metabolism to be realistic. Thus, the set of available reactions has to be extended in order to handle transmembrane transport (cytosol to nucleus), transcription (DNA to mRNA) and translation (mRNA to protein). Furthermore, a thermodynamic model of osmoregulation including variables such as turgor pressure and volume must probably also be included.

7.2 Analysis of real experimental data

At this stage the algorithm has not been tested on real experimental data from the HOG pathway. As mentioned in section 7.1, the modelling itself demands additional components in order to be realistic. Besides, there is a gap between the structure and data requirements of the algorithm on one side and the available experimental data on the other side. The gap is due to several different causes, which indicate the limitations of the present version of the algorithm, as well as the limitations of available experimental data. The limitations discussed below are divided into two groups: minor and major limitations.

Minor limitations

- The algorithm allows at most two states of each substance. In biological systems several states may be present. For instance, in the HOG signalling pathway there are at least three different states of Hog1; Hog1, Hog1^{P} and Hog1^{PP}. It is easy to allow for several states in the algorithm, but the demand for data would increase. In the above example, data for at least two of the three states would be necessary. In general, data from (n-1) out of n states are required.

- The algorithm requires experimental data points given with correct units (non-normalised). In reality, time series data are normalised between 0 and 1. Although the data is normalised, the structure is not dependent of the scaling. Thus, the structure will be correct but

the parameters will not. Rescaling of the parameters might adjust for that, if partial knowledge of the correct model is known. Such knowledge include steady-state distribution of the states, that is, what fraction of the molecules is in state i at steady state. In principle, it is easy to construct such an algorithm.

- The total concentration X_{tot} of each substance is assumed known in the algorithm. The real concentrations are not known but can be estimated from the literature, see section 2.4.

- The principle of incrementally adding one reaction to the model, may not be sufficient in all situations. It is possible to come up with situations where it is necessary to test all different combinations of two reactions in order to get the correct result. It is simple to change the top level algorithm to do this. The cost is an increase in computing time.

Major limitations

- The algorithm requires time series data for all substances in the model in order to be fast in practice. Presently, experimental time series data is only available for a couple of the substances involved in the HOG signalling pathway. Missing data is a fundamental algorithmic difficulty, which can be tackled in several different ways. In general, algorithms that can handle this are considerably slower, compared to the first algorithm presented (based on the least-square method parameter estimation).

  Using *Powell's method* for parameter estimation, we demonstrate that it is possible to run the algorithm with an incomplete dataset, but it is presently too slow to be attractive. However, there are ways of speeding up the method. It might also be possible to use the least-square approach for all possible situations and then automatically switch to methods like *Powell's method* for unresolved sub-problems.

  We would also like to emphasize that there are other ways to decrease the complexity of the analysis. For example, it could be possible to include constraints on the full model in order to limit the space of possible models. Such constraints could be extracted from public databases of the yeast proteins. The main difficulty is that the information is given in textual format. Thus, one has to translate the information into mathematical or logical form.

- Due to limited resources, time series data from biological experiments are usually collected from less than ten measurement points. Besides,

there are several sources of measurement error as discussed in section 4.2. This further restricts the capacity of the algorithm working on real data. The solution to these problems is not easy. Further development of the used error function (and/or the termination criterion), a good model of the measurement errors, proper filter and interpolation methods etc. help to extract the information. From the experimental side, new techniques such as protein chips, may lead to larger datasets with higher quality.

- As mentioned, the number of signalling proteins in a cell is not very high. Therefore, stochastic fluctuations may be large enough to affect the system. To measure the average value of several cells lead to a more deterministic shape of the experimental time series, but a systematic error may be present. This is especially true, if there are non-linear reactions in the model. As an example, the effects of stochastic fluctuations of proteins in *E. coli* cells have been studied by Bray and co-workers [28, 37, 38, 39]. They have built a differential equation model of the biochemical reaction steps behind the way the swimming behaviour of the cells. By introducing stochasticity into the model, they found that the model can predict the distribution of individual cells with different swimming behaviours. This example highlights the need to consider stochastic fluctuations in signalling systems. In order to include stochastic aspects in the model identification algorithm, estimates of the variances must be considered. One way of doing this is to perform series of stochastic simulations. However, that would be a very time-consuming strategy, since stochastic simulation requires more computing time than deterministic simulation do.

The result presented in this thesis is an important first step in order to realize the future plans, where real biological systems and real experimental data will be considered.

Appendix A

Plots of deterministic data

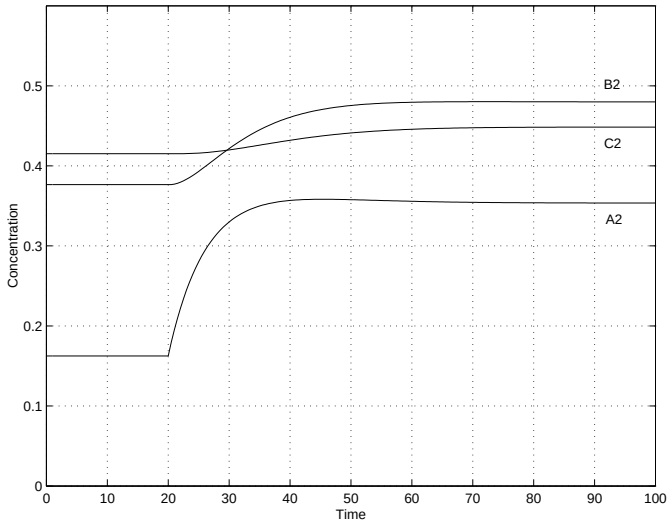


Figure A.1: *Test model 1, deterministic simulation of wild-type experiments, step(20) as physical effect.*

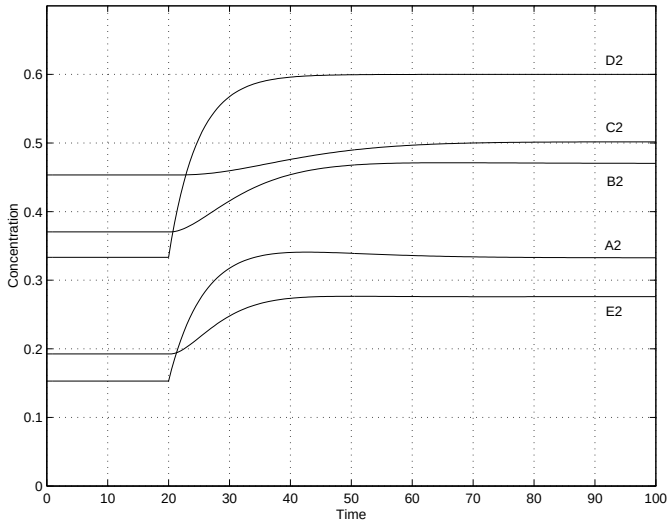


Figure A.2: *Test model 2, deterministic simulation of wild-type experiments, step(20) as physical effect.*

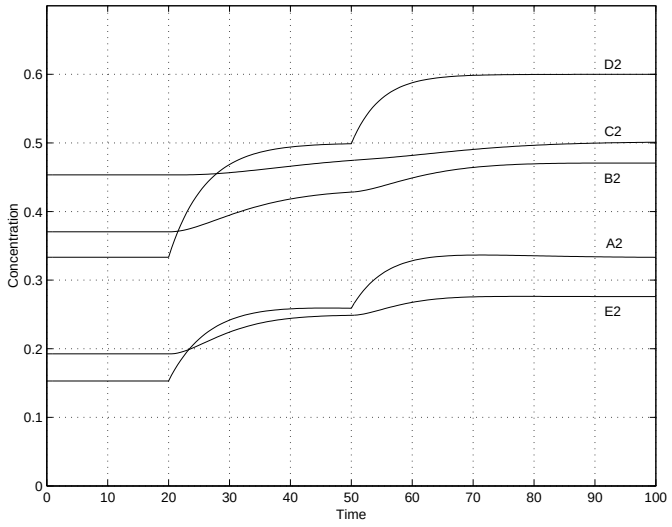


Figure A.3: *Test model 2, deterministic simulation of wild-type experiments, stairs(20,50) as physical effect.*

Appendix B

Plots of stochastic data

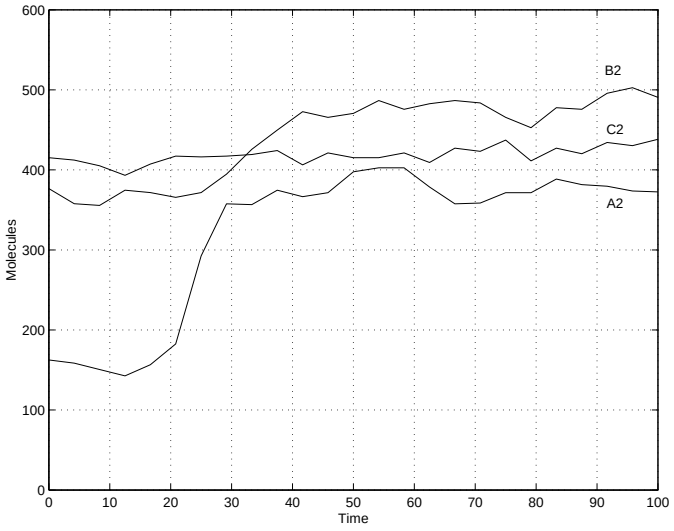


Figure B.1: *Test model 1, stochastic simulation of wild-type experiments, step(20) as physical effect. 25 measurement points for each experiment, 1 cell, no measurement noise (measurement noise constant=0).*

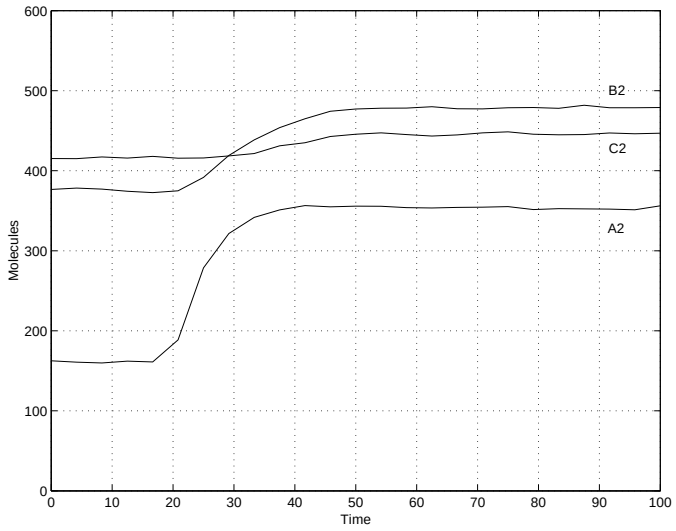


Figure B.2: *Test model 1, stochastic simulation of wild-type experiments, step(20) as physical effect. 25 measurement points for each experiment, average values of 50 cells, no measurement noise (measurement noise constant=0).*

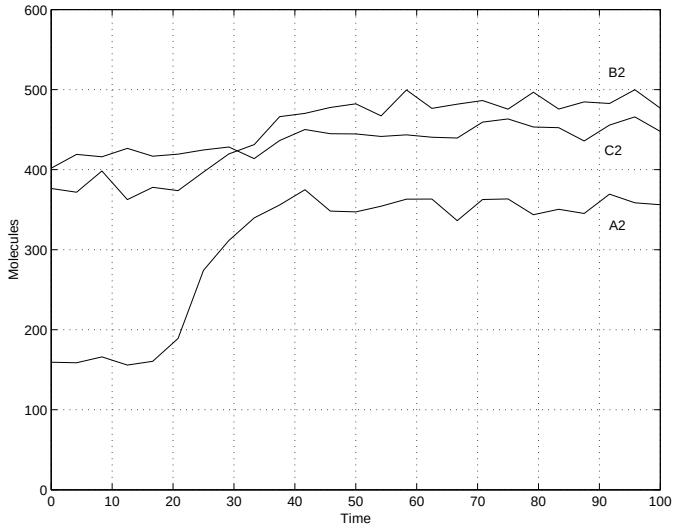


Figure B.3: *Test model 1, stochastic simulation of wild-type experiments, step(20) as physical effect. 25 measurement points for each experiment, average values of 50 cells, ,measurement noise is added (measurement noise constant=0.2).*

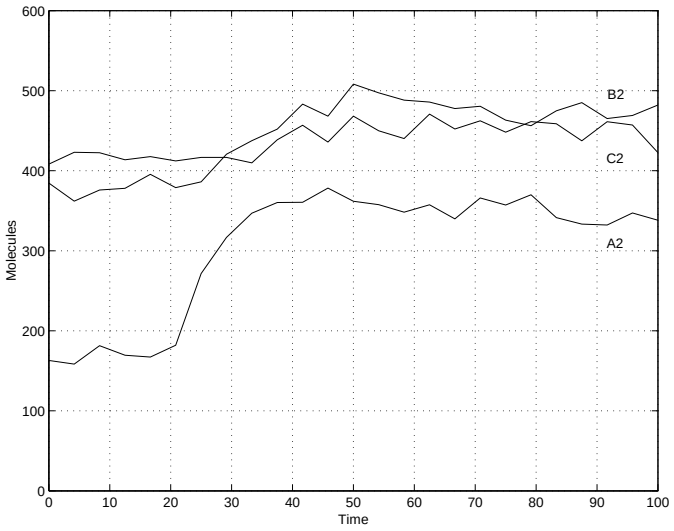


Figure B.4: *Test model 1, stochastic simulation of wild-type experiments, step(20) as physical effect. 25 measurement points for each experiment, average values of 50 cells, measurement noise is added (measurement noise constant=0.5).*

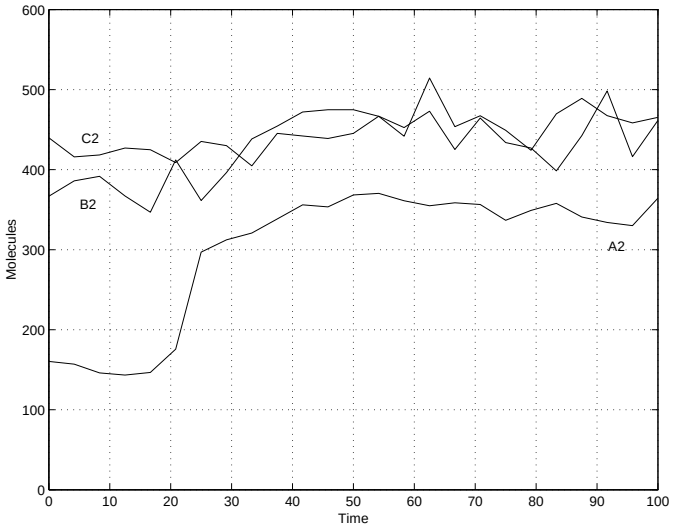


Figure B.5: *Test model 1, stochastic simulation of wild-type experiments, step(20) as physical effect. 25 measurement points for each experiment, average values of 50 cells, measurement noise is added (measurement noise constant=1).*

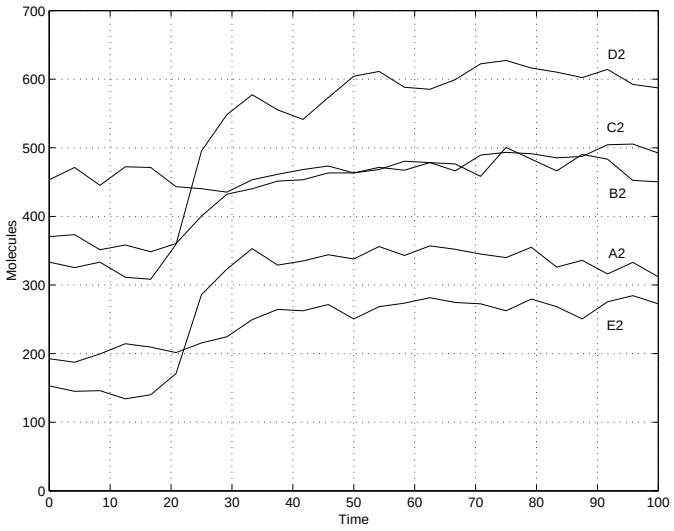


Figure B.6: *Test model 2, stochastic simulation of wild-type experiments, step(20) as physical effect. 25 measurement points for each experiment, 1 cell, no measurement noise (measurement noise constant=0).*

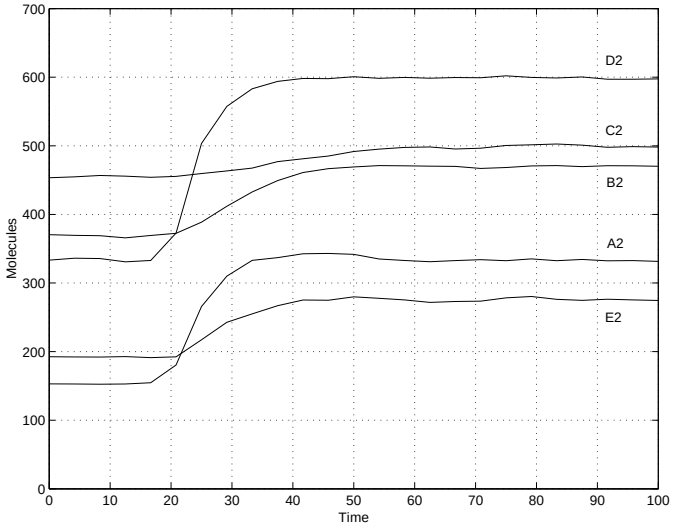


Figure B.7: *Test model 2, stochastic simulation of wild-type experiments, step(20) as physical effect. 25 measurement points for each experiment, average values of 50 cells, no measurement noise (measurement noise constant=0).*

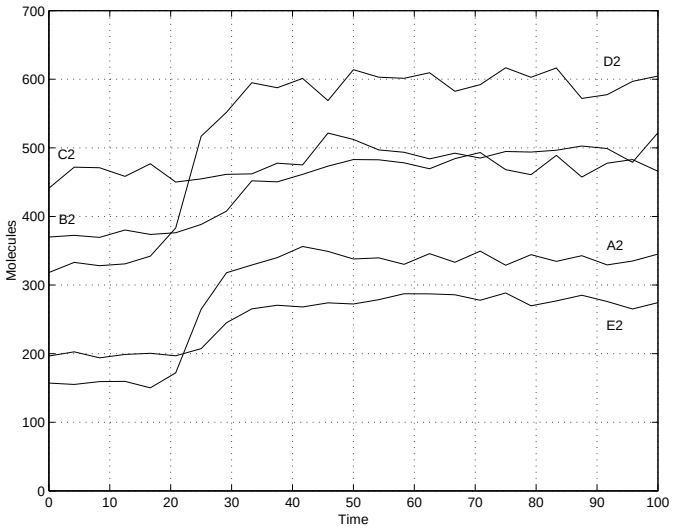


Figure B.8: *Test model 2, stochastic simulation of wild-type experiments, step(20) as physical effect. 25 measurement points for each experiment, average values of 50 cells, measurement noise is added (measurement noise constant=0.2).*

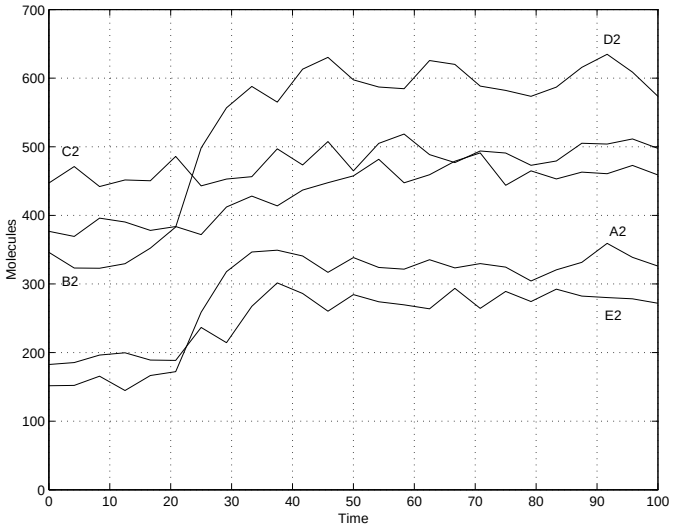


Figure B.9: *Test model 2, stochastic simulation of wild-type experiments, step(20) as physical effect. 25 measurement points for each experiment, average values of 50 cells, measurement noise is added (measurement noise constant=0.5).*

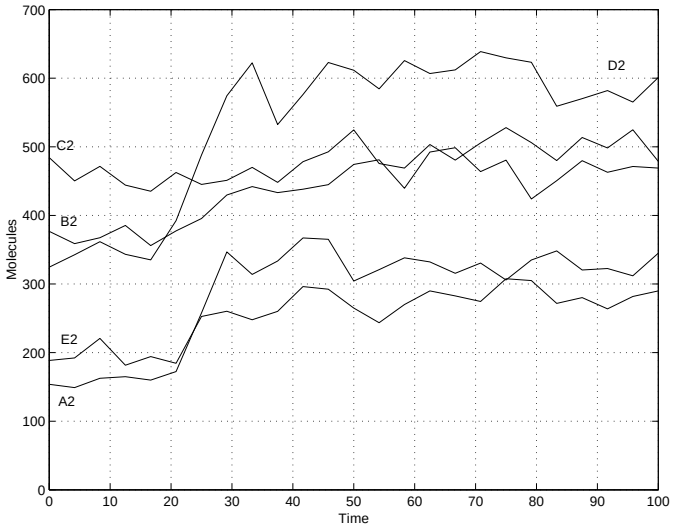


Figure B.10: *Test model 2, stochastic simulation of wild-type experiments, step(20) as physical effect. 25 measurement points for each experiment, average values of 50 cells, measurement noise is added (measurement noise constant=1.0).*

Appendix C

Parameter estimation from stochastic data

Type	**Substances**	**Correct parameter**	**Estimated parameter cells=1 c=0**	**Estimated parameter cells=50 c=0**	**Estimated parameter cells=50 c=0.5**
1	$A_1 \longrightarrow A_2$	0.04	0.026	0.026	0.029
2	$A_1 \longrightarrow A_2$	0.02	0.012	0.012	0.013
2	$A_2 \longrightarrow A_1$	0.02	0.012	0.012	0.013
2	$B_1 \longrightarrow B_2$	0.02	0.0083	0.019	0.0053
2	$B_2 \longrightarrow B_1$	0.06	0.021	0.060	0.010
2	$C_1 \longrightarrow C_2$	0.02	0.0012	0.013	0.00051
2	$C_2 \longrightarrow C_1$	0.06	0.0034	0.039	0.0014
3	$B_1 \longrightarrow B_2$ (A_2)	0.10E-3	0.0037E-3	0.040E-3	0.0034E-3
3	$C_1 \longrightarrow C_2$ (B_2)	0.06E-3	0.13E-3	0.13E-3	0.14E-3
3	$A_2 \longrightarrow A_1$ (C_2)	0.20E-3	0.036E-3	0.10E-3	0.016E-3

Table C.1: *Results from parameter estimation of Test model I given the correct structure and stochastic data, the number of experimental data points per experiment is 25 in all runs. cells = number of cells (simulations) from which the average value is calculated. c = measurement noise constant (see equation 4.1).*

Type	**Substances**	**Correct parameter**	**Estimated parameter cells=1 c=0**	**Estimated parameter cells=50 c=0**	**Estimated parameter cells=50 c=0.5**
1	$A_1 \longrightarrow A_2$	0.04	0.025	0.028	0.021
1	$D_1 \longrightarrow D_2$	0.08	0.044	0.050	0.045
2	$A_1 \longrightarrow A_2$	0.02	0.013	0.014	0.011
2	$A_2 \longrightarrow A_1$	0.02	0.012	0.013	0.0093
2	$B_1 \longrightarrow B_2$	0.02	0.0073	0.020	0.0080
2	$B_2 \longrightarrow B_1$	0.06	0.018	0.061	0.022
2	$C_1 \longrightarrow C_2$	0.02	0.014	0.018	0.040
2	$C_2 \longrightarrow C_1$	0.06	0.041	0.054	0.12
2	$D_1 \longrightarrow D_2$	0.04	0.022	0.025	0.023
2	$D_2 \longrightarrow D_1$	0.08	0.043	0.049	0.044
2	$E_2 \longrightarrow E_1$	0.06	0.025	0.060	0.020
3	$B_1 \longrightarrow B_2$ (A_2)	0.10E-3	0.029E-3	0.10E-3	0.036E-3
4	$C_1 \longrightarrow C_2$ (B_2)	k=0.06 K_M=200	k=0.036 K_M=120	k=0.065 K_M=340	k=0.19 K_M=630
3	$A_2 \longrightarrow A_1$ (C_2)	0.20E-3	0.13E-3	0.14-3	0.11E-3
3	$E_1 \longrightarrow E_2$ (D_2)	0.08E-3	0.039E-3	0.080E-5	0.029E-3
3	$E_2 \longrightarrow E_1$ (B_2)	0.14E-3	0.070E-3	0.14E-3	0.051E-3

Table C.2: *Results from parameter estimation of Test model II given the correct structure and stochastic data, the number of experimental data points per experiment is 25 in all runs. cells = number of cells (simulations) from which the average value is calculated. c = measurement noise constant (see equation 4.1).*

Bibliography

[1] Doyle JC Csete ME. Reverse engineering of biological complexity. *Science*, 295(5560):1664–9, 2002.

[2] Lanza G Yu J Keane MA Koza JR, Mydlowec W. Reverse engineering of metabolic pathways from observed data using genetic programming. *Pac Symp Biocomput*, pages 434–45, 2001.

[3] Kitano H Morohashi M. Identifying gene regulatory networks from time series expression data by in silico sampling and screening. *Proc. of the 5th European Conference on Artificial Life*, pages 477–86.

[4] Somogyi R Liang S, Fuhrman S. Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pac Symp Biocomput*, pages 18–29, 1998.

[5] Wedelin D. Efficient estimation and model selection in large scale graphical models. *Statistics and computing*, 6:313–323, 1996.

[6] Hohmann S. Osmotic stress signaling and osmoadaptation in yeasts. *Microbiol Mol Biol Rev*, 66(2):300–72, 2002.

[7] Alexander M Davenport K Gustin MC, Albertyn J. Map kinase pathways in the yeast saccharomyces cerevisiae. *Microbiol Mol Biol Rev*, 62(4):1264–1300, 1998.

[8] Maeda T Witten EA Thai TC Saito H Posas F, Wurgler-Murphy SM. Yeast hog1 map kinase cascade is regulated by a multistep phosphorelay mechanism in the sln1-ypd1-ssk1 two-component osmosensor. *Cell*, 86(6):865–75, 1996.

[9] Ammerer G Reiser V, Salah SM. Polarized localization of yeast pbs2 depends on osmostress, the membrane protein sho1 and cdc42. *Nat Cell Biol.*, 2(9):620–7, 2000.

[10] Saito H. Raitt DC, Posas F. Yeast cdc42 gtpase and ste20 pak-like kinase regulate sho1-dependent activation of the hog1 mapk pathway. *EMBO J*, 19(17):4623–31, 2000.

[11] Saito H Maeda T, Takekawa M. Activation of yeast pbs2 mapkk by mapkkks or by binding of an sh3-containing osmosensor. *Science*, 269(5223):554–9, 1995.

[12] Saito H Posas F. Activation of the yeast ssk2 map kinase kinase kinase by the ssk1 two-component response regulator. *EMBO J*, 17(5):1385–94, 1998.

[13] Witten EA Saito H Wurgler-Murphy SM, Maeda T. Regulation of the saccharomyces cerevisiae hog1 mitogen-activated protein kinase by the ptp2 and ptp3 protein tyrosine phosphatases. *Mol Cell Biol*, 17(3):1289–97, 1997.

[14] Saito H Posas F, Witten EA. Requirement of ste50 for osmostress-induced activation of the ste11 mitogen-activated protein kinase kinase kinase in the high-osmolarity glycerol response pathway. *Mol Cell Biol*, 18(10):5788–96, 1998.

[15] Ota IM Mattison CP. Two protein tyrosine phosphatases, ptp2 and ptp3, modulate the subcellular localization of the hog1 map kinase in yeast. *Genes Dev*, 14(10):1229–35, 2000.

[16] Johnson GL Garrington TP. Organization and regulation of mitogen-activated protein kinase signaling pathways. *Curr Opin Cell Biol*, 11(2):211–8, 1999.

[17] Saito H Posas F. Osmotic activation of the hog mapk pathway via ste11p mapkkk: scaffold role of pbs2p mapkk. *Science*, 276(5319):1702–5, 1997.

[18] Iyengar R Weng G, Bhalla US. Complexity in biological signaling systems. *Science*, 284(5411):92–6, 1999.

[19] Sternberg PW Levchenko A, Bruck J. Scaffold proteins may biphasically affect the levels of mitogen-activated protein kinase signaling and reduce its threshold properties. *Proc Natl Acad Sci U S A*, 97(11):5818–23, 2000.

[20] Iyengar R Bhalla US. Emergent properties of networks of biological signaling pathways. *Science*, 283(5400):381–7, 1999.

[21] Stryer L. *Biochemistry*. WH Freeman and Company, New York, 4th edition, 1995.

[22] Ferrell JE Huang CY. Ultrasensitivity in the mitogen-activated protein kinase cascade. *Proc Natl Acad Sci U S A*, 93(19):10078–83, 1996.

[23] Lauffenburger DA Asthagiri AR. A computational study of feedback effects on signal dynamics in a mitogen-activated protein kinase (mapk) pathway model. *Biotechnol. Prog.*, 17(2):227–23, 2001.

[24] Gilles ED Muller G Schoeberl B, Eichler-Jonsson C. Computational modeling of the dynamics of the map kinase cascade activated by surface and internalized egf receptors. *Nat Biotechnol*, 20(4):370–5, 2002.

[25] Lay S Bray D. Computer simulated evolution of a network of cell-signaling molecules. *Biophys J*, 66(4):972–7, 1994.

[26] Bhatt RR Ferrell JE Jr. Mechanistic studies of the dual phosphorylation of mitogen-activated protein kinase. *J Biol Chem*, 272(30):19008–16, 1997.

[27] Kholodenko BN. Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades. *Eur J Biochem*, 267(6):1583–8, 2000.

[28] Bray D Morton-Firth CJ. Predicting temporal fluctuations in an intracellular signalling pathway. *J Theor Biol*, 192(1):117–28, 1998.

[29] Bruck J Gibson MA. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem. A*, 104:1876–1889, 2000.

[30] Chang EC Cairns BR Thorner J Bardwell L, Cook JG. Signaling in the yeast pheromone response pathway: specific and high-affinity interaction of the mitogen-activated protein (map) kinases kss1 and fus3. *Mol Cell Biol 1996*, 16(7):3637–50, 1996.

[31] Gillespie DT. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Computational Physics*, 22:403–434, 1976.

[32] Vetterling WT Flannery BP Press WH, Teukolsky SA. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, 1993.

[33] Marquardt DW. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11:431–441, 1963.

[34] Wittmeyer-Koch L Elden L. *Numerisk analys - en introduktion*. Studentlitteratur, Lund, 3rd edition, 1996.

[35] Heath M.T. *Scientific computing, an introductory survey*. The McGraw-Hill Companies, Inc., 1997.

[36] Hicklin J. et al. *http://math.nist.gov/javanumerics/jama/*. Free linear algebra package for Java by The MathWorks and National Institute of standards and technology.

[37] Abouhamad WN Bourret RB Bray D Levin MD, Morton-Firth CJ. Origins of individual swimming behavior in bacteria. *Biophys J*, 74(1):175–81, 1998.

[38] Simon MI Bray D, Bourret RB. Computer simulation of the phosphorylation cascade controlling bacterial chemotaxis. *Mol Biol Cell*, 4(5):469–82, 1993.

[39] Schuster M Boesch KC Silversmith RE Bourret RB Abouhamad WN, Bray D. Computer-aided resolution of an experimental paradox in bacterial chemotaxis. *J Bacteriol*, 180(15):3757–64, 1998.