Linköping Studies in Science and Technology
Dissertations No. 1877

Model-Based Hypothesis Testing in Biomedicine

How Systems Biology Can Drive the
Growth of Scientific Knowledge

Rikard Johansson

Department of Biomedical Engineering
Linköping University, Sweden
Linköping 2017

Cover page: 2D density estimations of bootstrap distributions, as applied to model discrimination and hypothesis testing in insulin signaling.

Model-Based Hypothesis Testing in Biomedicine

© 2017 Rikard Johansson, unless otherwise noted

Department of Biomedical Engineering
Linköping University, Sweden

Linköping Studies in Science and Technology
Dissertation No. 1877

ISBN 978-91-7685-457-0 ISSN 0345-7524

Printed by LiU-Tryck, Linköping 2017

For my grandparents
Elis, the fighter
Sigrid, home and hearth
Lennart, the storyteller
Gun, the rebel

*“Ah, there's nothing more exciting than science.
You get all the fun of sitting still, being quiet, writing
down numbers, paying attention. Science has it all.”*

Seymour Skinner, the Simpsons

SUPERVISOR

Associate Professor Gunnar Cedersund, PhD

Department of Biomedical Engineering

Department of Clinical and Experimental Medicine

Linköping University

CO-SUPERVISORS

Professor Tomas Strömberg, PhD

Department of Biomedical Engineering

Linköping University

Professor Peter Strålfors, PhD

Department of Clinical and Experimental Medicine

Linköping University

FACULTY OPPONENT

Associate Professor Marija Cvijovic, PhD

Department of Mathematical Sciences

University of Gothenburg

ABSTRACT

The utilization of mathematical tools within biology and medicine has traditionally been less widespread compared to other hard sciences, such as physics and chemistry. However, an increased need for tools such as data processing, bioinformatics, statistics, and mathematical modeling, have emerged due to advancements during the last decades. These advancements are partly due to the development of high-throughput experimental procedures and techniques, which produce ever increasing amounts of data. For all aspects of biology and medicine, these data reveal a high level of inter-connectivity between components, which operate on many levels of control, and with multiple feedbacks both between and within each level of control. However, the availability of these large-scale data is not synonymous to a detailed mechanistic understanding of the underlying system. Rather, a mechanistic understanding is gained first when we construct a hypothesis, and test its predictions experimentally. Identifying interesting predictions that are quantitative in nature, generally requires mathematical modeling. This, in turn, requires that the studied system can be formulated into a mathematical model, such as a series of ordinary differential equations, where different hypotheses can be expressed as precise mathematical expressions that influence the output of the model.

Within specific sub-domains of biology, the utilization of mathematical models have had a long tradition, such as the modeling done on electrophysiology by Hodgkin and Huxley in the 1950s. However, it is only in recent years, with the arrival of the field known as systems biology that mathematical modeling has become more commonplace. The somewhat slow adaptation of mathematical modeling in biology is partly due to historical differences in training and terminology, as well as in a lack of awareness of showcases illustrating how modeling can make a difference, or even be required, for a correct analysis of the experimental data.

In this work, I provide such showcases by demonstrating the universality and applicability of mathematical modeling and hypothesis testing in three disparate biological systems. In Paper II, we demonstrate how mathematical modeling is necessary for the correct interpretation and analysis of dominant negative inhibition data in insulin signaling in primary human adipocytes. In Paper III, we use modeling to determine transport rates across the nuclear membrane in yeast cells, and we show how this technique is superior to traditional curve-fitting methods. We also demonstrate the issue of population heterogeneity and the need to account for individual differences between cells and the population at large. In Paper IV, we use mathematical modeling to reject three hypotheses concerning the phenomenon of facilitation in pyramidal nerve cells in rats and mice. We also show how one surviving hypothesis can explain all data and adequately describe independent validation data. Finally, in Paper I, we develop a method for model selection and discrimination using parametric bootstrapping and the combination of several different empirical distributions of traditional statistical tests. We show how the empirical log-likelihood ratio test is the best combination of two tests and how this can be used, not only for model selection, but also for model discrimination.

In conclusion, mathematical modeling is a valuable tool for analyzing data and testing biological hypotheses, regardless of the underlying biological system. Further development of modeling methods and applications are therefore important since these will in all likelihood play a crucial role in all future aspects of biology and medicine, especially in dealing with the burden of increasing amounts of data that is made available with new experimental techniques.

SVENSK SAMMANFATTNING

Användandet av matematiska verktyg har inom biologi och medicin traditionellt sett varit mindre utbredd jämfört med andra ämnen inom naturvetenskapen, såsom fysik och kemi. Ett ökat behov av verktyg som databehandling, bioinformatik, statistik och matematisk modellering har trätt fram tack vare framsteg under de senaste decennierna. Dessa framsteg är delvis ett resultat av utvecklingen av storskaliga datainsamlingstekniker. Inom alla områden av biologi och medicin så har dessa data avslöjat en hög nivå av interkonnektivitet mellan komponenter, verksamma på många kontrollnivåer och med flera återkopplingar både mellan och inom varje nivå av kontroll. Tillgång till storskaliga data är emellertid inte synonymt med en detaljerad mekanistisk förståelse för det underliggande systemet. Snarare uppnås en mekanisk förståelse först när vi bygger en hypotes vars prediktioner vi kan testa experimentellt. Att identifiera intressanta prediktioner som är av kvantitativ natur, kräver generellt sett matematisk modellering. Detta kräver i sin tur att det studerade systemet kan formuleras till en matematisk modell, såsom en serie ordinära differentialekvationer, där olika hypoteser kan uttryckas som precisa matematiska uttryck som påverkar modellens output.

Inom vissa delområden av biologin har utnyttjandet av matematiska modeller haft en lång tradition, såsom den modellering gjord inom elektrofysiologi av Hodgkin och Huxley på 1950-talet. Det är emellertid just på senare år, med ankomsten av fältet systembiologi, som matematisk modellering har blivit ett vanligt inslag. Den något långsamma adapteringen av matematisk modellering inom biologi är bl.a. grundad i historiska skillnader i träning och terminologi, samt brist på medvetenhet om exempel som illustrerar hur modellering kan göra skillnad och faktiskt ofta är ett krav för en korrekt analys av experimentella data.

I detta arbete tillhandahåller jag sådana exempel och demonstrerar den matematiska modelleringens och hypotestestningens allmängiltighet och tillämpbarhet i tre olika biologiska system. I Arbete II visar vi hur matematisk modellering är nödvändig för en korrekt tolkning och analys av dominant-negativ-inhiberingsdata vid insulinsignalering i primära humana adipocyter. I Arbete III använder vi modellering för att bestämma transporthastigheter över cellkärnmembranet i jästceller, och vi visar hur denna teknik är överlägsen traditionella kurvpassningsmetoder. Vi demonstrerar också frågan om populationsheterogenitet och behovet av att ta hänsyn till individuella skillnader mellan celler och befolkningen som helhet. I Arbete IV använder vi matematisk modellering för att förkasta tre hypoteser om hur fenomenet facilitering uppstår i pyramidala nervceller hos råttor och möss. Vi visar också hur en överlevande hypotes kan beskriva all data, inklusive oberoende valideringsdata. Slutligen utvecklar vi i Arbete I en metod för modellselektion och modelldiskriminering med hjälp av parametrisk ”bootstrapping” samt kombinationen av olika empiriska fördelningar av traditionella statistiska tester. Vi visar hur det empiriska ”log-likelihood-ratio-testet” är den bästa kombinationen av två tester och hur testet är applicerbart, inte bara för modellselektion, utan också för modelldiskriminering.

Sammanfattningsvis är matematisk modellering ett värdefullt verktyg för att analysera data och testa biologiska hypoteser, oavsett underliggande biologiskt system. Vidare utveckling av modelleringsmetoder och tillämpningar är därför viktigt eftersom dessa sannolikt kommer att spela en avgörande roll i framtiden för biologi och medicin, särskilt när det gäller att hantera belastningen från ökande datamängder som blir tillgänglig med nya experimentella tekniker.

PUBLICATIONS AND MANUSCRIPTS

PAPER I

Rikard Johansson, Peter Strålfors, Gunnar Cedersund.

Combining test statistics and models in bootstrapped model rejection: it is a balancing act.

BMC Systems Biology. 2014 Apr 17; 8:46.

PAPER II

David Julleson*, **Rikard Johansson***, Meenu R. Rajan, Peter Strålfors, Gunnar Cedersund.

Dominant negative inhibition data should be analyzed using mathematical modeling – Re-interpreting data from insulin signaling.

FEBS J. 2015 Feb; 282(4): 788-802.

PAPER III

Lucia Durrieu, **Rikard Johansson**, Alan Bush, David L. Janzén, Martin Gollvik, Gunnar Cedersund, Alejandro Colman-Lerner.

Quantification of nuclear transport in single cells.

In manuscript. Submitted to BioRxiv pre-print server**. 2014.

PAPER IV

Rikard Johansson, Sarah H. Lindström, Sofie C. Sundberg, Philip Blomström, Theodor Nevo, Karin Lundengård, Björn Granseth, Gunnar Cedersund.

Synaptotagmin 7 Vesicle Priming is Necessary and Sufficient for Explaining Facilitation in Murine Pyramidal Neurons.

In manuscript.

* These authors contributed equally to this work.

** Available on pre-print server. doi: <http://dx.doi.org/10.1101/001768>

ABBREVIATIONS

AIC	Akaike Information Criterion.
a.u.	Arbitrary units.
AUC	Area Under the Curve.
BDF	Backwards Differentiation Formula.
BIC	Bayesian Information Criterion.
BN	Bayesian Networks.
CNS	Central Nervous System.
DAG	Directed Acyclic Graph.
DFO	Derivative-free Optimization.
DGP	Data Generating Process.
DN	Dominant Negative.
FBA	Flux Balance Analysis.
FDM	Finite Difference Method.
FEM	Finite Element Method.
FIM	Fisher Information Matrix.
(f)MRI	(functional) Magnetic Resonance Imaging.
FPR	False Positive Rate. Also known as type I error rate.
FRAP	Fluorescence Recovery After Photobleaching.
FVM	Finite Volume Method. Method for solving PDEs.
GLUT4	Glucose transporter 4.
GOF	Goodness Of Fit.
GWAS	Genome-wide Association Studies.
ICU	Intensive Care Unit in a hospital.
IR	Insulin receptor.
IRS1	Insulin receptor substrate-1.
LHS	Latin Hypercube Sampling.
LGN	Lateral Geniculate Nucleus.
MCMM	Markov Chain Monte Carlo.
MD	Microdialysis.
mTOR	Mammalian target of rapamycin.
ODE	Ordinary Differential Equation.
OED	Optimal Experimental Design.
PCA	Principal Component Analysis.
PKB	Protein Kinase B. Also known as Akt.
PKPD	Pharmacokinetic and Pharmacodynamic modeling.
PNS	Peripheral Nervous System.
PPI	Protein-Protein Interaction.
ROC	Receiver Operator Characteristic. Measure of FPR vs TPR.
S6	A ribosomal protein.
S6K	S6-kinase.
SBTB	Systems Biology Toolbox.
SEM	Standard Error of the Mean.
syt7	synaptotagmin 7.
T2D	Type II diabetes.
TPR	True Positive Rate.
WT	Wild-type.
YFP	Yellow Fluorescent Protein.

MATHEMATICAL SYMBOLS

α	Significance level.
χ^2	Chi-square.
ε	Measurement error.
$C(\theta)$	Confidence interval for the parameter estimation of θ .
DW/dw	Durbin-Watson test.
$\mathcal{F}_f$	A generic probability distribution f .
$\mathcal{F}_f^{cdf}$	Cumulative density distribution function for f .
$\mathcal{F}_f^{cdf-inv}$	Inverse of a generic cumulative density distribution function f .
H	The Hessian (second derivative) of the cost function.
$\mathcal{H}$	An Hypothesis.
$\mathcal{H}_0$	The null hypothesis.
$\mathcal{I}$	Number of identifiable parameters.
$\mathcal{M}$	A model structure.
$\mathcal{M}(\theta)$	A specific model: A model structure with parameters.
v	Degrees of freedom.
p	Probability – p-value.
$r(t)$	Model residuals.
$rand$	A randomly generated number from some distribution.
σ	Standard deviation, or sample standard deviation.
T_f	Test statistic of a generic probability distribution f .
T_f^0	Threshold for rejection of generic probability distribution f .
θ	A parameter or parameter vector.
$\hat{\theta}$	Parameter estimate of the global minimum.
$t_{\mathcal{M}}$	Transcendence degree.
U	The uniform distribution.
u	Model input.
$V(\theta)$	The cost/objective function.
∇V	The gradient (first derivative) of the cost function.
$V^{PPL}(\theta)$	Extended cost-function for prediction profile likelihood.
w	numerical weight.
x	Model states.
$\dot{x}$	State derivatives in an ODE model.
y	Data.
$\hat{y}$	Model output.
$Z(\theta)$	A quantifiable model property / a core prediction.

TABLE OF CONTENTS

1	Introduction	1
1.1	Complexity	1
1.2	The Book of Life: from DNA to protein	3
1.3	Omics	5
1.4	Personalized medicine	7
1.5	Systems biology	7
1.6	Aim and scope	8
1.7	Outline of thesis	8
2	Science Through Hypothesis Testing.....	9
2.1	Facts, hypotheses, and theories	9
2.2	Verifications and falsifications.....	10
3	Mathematical Modeling	13
3.1	Modelling definitions and concepts	14
3.1.1	Model properties	15
3.1.2	Modeling frameworks	16
3.2	Ordinary differential equations	17
3.3	Black box modeling and regression models.....	19
3.4	Networks and data-driven modeling	21
3.5	Partial differential equations	23
3.6	Stochastic modeling	24
4	ODE Modeling Methods	29
4.1	The minimal model and modeling cycle approach	30
4.2	Model construction.....	32
4.2.1	Hypothesis and data	32
4.2.2	Scope and simplifications.....	32
4.2.3	Reaction kinetics and measurement equations.....	34
4.2.4	Units	34
4.3	Model simulation.....	34
4.3.1	Runge-Kutta, forward Euler, and tolerance	35
4.3.2	Adams–Bashforth.....	36
4.3.3	Adams–Moulton.....	37
4.3.4	Backward Differentiation Formulas.....	37
4.3.5	On Stiffness and software	38
4.4	Parameter estimation and goodness of fit	38

4.4.1	Objective function	39
4.4.2	Cost landscape	41
4.4.3	Local optimization	42
	Steepest descent, Newton, and quasi-Newton	43
	Nelder-Mead downhill simplex	44
4.4.4	Global Optimization	46
	Multi-start optimization	47
	Simulated annealing	47
	Evolutionary algorithms	49
	Particle swarm optimization	50
4.5	Statistical assessment of goodness of fit	51
4.5.1	The χ^2 -test	52
4.5.2	Whiteness, run, and Durbin-Watson test	53
4.5.3	Interpretation of rejections	54
4.6	Uncertainty analysis	54
4.6.1	Model uncertainty	54
4.6.2	Parameter uncertainty	55
	Sensitivity analysis	55
	Fisher information matrix	55
	Identifiability and the profile likelihood	56
4.6.3	Prediction uncertainty	57
4.7	Testing predictions	58
4.7.1	Core predictions	58
4.7.2	Validation data	58
4.7.3	Overfitting	59
4.8	Model selection	60
4.8.1	Experimental design and testing	61
4.8.2	Ranking methods and tests	61
	Information criterion	61
	The likelihood ratio test	62
4.9	Bootstrapping and empirical distributions	63
5	Model Systems	65
5.1	Insulin signaling system in human adipocytes	65
5.2	Cell-to-cell variability in yeast	67
5.3	Facilitation in murine nerve cells	69

6	Results	73
6.1	Modeling of dominant negative inhibition data	73
6.2	Quantification of nuclear transport rates in yeast cells	75
6.3	Quantitative modeling of facilitation in pyramidal neurons	78
6.4	A novel method for hypothesis testing using bootstrapping	80
7	Concluding Remarks	85
7.1	Summary of results and conclusions	85
7.1.1	DN data should be analyzed using mathematical modeling	85
7.1.2	A single-cell modeling method for quantification of nuclear transport	85
7.1.3	Facilitation can be explained by a single mechanism	85
7.1.4	A novel 2D bootstrap approach for hypothesis testing	86
7.2	Relevancy of mathematical modeling	86
7.2.1	Hypothesis testing	87
7.2.2	Mechanistic understanding	87
7.2.3	Design of experiments	87
7.2.4	Data analysis	88
7.2.5	Healthcare	88
7.3	Outlook	88
	Acknowledgements	91
	References	93
	Endnotes	102

1

INTRODUCTION

On July 4th, 2012, CERN announced that they have discovered the *Higgs boson*. Although final verification is still pending, this was the culmination of over forty years of searching for the final piece of the puzzle in the so called *standard model of particle physics* [1]. We do not have a *standard model* of biology. In fact, we are nowhere close. While no one would argue that physics is easy, it is from a certain perspective *simple*. That is, we are working with very basic building block and fundamental forces of nature, with corresponding universal constants that can be determined once and for all. In biology things are messier. Species, organisms, tissues, organs, and cells, are all in a state of constant flux, ever changing and with a lack of finality to any discovery made. How things are now, could always change in the future as evolution carries on its irrevocable march. The goal of the biologist and life scientist is therefore to find the common patterns and principles in life, rather than uncovering any underlying rule work or structure. In this chapter, I will go through some of the reasons why life science is so hard, and what approaches that have been tried historically and are being brought to bear on the topic right now.

1.1 Complexity

*“Those are some of the things that molecules do,
given four billion years of evolution.”*

Carl Sagan. Cosmos, 1980

Inside all living cells is a microcosm of weird structures, strange processes, and a seemingly never-ending chain of new peculiar phenomena. Here, at the molecular level, lies one of the frontiers of modern biology and medicine. The overarching goal is to understand how these molecules and structures work inside cells, and how cells work together to form living tissues, organs, and whole organisms. However, this task of unraveling the inner workings of life is hampered for at least two reasons. Firstly, many of the processes one wants to study are not directly accessible, or cannot be seen by the naked eye. Instead, scientists query the system with the aid of instruments and indirect measurements. Secondly, the number of different structures, molecules, interactions, and regulatory processes inside a cell are enormous. Figure 1 (top) shows a graphical summary of most major metabolic pathways, with a zoomed in version of the citric acid cycle in the bottom part. As you can see, the sheer number of reactions, substrates, and intermediaries is staggering. You should then also note that this picture includes almost no information on signaling, and how these reactions and the enzymes catalyzing them are regulated, nor how they fit into the bigger picture of the spatial or temporal structure of the cell.

Figure 1. Top: Metro-style map of major metabolic pathwaysⁱ. Single lines: pathways common to most lifeforms. Double lines: pathways not in humans (occurs in e.g. plants, fungi, prokaryotes). Orange nodes: carbohydrate metabolism. Violet nodes: photosynthesis. Red nodes: cellular respiration. Pink nodes: cell signaling. Blue nodes: amino acid metabolism. Grey nodes: vitamin and cofactor metabolism. Brown nodes: nucleotide and protein metabolism. Green nodes: lipid metabolism. Bottom: Zoomed in detailed map of the citric acid cycleⁱⁱ.

The picture in Figure 1 is the result of decades of painstakingly slow and gradual scientific research, where the topic of investigation often has been only a small set of reactions, or just a single reaction at a time [2]. However, since the early 1990s there has been an effort to move into large-scale, high-throughput data acquisition methods. This move can be seen as an effort to try to combat the problem of the complexity presented above: by simply collecting more and better data on all players involved. One of the earliest and most ambitious large-scale projects in this fashion was the Human Genome Project, which had as its objective to sequence the entire human genome, and identify all its genes and coding regions [3]. Genes code for proteins, and proteins determine the makeup of all cells, and in extension, all tissues, organs, and the whole body. The rationale behind the Human Genome Project was then that understanding the genome would be an important first step in solving the biological puzzle. We will talk more about genes, proteins, and the Human Genome Project in the next section.

1.2 The Book of Life: from DNA to protein

“We are here to celebrate the completion of the first survey of the entire human genome. Without a doubt, this is the most important, most wondrous map ever produced by humankind.”

Bill Clinton, June 2000

The quote above were uttered by the then president of the United States of America, Bill Clinton, during a press release in June 2000. The press release concerned the near completion of the Human Genome Project [3]. The media reported that *The Book of Life* had been found [3]. The Human Genome Project is of course one of the most important scientific collaborations of all time, and its accomplishments should not be understated. However, the metaphors used by both president Clinton - a map - and by the media - *The Book of Life* - were misleading because they did not accurately depict the role of genes and proteins in the human body.

To understand why we in fact did not have *The Book of Life*, we first need to understand what a gene and a genome is. In simple terms, the genome is the collection of all genes in an organism. A gene is a region of the DNA that codes for the production of a particular protein. Proteins are large biomolecules that perform various tasks inside your cells. Some of the functions performed by proteins are: to serve as enzymes, catalyzing the breakdown and production of different compounds; as transport molecules; as sensors to various external or internal stimuli; as specific scaffolding structures within the cell. Proteins are built up of chains of amino acids in a sequential manner that fold into a particular three-dimensional configuration and structure. To a first approximation, the sequence of the amino acid chain that builds up the final protein is uniquely determined by its gene (Figure 2). However, knowing the sequence still leaves us with two major problems. The first problem is that this sequence is not enough information to determine the final three-dimensional structure of the protein, or its immediate function. The second problem is that we also do not know where it will be located in either space or time, and therefore cannot determine its wider biological role.

Concerning the first problem, that of determining protein structure and function, it turns out that predicting the three-dimensional structure, also known as folding, of a protein is a difficult problem. Within the field of *bioinformatics*, there is a whole sub-field dedicated to this problem, using various tools such as direct modeling of the protein molecule and that energy states of various folding options, to inferring parts of the structure due to known motifs, or related proteins where the structure is known. Knowing the structure of a protein is sometimes enough to deduce its function, or at least say something about what kind of role it plays. More often however, we also need to know where and when a protein is expressed, and which other proteins and molecules it interacts with.

Figure 2. Information flow From DNA to proteinⁱⁱⁱ. A gene in the DNA is transcribed to create mRNA. mRNA in turn is used as a template for protein synthesis, a process called translation. The protein consists of a chain of amino acids which is determined by the stored information in the gene. However, the final three-dimensional structure, and ultimately protein function, cannot be determined from the known sequence alone.

This brings us to the second major problem. Proteins can be active at different locations within the cell, such as inside the nucleus, in the cytosol, in the cell plasma membrane, inside specific organelles and so on. Different cells, such as cells from different tissues, can turn genes on or off, and therefore regulate to which degree a certain protein should be present or active within a cell, if at all. In other words, some proteins can only be found in specific cell-types, in specific parts of your body. Likewise, some proteins have a temporal aspect to their expression. Some proteins are only produced during specific stages of life, for example during embryonic development or puberty. Other proteins vary with specific cycles, such as the cell cycle, the changing of the seasons, or night and day. Alternatively, a protein can be expressed as a triggered response to a specific environmental input.

In summary, we can from a specific gene sequence determine the amino acid makeup of a protein, but we cannot directly determine what the final structure of a protein will be, what the function of it will be, or where and when it will be expressed. In other words, the analogy of *The Book of Life* seems to somewhat halter. Having the genome sequence is rather like a chef having a list of ingredients but not having the recipe for producing the final meal. Additionally, the chef would not know if the meal in question were a starter, a main course, or a dessert, or how many people it is intended for. Expecting the chef to somehow recreate the original meal from only the ingredients would therefore be unrealistic even under the best of circumstances.

1.3 Omics

The Human Genome Project was in many ways the start of what is now called the *omics* era. The suffix *ome* is used for indicating the whole collection of something in a particular category. Even though the word *genome*, i.e. the collection of all genes, have been used since 1920 [4], the word *genomics* however, which means the study of the genome, was coined as recently as 1986 [5]. Soon after, more omics were to follow. Since the introduction of genomics, the concept of *omics* has been applied to distinguish several different topics of study such as *proteomics*, *metabolomics*, *transcriptomics*, and so on, often with additional sub-omics for each field, such as *phosphoproteomics* etc. Table 1 lists some commonly encountered *omics*.

Figure 3. A schizophrenia interactome network^{iv}. All nodes are genes, and all edges (connecting lines) in the network are protein-protein interactions (PPI). Dark-blue nodes are known schizophrenia associated genes, light-blue nodes are known interactors, and red nodes are novel interactors found by Ganapathiraju et. al. [6].

The common trends for all these omics are that they are very holistic in their nature, and typically revolve around methods that are high-throughput and result in large quantities of data. Furthermore, the research questions are often data-driven, rather than hypothesis driven.

In genomics and proteomics, for example, research is often centered around constructing large networks of genes and/or proteins, and possibly their means of interaction. The goal might for example be to find specific sub-clusters and hubs that are important for a particular disease or condition, such as allergy, metabolism, cancer etc. Figure 3 shows an example regarding schizophrenia [6]. Here the researchers studied the *interactome* of genes and proteins associated with schizophrenia. By constructing this network, they could identify 504 novel genes that could be involved in the disease.

Noun	Set	Field of study
Gene	Genome	Genomics
RNA Transcript	Transcriptome	Transcriptomics
Protein	Proteome	Proteomics
Lipid	Lipidome	Lipidomics
Metabolite	Metabolome	Metabolomics

Table 1. A list of some commonly encountered omics.

The demand for large-scale and high-throughput data acquisition methods has also led to a significant cost reduction in acquiring said information. For example, the cost of sequencing an individual genome has gone down from roughly a hundred million dollars, in 2001, to \$1000 dollars in 2016 (Figure 4). This means that in the near future, fully sequencing your genome might be a part of standard clinical treatment, provided that we can use the information to help inform treatment and decisions.

Figure 4. DNA sequencing costs^v. The cost of sequencing a human genome has decreased rapidly by several orders of magnitudes during recent years (circles, solid line). This reduction can be compared to e.g. Moore's law, where the price halves every two years (dashed line). The cost for sequencing a full genome is estimated to have dropped below \$1000 during 2016, and can therefore be expected to be part of regular treatment in health-care in the near future.

1.4 Personalized medicine

Health care has always tailored treatment to the individual to some degree, such as taking into account of a patient's gender, age, weight, family disease history, and so on. With recent advancements in data acquisition, the topic of tailoring treatment towards the individual, called personalized medicine, is more relevant than ever [7,8]. For instance, Genome-wide Association Studies (GWAS) have been able to link specific genetic variants with increased propensity for varying diseases, and knowing whether an individual has the particular genetic variant or not would then affect the treatment of the patient. With sequencing costs for a full genome predicted to be less than \$1000 in the near future (Figure 4) such sequencing is likely to become standard procedure for many diagnostic and treatment applications.

Apart from the increased availability of genetic data, clinical facilities in general have an increasing capability of producing more individual data from each patient. Patients in the Intensive Care Unit (ICU) for example, are monitored closely. This monitoring includes heart rate and breathing, but also more invasive procedures such as taking blood samples to check for levels of various metabolites, the most important one being blood glucose. Metabolites are sometimes also monitored using less invasive techniques such as glucose sensors, or microdialysis (MD) methods. For neurological disorders, the use of functional Magnetic Resonance Imaging (fMRI) is used to determine which part of a patient's brain that is functional, damaged, or behaving abnormally. MRI is also used together with various contrast agents to study various bodily functions, such as liver function [9].

This new abundance of individual patient data also come with new challenges. The sheer amount of data can be overwhelming, and it will be increasingly hard for any individual physician to make use of such data. What is needed are sound and reliable methods for analyzing and combining multiple data-sources, and to weigh their uncertainty together to make an informed decision. This challenge of interpreting data can be approached from several directions. Bioinformatics, classical statistics, and Bayesian inference methods look at correlations between different factors and biomarkers. Another approach is the use of mechanistic mathematical modeling and systems biology.

1.5 Systems biology

*“The beauty of a living thing is not the atoms that go into it,
but the way those atoms are put together.”*

Carl Sagan. Cosmos, 1980

Systems biology emerged as a new inter-disciplinary field in the early 2000s [10]. Instead of the classical reductionist approach of focusing on each component in isolation, common for instance in molecular biology, systems biology focused on a holistic understanding of the biological systems, and on how emergent properties arise when putting individual components together. The desire to understand the system as a whole has two important components which

both are central to systems biology: a) systems-wide data acquisition, and b) mathematical modeling.

The first component, systems-wide data acquisition, comes from the fact that for a system-level understanding to be obtained, many things from the intact system need to be measured in parallel. Such data, are gathered using techniques from some of the earlier mentioned *omics* fields, such as metabolomics, genomics, proteomics, etc. As already mentioned, these fields generate large amounts of data, with new challenges for data interpretation and storage.

The second component, and a central aspect to systems biology studies, is mathematical modeling. Mathematical modeling is not only well-suited for handling complex data sets, but also for the analysis of complex explanations. Since systems biology attempts to understand intact systems, the studied systems will typically correspond to networks of interconnected components, and this means that the various corresponding explanations will be complex as well. These explanations will almost always be too complex to be grasped using classical reasoning, but may sometimes still lie within the reach of mathematical modeling. This is known as *mechanistic modeling*, where both the components of the model, and their emergent behavior can be understood in mechanistic, i.e. biological, terms. The current limited use of mathematical modeling in biology is partly due to historical differences in training and terminology, as well as in a lack of awareness of showcases illustrating how modeling can make a difference, or even be required, for a correct analysis of experimental data.

1.6 Aim and scope

The aim of this thesis was to explore the possibilities and limitations of mathematical modeling, with a focus on Ordinary Differential Equations (ODEs), pertaining to its application in biology and medicine. In particular, I wanted to investigate the ability of mechanistic modeling as a tool for hypothesis testing, data analysis and experimental design, design new methods for the applicability of mechanistic modeling, and finally to apply these methods on relevant systems from biology and medicine.

1.7 Outline of thesis

In this thesis, using examples from my own publications, I will show how systems biology approaches can be used to answer biological and medical questions. To understand this in its correct framework, I will first briefly in Chapter 2 discuss the nature of science, scientific hypotheses, and the growth of scientific knowledge. In Chapter 3, I will discuss mathematical modeling and make a brief recount of various neighboring fields of modeling and then, in Chapter 4, describe in more detail the methods I have primarily used during my thesis. In Chapter 5, we will look at the biological systems on which my results are based. In Chapter 6, I will summarize my results and publications and how they fit into this overall picture, and finally in Chapter 7, draw some concluding remarks and some possible outlooks for the future.

2

SCIENCE THROUGH HYPOTHESIS TESTING

Arguably, what we today call modern science started with the so called *scientific revolution*, recognized for amongst other things overthrowing the thousand year old paradigm of Aristotelian physics [11]. This overthrow started with the presentation by Copernicus of the heliocentric model of the solar system^{vi}, gained momentum and support through the works of renown scientists such as Galileo and Kepler, and finally, roughly one and half century later, ended with Newton's publication of *Principia*^{vii}.

What exactly is this scientific method that was supposedly used to change our understanding of the solar system and the universe from the stationary geocentric world view put forth by Aristotle and Ptolemy, to the dynamic heliocentric world view proposed by Copernicus? Science is often said to be a cumulative and error correcting process. So how did we replace the Aristotelian understanding of motion with Newton's laws? How were Newton's law themselves later on succeeded by Einstein's theory of relativity? What does it mean that something is scientific? One answer was given by the philosopher of science *Karl Popper* in the early 1930s. His answer was: *Verifications and falsifications*. These two concepts are crucial to understanding *hypothesis testing*. In the following sections I will briefly discuss the road up to, and beyond Karl Popper, from the viewpoint of the philosophy of science.

2.1 Facts, hypotheses, and theories

It is said that science is derived from the facts, but what is a fact, and how is this derivation accomplished? A simplistic definition would be that a fact is just an observation about the world from which we construct scientific hypotheses and theories (Figure 5) [11].

Figure 5. A simplistic interpretation of how science could be derived from the facts.

However, a fact is much more than this. For example, an observation usually entails the use of some measuring device, or visualization tool. An underlying assumption about the observation being made is that the tool and instruments being used are functional, and adequate for the task. Often the use of the instruments entail assuming auxiliary assumptions and some theory about the instrument itself. For example any use of a microscope or telescope relies heavily on the

laws of optics and so on [11]. Establishing a fact from an observation is also knowledge dependent. For instance, if you wished to catalogue the various kinds of species that inhabited a forest, a botanist would be able to assess the facts about which species that lived in the forest to a much higher degree than a mere layman [11]. The superior assessment by the botanist is of course because the botanist has the relevant background knowledge for making useful observations (Figure 6).

Figure 6. Inductivism. A more realistic view of how knowledge is derived from facts and observations. Observations depends on background knowledge and auxiliary hypotheses, such as how measurement devices work and so on. New knowledge gained through this process then become part of a new and growing background base of knowledge.

Background knowledge, as exemplified by the botanist above, usually entails other established facts, but also a larger framework of theories and hypotheses. A hypothesis is a proposed explanation for how a particular phenomenon can be explained. Hypotheses can vary in both scope, detail, and explanatory power. A hypothesis, or set of hypotheses that has withstood rigorous testing and been accepted by the scientific community at large, is considered a theory. Finally, this newly gained information feeds back into the commonly available background information and is the basis for future scientific work (Figure 6).

2.2 Verifications and falsifications

In the early 20th century, the main school of thought regarding scientific knowledge was the so called *logical positivism* championed by the group of philosophers known as the

Vienna Circle [11]. The main idea was that science was derived from the facts much as depicted in Figure 6 using induction. This school of thinking was challenged by the philosopher Karl Popper in 1932 [12].

Popper noticed several problems especially in the social sciences where certain theories were so flexible that they could accommodate any new finding. Popper contrasted this with the physical sciences such as the testing of general relativity by Eddington in 1919 [13]. In this famous testing of general relativity there was a prediction that light would bend by a certain degree when passing close to a strong gravitational source, something that could be observed and measured during an eclipse. This prediction turned out to be right, but Popper noticed that it equally well might have been wrong. If the latter had been the case, the theory would have to have been discarded [11]. This lead Popper to propose the criteria of *falsifiability*.

Figure 7. Falsificationism. Science progresses by the testing of hypotheses. A prediction is made from the relevant background information and the hypothesis under testing. This prediction is then tested experimentally, and the outcome either supports the hypothesis, in which case we do not reject it, or does not support the hypothesis, in which case we reject it. If we do not reject the hypothesis, we can try to come up with new experiments to test the hypothesis. Only if the hypothesis has survived many such iterations do we tentatively accept it. Conversely, if we reject the hypothesis, we can see if a modification of the hypothesis can account for the discrepancy between prediction and observation, in which case we start the whole cycle over again.

According to Popper, something could be said to be scientific only if it could be falsified, i.e. be proven wrong, and the more ways it could be tested in and the more specific the predictions were the better the hypothesis was. Furthermore, Popper argued that the whole inductivist school of thought was fundamentally flawed. How do you go from a finite set of observations to make statements of the whole? In philosophy, this is known as *the problem of induction*. Popper's response to the problem of induction was to side-step the issue and declare that in the strictest sense, science can never prove anything, it can only disprove, i.e. falsify theories and

hypotheses. Popper argued that science progresses by the growing set of discarded hypotheses, and those who survives are only tentatively held as true (Figure 7). More specifically Popper argued that science advanced with the falsification of trusted hypothesis, and the verifications of bold new conjectures [11].

During the second half of the 20th century the scientific method was viewed through yet a new lens, focusing on the social aspect of science. Thomas Kuhn argued in his work *The Structure of Scientific Revolutions* that both the inductivists and the falsificationists failed to describe the known history of science [14]. Another of the main criticism of falsification was that when an observation is made, and if it conflicts with the hypothesis under testing, it is impossible to separate the testing of the hypothesis from all the auxiliary hypotheses regarding measurement devices and so on (Figure 6). That is, in practice one could always keep the hypothesis under testing under the assumption that something must have been wrong with the experimental setup. Kuhn introduced the concept of groups of scientists working within a specific *paradigm*. According to Kuhn, science advanced when an old paradigm was replaced by a new one. This revolution, as he called it, would happen after the old paradigm had accumulated a specific number of anomalies such as to no longer function coherently. For example, the Aristotelean school of thought would be an old paradigm, which was replaced by the newer Galilean-Newton paradigm.

The critique of the failure of falsificationism to account for the societal aspects of science led to a battle that is still raging within the philosophy of science. For instance, it led the philosopher Paul Feyerabend in 1973 to propose the *Anarchist View of Science* [11,15]. Feyerabend argued in *Against Method* that if there was any rule in science, it was the rule that there are no rules: i.e. anything goes.

Popper refined his views on falsificationism in later works. Whether he was successful enough to salvage his view on the scientific method or not, Popper has left a lasting impression in the scientific tradition. The criterion of *falsifiability* and the concept of *hypothesis testing* has stuck within all the natural sciences as a trademark of good scientific practice. In the next chapter, we will look at the tool known as mathematical modeling, and how this tool is especially appropriate for making predictions that can lead to rejections or tentative verifications.

3

MATHEMATICAL MODELING

“The most incomprehensible thing about the universe is that it is comprehensible.”

Albert Einstein, 1936 [16]

In his 1960 paper *The unreasonable effectiveness of mathematics in the natural sciences* the physicist Eugene Wigner described the peculiar fact that the language of mathematics was so widely suitable for use in the natural sciences [17]. There are two main categories of explanations for this phenomenon. The first category of explanations is that mathematics only seems to be particularly useful and successful because it is the toolkit we are bringing to bear on the problem. This is known as the *law of the instrument* or *law of the hammer* which can be summarized as: “If all you have is a hammer, every problem looks like a nail” [18,19]. The second category of explanations say that mathematics is successful at describing natural phenomenon precisely because reality is inherently mathematical, often referred to as *The mathematical universe hypothesis* [20,21]. One could argue that this second view, that reality is inherently mathematical, has been the dominant view historically. Towards demonstrating this, I will give a couple of examples to prove my point.

We know that many ancient civilizations used mathematics to study the heavens, and to predict the motions of planets and the turning of the seasons. In this tradition, around 240 B.C, the Greek polymath Eratosthenes calculated the circumference of the earth by comparing the angle of shadows cast at noon in Alexandria and Syene [22]. Eratosthenes assumed that the earth was spherical, that light rays hit the earth from optical infinity, then used measurements of the distance between Alexandria and Syene to predict within a 10% margin of error the circumference of the earth.

My second example is from almost 2000 years later. In the early 18th century, the English astronomer Edmond Halley discovered that several historical sightings of comets could be explained with the reoccurrence of a single comet returning with a 75-year periodicity. Following the publication of Newton’s laws of gravity and motion, Halley could in 1705 calculate the orbit for what is now known as Halley’s comet and predict its return in 1758. Halley died before his comet made its return in December of 1758. With its return, however, the comet verified for the first time that things other than planets could orbit the sun, and was one of the first tests of Newtonian physics [23].

My third and final historical example of using mathematics to make a prediction was mentioned already in Section 2.2, namely the deflection of light by the sun. Newtonian gravity predicted that the sun should bend light from distant stars by a certain degree, and later Einstein calculated

that under general relativity, the degree of light bending should be twice as much. This led to the aforementioned measurement by Eddington in 1919, which measured the deflection during a solar eclipse, and found that it confirmed the prediction under general relativity, and that Newtonian gravity could be rejected [13].

The common denominator for the above examples is that in all the cases, some underlying structure about the world has been assumed, i.e. a model, and some property of this structure has been calculated in order to make a prediction. We will now in more detail examine what constitutes as a mathematical model and what kind of models that are used within the field of systems biology.

3.1 Modelling definitions and concepts

“Essentially, all models are wrong, but some are useful.”

George Box, 1987 [24]

In the introductory section to this chapter, I talked about the use of mathematics in the natural sciences, and gave historical examples of modeling. However, perhaps the most pertinent question remains: What, exactly, is a model? A model is a representation, often abstract, of an object or a process, much in the same way as a map is representation of the world. A model is almost per definition a simplification of reality. This simplification is in and of itself neither a bad nor a good thing, but it is relevant for the *scope* of the model: i.e. the regime the model is designed to work in. A model could be a graphical representation of an interaction network, which qualitatively describes relations between its various components, or it can be of a more quantitative nature, with a specific set of equations governing its behavior. For example, if we look again at the interaction graph in Figure 1 (p. 2), this is a graphical model of the underlying biological pathways. If we provide details about the amounts of enzymes, reactants and coefficients for the reactions, we would also have a quantitative model which we could simulate and use to predict how the system would respond to perturbations or changes in its initial conditions.

In *Systems biology: a textbook*, the authors give 10 reasons and advantages for using modeling in biology [25]. Below I have summarized their reasons condensed to one-sentence arguments:

1. Conceptual clarification by writing down verbal relations as rigorous equations
2. Highlighting gaps in knowledge
3. Independence of/from the modeled object
4. Zooming in time and space at your own discretion
5. Algorithms and computer programs can be used independently of the system
6. Modeling is cheap compared to experiments
7. Ethical reasons: models do not harm the environment, animals, or plants.
8. Replacing and/or assisting experimentation.
9. Precise formulation leads to easy generalization and visualization.
10. Enabling well-founded and testable predictions.

From the viewpoint of using mathematical modeling as a tool for hypothesis testing, number ten is of particular interest to us, since testable predictions lie at the core of hypothesis testing.

3.1.1 Model properties

Models exist in various forms and frameworks, depending on their usage and domain of applicability. Table 2 lists various properties a model could have, along with their opposite attribute. These attributes are either applied to a model as a whole, or to parts of the model. Let us quickly go through their meaning.

Model aspect	Opposite aspect
Qualitative	Quantitative
Linear	Nonlinear
Static	Dynamic
Explicit	Implicit
Discrete	Continuous
Deterministic	Stochastic
Black Box	Mechanistic

Table 2. Pairs of opposing model attributes.

A *qualitative* model is a model that lacks *quantitative* precision. This might be that the model uses *fuzzy* categorical descriptions [26], or that it reduces data and model components to only binary or discrete values [27]. However, it might also be that the model depicts components in arbitrary units and/or only work with relative data, or data with an unknown scaling factor. Ultimately, the *qualitative* and *quantitative* epithets are not a strict binary. Rather, models and different modeling frameworks all exist along a spectrum with varying degrees of *quantitativeness*. The properties of *linear* and *nonlinear* covers a wide array of models and vary from topic to topic, and therefore has slightly different definitions depending on the context. In the case of Ordinary Differential Equations (ODEs), which will be the main type of models discussed in this thesis, it means that the state derivatives can be written as a linear combination of all the other states. We will define ODE model in Section 3.2, and then spend the majority of Chapter 4 on details about working in an ODE framework. Continuing with the attributes in Table 2, a model is said to be *static* if there is no explicit time in the model. A *dynamic* model, on the other hand, explicitly models a time evolution of some system. Some models have a dependency of the history of the system, whereas the evolution of other models can be described completely from what is known about the state of the model in the present. *Static* models are often *explicit*. This means that given the input, the output can immediately be calculated. *Dynamic* models on the other hand are usually *implicit*, which means that their output cannot immediately be calculated but needs to be iteratively solved by some step-method. A model is said to be *deterministic*, if for a given input, the model always returns the same output. Conversely, a *stochastic* model has some random process in it that results in different output each time the model output is simulated. Finally, a model is said to be *black box* if it is constructed in such a way that the functions and equations that goes from the input to the output lack any interpretability of the physical system modeled. These models are typically used when one only wishes to have a model with great predictive ability, but where an understanding of the actual system is of no interest. In a *mechanistic* model, on the other hand, some steps, if not all, from input to output have some degree of interpretability in terms of the natural

phenomenon the model describes. This can for example be reactions inside a cell, such as phosphorylation events. In these kinds of models, the mechanistic processes are often of as much importance as the model's predictive capability.

3.1.2 Modeling frameworks

The remainder of this chapter will be dedicated to putting into perspective different modeling frameworks within, or bordering, the field of system biology. These different frameworks range in scope and complexity and have various strengths and drawbacks. We will start by introducing the ODE framework, and then proceed to discuss other alternatives. We start our exploration with ODE models for two related, but albeit distinct, reasons. The first and major reason is because this is the framework and methods I have used and deployed in my research projects (see attached papers). The second reason, and arguably the major contributor for my choice in working with ODE models, is that ODE models are the most frequent type of modeling approach used within systems biology [28,29]. ODE models are therefore a natural starting point for discussing other alternatives.

Figure 8. Prevalence of different modeling frameworks. Graph shows the number of publications describing systems biology as applied to biochemistry in the years 2000–2010 using a specific computational modeling approach. Originally published as Figure 7 in [29]. Included with the permission of the authors.

For a comparison in methodology usage, Figure 8 shows an analysis of papers published between the years 2000-2010 of systems biology models in biochemistry. In first place we have ODE models, which dominate the field with more than 65% of all publications. In second place comes stoichiometric modeling, which is synonymous to flux balance analysis, which will be briefly discussed in Section 3.4. The third and fourth position is taken by PDE models and stochastic models respectively. These will be covered in Sections 3.5 and 3.6. I will discuss logic models, the fifth position, also in Section 3.4. Petri nets, the sixth position, is a type of network modeling, and the last position, hybrid, is where researchers have used combinations of different modeling approaches. Such combinations have various advantages and disadvantages. Neither petri nets nor hybrid approaches will be discussed any further here.

3.2 Ordinary differential equations

Models based on Ordinary Differential Equations (ODEs) are a type of *implicit* models (Table 2) which are used frequently within biology, and especially within systems biology [29]. ODEs are implicit because rather than describing the components of a model directly, an ODE describes the *rate of change* of the component. This rate of change is most often with respect to time, which makes ODE models very suitable for describing *dynamic* phenomena. Concordantly, ODEs have a long and successful history of being used within biology. As early as 1837 the French mathematician *Pierre Franois Verhulst* used ordinary differential equations to describe population growth under limited resources, also known as the logistic growth model [30]. In 1952, *Hodgkin* and *Huxley*, developed their world-famous model for the initiation and transmission of the action potential in neurons, based on a series of non-linear differential equations, which later resulted in a Nobel Prize in Physiology or Medicine in 1963 [31]. The Hodgkin-Huxley paper can in hindsight arguably be considered one of the first applications of system biology.

The major components of an ODE model are called states and are usually denoted by the letter x . The state derivatives with respect to time, dx/dt , are usually shortened to $\dot{x}$ and are governed by a smooth non-linear function f , which takes as input the state values, the model parameters θ , and the input to the model, u .

$$\dot{x} = f(x, \theta, u) \quad (1)$$

A state could for example correspond to the amount, or the concentration, of a compound or molecule, such as a metabolite or a signaling protein. The function f is usually obtained by summing up the kinetic rate expressions of all the reactions of the involved compounds. These rate expressions are formulated from basic chemistry principles such as the law of mass action, Michaelis-Menten kinetics, and Hill kinetics [32].

An ODE model is solved numerically using a specific step-function that takes a state vector as a starting point. This starting vector is known as the initial conditions of the model, and is denoted x_0 . Usually x_0 is grouped together with the model parameters θ . The output of the model, denoted $\hat{y}$, is given by another smooth non-linear function g .

$$\hat{y} = g(x, \theta, u) \quad (2)$$

The specific data the model is intended to fit is denoted by $y(t)$. If one assumes only additive measurement error, we obtain

$$y(t) = \hat{y}(t, \theta) + \varepsilon(t), \quad \varepsilon \sim \mathcal{F}_f \quad (3)$$

where ε is the measurement noise which follows some distribution $\mathcal{F}_f$. The most frequently encountered variants of the distribution $\mathcal{F}_f$ within systems biology are normal or log-normal distributions [33]. An ODE model, $\mathcal{M}(\theta)$, is defined by the model structure, $\mathcal{M}$, and the parameters θ , where $\mathcal{M}$ is given by the specific choices of the functions f and g , and where the values of θ is chosen from literature or fitted to the model using optimization techniques.

Figure 9. $\mathcal{M}_{i,b}$ - A simple ODE model of insulin binding to the insulin receptor, with subsequent receptor phosphorylation, internalization, and recycling [34,35]. (A) Interaction graph showing how the insulin receptor transitions between different states when insulin is present. (B) Western blot data (*, error bars) of the insulin receptor phosphorylation levels (a.u.) after stimulation with insulin (the model input). The $\mathcal{M}_{i,b}$ model was fitted to this data to provide as good an agreement between simulation (solid line) and data as possible. (C) A reaction list description of the model. (D) A full state-space description of the model. Notably, by design insulin is here considered to be present in such excess that any dynamics in insulin due to internalization can be ignored. Insulin is therefore considered to be constant in the model, and does not need to be explicitly modeled like the receptor states do. To simulate this model, one also needs to specify numeric values for the θ s, including the initial conditions.

Equations (1) - (3) are called the state space description of an ODE model, and with specified values for the model parameters, including initial conditions, they give a full description of the model. ODE models are often visualized using an interaction graph. The main purpose of an interaction graph is to serve as a visual aid for the ingoing components, reactions, and interactions of the ODE model. An interaction graph of an ODE network usually has states as the nodes, and fluxes or rate constants along the edges of the network, and where the edges themselves are either reactions or governing interactions. Figure 9 shows a small real-life

example of an ODE model as applied to describing events pertaining to the activation of the insulin receptor in response to insulin in primary human adipocytes, i.e. fat cells from real human patients. In Figure 9A, this model is depicted using an interaction graph. Another, less graphic, way of depicting an ODE model is a reaction list (Figure 9C), where one simply lists all the reactions that take place in the model. A full description of the model, however, can only be found in the specific details that make up the state space description (Figure 9D).

3.3 Black box modeling and regression models

As already mentioned in previous sections, black box models are models that focus on describing data, but that lack any internal interpretability. That is, you cannot look inside the model to understand what is going on (Figure 10).

Figure 10. Black box models map model inputs to model outputs, but where the inner workings of the model lack any mechanistic interpretations, either because the underlying system is poorly understood, or because it is not important for the current task.

Black box type of models are common in many engineering fields, where the model serves as a tool for regulating a specific process or machine. However, these kinds of models are also used within biology to some extent. Perhaps even more importantly, black-box models are common in fields like *control theory* and *system identification*, which are neighboring fields that have informed the field of systems biology on several levels such as methods and vocabulary [10,25,36].

I am in this thesis, with black box models, also grouping all the various kinds of statistical models that lack interpretation of the actual details of the model. These are models that for example try to establish links between a response variable and various predictor variables. Most common amongst these are statistical regression models. These models work by defining a *response variable*, i.e. the thing you want to model, as a function of one or more *predictor variables*. One then fits a specific model structure, i.e. an assumption of how the response variable depends on its predictor variables, to the available data. This fitting procedure usually tries to minimize the sum of squares between model prediction and observed data. We will return to the issue of minimizing sum of squares in more detail in Chapter 4. The simplest version of these kinds of regression models is a linear model with one response variable and one predictor variable. In Figure 11, I have used the genome sequencing data cost from Figure 4 as a toy example. Using only the data from 2001 to 2008, I fitted a linear model to a log-transformation of this data:

$$\log_{10}(\text{cost}) = \theta_1 * \text{time} + \theta_2 \quad (4)$$

Here θ_1 is the slope of the curve, and θ_2 is the intercept of the curve with the y-axis. These θ s are known as the coefficients, or the parameters, of the model. Again, focusing on the time period of 2001-2008, this model achieves a good fit to the data (Figure 11B, solid line). The quality of the fit is numerically assessed by looking at the aforementioned sum of squares, and also by looking at p-values of the individual parameters (Figure 11A, table). We will return to the issue of interpreting p-values in Chapter 4. For now, it will be suffice to say that a low p-value signifies that the parameter is important for the fit of the model to the data.

Figure 11. Simple linear regression example. (A) Output excerpt from the computational program MATLAB when fitting a linear regression model to data from the genome sequencing example in Figure 4. The model consists of a linear relation between the cost of sequencing and time, using the Wilkinson-Roberts notation [37]. The model has two coefficients, i.e. parameters, which are estimated: The intercept with the y-axis and the slope of the curve (both in log-space). (B) Plot of data (o), model fit (solid line), and model prediction (dashed line). The model was trained on data from 2001 to 2008 (left, white region). The model fails to predict data from 2008 and onwards (right, gray region), where a shift in sequencing technology reduced the price of sequencing at a completely new rate of reduction per year.

This sequencing cost example also illustrates a problem that is common with black box modeling, namely the inability to accurately predict qualitatively different data than the one it was trained on. After 2008, the cost per genome for sequencing took a dramatic shift towards

lower costs (Figure 11B, circles). This was mainly due to new sequencing technology being developed and implemented. However, with this kind of regression model, there is no way in which the model could predict such an event, which leads to the noticeably bad prediction for the years 2008-2016 (Figure 11B, dashed line). Models with more mechanistic details, e.g. ODE models, sometimes fare better at predicting data outside the regime they were trained on. However, there is never any guarantee that any model will be useful for novel situations. Optimally, when developing any model, one tries to separate one's available data into two sets, one set of data used for training, and one set of data used for *validation*, a topic which we will also visit again in Chapter 4.

3.4 Networks and data-driven modeling

In Section 1.1, we discussed briefly how networks such as those depicted in Figure 1 (p. 2) have traditionally been put together one piece at a time. However, with the new *omics* techniques discussed in Section 1.3, a new type of network reconstruction emerged, one that was data-driven in nature [2]. Several methods and modeling frameworks have been applied to this task. I will now go through the most frequently used of these methods.

First, we have the black-box-like type of approaches. These approaches are mainly statistical methods that focus solely on some specific data aspect. For instance there are clustering algorithms that try to group genes with similar expression patterns into clusters [2,38,39]. Identifying components that appear to be co-regulated then generates a hypothesis that these genes or proteins etc. are interacting with each other, either directly or via some shared interactor. Another often used black box type methods is that of Principal Component Analysis (PCA). PCA is a regression method that for each output, or phenotype, of the network, reduces the number of components into only a small number of the most important, i.e. principal, components [2,39].

Bayesian Networks (BN) are a type of directed acyclic graphs (DAG) of conditional probabilistic dependencies, based on Bayes Theorem [2,40]. That is, each node in BN has a probability of being active given some input nodes. Generally, a BN is static. However, time-dependent BNs are possibly by replicating the network for each time point desired and allow for historical dependencies. This expansion also allows for feedbacks, which is generally not available otherwise due to the acyclic nature of BNs. One of the main advantages of the Bayesian framework is for the allowance of prior knowledge to enter into the network, and affect the a priori conditional probabilities. The final network is inferred from the combination of prior knowledge and the updating of the new posterior conditional probabilities after new data is added [40].

Boolean models are a special set of logical models [2,26]. Boolean models are used primarily when analyzing very large networks, or networks where the quality of the data is limited or poor [27]. In this framework, all nodes in the network, and all data considered, are discretized to values of 0 or 1, i.e. TRUE or FALSE. In a biological context, this can for example be interpreted as a gene being turned on or off. In Figure 12, a small toy example of a Boolean network consisting of three nodes is depicted. Each node has a corresponding updating function that takes the binary values of all its interactors as input, and then returns a TRUE or FALSE output for the new value of the node.

Figure 12. A small Boolean model example. (A) This network consists of three nodes, N_1 - N_3 , that each take a binary value of 0 or 1 (TRUE or FALSE). The dashed lines show interactions: A sharp arrow head indicates a positive interaction, and a blunt head a negative interaction. (B) The network is updated using three corresponding Boolean functions, B_1 - B_3 . These functions are written using logical statements such as AND, OR, and NOT etc. The Boolean functions takes the values of all interactors as inputs, and return the new state value of the node. (C) These rules can also be written as truth tables, using all possible combinations of inputs. (D) Using a so called synchronous updating scheme (Equation 5), a transition path can be traced from each of the eight possible starting values of the networks. In this example, there exists three attractors (★). {000} and {011} are fixed point attractors. Additionally, there exist a limit cycle between {001} and {010}.

The rules for updating the network are constructed using prior knowledge and databases with known interactions. Given a specific starting condition, which can be known or evaluated over several permutations, the network is updated iteratively using a specific scheme. The most common scheme is the synchronous updating scheme, where all nodes are updated simultaneously using the values of all the nodes in the previous step (Figure 12) [27]:

$$N_i^{t+1} = B_i(N_1^t, N_2^t \dots N_n^t) \quad (5)$$

Where N_i is a node in the network, evaluated at different discrete time points t , using a node specific update function B_i which is formulated using logical operators such as AND, OR, and NOT etc. (Figure 12B). Given a specific starting condition, a Boolean model is simulated by performing the updating step until an attractor is reached. An attractor is a state space where no further changes will arise from updating the model, or where the model has reached a periodic cycle of states that repeat indefinitely (Figure 12D). Boolean models are typically evaluated for several different starting conditions. Usually one tries to find all the different attractors and the possible path to transition to that attractor, and compare these paths to data or to predict new experiments.

Fuzzy models are another type of logical models that are similar to Boolean models [2,26]. Fuzzy models differ from Boolean models in that they either utilize non-binary (>2) discrete levels for representing node activity, or that they have a continuous representation between 0 and 1, using specific transfer functions when updating and evaluating the networks [26]. Fuzzy models get their names from that they also can incorporate qualitative information such as classification names or other nomenclature [2,26].

Lastly, I wish to mention Flux Balance Analysis (FBA). FBA is about the stoichiometric modeling of reaction fluxes in metabolic networks [29]. The main advantage with FBA is that it is computationally very fast, since most models can be formulated as linear programming problems. One of the limitations however is that it requires steady-state assumptions to be true, and thus cannot model *dynamic* systems.

3.5 Partial differential equations

When it comes to mechanistic modeling, ODE models are the most frequently used type of models (Figure 8, p. 16). However, ODE models require some specific assumptions to be fulfilled, at least approximately, which might not always hold true. For instance, when you use a system of ODEs to describe a compound in any given compartment, it is assumed that any changes concerning the compound is instantly homogenized within the compartment, without any delay. In other words, you are claiming that diffusion is instantaneous, or at least negligible to the overall evolution of the system (Figure 13A). However, this is not always true. If the specific compartment you are considering is large enough, or the time scale small enough, the time it takes for a compound to homogenize within a compartment is not negligible. What is then needed to describe the system is a concentration inhomogeneity within the compartment that is dependent on an additional independent variable: space (Figure 13C).

Figure 13. Spatial components of ODEs and PDEs. (A) Within a single compartment in an ODE model the concentration of a particular compound (grey) is homogenous. Diffusion is considered to be instantaneous, or on a time-scale much faster than the other dynamics described in the model, e.g. chemical reactions etc. (B) Spatial heterogeneity in an ODE model can be achieved by adding more compartments. A natural sub-division might for instance be adding a specific organelle such as the nucleus with transport reactions between the cytosol (light grey), and nucleus (dark grey). (C) When more complex spatial details are needed, such as a concentration gradient, the system will depend on both time and space. In this case, one need to describe the system with Partial Differential Equations (PDEs).

Here then, lies the problem with ODE models. They can only work with one independent variable (time) whereas we in this kind of scenario need a dependency also on a second variable (space). Sometimes, an ODE model could still be salvaged if the spatial inhomogeneity can be represented by sub-dividing your compartment into several smaller compartments. For instance, if you have several smaller organelles inside a cell, those organelles would be allowed to have different concentrations of your specific compound than in the cytosol if they are implemented as separate compartments (Figure 13B).

However, sometimes a spatial description turns out to be a necessary part of the model description that cannot be left out. In such cases, one needs to turn to Partial Differential Equations (PDEs). The PDEs most used to describe these kinds of systems are *reaction-diffusion* equations [41]:

$$\dot{x} = f(x, \theta, u) + D \sum_{j=1}^m \frac{d^2 x}{dz_j^2}, \quad 1 < m < 3 \quad (6)$$

where $\dot{x}$ and $f(x, \theta, u)$ are given as before (Equation 1, p. 22), D is a diffusion coefficient, z_j represents a spatial dimension, and m is the number of spatial dimensions modeled. A separate diffusion coefficient D is needed for each species in x . For most species, this coefficient is not experimentally known, and/or hard to estimate experimentally, and must therefore be estimated together with the other parameters [42]. This means that as a general rule of thumb, PDE problems are always of a higher parametric dimensionality than ODE problems. Higher dimensionality, in turn, implies a higher computational cost, which will be discussed in further detail in Chapter 4. The computational cost of solving PDE problems is further increased by the need to specify concentrations and fluxes at the boundaries of the space being modeled [25]. This problem becomes especially large if the geometry under consideration is complex [41]. PDEs are solved almost exclusively numerically [41]. The three most common approaches are the *Finite Element Method (FEM)*, the *Finite Difference Method (FDE)*, and the *Finite Volume Method (FVM)* [42–45]. These methods discretize the problem by subdividing it into smaller parts. For example, the Finite Volume Method creates a three-dimensional mesh-grid and approximates a small volume around each point on the grid. It should be noted that due to these approximations when solving PDEs, it is at least in principle possible to construct ODE models of similar complexity by introducing a sufficient number of compartments in the model, whether they represent a physical barrier or not.

3.6 Stochastic modeling

An assumption that is made in ODE models is that the states in the model can be described by a continuous state value x . In reality, we know of course that the number of molecules of a particular compound in a specific volume is an integer value, and therefore that the true value of the concentration of this compound must be discrete, and not continuous. However, even for a small number of molecules, a continuous state description is a valid approximation. In fact, this approximation is valid as soon as you have a couple of hundred copies of a compound in your model [46]. If you have less than this number, you might consider using what is known as *stochastic modeling* (Table 2, p. 15). Most metabolites and proteins exist in quantities several orders of magnitude above this. However, there are some components within the cell that exist

in such low copy numbers that stochastic modeling is necessary. The most glaring example is of course the DNA in the nucleus, which exists in only two copy numbers, of which mostly one is active at a given time. If one wished to model the interaction between the DNA and a particular transcription factor, for example, it immediately becomes obvious that it is incoherent to talk about how large a fraction of the DNA that exist in a particular state when in reality the possibilities are binary [47].

Stochastic models are formulated in many ways similar to ODE models, but one specifies the specific reactions of the model, rather than the state derivatives. Importantly one needs to specify the reaction probabilities, c , which are similar in nature to the kinetic rate constants of an ODE model. In other words, they are part of the parameter vector θ . Figure 14 shows a very basic example of two molecules, A and B , that can spontaneously turn into each other back and forth. Instead of reaction rates we have the probability constants c_1 and c_2 , and the initial conditions $A(0)$ and $B(0)$ (Figure 14B).

Figure 14. Stochastic modeling example. (A) Interaction graph. (B) Reaction List. In this example, A has a probability, c_1 , to spontaneously turn into B . B , in turn, has a probability, c_2 , to turn back into A . (C) List of model parameters. In this example, c_1 is 5 times larger than c_2 . (D) A single run of the model using the stochastic simulation software of the MATLAB stand-alone toolbox SBtoolbox2. Due to the inequality between c_1 and c_2 the model tends to hover around 2 and 8 for A and B respectively.

The most common way to simulate a stochastic model is by implementation of Gillespie's algorithm [46,48–50]. Given a stochastic model, the algorithm can be summarized into the following steps:

1. Chose a starting condition for the model, and a simulation end time.
2. Determine the reaction propensities of all the reactions in the model.
3. Generate two random numbers from a standard uniform distribution.
4. Use the first random number to determine the time until the next reaction.
5. Use the second random number to select which reaction happened.
6. Update the reactants with corresponding new values. Update the time.
7. If the end time has been reached, stop, otherwise go back to Step 2 and repeat.

Step 1 is simply choosing the model's initial conditions, similarly to any other modeling framework. The only major difference here is that the states are in molecular numbers, rather than concentrations, and need to be specified in whole integers. Step 2 requires us to determine something called the reaction propensity for all reactions in the model. Simply put this is the combination of the amounts of the ingoing reactants, together with the intrinsic reaction probability. In the case of the simple model in Figure 14, the reaction propensities, a , are:

$$a_1 = c_1 * A \quad (7)$$

$$a_2 = c_2 * B \quad (8)$$

$$a_{total} = a_1 + a_2 \quad (9)$$

The actual stochasticity of the model enters in Step 3. Here we generate two random numbers which we will use to determine the time between reactions, and to select which reaction that happened.

$$rand_1 \sim U(0,1) \quad (10)$$

$$rand_2 \sim U(0,1) \quad (11)$$

The mean time for *any* reaction to happen in this system is $1/a_{total}$. The *actual* time until the next reaction is called τ and is determined in Step 4 in the following manner:

$$\tau = \frac{1}{a_{total}} \cdot \ln \frac{1}{rand_1} \quad (12)$$

In Step 5, we determine which reaction happened in the following manner: From a list of all reactions, we choose the Q th reaction, where Q is obtained by summing reaction propensities until this sum of reaction propensities is greater than $a_{total} \cdot rand_2$:

$$f(q) = \sum_{j=1}^q a_j, \quad q \in \mathbb{Z}^+ \quad (13)$$

$$Q := \operatorname{argmin} f(q) \text{ s.t. } f(q) > a_{total} \cdot rand_2 \quad (14)$$

This means that each reaction has a probability of being chosen proportional to its reaction propensity. In Step 6, we perform the reaction chosen and update the values of all states that are affected. For example, if reaction 1 in our simple stochastic model was chosen, we would reduce the number of A molecules by 1, and increase the number of B molecules by 1. Afterwards, we would update the time to $t + \tau$. If this updating of the time results in that the

end time has been reached, we terminate the simulation in Step 7. If not, we go back to Step 2 and repeat these step until the end has been reached.

The result from performing Gillespie's algorithm is a single simulation run. An example of such a run can be seen in Figure 14D. Typically, one performs a large number of such simulations in order to determine the model's uncertainty range when comparing the model to data. Additionally, one needs to take into account that the number of time steps between each simulation run will differ, and the exact time points when things happened is also different, both due to the intrinsic stochasticity of the model. Due to these facts, stochastic simulations are computationally heavy and take a lot of time and computer power to perform.

4

ODE MODELING METHODS

As we saw in Figure 8 (p. 16), ODE models are the most commonly used models in systems biology. One reason for this is of course a long tradition of applying ODE models to these kinds of biological systems. However, there is a good reason for using ODE models in the first place, and that is that they strike a good balance between model complexity, a quantitative agreement with data, and computational cost.

Biological Realism / Complexity
	Desired Trait	Black-Box	ODEs	PDEs +
1	Quantitative accuracy	Yes	Yes	Partly
2	Optimizable	Yes	Yes	No
3	Biological realism	No	Yes	Yes
4	Detailed description	No	Partly	Yes
5	Avoids unsupported Assumptions	Yes	Partly	No
6	Identifying sufficient components	No	Yes	No
7	Hypothesis testing	No	Yes	No
8	Predicting new experiments	No	Yes	Partly
Data-driven / Computational Speed				

Table 3. Comparison of ODE models (middle column) to less complex (left column) and more complex (right column) modeling frameworks. The rows of the table correspond to different qualities, desired abilities, and usages of models. Red (“No”) means that the quality or ability is lacking. Yellow (“partly”) means that the modeling framework partly supports it, or supports it within a limited range. Green (“yes”), means that the featured is fully functional in the framework. ODE models in general strike a good balance between quantitative accuracy and biological realism. Modified from Figure 1 in Paper IV.

In contrast, models of low complexity are often of a black-box character (Figure 10, p. 19). Black-box models are characterized by the fact that while they may describe the data quantitatively, the description is not based on the underlying mechanisms believed to be at play in the system, and they therefore cannot be used to gain mechanistic insights about the system. At the other end of the spectrum, we have models that are too complex to allow for a good quantitative analysis. These models often use PDEs and/or stochastic processes. In addition, these models usually include a detailed spatial description, which is often assumed somewhat

arbitrarily, since the detailed structure is unknown. Furthermore, these complex models tend to be primarily descriptive, i.e. they include all, or at least most, known mechanisms of the system, with no reliable way to sort the important and necessary mechanisms from the unimportant ones. Perhaps most importantly, these models are typically so complex that they cannot be fitted to data. Instead the parameters of the models are manually fixed or tuned to produce a qualitatively good agreement with data, but not a quantitatively good agreement. Therefore, also these more complex models can be limited in their ability to generate mechanistic insight.

In Table 3, I have summarized some of the important modeling framework capabilities and compared them from the viewpoint of ODE models to less and more complex modeling frameworks respectively. As can be seen, ODE models succeed at, or at least pass, most of the important aspects a modeling framework requires for being useful. This chapter will be dedicated to describing these capabilities, and the various methods used when modeling in an ODE framework. Some of these methods are applicable also to other modeling frameworks, whereas others are specific for the case of ODE modeling. I will start by introducing two important concepts: *the minimal model* and *the modeling cycle*.

4.1 The minimal model and modeling cycle approach

“Entities must not be multiplied beyond necessity.”

Occam’s Razor according to John Punch, 1639.

A *minimal model* is a model that makes as few assumptions as possible while still explaining all *relevant* data. The word *relevant* is important because there is always data available on a multitude of biological components, but whether they are relevant or not depends on the *scope* of the model. For instance, if you wished to model the propagation of an action potential along a neuron, like Hodgkin and Huxley did, having data available for the blood flow to the brain is mostly irrelevant. On the other hand, if you wish to make a large-scale model of the brain, including blood flow and oxygenation is highly relevant. A one-sentence summary of the minimal model approach would be: *Have as few components, and as simple relationships between them, as possible*. The minimal model is a consequence of the *law of parsimony*, also known as Occam’s razor, named after the twelfth century theologian and philosopher William of Ockham. This principle is used widely within science and can be reduced to the facts that when choosing between several explanations, the explanation that makes the fewest assumptions is preferable to an explanation that makes more assumptions.

The construction of a minimal model is an iterative process, where one adds components and/or complexity only when needed. This iterative process is part of what we call the *modeling cycle* (Figure 15). The *modeling cycle* is an integrated framework for minimal model construction, data analysis, model-based hypothesis testing, model analysis, predictions, and experimental design. On a small scale, the modeling cycle depicts how you should apply model analysis to data and theory. On a large scale, the cycle is an instruction for how to work cooperatively and iteratively between the lab that produces the data, and the modeler(s), who analyzes it and tests

explanations. In Figure 15, I have condensed the various tasks involved into five major steps, divided into two phases. The first phase consists of two steps and concerns itself with finding a hypothesis and model that can explain the data. The second phase consists of the last three steps and concerns the analysis of the model once a sufficiently good model has been found.

Figure 15. The modeling cycle can be summarized by the following steps: (1) Given some data and initial hypothesis, construct a mathematical model. (2) Fit the model to the data and test the goodness of fit statistically. Either a), reject the model, or b), don't reject the model. If a), go back to (1) and try to modify the model or hypothesis. If b), proceed to (3) and analyze the model with respect to parameter identifiability and uniqueness of predictions. (4) Identify a testable core prediction, i.e. a prediction with low uncertainty. (5) validate the model against new data, or other independent data not used in the fitting procedure. If validation fails, start over at (1). If the model survives validation, tentatively accept the hypothesis or use the knowledge gained to further develop the model and test it in novel ways.

It has probably not escaped the reader that the modeling cycle (Figure 15) bears a strong resemblance to the falsificationist view of science promoted by Popper we discussed in Section 2.2 (Figure 7, p. 11). This is no accident. In fact, the modeling cycle could and should be thought of as falsificationism in a modeling framework, where our knowledge grows by the rejection and experimental testing of models.

In the remainder of this chapter, we will look in more detail at some aspects of the individual steps of the modeling cycle. Naturally, these steps involve several important sub-steps and methods, of which an exhaustive coverage lies beyond the scope of this work. I have chosen to focus on seven specific topics of the modeling cycle. These topics, and how they fit into the larger picture of the modeling cycle, are summarized in Table 4.

Additionally, in Section 4.8, we will discuss the topic of model selection, i.e. how to choose between competing hypothesis that have not been rejected.

Section	Topic	Part of
4.2	Model construction	Step 1
4.3	Model simulation	Step 2
4.4	Parameter estimation and goodness of fit	Step 2
4.5	Statistical assessment of goodness of fit	Step 2
4.6	Uncertainty analysis	Step 3-4
4.7	Testing predictions	Step 4-5

Table 4. Important sub-steps in the modeling cycle.

4.2 Model construction

The first objective in model construction is the identification of which components that should be included, and which components or details that can be left out. This is usually done by starting from a larger known biochemical network, such as the one in Figure 1 (p. 2), and condensing it down to a small interaction graph of only the most relevant players. Typically, one has data for some parts of the model, for example phosphorylation levels of one or more protein. This data corresponds to the output signal of the model, and the goal is to connect the output signal to the input signal, with as few intermediary components as possible. Using the minimal model cycle approach, one can later add more detail when needed. Let us now look at a concrete example from insulin signaling, using a model from my second paper known as the S6K-model [51]. While the details will differ, the overall approach is generalizable to any type of modeling task.

4.2.1 Hypothesis and data

In this example, we wanted to test a specific hypothesis, namely that the protein S6K is the kinase responsible for phosphorylating another protein IRS1 on the serine residue 307 in response to a dose of insulin. Data pointed towards the opposite conclusion, since a known readout of S6K, S6, displayed a reduced activation when S6K was inhibited, whereas IRS1-ser307 displayed no such reduction. The question was if this data was conclusive, or if there was some way to still salvage the hypothesis that IRS1-ser307 is indeed phosphorylated by S6K. We therefore constructed a model from the hypothesis that S6K phosphorylates IRS1-ser307 (Figure 16, $\mathcal{H}_0$).

4.2.2 Scope and simplifications

To construct the S6K-model (Figure 16), we needed to include S6K, its known substrate S6, the putative substrate IRS1-ser307, and a way to account for the inhibition experiment performed. The inhibition was modelled by including competition for the upstream protein mammalian target of rapamycin (mTOR). Lastly, we needed to connect these proteins to the input, a dose of insulin. For this reason, we added an insulin receptor module, and its downstream substrate IRS1. Two things need to be mentioned here. Firstly, the insulin receptor module we added was knowingly a simplification, with several important steps left out [52–

54], but it was sufficient for generating the desired input-output response needed to propagate the signal to the target of interest, namely the interaction between mTOR, S6K, and the dominant negative (DN) inhibitor version of S6K. Depending on the scope of the model, such simplifications are often used. However, if one wished to draw conclusions, or study in detail, the behavior of the insulin receptor, one needs to insert a more realistic module in its place. Secondly, IRS1 has a dual role in the signaling network, being both an upstream activator of

Figure 16. Choosing a condensed interaction graph to use for model construction. Example from insulin signaling known as the S6K-model [51]. In this model, S6Kp is the kinase responsible for phosphorylating both IRS1-ser307, as well as the known target S6. The model includes an upstream insulin signaling module, comprising the response of the Insulin Receptor (IR) to insulin, its subsequent internalization, and activation of the Insulin Receptor Substrate 1 (IRS1) on several tyrosine residues. Downstream the insulin signaling module is the mammalian Target Of Rapamycin (mTOR), which in the presence of the transfected dominant negative (DN) variant, S6K_{DN}, is sequestered into a complex (dashed gray box). An additional assumption made is that IRS1-ser307 and its phosphorylated form are independent of the tyrosine phosphorylations, and therefore implemented as separate states. This specific case is discussed in further detail in Sections 5.1, 6.1, 7.1.1, and the entirety of Paper II.

mTOR via its tyrosine-regulated kinase activity, and by being the downstream target of S6K. It is known that the tyrosine-dependent activity of IRS1 is the subject of complex regulation by modifications on several serine residues, of which much is not understood [55]. In this paper, we chose to implement the serine phosphorylation state of IRS1 as an independent state, with the possibility of adding a more complex interaction later if it was needed.

4.2.3 Reaction kinetics and measurement equations

Continuing with our example model (Figure 16), all reactions were implemented using mass-action kinetics. In Section 3.2 (p. 17), I briefly mentioned other alternatives, such as Michaelis-Menten and Hill kinetics [32]. However, none of these more advanced variants were needed to fit the data, and were therefore not necessary. The S6K-model additionally had many measurement equations used for fitting the model to the available data. This included data of the primary observation: phosphorylation data of IRS1-ser307 and S6 in response to insulin, with and without DN inhibition of S6K. The data also included phosphorylation time courses and dose-response curves for most of the proteins in the model. For more detail on the available data, I refer the reader to Paper II [51].

4.2.4 Units

When it comes to the units for the states and parameters in the S6K-model (Figure 16), the following choices were made: The states have been implemented as arbitrary concentration units, also known as nondimensionalization [56,57]. This is because we for most components do not know, or can choose to ignore, the real concentrations of these states. Concordantly, we also did not know or need to specify the exact volume of the single compartment covered by the model. This rescaling of concentrations and volume has some side effects that one should notice. Firstly, it means that the exact values, and units, of the compounds and parameters lack a direct interpretation in the model. However, one can achieve such an interpretation by setting fixed values for concentrations and volumes, if such an interpretation is desired. Secondly, this allows for a wide range of concentrations and volumes to be described by the same unitless model. This means amongst other things that if we can reject such a model, we can reject it for all possible combinations of concentrations and volumes, and not just the specific ones which would otherwise be the case. The downside of nondimensionalization, is of course that failure to reject a model might be due lack of detail. In other words, if you would specify known quantities and volumes, you might get a result with unreasonable, e.g. not biologically feasible, parameter values and so on. One should therefore, as always, be wary about over-interpreting non-rejected models.

4.3 Model simulation

ODE models are solved, i.e. simulated, using numerical step-methods. These methods can generally be divided into one-step and multi-step solvers. One-step solvers only require the state information in the proceeding step to calculate the next-step, whereas multi-step solvers use several sequential points for calculating the next step. These methods can also be categorized by whether they are explicit or implicit. Explicit methods calculate the state of the

system at the next step from the state of the system at the current and/or previous steps. Implicit methods on the other hand find a solution by solving an equation involving both the current and a later state of the system.

In this section, I will briefly go through four different groups of methods used for solving ODE systems. These are: Runge-Kutta methods (one-step explicit), Adams–Bashforth methods (multi-step explicit), Adams–Moulton methods (multi-step implicit), and Backward Differentiation Formulas (BDFs, multi-step implicit) [58].

4.3.1 Runge-Kutta, forward Euler, and tolerance

The Runge-Kutta methods, are arguably the most widely used methods for solving ODEs. They were developed around the start of the 20th century by the German mathematicians Carl Runge and Martin Wilhelm Kutta [25]. The Runge-Kutta-methods are ranked by their order. The forward Euler method, developed by Leonhard Euler in the late 1760s, is a special case of the Runge-Kutta methods, namely the first order implementation.

Figure 17. The forward Euler method (First Order Runge-Kutta) for simulating ODEs. The figure shows a toy ODE example simulated using an adaptive step length method (solid black line), and a with the forward Euler method with an exaggerated fixed step length, Δt (dashed arrows). In each point, the derivative is calculated and one then steps in the direction of the derivative with step size Δt . With this large step size, large discrepancies can be seen already after a few steps. Also for small step sizes, errors eventually accumulate. Therefore, higher order Runge-Kutta methods are typically used.

The first order Runge-Kutta method is straight forward. Given a known initial condition, x_0 , a starting time, t_0 , and a step size, Δt , the model is simulated by simply stepping in the direction of the gradient at each time point (Figure 17):

$$x(0) = x_0 \quad (15)$$

$$t_n = t_o + n \cdot \Delta t \quad (16)$$

$$x_{n+1} = x(t_n) + \Delta t \cdot k_1 \quad (17)$$

where k_1 is the first order derivative, given by Equation (1) (p. 17).

The forward Euler method rapidly accumulates numerical errors and requires a small step-size to maintain precision (Figure 17). Runge and Kutta therefore instead proposed the fourth order method for solving ODEs:

$$x_{n+1} = x_n + \frac{\Delta t}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4) \quad (18)$$

where k_2 , k_3 , and k_4 are given by:

$$k_2 = f\left(x_n + \frac{\Delta t}{2} \cdot k_1, \theta, u\right) \quad (19)$$

$$k_3 = f\left(x_n + \frac{\Delta t}{2} \cdot k_2, \theta, u\right) \quad (20)$$

$$k_4 = f(x_n + \Delta t \cdot k_3, \theta, u) \quad (21)$$

In other words, the change in x from one step to the next is the weighted average change given by four different slopes. k_1 and k_4 represent the slopes at the beginning and at the end of the step interval respectively. k_2 and k_3 on the other hand represent the slopes at the mid-point of the step interval, and are given higher weight to the overall step (Equation 18).

The choice of Δt is not trivial. If you choose a too large step size, the derivatives are no longer a good approximation of the true system (Figure 17). On the other hand, if you choose a too small step size, numerical errors will accumulate and introduce errors. Modern methods and software utilize what is known as variable and adaptive step lengths to mitigate this problem. A known example is the Runge-Kutta-Fehlberg method, also known as RKF45 or simply ode45. This method utilizes the absolute value difference, δ , between the fourth and fifth order Runge-Kutta method to either increase or decrease the step size [59]. The step size is increased if δ is smaller than the total tolerance, and reduced if δ is larger than the total tolerance, where the total tolerance is obtained by the absolute and relative tolerances according to:

$$tol_{tot} = tol_{abs} + tol_{rel} \cdot |x(t_{n+1})| \quad (22)$$

and where the absolute and relative tolerances are specified by the user, or by some default value in the software [60]. The absolute tolerance concerns the absolute size of δ , and will be the dominant term in the total tolerance for values of x close to zero. The relative tolerance is the size of δ , as compared to the size of x , and will be the dominant term in the total tolerance for large and intermediary values of x .

4.3.2 Adams–Bashforth

The Adams–Bashforth methods are a group of explicit multi-steps methods. If the explicit dependency of f on θ and u in Equation (1) are dropped from the notation, the equations for calculating the next step for the first three orders of the methods are:

$$AB1: \quad x_{n+1} = x_n + \Delta t \cdot f(x_n) \quad (23)$$

$$AB2: \quad x_{n+2} = x_{n+1} + \Delta t \left(\frac{3}{2} f(x_{n+1}) - \frac{1}{2} f(x_n) \right) \quad (24)$$

$$AB3: \quad x_{n+3} = x_{n+2} + \Delta t \left(\frac{23}{12} f(x_{n+2}) - \frac{4}{3} f(x_{n+1}) + \frac{5}{12} f(x_n) \right) \quad (25)$$

where the first order Adams–Bashforth (Equation 23) is the forward Euler method. Similarly, the method can be expanded for higher orders. One should note that the coefficients for the f -function are not chosen arbitrarily, but are determined by solving a Lagrange form polynomial interpolation problem, and using this polynomial to determine the coefficients [58]. The details of how to do this will not be further expanded upon here.

4.3.3 Adams–Moulton

The Adams-Moulton methods are similar to the Adams–Bashforth methods, but for each order, the solution includes the very term one is solving for, which makes these methods *implicit*. Looking again at the first three orders and omitting the dependency of f on θ and u we get:

$$AM1: \quad x_n = x_{n-1} + \Delta t \cdot f(x_n) \quad (26)$$

$$AM2: \quad x_{n+1} = x_n + \frac{1}{2} \Delta t (f(x_{n+1}) + f(x_n)) \quad (27)$$

$$AM3: \quad x_{n+2} = x_{n+1} + \Delta t \left(\frac{5}{12} f(x_{n+2}) + \frac{2}{3} f(x_{n+1}) - \frac{1}{12} f(x_n) \right) \quad (28)$$

where the first order Adams-Moulton (Equation 23) is also known as the backward Euler method. The coefficients for the f -functions are obtained in very similar way as to that of Adams–Bashforth [58]. The solution for the Adams-Moulton equations does imply an extra step and computational cost. Commonly, the solution is found by an iterative root-finding method such as Newton-Raphson.

4.3.4 Backward Differentiation Formulas

BDFs are another type of implicit multi-step formulas, which are commonly used when solving *stiff* ODE systems (Section 4.3.5). The first three orders when omitting the dependency of f on θ and u are given by:

$$BDF1: \quad x_{n+1} - x_n = \Delta t \cdot f(x_n) \quad (29)$$

$$BDF2: \quad x_{n+2} - \frac{4}{3} x_{n+1} + \frac{1}{3} x_n = \frac{2}{3} \Delta t \cdot f(x_{n+2}) \quad (30)$$

$$BDF3: \quad x_{n+3} - \frac{18}{11} x_{n+2} + \frac{9}{11} x_{n+1} - \frac{2}{11} x_n = \frac{6}{11} \Delta t \cdot f(x_{n+3}) \quad (31)$$

where the first order BDF (Equation 29) is again the backward Euler method. The conceptual difference with regards to the Adams-Moulton methods is that the BDFs utilizes the direct value of x in the previous points, whereas the Adams-Moulton methods utilizes $f(x)$ of the previous points. The coefficients for the f -functions are obtained in very similar way as to that of Adams–Bashforth and Adams-Moulton [58], and since BDFs are implicit, one also here needs to solve the problem using an iterative method such as Newton-Raphson.

4.3.5 On Stiffness and software

Some ODE systems behave in a fashion that is called *stiff* [60]. A stiff system is recognized by the fact that the numerical integrator sometimes is forced to take extremely small step-sizes to prevent the numerical solution to diverge from the true solution. This may be the case even if the true solution is smooth in the interval investigated, but is especially true if the model can undergo rapid changes, for instance a trigger of some event. In the case of a stiff system, implicit ODE solvers are typically used because even though they require an additional step for the solver, one can take much greater step-sizes and thus the total computational time will be much lower.

In practice, ODE models are simulated using some type of simulation software, of which there is an abundance to choose from [61]. The preferred software of my choice throughout this thesis and my papers have been the Systems Biology Toolbox (SBTB) 1 and 2^{viii} for MATLAB [62,63]. The standard ODE solver in MATLAB^{ix} is the Runge-Kutta *ode45*. The default for the Systems Biology Toolbox is the *ode23s* solver. However, SBTB has the functionality of converting models to MATLAB executable files, also known as MEX-files. This uses C, C++, and Fortran code. This conversion involves the CVODE suite from Sundial^x. The CVODE suite uses Adams-Moulton methods varying between order 1-12 for non-stiff problems, and BDFs of varying order 1-5 for stiff problems. Unless otherwise specified, all model simulations shown in this work is simulated using the CVODE suite, and where the default is that the systems are considered stiff.

4.4 Parameter estimation and goodness of fit

Parameter estimation is the process by which one tunes the parameters, θ , of a model to make the model better agree with data. Parameters that can be tuned in this way are called *free parameters*, while other parameters in the model might be locked into a specific value. If a parameter is not free, this could be because the parameter is known by definition, e.g. resulting from an experimental design property, or maybe, if it is a kinetic parameter, that the rate constant of a reaction has been estimated previously to a high degree, and/or is known from literature. In these cases, the term *constant* might be used, but typically the word parameter is used interchangeably, irrespective of whether it is a free parameter or not. Sometimes the term *coefficient* is used to refer to the parameters of the model, as was the case in our regression example (Figure 11, p. 20).

Goodness of Fit (GOF) is a general term for describing how good the agreement is between a model and a set of observations, i.e. data. In general, one looks at two specific aspects when judging the GOF of a model. The first, and major, aspect is the size of the *residuals*, r . The residuals are defined as the measurement value, y , minus the simulated value, $\hat{y}$ [35].

$$r(t) = y(t) - \hat{y}(t, \theta) \quad (32)$$

Small residuals are the result of a good numerical agreement between the data and model, whereas large residuals, in absolute terms, imply that the model has a poor agreement. The second aspect one considers when judging GOF is the sign and correlation of the residuals. A model where there is no correlation between residuals, is preferable to one where there is a

significant correlation. Figure 18A shows a case where the residuals are uncorrelated, but large, whereas Figure 18B shows a case where the residuals are small, but with a high degree of correlation. In practice, one wishes to avoid both situations, but a preference to minimizing the size of the residuals is often given priority to avoiding correlation. This is because when maximizing the GOF one uses what is known as an *objective function*, or *cost function*, and this function typically revolves around utilizing the combined sum of all the normalized and squared residuals as a measurement for the overall GOF. However, it is in principle possible to also include the correlation of the residuals in the objective function. We will now discuss in more details the specifics of constructing your objective function.

Figure 18. Size and correlation of residuals. Simulation (solid line) and data (*) for a toy example. (A) The residuals are uncorrelated, which is recognized by the many sign changes ($\pm$ - boxes), but large (long arrow). (B) The residuals are small (short arrow). However, they are correlated, which is recognized by the fact that the sign of the residuals only changes once: the simulation lies above the data for the first half of the simulation, and under the data for the second half of the simulation.

4.4.1 Objective function

The objective function, also known as the *cost function*, takes the free parameters of a model as input, and returns as a single value, the associated cost of those parameters. To do this, the model first needs to be simulated using the new parameters, θ . Here it should be noted that some model structures can result in a situation where the initial conditions, x_0 , are dependent on the θ -vector. This can either be explicitly, as when one or more of the initial conditions are treated as a free parameter, but it can also be an implicit dependence. Consider again the example from insulin signaling in Figure 16 (p. 33). Let us look at the top-most reaction: the phosphorylation of the insulin receptor. In the interaction graph, this reaction is in fact the lumping of two separate terms. The first term is the insulin receptor becoming phosphorylated from insulin, as indicated by the interaction graph. However, there is also a second term, which represents the basal auto-phosphorylation of the insulin receptor, and the rate for this reaction is a free parameter. This means that depending on the value of this parameter, the basal phosphorylation state of every downstream protein is affected. Using steady-state assumptions, the new, θ -dependent initial conditions can in simple cases be calculated directly. In many cases

however, one needs to solve for the new steady-state solution, or make an initial simulation of the model, without any input, until it reaches a steady-state which can serve as the new initial condition.

Figure 19. The objective function. The purpose of the objective function is to return the cost, $V(\theta)$, of any parameter vector, θ , where the cost is the quantitative agreement of the model simulation to the data and prior knowledge. (A) First the model is simulated. If the initial conditions of the model depend on θ , this may require an additional simulation to achieve a steady-state before simulating the model for your desired input. (B) Once the model has been simulated, the simulated output is compared with data, typically by punishing large residuals using a scheme such as Equation (33). Additionally, one can add punishment for model behavior that does not agree with prior knowledge, sometimes referred to as *ad hoc* punishments.

Once the initial conditions, given θ , have been determined the model is simulated as described in Section 4.3 (Figure 19A). The output from that model simulation, i.e. the measurement equations, is then used to give a score of the agreement with data (Figure 19B), also known as the *cost* of the parameters. The most common and straight forward variant of determining this cost is to calculate the so called weighted sum of all the residuals. The cost, $V(\theta)$ is then given by:

$$V(\theta) = \sum_{\forall i,j} \left(\frac{y_i(t_j) - \hat{y}_i(t_j, \theta)}{\sigma_i(t_j)} \right)^2 \quad (33)$$

where one sums over all measurement signals, i , and time points, j . As before, $y_i(t_j)$ is the experimental data, and $\hat{y}_i(t_j)$ is its simulated counterpart [35]. Finally, $\sigma_i(t_j)$ is the sample standard deviation of the experimental data, which is replaced by the standard error of the mean (SEM) if the data are a mean of several repetitions. If $n_i(t_j)$ is the number of observations, i.e. repetitions, made, then the SEM is given by:

$$SEM_i(t_j) = \frac{\sigma_i(t_j)}{\sqrt{n_i(t_j)}} \quad (34)$$

A large cost, $V(\theta)$, implies a poor fit to data, whereas a small cost implies a good fit to data. This is because when the residuals are large, the numerator in Equation (33) becomes large.

Alternatively, the cost can become large if the denominator is small, i.e. if the uncertainty of the data is very small. This is what is implied by the term *weighted residuals*, namely that they are weighted by the uncertainty. The motivation behind the exact choice of Equation (33) is that it coincides with the statistical χ^2 -test for normally distributed noise, which we will return to in Section 4.5 [35]. However, other choices are also possible if the scoring method punishes large residuals. One can also add additional costs if the simulation violates prior knowledge. This might for example be that the model can potentially display an unrealistic behavior, which one does not necessarily have data for, but is known *a priori* to not be the case, i.e. to be unrealistic.

4.4.2 Cost landscape

From Equation (33) we can see that $V(\theta) = 0$ is the lowest cost attainable by any model, which happens when the model simulation overlaps perfectly with data so that all residuals are zero. However, unless you are dealing with artificial data that has no noise, this situation never arises. In fact, even when you have the true model, and the presence of any level of noise, we do not expect the cost to be zero. In fact, a too low cost might be indicative of a phenomenon known as *overfitting*, which we will return to later in Section 4.7.3. From these facts, it follows that any given model also will have a minimal cost, i.e. a best fit, which is achieved when the model achieves its smallest weighted sum of residuals. We call this minimal cost the *global minimum* (Figure 20), and one should note that in general different models will have different global minimums, i.e. one model may fit the data better than another. One should also note that, depending on the model structure, the global minimum might be achieved by several different values for θ .

The entries of θ , and the associated cost $V(\theta)$, together span a manifold which we call the cost landscape. Figure 20 shows the hypothetical case of how a cost landscape might look like from the perspective of one parameter versus the cost. This is either a one-parameter model, or as more often is the case, we are looking at one parameter of a multi-dimensional parameter space at a time. In this scenario, we have two (non-global) *local minima*. A local minimum is a place in the parameter space where, if you move away from it in any direction, the cost increases, or in the case of a plateau, stays the same. A local minimum is not the *global* minimum if you can move away from it far enough, in the right direction, and find a point or region in the parameter space that yields a lower cost. Such is the case for the two local minima in Figure 20. For the left-most local minima, the global minimum can be reached by going in the direction of increasing θ_I , even though this initially results in a higher cost. Similarly, the reverse is true for the right-most minima. The *global minimum*, which is technically also a local minimum, is where the cost function takes on its lowest value for any points in the parameter space. When it comes to the global minimum, we have in this case (Figure 20) not a single point, as is the case with the two local minima, but a plateau of equally good costs for specific interval of θ_I . A plateau in the cost landscape typically happens when one parameter in the model can compensate for the change in another parameter. For example, if one parameter regulates the activation of a specific reaction, and another regulates the inhibition of the same reaction, then this case can arise, where the change in one parameter can be balanced by the change in another. If this compensation is limited, i.e. it can only compensate up to a certain point we can get the situation in Figure 20. However, if this second parameter can always compensate fully for the

change in the first, the cost landscape turns into a flat line and we have situation known as *unidentifiability*, which will be discussed in Section 4.6.

Figure 20. The Cost Landscape for a hypothetical case. In this scenario, we have one global minimum (grey, shaded area), and two local minima. Unlike the local minima, which are single values, the global minimum is attained by several different values of θ_1 . Since this area is continuous, I here classify it as the same global minimum. The points p_1 and p_2 represent different starting points for a generic optimization algorithm. A local optimization will not search “uphill” and might therefore get stuck in local minima. This means that if you choose p_1 as the starting point for your optimization, you may get stuck in the left local minimum and never find the global minimum. However, if you choose p_2 as your starting point you would find the global optimum. A global optimization algorithm, can go uphill in the cost landscape, and might therefore get over the hump between the left local minimum and the global minimum. However, global optimization techniques come with increased computational costs and performance uncertainty, such as being stochastic etc.

The goal of a parameter estimation enterprise is to find the global minimum. The best estimate of the global minimum, $\hat{\theta}$, is defined as:

$$\hat{\theta} = \operatorname{argmin} V(\theta) \quad (35)$$

To perform this estimation, you typically employ one of several available global optimization techniques. However, let us first define what a local optimization method does.

4.4.3 Local optimization

There exist several different local optimization methods. Given a starting guess, θ^0 , these methods find the local minima in the region surrounding θ^0 . Local optimization techniques are

typically *deterministic*, i.e. they reach the same solution every time. In contrast, global optimization techniques usually involve some stochasticity in their procedures which is part of their strategy for not getting stuck in a local minimum. Take the case in Figure 20 again. When performing local optimization, depending on your starting guess θ^0 , you can get very different results. If, for example, you start at the points p_1 , a local optimization algorithm would find the left-most local minima and stay there. On the other hand, if you chose p_2 as your starting point, you would be able to find the global minimum.

Steepest descent, Newton, and quasi-Newton

The simplest approach for finding a local minimum is the so called *steepest descent*, or *gradient descent*, method. Here one simply steps iteratively in the negative direction of the gradient until it converges, or some other termination criteria has been fulfilled.

$$\theta^{n+1} = \theta^n - h \cdot \nabla V(\theta^n) \quad (36)$$

where h , is the step size, and $\nabla V(\theta_n)$ is the gradient at the previous step.

Figure 21. Path Comparison of two local optimization techniques^{xi}. The figure shows a two-dimensional cost landscape with contour-lines, with the paths taken by the steepest descent method (grey), and the Newton method (black). The steepest descent method always steps orthogonally to the contours lines, while the Newton method utilizes information in the curvature to take a more direct path from the starting point, θ^0 , to the local minima, $\hat{\theta}$.

The steepest descent method converges linearly, which is slower than for example the Newton method, which converges quadratically. The Newton method, which is based on the Newton-Raphson method for root-finding, in contrast to the steepest descent method, also takes

the curvature of the space into account to faster converge on the solution. However, the Newton method requires not only that you are able to calculate the Jacobian, i.e. the first derivative of the cost function, but also the inverse of the Hessian, i.e. the second derivative of the cost function. The Newton method steps iteratively from an initial guess using the following formula [25]:

$$\theta^{n+1} = \theta^n - h[HV(\theta^n)]^{-1} \cdot \nabla V(\theta^n) \quad (37)$$

where the content of the square brackets is the inverse Hessian of the cost function, and with the Hessian matrix defined for all parameters i and j in θ as:

$$H_{ij} = \frac{d^2V}{d\theta_i d\theta_j} \quad (38)$$

The gain from using the extra information contained within the curvature is visualized in Figure 21, where we can see that the Newton method takes a shorter path, i.e. fewer steps, to reach the local minimum. The drawback to this method is that calculating the inverse Hessian may be extremely costly, in terms of computational time. Some methods, called quasi-Newton methods, therefore approximate the Hessian, and/or the Jacobian, by the use of successive gradient evaluations and some updating scheme [64]. There are various ways of doing this. In systems biology, the two most common ways for doing these approximations are the Finite Difference Method (FDM), and the simultaneous simulation of the sensitivities equations [56,65]. For the first approach, FDM, which was already mentioned in a different context with regards to solving PDEs (Section 3.5, p. 23) the gradient is approximated using Taylor's theorem. The reader is referred to [64] for further detail. With regards to the second approach, the sensitivity equations, this involves formulating time derivatives for the sensitivities, i.e. the inner derivatives with respect to θ , as part of the ODE system to be solved [56,65].

Nelder-Mead downhill simplex

Estimating the derivatives is not always feasible. It may, for example, require too many function evaluations, or these methods may be sensitive to inaccuracies in the function evaluation, i.e. noise [56,64]. Therefore, there has been developed a range of so-called Derivative-free Optimization (DFO) techniques. I will here briefly mention one such technique, namely the Nelder-Mead, or *downhill simplex*, method [66]. The downhill simplex method uses what is known as an *active set* of guesses for the optimum. The method successively updates this set of guesses by replacing the worst guess with a new better one. The size of this set of guesses is $n+1$ where n is the dimension of θ . These guesses together span a polytope of $n+1$ vertices, which is known as a simplex. In one dimension, it is a segment of a line, in two dimensions a triangle on a surface, and in a three-dimensional space a tetrahedron, and so on.

A general flow-chart for a single iteration of the downhill simplex method is shown in Figure 22, and can be divided into six major steps. There are five possible outcomes from one such iteration [64]. The first four such outcomes are the replacements of the worst vertex with the reflected point, the expanded point, the outside contracted point, or the inside contracted point respectively (Figure 22). These four points all lie along the line connecting the worst point

through the centroid. The fifth outcome, if none of the previous four is successful, is the shrinking of all points, except the best point, towards the best point.

Figure 22. Flow-chart of the downhill simplex method. (1) Sort the vertices of the active set (triangle) in order of increasing cost value. (2) Calculate the centroid of all vertices but the one with worst cost. (3) Create a reflected point by reflecting the worst point through the centroid. If this reflection is neither best, nor worst, we simply replace the worst point in the active set with this new point and start over. Otherwise, if the reflected point is the best point so far, we go to step 4, else step 5. (4) Here the reflected point is the best point so far, therefore we try to expand further in that direction by creating the expanded point. We then replace the worst vertex with the expanded or reflected point, whichever was best, and start over at step 1. (5) Here the reflected point is either worse than the worst point in the active set, in which case we perform an inside contraction, or the reflected point lies between the worst point and the second worst point in the active set. In the latter case, we perform an outside contraction. If the contracted point is better than the worst point, replace the worst point in the active set with the contracted point and start over at step 1, otherwise go to step 6. (6) If none of the above resulted in an updated active set, we shrink the simplex. Shrinking is done by replacing all points, except the best, by contracting them towards the best point. We then start over at step 1.

Figure 23 shows three iterations of a hypothetical case of the downhill simplex method. In Figure 23A, we can see the initial simplex as a black triangle, along with some points of interest, namely the centroid, as well as the reflection and expansion points. Going through the steps of the downhill simplex algorithm iteratively results in an expansion (Figure 23B), a reflection (Figure 23C), and a contraction (Figure 23D).

The distances one reflects, expands, contracts, and shrinks the simplex are determined by their respective coefficients. For reflection, the standard value of this coefficient is 1, meaning that the simplex keeps its shape as it is reflected through the centroid. For expansion, the standard coefficient is 2, meaning that the simplex is elongated by a factor of 2 in the direction through the centroid. For the contraction, the standard coefficient is $\frac{1}{2}$, meaning that the simplex is shrunk

by a factor of 2 in the direction through the centroid. Lastly, for the shrinking, the standard value is also $\frac{1}{2}$, meaning that all the vertices are halved. These standard values are the values used for the example in Figure 23, although other choices for the coefficients are also possible [64].

Figure 23. Example steps of a downhill simplex. Shown is a hypothetical two parameter cost landscape with contour lines indicating the magnitude of the cost. (A) An initial simplex, black triangle, with the vertices θ^1 , θ^2 , and θ^3 , sorted in order of increasing cost. Shown as black balls are five points of interest: θ , the centroid; r , the reflection point; e , the expansion point; and c , the two contraction points. Shown as a checkered smaller triangle is the shrunk simplex, that the method defaults to if none of the points are an improvement. In the case in (A), the reflection point, r , is better than both θ^1 and θ^2 so therefore an expansion, e , is tried and accepted. (B) Shown is the new simplex (black triangle) after the expansion, and the simplex from the previous step (grey triangle). Performing the steps of the simplex method in this case results in a reflection. (C) The newly reflected simplex. (D) When reflecting the worst point of the simplex in (C) one steps too far, but not so far as to result in the worst point amongst all. We therefore do a contraction from the outside, which results in D).

4.4.4 Global Optimization

The local optimization techniques discussed above all have their strengths and their weaknesses. One such weakness is their tendency to get stuck in a local minimum. This is no accident: by design these methods cannot go uphill in the cost landscape, but only downwards.

In Figure 20 (p. 42), we saw how the two different starting points, p_1 and p_2 , resulted in different outcomes when using a local solver. The main objective of a global optimization method is to have some method for escaping local minima, and climb uphill in the cost landscape, in the hope of arriving at better guesses elsewhere. This strategy for a global search is known as a *metaheuristic* [67]. Most metaheuristics are stochastic, and many draws inspiration from natural phenomena. The method *simulated annealing* tries to mimic the process of repeated heating and cooling used in metallurgic processes to harden a metal [68]. The group of methods called *evolutionary algorithms* draw inspiration from the process of evolution, mimicking steps like mutation, cross-breeding and selection. Perhaps the most well-known evolutionary algorithm is the *scatter search method* [69]. A final example comes from *swarm behavior*, where the most noteworthy example is the *particle swarm method* [70].

Multi-start optimization

Before elaborating on the approaches used for global optimization mentioned above, it should be pointed out that it is possible to sub-divide the parameter space into a fine grid and do a local optimization within each grid space. However, if each parameter is divided into m spaces, we would have to perform m^n , local optimizations, where n is the number of parameters [25]. Such a strategy is called a *multi-start* strategy. When you have many parameters, performing an optimization in each grid space quickly becomes impractical. However, a multi start approach can still be useful using a smaller number of starting points, provided that the initial starting points are chosen in a well-thought-out manner, such as Latin Hypercube Sampling (LHS), which guarantees that the starting point span out the parameter space and do not cluster. Indeed, such methods have been shown to be effective when coupled with local solvers using the gradient information [56].

Simulated annealing

Simulated annealing is a global metaheuristic that was inspired by the metallurgic process of *annealing*, where a metal is repeatedly heated, then cooled, which allows for the crystals of the metal to adopt their lowest energy confirmations [25,67,68]. This process is mimicked in simulated annealing by a global parameter called the temperature. The temperature is a measure of how accepting the algorithm is of worse solutions, as compared to the optimal found so far. Initially the temperature is set to be high, and the algorithm allows for large steps uphill in the cost landscape. The temperature is then iteratively decreased until it reaches zero, where no uphill moves are allowed.

The algorithm for simulated annealing is a modified Metropolis-Hastings algorithm [71,72]. Metropolis-Hastings is in turn a Markov Chain Monte Carlo (MCMC) approach, which are frequently used for sampling unknown probability distributions. In particular, MCMC approaches, including Metropolis-Hastings, are frequently used within the field of Bayesian statistics and modeling [40,73].

Figure 24 shows a generalized form of the simulated annealing algorithm. In short, we start with an initial guess and a high temperature, then iteratively reduces the temperature, all the while exploring the parameter space using some stepping function that can go uphill in the cost landscape. As the temperature decreases, the likelihood that an uphill move will be accepted goes down [25,67]. This is conceptually illustrated in Figure 25, using our previous one-parameter example from Figure 20 (p. 42). The regions of the parameter space that has an associated high cost (orange - red) are searched only in the beginning of the algorithm, when

the temperature is still high. As the temperature is decreased, only regions with a relatively low cost (blue – green) are still being explored.

Figure 24. Basic steps in the simulated annealing algorithm, written in pseudo-code. (1) Define the parameter space, an associated cost function and a start guess. Also decide on how the temperature will be updated, i.e. decreased in a step-wise manner during the algorithm, and how many iterations should be performed per temperature step. An important decision lies in the choice of the step function $S(\theta^i)$. For instance, this could be a version of one of the local solvers discussed earlier, but modified to be able to propose steps going uphill. The scheme for whether to accept a proposed step or not goes as following: If the proposed step is better than the current step, we accept it automatically. If the proposed step has a higher cost than the current, we use the acceptance probability function a , and compare it to a randomly drawn number, q , from a standard uniform distribution. The common choice for a is $a(\Delta V, T) = e^{\Delta V/T}$. (2) The algorithm now loops iteratively over all temperature steps t , performing k iterations of proposing a new step, and then deciding on whether to keep or discard the new step. As the temperature goes towards 0, so does the likelihood of accepting worse solutions. When all steps have been performed, we return the best guess found, $\hat{\theta}$.

The choice of the step function is an important part of the algorithm. All instances of simulated annealing used during my projects have been with a version of simulated annealing available from the Systems Biology Toolbox, called *simannealingSB*^{xii} [56,62]. This particular implementation of simulated annealing couples the algorithm with the Nelder-Mead downhill simplex. The main difference with this implementation to the general forms described in Figure 24 is that rather than just stepping from one point to the next in the parameter space, this simplex version of simulated annealing instead uses the non-deterministic scheme described in Figure 24 to decide on whether the new point should be added the existing simplex or not.

Figure 25. The search space for simulated annealing as function of temperature. The color code indicates the lowest temperature at which a region in the parameter space is likely to be searched. High-cost regions (red) are likely to only be searched in the beginning of the algorithm, when the temperature is high. As the temperature is decreased, only regions with a relatively low cost (blue) are still being explored.

Evolutionary algorithms

Evolutionary algorithms are a group of metaheuristics that draw inspiration from the biological process of evolution. Typically, evolutionary algorithms start with an initial population of individuals, i.e. parameter vectors, and then iteratively try to evolve this population towards a better fitness score, where the fitness is defined by the objective function. This evolution is done over a series of generations, where, in each generation, new individual solutions are generated from a breeding parent generation in two processes called *crossover* and *mutation*, and where unfit individuals are removed from the population, or not allowed to breed [74,75]. A crossover change could for example be the exchange of some of the entries of a parameter vector from one parent with the corresponding entries from another parent, to create a new individual with entries from both parents. A mutation would instead be the introduction of some kind of random noise or change to a single parameter value. A general scheme for an evolutionary algorithm is presented in Figure 26 [76].

Scatter search is an evolutionary algorithm that works very similarly to the general scheme in Figure 26. Scatter search was first introduced in 1977, and works with a population called the *reference set*, which is continually updated throughout the algorithm [69]. The main difference to the generic scheme presented in Figure 26 is that scatter search also utilizes some additional functions for generating new diverse solutions. Scatter search in addition also performs a local optimization on each individual solution [77].

Figure 26. General scheme of a genetic algorithm [76]. (1) Generate an initial population. (2) From the entire population, determine the breeding parent fraction using some fitness criteria, e.g. cost value. (3) Generate offspring by combining the information from two parent parents. (4) Introduce mutations into the offspring and (5) insert the offspring into the population. (6) If the new generation fulfill a stopping criterion, end the algorithm (7). Otherwise, go back to step (2) and repeat.

Particle swarm optimization

The *particle swarm* metaheuristic draws inspiration from a related natural phenomena, namely that of swarm behavior, such as a flock of birds [70]. Like the evolutionary algorithms described above, the particle swarm algorithm also works with a population of individuals, but instead of having the individuals produce offspring, each particle in the swarm moves around in the parameter space with a velocity that is the weighted average of several other velocities combined with a stochastic element. These other velocities depend on the neighborhood around the particle. Which neighboring particles that have an influence on the particle is called the topology of the swarm [78]. In the original publication, only two other positions were considered when determining the new velocity, namely the particles own personal best, and the best global solution found so far [70]. This topology is called the *global best topology*. Additionally, the authors added what is known as an inertia velocity to the topology in a follow up study [79]. Since then, more complicated topologies have been considered such as ring topology, wheel topology and pyramid topology, all with different rules for how an individual particle should learn information from its neighbors [78]. For example, in some methods the particles only have access to the local best, as to avoid getting stuck in a local minimum, which can happen when all particles have access to the global best position.

Figure 27 shows an example of the global best topology as applied to the same cost landscape used in Figure 24 (p. 48). Here we can see how three different velocities are combined to form the particles new velocity and next position. These three velocities are the inertia – the particles current velocity, the cognition velocity – the vector towards the particle’s own personal best, and the socialization velocity – the vector to the best global solution found so far. The importance of each contributing velocity is determined by its weight, ω , which is a design parameter chosen by the user [78]. Additionally, each of these three vectors are weighted in each dimension of θ by r , a randomly generated vector of numbers from a uniform distribution.

In some implementations, this random number is the same for each dimension, and r becomes a scalar. This is called linear particle swarm optimization [78] .

Figure 27. Conceptual visualization of the particle swarm optimization metaheuristic, using the same two-parameter cost landscape as in Figure 24. The swarm consist of several particles (grey circles), each with their own velocity and position. In each iteration, the position of each particle is updated using information from its neighbors. In the global best topology shown above, the new velocity of each particle is weighted together from the inertia - its own current velocity, the cognition velocity - the vector to its own personal best, and the socialization velocity – the vector to the best global position. The figure shows the current position (black circle), next position (dotted circle), and velocities (arrows) for an example particle, with its own personal best (blue diamond), and the global best (red square). The weights ω determine the relative importance of each velocity vector, together with a stochastic element, r .

4.5 Statistical assessment of goodness of fit

Let us look back to the modeling cycle (Figure 15, p. 31). We have in previous sections talked about how to construct an ODE model, how to simulate it, and how to optimize the model for a better agreement with data. The last task in phase 1 of the modeling cycle is to check whether the GOF achieved after optimization is good enough, or if we need to reject the model. Formally this is done by checking the null hypothesis, $\mathcal{H}_0$, that the model of interest, $\mathcal{M}(\theta)$, with a specific choice of θ , has generated the observed data. Specifically, this also includes an assumption about the measurement noise, for instance that it is normally distributed (Equation 3, p. 18).

4.5.1 The χ^2 -test

The χ^2 -test is a statistical test for the size of the residuals. In the case of normally distributed noise, we have that a sum of squares of n independent standard normally distributed variables follows a χ^2 -distribution [80]:

$$\sum_{i=1}^n z_i^2 \sim \mathcal{F}_{\chi^2} \quad (39)$$

where z_i are independent standard normally distributed variables, and the degrees of freedom of the χ^2 -distribution equals n . In the case of normally distributed noise, it follows by definition from Equations (3) and (32) that the residuals, r , also are normally distributed and with mean zero, and therefore that their weighted sum of squares, Equation (33), also follows a χ^2 -distribution [35,52]. This means that, provided that the cost function, $V(\theta)$, does not introduce any extra costs or *ad hoc* punishments, the cost value serves as the χ^2 -test statistic, $T_{\chi^2}^{\mathcal{M}}$. The question of whether to reject a model $\mathcal{M}(\theta)$ is then answered by comparing the χ^2 -test statistic to a threshold value, $T_{\chi^2}^0$, given by the χ^2 cumulative density function with significance level α , and v degrees of freedom.

$$T_{\chi^2}^0 = \mathcal{F}_{\chi^2}^{cdf-inv}(1 - \alpha, v) \quad (40)$$

$$T_{\chi^2}^{\mathcal{M}} = V(\theta) \quad (41)$$

$$\text{if } T_{\chi^2}^0 < T_{\chi^2}^{\mathcal{M}}, \quad \text{reject } \mathcal{M}(\theta) \quad (42)$$

As an alternative to checking whether the test statistic is under your desired threshold, we can test if the p-value for our test statistic is smaller than our significance level α . The p-value is obtained from the χ^2 cumulative density function evaluated at our test statistic:

$$\text{if } p(T_{\chi^2}^{\mathcal{M}}) := \mathcal{F}_{\chi^2}^{cdf}(T_{\chi^2}^{\mathcal{M}}, v) < \alpha, \quad \text{reject } \mathcal{M}(\theta) \quad (43)$$

We get the χ^2 -test statistic of our model from the cost function, and the significance level α is typically chosen beforehand. What remains to be determined is the degrees of freedom v . If the model was tested on independent *validation data* (discussed in Section 4.7.2), then we would have independent residuals and the degrees of freedom would be equal to the number of data points, n . However, it is common that the model is tested on the same data that it was trained on, i.e. the data it was optimized to fit towards. In this case, the residuals are not all independent [52]. We can compensate for the unfulfilled criteria of independence by subtracting one degree of freedom for every *identifiable* entry in θ [52].

$$v = n - \mathcal{I} \quad (44)$$

The concept of identifiability concerns whether we can specify a range of the parameter space in which we with a specific certainty can say that the parameter must lie. If we can specify such a range, the parameter is said to be *identifiable*, and if not, it is *unidentifiable*. For a more detailed description regarding identifiability, the reader can skip ahead to Section 4.6.2. Given that all parameters in θ are identifiable we have:

$$\mathcal{I} = \dim(\theta) \quad (45)$$

where $\dim(\theta)$ is the dimension of θ , i.e. the number of parameters in the model. Parameters can be *structurally unidentifiable*, which roughly means that the parameters in question can be described as a function of the other parameters in the model, and technically be removed from the model. In this case, $\mathcal{I}$ is modified by $t_{\mathcal{M}}$, the transcendence degree, which is the number of such superfluous parameters:

$$\mathcal{I} = \dim(\theta) - t_{\mathcal{M}} \quad (46)$$

Alternatively, parameters can be *practically unidentifiable*. In this case, the reason the parameters cannot be identified with the specific certainty is due to lack of quality in the available data [52]. In this case, one need to try to estimate directly the number of *identifiable* parameters, $\mathcal{I}$. This will be discussed further in Section 4.6.2.

4.5.2 Whiteness, run, and Durbin-Watson test

In Figure 18 (p. 39), we discussed two ways in which the residuals could be unrealistic. Above, we dealt with the first case where the residuals are too large (e.g. a χ^2 -test). As for the second case of unrealistic residuals, there are several ways in which to test whether the residuals are too correlated.

The most common of the tests for correlation of residuals is perhaps the *whiteness test* [35,52,81]. The whiteness test assumes as the null hypothesis that the residuals are uncorrelated. One thereafter calculates what is known as the correlation coefficients, which are further used to calculate the actual whiteness test statistic. The whiteness test statistic follows a χ^2 -distribution. The details of how to calculate the correlation coefficients and the test statistic can be found in [52].

A second test that is common is the *run test*. In this test, we count the number of times the residuals change signs, and compare this to expected number of sign changes. For a small number of data points this number is compared to a cumulative binomial distribution, and for large number of data points, by a cumulative normal distribution [52].

A third test, which we made extensive use of in Paper I, is the *Durbin-Watson test* [35,81]. The Durbin-Watson test is a two-sided test that checks whether the residuals are too correlated, or too anti-correlated. The test statistic is given by:

$$T_{dw}^{\mathcal{M}} = \frac{\sum_{\forall i, \forall j \geq 2} (r_i(t_j) - r_i(t_{j-1}))^2}{\sum_{\forall i, j} r_i(t_j)^2}, \quad 0 \leq T_{dw}^{\mathcal{M}} \leq 4 \quad (47)$$

A value close to zero indicates a positive correlation, whereas a value close to four indicates a negative correlation. In this paper, we calculated the p-value for the Durbin-Watson test statistic empirically using something called *parametric bootstrapping*, which will be discussed later in Section 4.9.

4.5.3 Interpretation of rejections

“It’s only a model.”

Patsy. Monty Python the Holy Grail, 1975.

The different ways of achieving a model rejection, described in the previous sections, are only for a model $\mathcal{M}$ with the specific parameters θ . To reject the whole model structure $\mathcal{M}$, an exhaustive search of the entire parameter space is technically needed. Since this is often impossible in principle, you typically settle for performing multiple optimizations, with many different start-guesses, and possibly different optimization algorithms. Furthermore, in order to claim that your entire hypothesis, $\mathcal{H}$, has been rejected, you need to demonstrate that all relevant model structures have been tested, which is an even harder claim to make.

4.6 Uncertainty analysis

Uncertainty enters into the modeling enterprise in several ways. The first and most immediate way is in the uncertainty of the data. As we saw in Equation (3) (p. 18), we typically assume some kind of error model when doing the model analysis, for example that the residuals are normally, or log-normally distributed. Occasionally, other distributions need to be assumed, and might even be beneficial to do in some cases, such as when having outlier-corrupted data [33]. We will now discuss some ways in which uncertainty relates to our actual models.

4.6.1 Model uncertainty

Model uncertainty is a concept most frequently used within the field of *Bayesian statistics* [73]. We saw in Equation (43) how we could calculate the p-value for a χ^2 -test statistic. The interpretation of the p-value is the probability of observing the data, given that the model is true, and is therefore indirectly used to argue for or against the particular model. In Bayesian statistics you can use Bayes’ rule to ask yourself another question: What is the probability of the model given the observed data? Bayes’ rule uses what is known as *conditional probabilities*, and is given by:

$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)} \quad (48)$$

In words, this rule states the probability of observing event A, given that B is true, is equal to the probability of observing B, given that A is true, times the probability of A, divided by the probability of B. For the case of a model, $\mathcal{M}$, and some data y we have:

$$p(\mathcal{M}|y) = \frac{p(y|\mathcal{M}) \cdot p(\mathcal{M})}{p(y)} \quad (49)$$

The probability of the model given the data, $p(\mathcal{M}|y)$, is called the *posterior probability* of $\mathcal{M}$ while $p(\mathcal{M})$ is called the *prior probability* of $\mathcal{M}$. The prior probability needs to be somewhat

arbitrarily assumed. The probability of the data given the model, is known as the likelihood, and is similar to calculating the cost of the parameters. The likelihood is straight forward to calculate if we know θ . However, if we are also estimating θ , the likelihood needs to be marginalized with respect to the parameters, using for example methods such as MCMC-approaches, which were mentioned when discussing simulation annealing (Section 4.4.4, p. 46). This process is an entire field of itself called *Bayesian inference* and *Bayesian parameter estimation* [73], which will not be discussed in further detail here.

A related concept to model uncertainty is that of model selection, in other words how to choose between competing models. This topic is common in non-Bayesian statistics, also known as *frequentist* statistics. We will return to model selection in Section 4.8.

4.6.2 Parameter uncertainty

We called our best estimate of the parameters $\hat{\theta}$ (Equation 35, p. 42). However, this value also comes with some uncertainty. For instance, as we saw in Figure 20 (p. 42), the global minimum can lie at a plateau, and therefore be cost-insensitive in certain directions of θ . If this plateau is boundless in some directions, we say that the parameter is *unidentifiable*. Even in the case of minima that are not plateaus, we still can say that the parameter has some uncertainty, because values close to the optimal value does not differ significantly in the cost. Taken together, this means that for every parameter entry in θ , we wish to determine an upper and lower boundary for that parameter, where the boundary is specified as the point at which the model no longer adequately can describe the data, meaning that the cost of the objective function has become too high. Parameter uncertainty is important because without it, we cannot know whether the model behavior for a specific set of parameter values, is *necessary* and/or *sufficient* for achieving the fit of the model to the data. Further on, as we will discuss in the next session, to do reliable predictions, we also need to take parameter uncertainty into account.

Sensitivity analysis

Historically, a common substitute for estimating the parameter uncertainty has been to do what is known as a sensitivity analysis [2,65]. The question asked here is: What happens to the model output, if we change the value of a parameter slightly? Traditionally this has been done by changing the parameter value, simulate again, and comparing the difference. However, a much more efficient way is to model the sensitivities as extra states in your model, and solve the equations for these states simultaneously with the original states [65]. These sensitivity equations also have the added benefit that they can be used for optimization solvers when approximating the Hessian.

Fisher information matrix

The sensitivities can also be used to estimate an upper and lower boundary for the θ s. This is done using the so-called Fisher Information Matrix (FIM) [81]:

$$\text{FIM} := \sum_{i=1}^N \left(\frac{\partial y(t_i)}{\partial \theta} \right)^T \Sigma(t_i)^{-1} \left(\frac{\partial y(t_i)}{\partial \theta} \right) \quad (50)$$

where $\partial y / \partial \theta$ are the sensitivities, and where $\Sigma(t_i)$ is the diagonal matrix containing the variance of the measurement noise. A confidence interval, $C(\theta_i)$, under some standard assumptions, for the i :th entry in θ is then given by [82]:

$$C(\theta_i) = \{\hat{\theta}_i - T_t(v, \alpha) \hat{\sigma} \sqrt{FIM_{ii}}, \quad \hat{\theta}_i + T_t(v, \alpha) \hat{\sigma} \sqrt{FIM_{ii}}\} \quad (51)$$

where $T_t(v, \alpha)$ is the student's t-test with v degrees of freedom (Equation 44, p. 52), and confidence level α , and where $\hat{\sigma}$ is an estimate of the true variance of the data [81]. Finally, FIM_{ii} is the i :th diagonal entry of the FIM.

Identifiability and the profile likelihood

Confidence intervals gained from the FIM rely on a number of assumptions to be fulfilled. For instance, it is assumed that the cost-landscape around the global optimum can be described as an ellipse, and therefore that all parameters are *identifiable*. Figure 28 shows three different two-parameter cost landscapes (A,C,E), where only the last one is identifiable and fulfills the necessary criteria for using the FIM approximation of the parameter confidence intervals. In Figure 28A, we have a scenario known as *structural identifiability*. This happens when the parameters enter into the model in a functional relation, and when it only is this relation that affects the cost [83]. For instance, such a relation could be that we only can measure the effect of the sum, or the product, of two parameters, and therefore that an increase in one parameter can be entirely compensated for by a decrease in the other. In Figure 28A, we have the functional relationship h :

$$h(\theta_1, \theta_2) = \theta_1 \cdot \theta_2 - 10 = 0 \quad (52)$$

Figure 28. Parameter identifiability and the profile likelihood. (A,C,E) shown are contour plots of three different two-parameter cost landscapes (solid lines). The thick solid lines indicate the cut-off threshold (Δ_α) for statistically acceptable solutions. Shown as dashed lines are the traces for the profile likelihood. (B,D,F) The corresponding χ^2 -cost along the traces in (A,C,E). (A,B) Structural unidentifiability, θ_1 is unbounded while stepping along the trace of h . (C,D) Practical unidentifiability, due to lack of quality in the data, an upper boundary cannot be established for θ_1 . (E,F) θ_1 is identifiable. Originally published as Figure 4 in [83]. Included with the permission of the authors.

If we explore this profile and step along the trace of h (Figure 28B), we observe that the corresponding χ^2 -value is flat, and there is no upper or lower boundary for θ_1 . This situation

cannot be remedied by improving the quality of the data, but could be remedied if some completely new data, that effects the model output and cost as a function of θ_1 alone, was introduced.

In Figure 28C, we have another situation, known as *practical unidentifiability*. Here we do have a unique optimum, but the quality of the data is not sufficiently good to bound the parameter. If we look at the trace of the profile (Figure 28D) we see that there is a clear lower boundary, but not an upper boundary. We also see that if the cutoff bar would be lowered, potentially an upper limit could be established. This could happen if the quality of the data improved, or more relevant data was collected.

Figure 28 employs something known as the profile likelihood [83,84]. It is method that has gained in popularity over recent years as computer power has increased. The reason for this is that the method is computationally heavy. The profile likelihood method works by starting in $\hat{\theta}$ and then for each parameter stepping in small increments of that parameter, and reoptimize all the other parameters in the model. We do this stepping and re-optimization procedure until the cost crosses a threshold, Δ_α . Using this method, we can define the confidence interval of the i^{th} entry of θ as:

$$C(\theta_i) := \forall \theta_i \text{ s. t. } T_{\chi^2}^{\mathcal{M}(\hat{\theta}_i)} - T_{\chi^2}^{\mathcal{M}(\hat{\theta})} < \Delta_\alpha \quad (53)$$

$$\Delta_\alpha = \mathcal{F}_{\chi^2}^{cdf-inv}(1 - \alpha, \nu_{PL}) \quad (54)$$

where, $\hat{\theta}_i$ is the re-optimized parameter vector where the i^{th} entry has been locked according to the stepping procedure. The threshold value, Δ_α , is given by a χ^2 -distribution, where $\nu_{PL} = 1$ for single-parameter estimates, or $\nu_{PL} = \dim(\theta)$ for a simultaneous confidence interval estimate of all parameters.

4.6.3 Prediction uncertainty

The parameter profile likelihood described above can be extended to not consider only the dependency of any single parameter, but on any given property $Z(\theta)$ in the model [85]. Such a property could for example be the EC_{50} value of dose-response curves [86], the relative levels of total protein phosphorylation (Paper II), the residual calcium concentration between stimulations (Paper IV), or the Excitatory Post-Synaptic Current (EPSC) response after a gene knockout (Paper IV). In other words, $Z(\theta)$ could be anything that could be quantifiable extracted from the model simulations.

The prediction profile likelihood can be estimated using a modified version of the cost function [87]:

$$V^{PPL}(\theta) = V(\theta) + w * |Z(\theta) - Z'| \quad (55)$$

where $V(\theta)$ is the original cost, w is a weight chosen to be sufficiently large to force the right term to approach 0, $Z(\theta)$ is the value of the chosen property for the observed parameter θ , and where Z' is the incremental stepping in the direction of the property Z . A trace for the prediction profile likelihood is achieved by stepping in decreasing and/or increasing values of Z , and optimizing over the modified cost function.

4.7 Testing predictions

With the inclusion of the prediction profile likelihood described above we have now reached the last two steps of our modeling cycle (Figure 15, p. 31). We know how to get upper and lower boundaries for our predictions (Equation 55), but how do we identify an interesting prediction to begin with, and what do we use it for?

4.7.1 Core predictions

In *Cedersund 2012* [88], a *core prediction* is defined as:

“A prediction, $Z(\theta)$, is ... a core prediction, if it is associated with an uncertainty Δ_Z , ... and where Δ_Z implies a biologically interesting, experimentally testable, or in some other way useful and necessary property in the model.”

Furthermore, a *strong core prediction* is defined as a property which must be true if the model is true. In other words, there are no parameters, for which the model can describe the data, that does not have the property Z . A three-step method for identifying and using core predictions is suggested in [88]:

1. During optimization, save all acceptable parameters and look for shared, interesting, and well-defined properties among them.
2. Estimate the prediction profile likelihood for any property found.
3. Perform a validation experiment testing the property and re-optimize the model to the new extended data-set.

Useful and interesting, are of course highly subjective terms, but they are not vacuous, and can be determined with the assistance of an experimental collaborator, or someone with knowledge of the biological system tested.

4.7.2 Validation data

Validation data is any data that was not used in the estimation process of the parameters θ . Ideally you have so much data that you can train the model on one part of the data, called *estimation data*, and test it an independent data set called *validation data*. The benefit of this is two-fold. Firstly, it side-steps the need for estimating the number of identifiable parameters when determining the number of degrees of freedom for testing your optimal parameters (Equations 39-43, p. 52). Secondly, it is the only safe way of guarding against overfitting, which will be discussed in the next section.

In many systems biology applications however, you are not fortunate enough to have large quantities of data available, and you therefore need to put all available data into the estimation process. In this case, the validation data is either produced after a potential core prediction has been identified, or data from another lab, can be used to test the prediction. Many times, this

can introduce additional uncertainty into the data, which makes comparison between validation data and simulation somewhat less quantifiably stringent.

Ideally, as described in the modeling cycle (Figure 15, p. 31), the modeling work and the experimental work can go hand in hand, and the result from a core prediction analysis will feed directly into a new experimental setup. When this is the case, the final validation experiment will decide whether we ultimately reject our model and hypothesis, and start over in the modeling cycle, or if we tentatively accept it for now.

4.7.3 Overfitting

Overfitting is a phenomenon that occurs when a model is overparametrized as compared to amount of data it is fitted to [52]. This phenomenon manifests itself as an ability of the model to fit too well to the estimation data, to the detriment of not being able to describe validation data. This is often a danger within systems biology applications since the models often tend to be overparametrized and the availability of data scarce.

Figure 29. Example of overfitting. Shown in black error bars are noisy data generated from a 3rd degree polynomial. In solid lines, we have polynomial fits of a 2nd degree (red), 3rd degree (blue), 4th degree (green), and 7th degree (purple) polynomial. The higher the degree of the polynomial, the better is its ability to fit to any data point, including outliers. This phenomenon will result in a bad performance on validation data for models that are too complex, such as the 7th degree fit above.

In Figure 29 above, we have a simple polynomial example demonstrating the flexibility of too complex models. Here the underlying truth generating the actual data is a 3rd degree polynomial. As can be seen, the 2nd degree polynomial is quite poor, while the 4th degree is similar to the 3rd. The 7th degree polynomial however is very flexible and goes through almost all data points,

which gives the model a low objective function value on the training data, but possibly a very high value on independent validation data.

In Figure 30, I calculated the mean cost for 100 data realizations of the type used in Figure 29, and plotted them against the polynomial degree for the estimation data (solid line), and validation data (dashed line). As you can see, a more complex model can always fit the same data equally well or better than what any sub-model can. This is the reason the cost always goes down for estimation data with increased model complexity. However, one reaches a point where a more complex model will actually be worse at predicting any validation data than the true model. For these reasons, it is always recommended to end your modeling enterprise with some sort of validation if possible.

Figure 30. The price of overfitting. Shown is the cost, i.e. the objective function value, for estimation data (solid line) and validation data (dashed line) for 8 different polynomial degrees using the same underlying truth as in Figure 30. As model complexity goes up, the cost for the training data always goes down. This is also the case for the validation data for sufficiently non-complex models. However, at a certain point, here at the 4th degree, adding complexity to the model will only make it worse at predicting new validation data.

4.8 Model selection

So far, when we have been discussing the steps of the modeling cycle (Figure 15, p 31), it has been from the viewpoint of having only one hypothesis, and only one model at a time. It is quite common however, that you are considering several competing, or nested, hypotheses at a time, and that you have several model implementations of each model to consider. If you perform the steps of the modeling cycle in parallel you might therefore end up with several models that have passed your statistical test, and perhaps even some validation data. In this scenario, one can

look at two different alternatives. The first alternative is experimental design and testing. The second alternative is to perform a test or ranking, to see if any model *outperforms* any other.

4.8.1 Experimental design and testing

The right experiment is arguably the best way to discriminate between two or more models. This method starts by identifying an experiment, whose outcome will result in the rejection of at least one model. This can for example be done by identifying a core prediction(s), whose boundaries are non-overlapping for at least two of the model structures that you are testing. If such a core prediction can be found, and the experiment performed, we know that we are guaranteed to learn something from performing that experiment. Just as with identifying a regular core prediction, identifying one that is optimal for model discrimination is even harder. There is an entire field, called Optimal Experimental Design (OED), based on methods for identifying and setting up the best experiments for various tasks, such as reducing parameter uncertainty in your model, or the best discrimination experiment [89–91].

4.8.2 Ranking methods and tests

Sometimes performing an experiment to discriminate between model is not always an option. One scenario is that no experiment could be identified with the desired outcome. Another scenario is that acquiring new data is costly in either money, time, or both, and any non-expensive way to reduce the number of available models and hypothesis is a preferable first step, before allocating money that will only result in the rejection of unlikely models. For these, and many other reasons, one often needs to perform some kind of *model selection* method, to rank, or possibly reject, competing hypothesis, based on their relative performance.

Information criterion

The Akaike Information Criterion (AIC) is a method used for ranking models based on their ability to describe the data, while at the same time punishing model complexity as defined by their number of free parameters [92,93]. The AIC is calculated as following:

$$AIC(\mathcal{M}(\theta)) = \log \left(1 + 2 \frac{J}{n} \cdot V_n^{\mathcal{M}}(\theta) \right) \quad (56)$$

where J is the number of identifiable parameters, n is the number of measurement points, and $V_n^{\mathcal{M}}$ is the un-weighted cost function scaled by the number of data points:

$$V_n^{\mathcal{M}}(\theta) = \frac{1}{n} \sum_{j=1}^n \left(y(t_j) - \hat{y}(t_j, \hat{\theta}) \right)^2 \quad (57)$$

The model with the lowest AIC value is then chosen. Importantly, the AIC does not come with an accompanied uncertainty, and we therefore neither know how much better the first model was to the second, and we also do not know if any of the models is a good description of the data in the first place.

Another common information criterion methods encountered is the Bayesian Information Criterion (BIC), which can be said to be an implementation of the AIC for a Bayesian framework [94].

The likelihood ratio test

The likelihood ratio test was developed already in the 1920s, and it considered the relationship between nested models [95,96]. Nested models refer to the concept of one model being a sub-model of the other. This is the case if you can collapse the super-model into the sub-model by fixing certain parameter values, most often to 0 or infinity. A good example is the polynomial fits we saw in Figure 29 (p. 59), where any lower degree polynomial can be achieved by setting the coefficients of any higher order terms to 0.

Originally, the test had more complex definitions but in the 1930s it was shown that for a large number of data points, the likelihood ratio test statistic collapses to [97]:

$$T_{lhr}^{\mathcal{M}_1\mathcal{M}_2} = -2\log(\Delta_{lh}) \quad (58)$$

where Δ_{lh} is the ratio of the likelihoods, and that the test statistic follows a χ^2 -distribution:

$$T_{lhr}^{\mathcal{M}_1\mathcal{M}_2} \sim \mathcal{F}_{\chi^2}(v) \quad (59)$$

$$v = \dim(\theta^{\mathcal{M}_1}) - \dim(\theta^{\mathcal{M}_2}) \quad (60)$$

where the degrees of freedom are equal to the difference in number of free parameters between the super-model, $\mathcal{M}_1$, and the sub-model, $\mathcal{M}_2$. With the exception of some scaling, the log-likelihood of the normal distribution is identical to the weighted residual sum of squares used for calculating the χ^2 -cost. We therefore get:

$$T_{lhr}^{\mathcal{M}_1\mathcal{M}_2} = T_{\chi^2}^{\mathcal{M}_1(\theta^{\mathcal{M}_1})} - T_{\chi^2}^{\mathcal{M}_2(\theta^{\mathcal{M}_2})} \quad (61)$$

which is oftentimes referred to as the *log likelihood ratio test*. What this means in principle is that for two nested models, we only need to compare their χ^2 -costs and compare that difference to a standard χ^2 -distribution with degrees of freedom equal to the difference in number of parameters.

There are two possible outcomes of the likelihood ratio test, and also two different conceptual frameworks for viewing the outcome. The first outcome is that we reject the smaller model. This happens when the difference between the models is larger than the threshold given by the test. Alternatively, we reject neither model, which is the second outcome. We cannot however, reject the larger model or both models. From this first viewpoint, the likelihood ratio test is therefore equivalent to asking whether the smaller model is an allowed simplification of the larger model. The second, equally valid, viewpoint, is from that of the minimal modeling approach. Here we instead ask whether an addition to an existing model results in a significant improvement, as measured by the objective function, or not. If the addition, i.e. the more complex model, is significantly better, then we should choose that model. If it is not significantly better, we should choose the simpler model by the principle of *Occam's razor*. It is important to note that in this latter case, the larger model is not rejected, only selected against.

Since the 1930s the likelihood ratio test has also been generalized to the case of non-nested models [98,99], and to the case of neither model being true [96].

4.9 Bootstrapping and empirical distributions

There is a problem, which we have yet not addressed, with many of the statistical tests we have discussed so far in this thesis. The problem is that the analytical derivations for these tests are usually based on several assumptions, but many of these assumptions might not be true [52]. For instance, some common assumptions are that the experimental noise is (log-) normally distributed, that the parameter estimation process have converged, and that the parameters appear linearly in the model [96–100]. Typically, it is the case in systems biology problems that not all these assumptions will be fulfilled [100–106]. For example, the availability of data is often limited and the signal-to-noise ratio might be low. With regards to your model, the number of parameters that are identifiable and/or appears non-linearly might be high, and in the case of testing several models, such as with the likelihood ratio test, the tested models might not be nested.

Figure 31. Tests in two dimensions. Conceptual basis for the 2D bootstrapping approach developed in Paper I. The values of two generic test statistics, A and B, are plotted on the positive x- and y-axes, respectively. The colored points correspond to bootstrap samples of pairs of these values, and the color of the clouds represent the probability density of those points: red means high density, i.e. a high probability to find a point there, and blue low. The 1D projections of the cloud are plotted on the negative x- and y-axes. The strength of the 2D method developed in Paper I, is that it can discriminate between the two points p_1 and p_2 based on their 2D-density: p_1 lies outside the cloud while p_2 lies inside the cloud. From a 1D-density viewpoint however, the two points are equivalent. Originally published as Figure 1 in Paper I.

An alternative to using these analytical expressions, that have emerged with the increasing availability of computer power over the last decades, is to replace the analytical expressions with empirical distributions of the test statistics. One way to derive the empirical distributions is to use bootstrap approaches. There are two types of bootstrap approaches: parametric and

non-parametric [107–109]. Non-parametric bootstrap samples are generated from the original data set by drawing with replacement. Parametric bootstrap samples, on the other hand, are generated from a specific model, such as the ODE models discussed in this thesis. In other words, bootstrap samples are artificially generated data sets, where the distribution of the bootstrap samples are used to approximate the true Data Generating Process (DGP). There is a rich literature for both parametric and non-parametric methods and their applications to statistical testing in systems biology [110–118].

In Paper I, we made extensive use of parametric bootstrapping to approximate distributions for both χ^2 -tests, Durbin-Watson tests, likelihood-ratio tests, and two-dimensional combinations of these tests. As described above these tests have problem with many assumptions already in one dimension, but derivations of their two-dimensional distributions do not exist, and therefore needed to be approximated with bootstrapping (Figure 31).

5

MODEL SYSTEMS

The purpose of this chapter is to give some additional biological background for the specific modeling projects and systems encountered in my papers, as to be able to better put my results into their proper context. We will start by discussing the insulin signaling network, and its importance for understanding type II diabetes, which will give the necessary context to understand the results from Paper II, and parts of Paper I. After that we will shift gears to the model organism yeast, and talk about concepts such as parent-offspring heterogeneity and cell-to-cell variability. Lastly, we will discuss the studying of synapses in rats and mice, and how some synapses display an unusual phenomenon known as facilitation, which may play an important role in the understanding of epilepsy.

5.1 Insulin signaling system in human adipocytes

The insulin signaling system plays an integral part in understanding and devising treatment for type II diabetes (T2D). T2D is quickly becoming a global problem. For high-income countries, the prevalence of diabetes can be as high as 8% amongst adults, with over 90% of those 8% being T2D [119]. T2D manifests itself in two ways: firstly, as an insufficient release of insulin by the beta cells, and secondly as a resistance to insulin in the target tissues [120]. Insulin is the main anabolic hormone in the body, responsible for regulating the metabolism of carbohydrates, fats and proteins, with its chief effect being the uptake of blood glucose to muscle, liver, and fat tissue. During the initial stages of the T2D, the beta cells can compensate for the insulin resistance, but as the disease progresses, the beta cells are eventually not able to produce insulin in quantities necessary to overcome the increasing insulin resistance and properly regulate the blood glucose levels.

Here we will look at the insulin signaling system of primary human adipocytes (fat cells) from the adipose (fat) tissue. Primary means that we are using data from real patients, and not from a particular cell-line. The adipose tissue is of particular interest because insulin resistance in this tissue seems to be a consequence of an expanding adipose tissue, as more and more fat is stored. Furthermore, insulin resistance in the adipose tissue often come before insulin resistance in other tissues such as muscle and liver, and may have a causal part in the progression of the disease [120,121]. The main function of the adipose tissue is to act as an energy buffer. When energy is in abundance, the fat cells take up glucose from the blood, and store it in the fat cells as triglycerides in the form of a lipid droplet, and when energy is needed, these triglycerides can be broken down and be released out into the bloodstream to serve as an energy source.

The uptake of glucose into the fat cells starts with insulin binding to the insulin receptor (IR), which is a transmembrane receptor responsible for propagating the signal from the extracellular medium to the intracellular components. The insulin signaling network is vast, and highly interconnected with other networks [122], which makes establishing the individual roles and

mechanisms of the players in the network very hard, and thus suitable for modeling [34,53,54,123–126]. In Figure 32, a reduced network featuring some of the key players of the insulin signaling system is shown. In this system, the signal travels from the insulin receptor down to intracellular intermediates such as the Insulin Receptor Substrate-1 (IRS1), Protein Kinase B (PKB), mammalian Target of Rapamycin (mTOR) in complex 1 or complex 2 (mTORC1/mTORC2), S6 Kinase (S6K), and so on. Shown is also downstream effects on cellular processes such as protein synthesis, mitochondrial function, and autophagy. One of the most important consequences of the initiated signaling cascade is the recruitment of Glucose Transporter 4 (GLUT4) containing vesicles into the plasma membrane, which results in an increased uptake of glucose into the adipocyte.

Figure 32. Intracellular insulin signaling network. Blue arrows indicate reactions or propagation of signal. The green arrow indicates a positive feedback signal. The red arrows indicate functional effects of insulin-resistant insulin signaling in T2D. Included and modified with the permission of the authors from [126].

In Paper II, we focus on a particular aspect of the green arrow in Figure 32, namely the positive feedback from a downstream component (here shown as mTOR) up to IRS1. This feedback had been shown in other cell types to come directly from S6K, but experimental evidence suggest that this was not the case in human adipocytes [127]. In Paper II, we investigate whether this conclusion held up under further scrutiny, and which conditions must apply for it to be

considered valid. Finally, in Paper I, we looked at models describing the insulin receptor and the initial response to insulin stimulation.

5.2 Cell-to-cell variability in yeast

Baker's yeast (*S. cerevisiae*) is one of the most preferred eukaryotic models when it comes to cell and molecular biology. In 1996, yeast was the first eukaryotic genome to be fully sequenced [128]. Since then, the powerful available genetic tools, combined with the comparatively simple growth requirements has made yeast a powerful model organism for studies in cell and molecular biology. The knowledge gained from studies of yeast has high relevancy also for the study of other model organism, including humans, since many features are conserved throughout the eukaryotic kingdom.

Figure 33. Cell-to-cell variability and the problem with averages. (A) Hypothetical average response, e.g. from a cell culture, to some stimuli. The response looks continuous and sigmoidal. (B) Three individual cell responses, where each cell responds in a switch-like manner, but for different input strengths. (C) If the distribution of the cells is unimodal for a specific trait, the population average (dashed line) is the most common value, and might be a relevant approximation to use. (D) If the underlying population is more complex, such as a bimodal distribution, it may be the case that the number of cells behaving like the “mean cell” are virtually non-existent.

In Section 1.4 (p. 7), we discussed the topic of personalized medicine, and how to best address the differences between individuals when determining the best care for patients. Differences between individuals in a population, or between cells, is a topic that applies to all aspects of life and model organisms, including yeast. There are many various sources of population heterogeneity. In nature, the most obvious ones are environmental and genetic differences. However, even when cells have the same genotype and/or live in the same environmental conditions, they can differ in many ways. For multi-cellular organisms, some cells may be specialized to perform a specific function, but also for single cell organisms we still can see differences. Cells can differ because they are of different age, e.g. in yeast we can see differences between mother and daughter cells, they may be in different stages of the cell cycle, have different expression levels of specific proteins, and even possible epigenetic differences, such as inherited protein and/or DNA modifications. All of these differences add up, resulting in that specific cells can respond differently to the same stimulus, even though they have the same genotype.

One type of error that can arise from not taking cell-to-cell variation into account is shown in Figure 33. Here we consider the case where one has prepared an entire culture of cells, and then measures the mean response of this culture for various input strengths. The result then looks as a typical sigmoidal dose-response curve (Figure 33A). However, we can imagine a scenario where each cell in the culture has a binary response. The cells then either respond fully, or they do not respond at all, and the outcome of this decision comes at different input strengths for individual cells (Figure 33B). The combined average of such a culture could look still look exactly like the response in A. For example, imagine that the population response to a certain stimulus is 50%. Without further investigation, we cannot distinguish between the case where 100% of all cells respond with 50% response strength, or if 50% of all cells respond with 100% strength, or anything in between. Sometimes the mean response of a population is still relevant, and it can make sense talking about an average cell as a valid approximation (Figure 33C). Sometimes, however, it does not make sense to talk about the average cell. For instance, if we are dealing with distinct sub-populations, it may be the case that neither of these sub-populations overlap, and the population average would be representative of no cells (Figure 33D).

One source for cell-to-cell variability that I mentioned above is the one between mother and daughter cells. Yeast cells undergo something called asymmetric cell division. This asymmetry manifests itself in the mother cell being distinct from the daughter cell in several ways, such as size, number of bud scars, retention of damaged cell components and so on. Understanding asymmetric cell-division in yeast and pinpointing the similarities and differences to e.g. humans, is of importance for topics such as cancer and ageing.

Fluorescence Recovery After Photobleaching (FRAP, Figure 34) is an experimental technique we used in Paper III, combined with mathematical modeling to develop a method for quantifying nuclear transport rates in single cells. In Paper III, as a proof of concept, we look at the freely diffusing Yellow Fluorescent Protein (YFP), which is non-native and inert in yeast. With the proposed method now established, the next step is to look at specific YFP-tagged proteins of interest, such as transcription factors, and if their transport rates across the nucleus differ between mother and daughter cells.

Figure 34. The FRAP Experimental Technique. (A) A cell containing fluorescent (yellow) and non-fluorescent (grey) proteins, with an abundance of YFP located in the nucleus. A photobleaching in the nucleus is initiated (blue flash). (B) After the bleaching event, there is a relative shortage of fluorescent proteins in the nucleus. (C) Over time the balance of fluorescent to non-fluorescent proteins will somewhat recover due to continuous in and outflux (black arrows) of the YFP protein to and from the nucleus. The speed and amplitude of this recovery can be studied to determine the rate constant of the transport.

5.3 Facilitation in murine nerve cells

Sensing and responding to changes in the environment is a critical ability of all living organisms. In unicellular organisms, this usually takes the form of a specific receptors that trigger an intracellular signaling cascade in response to changes in external stimuli. Many single-celled organisms also communicate between each other by secreting specific signaling substances that neighboring cells can sense. In larger, multicellular organisms, such as mammals, these mechanisms still exist in various forms, but have also been complemented with sensory organs, such as eyes and ears. These organs are in turn coordinated and regulated by the nervous system.

The nervous system coordinates the bodily functions of all animals and transmits signals to and from different parts of its body. The nervous system consists of two main parts, the central nervous system (CNS) and the peripheral nervous system (PNS). The CNS contains the brain and spinal cord and is responsible for processing and integrating signals sent to the CNS, and for coordinating and regulating all the parts of the body. Under most definitions, the CNS also includes the visual system and the olfactory system.

Information is sent along the nervous system via neurons. These signals are in the form of electrochemical waves traveling along neuronal fibers called axons. Axons terminate in a synapse; a junction between two neurons where the information is chemically conveyed from one neuron to the next, via transmitter substances (Figure 35). The different connections between neurons form neural circuits and networks that regulates an organism's perception of the world and ultimately determine its behavior. When a synaptic signal is transmitted, it may result in an activation, an inhibition, or some other sort of modulation of the target. Which outcome that happens depends on many factors. One such factor is the history of the system, i.e. the number of previous stimuli and their lengths.

Figure 35. Propagation of signal in a synapse^{xiii}. (1) An action potential depolarizes the membrane in the presynaptic cell. (2) This causes voltage-gated ion channels (pink) to open and calcium to flow into the cell. (3) Vesicles containing transmitter substance (green) responds to the increased calcium concentrations by (4) fusing with the cell membrane and releasing their cargo into the synaptic cleft. (5) Transmitter substance binds to receptors (dark green) and initiate a depolarization and action potential in the postsynaptic cell (6). (7) Afterwards transmitter substance is transported back into the cells via transporters (orange).

External stimuli are often continuous, i.e. they are not a one-shot-effects. However, different synapses respond differently to repeated stimulation. Most synapses exhibit a decreased response over time, a phenomenon called depression, which can be thought of as adaptation over time (Figure 36B, bottom). Synapses displaying this quality are called phasic synapses or drivers [129,130]. The opposite phenomenon, where repeated stimulation leads to an increased response, is called facilitation (Figure 36B, top). Synapses with this behavior are called tonic synapses or modulators. One of the most pronounced cases of facilitation can be found in pyramidal nerve cells in layer six of the visual neocortex.

For the visual system, and most other sensory systems, sensory information is passed via the thalamus, which acts as a kind of signal hub. Photoreceptors in the retina of the eye send information to the Lateral Geniculate Nucleus (LGN) of the thalamus (Figure 36A). In the LGN, the signal is both processed and relayed to the visual cortex, and in return, the LGN receives strong feedback connections from the visual cortex that results in a complex interaction of excitations, inhibitions, and/or modulations. One such modulation is regulated by the facilitating neurons in the visual cortex. The function of this regulation is unknown, but recent finding indicates it is important for cognitive function [131]. It is also suspected that facilitation could be involved in epilepsy, by inducing a feed-back loop of constant re-firing of the signal [132].

Figure 36. Paired pulse depression and facilitation in rat synapses. (A) Schematic of the experimental setup of rat brain slices with the optic tract (OT), LGN, and the optic radiation (OR) from the cortex, electrode stimulation points (OR, OT), and patch-clamp recordings. (B) Top: Facilitation in OR synapse. Bottom: Depression in OT synapse. As measured by EPSC recordings. (C) Paired pulse facilitation (OR) and depression (OT) as a function of varying time delay between first and second stimulation. Originally published as Figure 1 in [133]. Included with the permission of the authors.

The underlying mechanisms of facilitation are unknown. Potential candidate hypotheses involves accumulation of residual calcium between stimulations [134], saturation of endogenous buffers [135], recruitment of vesicles to the active zone [129], and priming of vesicles for fusion [129]. Through mathematical models, and in some cases experimental testing, these mechanisms have independently been shown to be able to produce facilitation. It is, however, not known which of these mechanisms, if any, are used in modulators to achieve facilitation, and if the mechanisms differ between different modulating synapses. However, recently it was shown that the calcium sensor synaptotagmin 7 (syt7) is required for facilitation in at least four different synapses including the layer 6 pyramidal neuron [136]. Previously, it has been suggested that more than one mechanism might be necessary to account for facilitation in the layer 6 pyramidal neurons (Figure 36), due to the high levels of facilitation observed in these cells [9].

In Paper IV, we use a minimal modeling approach to test the viability of four different hypotheses as applied to data from facilitating synapses in pyramidal neurons in rats and mice. Using modeling tools as described in Chapter 4, we could show that a single mechanism was *sufficient* for explaining the available data, and we were able to reject all but one surviving hypotheses, which also could explain independent validation data.

6

RESULTS

Here I present the main findings from my papers. These will be presented in the following order: Paper II, Paper III, Paper IV, and Paper I. Paper II–IV are mainly applied modeling papers, whereas Paper I is mainly a method development paper, although with parts of the results pertaining to the early phases of insulin signaling.

6.1 Modeling of dominant negative inhibition data

In Paper II, we looked at an experimental technique known as Dominant Negative (DN) inhibition, which can result in only partial inhibition, and at how one has traditionally interpreted the results of such partial inhibitions [137,138]. Imagine three proteins: A, B, and C. We know protein A activates protein B, but are unsure if A activates protein C, or if it is another unknown protein that is responsible. The argument then goes like this: if we inhibit A, and see a partial decrease in both A and B, but no decrease in C, we can conclude that A does not activate C. At face value, this reasoning sounds logical, but there are reasons why this conclusion is at best premature, and even worse, might be completely wrong.

Figure 37. Modeling cycle approach to DN data interpretation. Given some initial data, we test a given hypothesis by translating it into a mathematical model and fit this model to the data. If the model cannot explain the data in a satisfactory way we modify the model and repeat this fitting procedure until we either have an acceptable model or until all plausible modifications have been exhausted, in which case we reject the hypothesis. If the model can fit the data in the basic observation we fit the model also to time-series data of different intermediates and repeat the previously described modification loop. If the model can accurately describe the combined data-set we analyze the fitted model for core predictions, which we can use to suggest simplifications of the model, or new experiments. Originally published as Figure 4 in Paper II.

We developed and proposed a method for how to interpret DN data, using the modeling cycle approach from Figure 15 (p. 31) as a template. The result can be viewed in Figure 37. We then applied this specific methodology to re-interpret data pertaining to the phosphorylation of IRS1 on serine residue 307. It had previously been shown that in primary human adipocytes, unlike other known cell lines, S6K inhibition by DN transfection did not result in any effect on IRS1-ser307 phosphorylation [127]. This lack of response was used as evidence to suggest that S6K was unlikely to be the kinase responsible for IRS1-ser307 phosphorylation in this cell type. However, a complete model analysis was not part of this conclusion, so a more thorough investigation was warranted.

We therefore constructed a model corresponding to the hypothesis that S6K was the kinase responsible for this phosphorylation (Figure 38A). We fitted this model to DN data, as well as time-step and dose-response data from several intermediaries in the insulin signaling network, and showed that this model could fit both the DN data (Figure 38B) as well as the remaining data set (Figure 38C).

Figure 38. Modeling of DN inhibition of S6K. (A) Interaction graph of our implemented ODE model, see Figure 16 (p. 33) for a larger version and description. (B) Using standard assumptions, we showed that our model could explain the basic observation of reduced phosphorylation in one downstream target, but not in another. See Figure 3 in Paper II for more detail. (C) The model could simultaneously fit to all relevant data from insulin signaling intermediaries. Shown are time-step curves and dose-response curves. See Figure 3-7 in Paper II for additional details.

We analyzed the properties of the model to find out what was the core explanation for the observed behavior. In Figure 39A, we see the simulated fraction of total IRS1-ser307 in Wild Type (WT) conditions (blue), and under DN inhibition (red), for a first approximation of

the entire parameter space. As can be seen, the total fraction of IRS1-ser307 phosphorylation is very high, >0.9 , which means that the un-phosphorylated form is near depletion. We investigated this property using prediction profile likelihood analysis and identified the core prediction that more than 85% of IRS1-ser37 needs to be phosphorylated at 90 min after insulin stimulation, for this model to be true (Figure 39A).

Figure 39. IRS1-ser307 needs to be close to depletion, if the model is true. (A) Simulation trajectories under WT-conditions (blue) and DN inhibition (red) using a first approximation of the acceptable parameter space. The phosphorylated form of IRS1-ser307 is very high, $>95\%$. (B) We investigated the property identified in (A) using prediction profile likelihood and found the lower boundary for this prediction to be $\sim 85\%$. Taken from Figure 7 and 9 in Paper II.

We did not test this prediction experimentally for a number of reasons. Biologically it was unrealistic that a protein would display a near to maximal response already for intermediate physiological levels of insulin. Secondly, measuring complete protein fractions require measuring total protein abundance, which is cumbersome and costly. Thirdly, a failure to detect a depletion does not rule out any sort of local depletion, such as a spatial or temporal saturation effect. We do however suggest that if one is unsure about any of these conditions, the best experiment to perform is to stimulate the DN inhibited cells with several different stimulation strengths. If your protein is near to depletion for a lower stimulation strength, we should see no increase at higher stimulation strengths.

6.2 Quantification of nuclear transport rates in yeast cells

In Paper III, we combined the experimental FRAP technique with mechanistic modeling and parameter estimation to get reliable quantifications of transport rates of YFP molecules across the nuclear membrane in different yeast strains. It is of importance to be able to quantify not only the abundance of a particular protein, such as a transcription factor, in the nucleus, but also the rate at which these proteins enter and leave the nucleus [139,140]. Nuclear localization of a protein may be due to either static or dynamic mechanisms. In the first case, it might be because the protein is binding to another molecule in the nucleus, or that it lacks any means of export. In the second case, it is because the rate constant for import is higher than for export, resulting

in a relative abundance of the protein in the nucleus [140]. Since proteins can undergo modifications in the nucleus, which they can bring back with them to the cytosol once they have recycled through the nucleus, the ability to determine the rate of nuclear shuttling becomes important.

In this study, we used a single two-compartment model to describe the transport of unfused YFP across the nuclear membrane. The model consisted of two ODEs, with mass action kinetics, and two kinetic parameters: import (k_I) and export with volume fraction (k_{EV}). The model had 4 extra parameters for controlling the resetting of the initial conditions after each FRAP (Figure 40). We settled on this very simple model because it followed by necessity from some simple assumptions about the underlying system. We also tested more advanced models and could rule them out as they did not result in any improved fit.

Figure 40. A train of FRAPs improves parameter estimation. (A) General scheme for performing a train of bleaching and recovery events. Here shown with 2 FRAPs. (B) Simulation study showing the recovery of the true parameter value (grey line) as a function of different number of trains. For four trains, the most robust effect is seen on recovering true parameter values. Taken from Figure 1 in Paper III.

We first did an *in-silico* simulation study with various noise levels to investigate the optimal experimental setup for efficient parameter recovery. From the simulation study, we found that a sequential train of FRAPs was preferable for a reliable recovery of the true parameter values (Figure 40A), and that four FRAPs performed most reliably under situations of high noise (Figure 40B).

We next performed the suggested experimental procedure on yeast cells from two different strains: BY4741, and W303. To verify that this resulted in identifiable parameters also with data from *in vivo* experiment, we investigated the profile likelihood of the two kinetic parameters. Figure 41C shows the results from a profile likelihood analysis of a typical cell: the two parameters are identifiable. The results from the parameter fitting as applied to all cells can be seen in Figure 41A (BY4741) and Figure 41B (W303). As expected for a protein diffusing freely through the nuclear pore complexes, the scatterplots of k_I vs. k_{EV} lie along the identity line, but with a large inter-variability for each cell line.

Figure 41. Quantification of nuclear transport parameters and nucleosol to cytosol volume ratio. (A-B) Single-cell rates of nuclear transport in strains BY4741 and W303 respectively. The circles indicate individual cells, the black stars show the mean. (B) Both transport parameter in the model are identifiable. Shown is the profile likelihood analysis as performed on one example cell. (C) The cytosol to nucleosol diffusion-available volume ratio is similar in strains W303 and BY4741. Taken from Figure 2 and 3 in Paper III.

Unfused YFP is a non-native protein to yeast, which interacts poorly, if at all, with any cellular constituents. YFP diffuses freely across the nuclear pore complexes, which means that the ratio of the two kinetic parameters gives us the diffusion-available volume ratio between the nucleosol and the part of the cytosol that is available for diffusion. This fraction is not easy to estimate since a large portion of the cytosol is occupied by organelles and other spaces. However, we can now use our proposed method for rate quantifications to also estimate this fraction. In Figure 41D, we show this volume fraction for both strains (Average 4.77, CV = 0.20).

6.3 Quantitative modeling of facilitation in pyramidal neurons

In Paper IV, we evaluated four main hypotheses pertaining to the observed facilitation in pyramidal neurons (Figure 42). We created a general minimal model framework for the presynaptic neuron and then translated the biological hypotheses into mathematical models which fit into this framework. The four hypotheses and their corresponding models were: Accumulation of residual calcium ($\mathcal{M}_r$), Saturation of endogenous buffer molecules ($\mathcal{M}_b$), Recruitment of additional vesicles to the active zone ($\mathcal{M}_v$), and Syt7-priming of vesicles for facilitated vesicle fusion ($\mathcal{M}_s$).

Figure 42. Schematic overview of active mechanisms at the presynaptic neuron, and the investigated possible mechanisms behind facilitation in pyramidal nerve cells. (1) Accumulation of residual calcium between simulations. (2) Saturation of endogenous buffer molecules. (3) Recruitment of additional vesicles to the active zone. (4) Priming of vesicles for facilitated vesicle fusion and transmitter release. Taken from Figure 2 in Paper IV.

Figure 43. Fits and properties of the four models. Row (A) – $\mathcal{M}_r$, the residual calcium accumulation model. Row (B) – $\mathcal{M}_b$, the buffer model. Row (C) – $\mathcal{M}_v$, the vesicle recruitment model. Row (D) – $\mathcal{M}_s$, the syt7 vesicle priming model. Column (1) - Best fit of respective model to EPSC data using 5 different simulation frequencies. Column (2) – Prediction profile likelihood corresponding to testing for asynchronous release. Column (3) – Prediction profile likelihood for testing of peak amplitude during syt7-deletion. All models but $\mathcal{M}_s$ are rejected. See Figure 3 in Paper IV for additional details.

Figure 43 shows the model fits and model properties of all four models. As can be seen from column 1, all models are able to produce quantitatively good fits. They are all also able to pass a χ^2 -test. However, all models but $\mathcal{M}_s$ can be rejected because they entail necessary predictions that contradict *a priori* biological knowledge. Both $\mathcal{M}_r$ and $\mathcal{M}_b$ are rejected because they have as a necessary consequence, as determined by prediction profile likelihood analysis, a high level of asynchronous release (Figure 43, A2-B2). $\mathcal{M}_v$ does not have this problem but in turn is unable to accurately predict the behavior of the system during a *syt7* knockout (Figure 43, C3). Only $\mathcal{M}_s$ is able to quantitatively fit to the available data without resulting in a biologically unrealistic prediction (Figure 43D).

We show that the $\mathcal{M}_s$ mechanism is sufficient for explaining all available data, and that a more complex model including the two best mechanisms tested ($\mathcal{M}_s$ and $\mathcal{M}_v$), does not improve the fit. Lastly, we try to validate the $\mathcal{M}_s$ model using data from mice. We retrain the model on this data and show that $\mathcal{M}_s$ can satisfactorily fit to equivalent ESPC data from mice. Additionally, we show that we can explain the altered behavior in a mutant strain by locking all parameters in the model but one, and fit the data by simply scaling the *syt7* priming parameter to 50% of its original value. As a final validation, we satisfactorily predicted the model behavior under a more physiological stimulation firing pattern.

6.4 A novel method for hypothesis testing using bootstrapping

Based on work by *Hinde* in 1992, *Cedersund and Roll* had previously suggested that producing two-dimensional bootstrap distributions of χ^2 -costs from two competing models, could be an informative method for discriminating between models [52,118]. However, they did not develop any framework for quantifying such distributions into statistically meaningful p-values.

In Paper I, we expanded this concept from simply combining two χ^2 -costs into a more general framework of combining any two different test statistics. These two test statistics could be two different statistics from the same model, or the same statistic from two different models [35]. The reason for why this would be beneficial is illustrated in Figure 31 (p. 63).

In the paper, we analyzed the tests we developed on simple test cases: *static* or *dynamic*, and *linear* or *nonlinear* cases (Table 2, p. 15). From these test models, we generated artificial data to which we trained our models and generated bootstrap distribution. The models were chosen to be fast enough to perform this kind of analysis thousands of times, and since the data was artificial, to know the underlying truth. The combination of these two properties allowed us to investigate meta qualities of the proposed tests such as the True Positive Rate (TPR) and False Positive Rate (FPR), which we used to evaluate the effectiveness of our tests.

Estimating distributions in two dimensions is difficult. We therefore first investigated several ways in which you could combine these tests into one-dimensional tests. With the exception of the likelihood ratio test, which is not applicable to the case of only one model, there is no obvious way of how to collapse two distinct test statistics into one dimension. Instead we decided to look at the one-dimensional distributions of the χ^2 -test and the Durbin-Watson test and see if their respective p-values could be combined into a useful metric. The considered combinations of p-values were the *min*, *max*, *mean*, and *product*. The *min* combination

corresponds to rejecting if either of the individual tests reject, whereas the *max* combination corresponds to rejecting only if both tests reject. The *mean* combination is a balance between these two extremes, while the *product* combination, could be thought of as the joint probability of the tests.

It turns out, however, that all these *simplistic combinations* of one-dimensional p-values are unsound. For a statistical test to be useful, the obtained p-value from the test needs to be consistent with the observed number of false positives. When we plotted the expected TPR, i.e. the p-values, against the actual observed FPR, we obtained satisfactory results only for the single tests (Figure 44, red triangles and blue circles). All combinations of two tests lay away from the identity line, which means that the tests were either too conservative (max, mean) or too liberal (min, product).

Figure 44. Simplistic combinations are unsound. Type I error rate plots for simplistic combinations of single test statistics. Type I error plots show the Expected False Positive Rate (EFPR), which is the stated number of erroneous rejections of a true model, versus the observed number of rejections, Observed False Positive Rate (OFPR). Only the single test statistics, and none of their combinations, lie along the identity line. The simplistic combinations were therefore disregarded from further analysis. Taken from Figure 4B in Paper I.

We next looked at two-dimensional combinations of tests statistics from one model. For this, we needed to develop a method that could take as input the two-dimensional distributions and the original data, and return a p-value as output, without being unsound. The method we developed (Figure 45, left) estimates a p-value from the two-dimensional distribution in two steps. In the first step we perform a density estimation of the cloud on a grid using a smooth Gaussian kernel to construct a two-dimensional histogram [141]. The volume of this density estimation is normalized to 1, and the p-value is then obtained by summing over all bins that has a density lower than the density at the point of the original data set (Figure 45, red box in step 3).

We show in the paper that this way of estimating p-values is *sound*, and does not lead to problems with consistency. That is, it is not too liberal or conservative (Paper I, Figure 5-6). Furthermore, when we look at the Receiver Operator Characteristic (ROC) curve, a two-dimensional test of χ^2 vs. DW outperforms both single tests (Figure 46A). In the

ROC curve, the FPR is plotted against the TPR, and for a test to be informative, i.e. being good at discriminating between a true and a false model, the resulting curve should lie away from the identify line and result in a large Area Under the Curve (AUC). In other words, the curve should lie above all the other curves to be considered the best. In Figure 46A, we see how the 2D combination of χ^2 vs. DW (green) lies above both the single tests.

Figure 45. Schemes for two-dimensional hypothesis testing using bootstrap distributions. (Left) Method using one model and two different test statistics (χ^2 and Durbin-Watson). (Right) Method using two models and one test (χ^2). The method steps are: (1) Initial fit to experimental data for the model(s). (2) Generate bootstrap samples from the model(s). (3) Generate clouds by fitting model(s) to the samples and calculate the test statistics (Left: χ^2 -DW, Right: χ^2 - χ^2). (4) Estimate the density distribution(s). Calculate p-values(s), equal to the portion below the cutoff plane(s). (5) Decision: Reject or not, based on calculated p-value(s). Possible outcomes are: (Left) Reject or do not reject the model; (Right) Reject either model, reject both models, or reject neither model. Taken from Figure 2 and Supplementary Figure S1 in Paper I.

Next, we looked at the combination of two χ^2 -costs from two different models (Figure 45, right). This second model being added to the picture can be an otherwise uninteresting model. We call

such a model a *help model*. In this scenario, we are not interested in the help model as an explanation of the original data, but only in how it interacts with the model being tested. In the terms of the method, a consequence is that we do not need to generate the secondary blue cloud in Figure 45, but only need to consider the green cloud. More interesting however, is the case when the second model is not a help model. For instance, the second model could be a competing model and its ability to describe the original data is then of equal interest as that of the first model. If this latter situation is the case, we do generate both clouds in Figure 45 and calculate the corresponding p-values for both models.

Figure 46. 2D combinations are informative. Receiver Operator Characteristic (ROC) Curves for: (A) One model and two tests. The 2D combination of χ^2 and Durbin-Watson tests (green) is more informative than the single tests by their own, as measured by their Area Under the Curve (AUC). (B) Combinations involving two models, 2D χ^2 vs. χ^2 (purple) and 1D likelihood ratio (LHR, orange) are most informative. For comparison, the green χ^2 -DW curve in B is the same as in (A). (C) 2D χ^2 -DW applied to the model $M_{i,c}$ from insulin signaling. $p = 0.07$. (D) 2D χ^2 vs. χ^2 using $M_{i,b}$ from insulin signaling as a secondary help model. $p = 0.0008$. Taken from Figures 5C, 6C, and 7C-D in Paper I.

In Figure 46B, the ROC curves for our tests involving two models are shown. As a comparison, the green χ^2 vs. DW curve from (A) is included. Our 2D χ^2 vs. χ^2 (purple) clearly outperforms this method. It turns out however that a 1D bootstrapped log-likelihood ratio test is the best test, of all we tested (orange). We showed that this is because all the information contained within the χ^2 vs. χ^2 cloud can be represented by projecting the cloud onto a rotated LHR axis (Paper I, Figure 7B), and that no information is contained within the sum of the likelihoods. This means that even though the same information should be contained also in the 2D plot, there are some inherent drawbacks of working with extra dimensions, such as the density estimation etc. These drawbacks results in the 2D analysis not performing as well as the 1D LHR. The 2D analysis is however more robust to poor choices of help models, and is also a useful visualization tool.

Importantly, these methods, although reminiscent of model selection methods, are actually model *rejection* methods, where we can have all four possible outcomes of rejecting neither, both, or either of the two models. This should be contrasted to the traditional likelihood ratio test that only has two outcomes: the rejection of the simpler of the two models, or no rejection. Additionally, we do not need to consider assumptions about nestedness and so on.

Lastly, we tested our sound methods on a real modeling example from insulin signaling. Here we revisited two models $\mathcal{M}_{i,c}$ and $\mathcal{M}_{i,b}$ from [34]. Importantly, we wanted to reevaluate the rejection of $\mathcal{M}_{i,c}$, which from a single χ^2 -test has a p-value ~ 0.06 . Here, we do not know the underlying truth, but due to subsequent experimental data, both $\mathcal{M}_{i,c}$ and $\mathcal{M}_{i,b}$ have been rejected in later analyses. We tested the 2D χ^2 vs. DW (Figure 46C), 2D χ^2 vs. χ^2 (Figure 46D), and 1D LHR (not shown) on $\mathcal{M}_{i,c}$, using $\mathcal{M}_{i,b}$ as help model. Specifically, using the 2D χ^2 vs. χ^2 and 1D LHR improved our confidence in the rejection of $\mathcal{M}_{i,c}$ ($p = 0.0008$ and ~ 0 respectively). This shows that our more advanced statistical tests moved a borderline rejection to a clear-cut case.

7

CONCLUDING REMARKS

In this chapter, I will recapitulate the main findings from my papers. I will then discuss how these results fit into the greater picture of mathematical modeling as it is applied within the fields of biology and medicine today, and provide some outlook on the future.

7.1 Summary of results and conclusions

A brief summary of the results and main conclusions from my papers now follow. Using the same order as in Chapter 6, Paper I will be discussed after Paper II-IV.

7.1.1 DN data should be analyzed using mathematical modeling

In Paper II, we develop a general method for how to interpret data from dominant negative inhibition experiments. We demonstrate that previously analyzed DN data are not conclusive evidence about kinase activity, and that in the absence of other proofs, S6 kinase still may be the kinase responsible for phosphorylating IRS1 on serine residue 307. Using model simulations and predictions with uncertainty analysis, we show that an alternative explanation is possible. This alternative explanation is centered around depletion of substrate, and we suggest different ways to test this experimentally. Our analysis illustrates the benefits of using mathematical modeling to fully understand the implications of biological data, and the potential pitfalls associated with not using it. Taken together, this demonstrates the appropriateness of involving modeling in both data analysis and experimental design, even for a small system with relatively simple data.

7.1.2 A single-cell modeling method for quantification of nuclear transport

In Paper III, we developed a method for determining molecular transport rates using FRAP and mechanistic mathematical modeling. This method is applicable to any system that can be investigated using FRAP. We determined the average YFP transport rate in yeast cells to be 0.15 s^{-1} and found a large cell-to-cell variation ($\text{CV} = 0.79$), which is a much larger variation than expected. We speculate that this variability manifests itself as different number of pore complexes and/or in their permeability. We finally estimated the average diffusion-available volume ratio to 4.77 ($\text{CV} = 0.20$).

7.1.3 Facilitation can be explained by a single mechanism

In Paper IV, we perform quantitative analysis of observed facilitation in pyramidal nerve cells from layer six in the neocortex from rats and mice. We investigate four known mechanisms that

can give rise to facilitation and show that only one mechanism can describe all data while not violating *a priori* knowledge. The surviving hypothesis and model ($\mathcal{M}_s$) is based on the priming of vesicles for fusion by *syt7*. We show that the $\mathcal{M}_s$ mechanism is sufficient for explaining all available data, and that a more complex model including the two best mechanisms tested ($\mathcal{M}_s$ and $\mathcal{M}_v$), does not improve the fit. Lastly, we show that $\mathcal{M}_s$ can satisfactorily fit to equivalent data from mice, and predict further validation data from mutants, and stimulation under a more physiological firing pattern. These results imply a new type of clarity regarding the precise mechanisms active in a specific cell type, as opposed to previous more qualitative modeling, only giving generic and non-specific insights.

7.1.4 A novel 2D bootstrap approach for hypothesis testing

In Paper I, we developed a novel method for combining different test statistics into joint two-dimensional distributions, and how to analyze these distributions in a sound manner. We show that the two-dimensional combination of the χ^2 -test and the Durbin-Watson test is an informative and useful test to perform in the case of a single model. We also show that any simplistic combinations of one-dimensional tests are unsound. We further show that if you can introduce a secondary help model to the test, a two-dimensional χ^2 vs. χ^2 test is even more informative, and that the one-dimensional bootstrapped log-likelihood ratio test is the most informative. This is because the one-dimensional test captures all the relevant information from the two-dimensional case, but does not suffer any of the negative consequence resulting from the added dimension. This gives a new motivation for the usage of likelihood ratio test in the case of non-nested models, and a fundamentally new usage as a model rejection tool, in addition to its previous usage as a model selection tool.

7.2 Relevancy of mathematical modeling

In Chapter 3, I presented the following list of useful applications of modeling:

1. Conceptual clarification by writing down verbal relations as rigorous equations
2. Highlighting gaps in knowledge
3. Independence of/from the modeled object
4. Zooming in time and space at your own discretion
5. Algorithms and computer programs can be used independently of the system
6. Modeling is cheap compared to experiments
7. Ethical reasons: models do not harm the environment, animals, or plants.
8. Replacing and/or assisting experimentation.
9. Precise formulation leads to easy generalization and visualization.
10. Enabling well-founded and testable predictions.

During the writing of this thesis, and the included papers, we have touched on almost all of these items in some way or another. I will now highlight a number of areas where I think modeling has a major role to play, and discuss these areas in the light of the above list and my main results.

7.2.1 Hypothesis testing

It should come as no surprise by now that mathematical modeling is a highly useful tool when it comes to hypothesis testing. After all, model-based hypothesis testing is the main thread throughout this thesis. If we look at the list above, the reason for this effectiveness is a combination of several, if not all, of the reasons given above.

In each of my papers, hypothesis testing has played some role. Paper I was entirely dedicated to developing methods to use for hypothesis testing, and we applied it to a previously borderline conclusion from insulin signaling. In Paper II, we looked at how underlying hypotheses plays into the aspect of data analysis. We showed how the lack of response in one protein was not conclusive of its involvement, but how, under certain circumstances, an alternative hypothesis was valid. In essence, we demonstrated a classic case of *absence of evidence is not evidence of absence*. In Paper III, we looked at more complex models for the nuclear transport in yeast, but rejected them due to lack of improvement. Finally, in Paper IV we analyzed four competing hypotheses, of which only one survived to the end.

7.2.2 Mechanistic understanding

Having a mechanistic model helps in achieving a mechanistic understanding of how the underlying system works. We can, through uncertainty analysis, learn about upper and lower boundaries for specific model properties. We can investigate states of the model that is inaccessible by current measurement techniques, and we can investigate the relative importance of various parts of the model. Sometimes the explanation is revealed first after the model is fitted to the data. That is, if a model is complex enough, it may fit to the data in some qualitatively new or unpredictable way that was not expected. This is to be contrasted to the case where one tries to build a model around a preconceived idea of how a mechanism might work. A good example is how the S6K-model in Paper II managed to fit to the DN-data by placing one of the proteins near depletion. This was an explanation that we had not considered beforehand, that we realized because of the modeling analysis.

7.2.3 Design of experiments

Modeling has an important role to play in the design of experimental setups. Almost always there are some trade-offs to consider when one is doing experiment. You may be limited fiscally, or by time. You can be limited in the number of samples you can get access to, or when and where to measure. All of these factors make choosing exactly which experiment to perform an important task. If we also consider ethical considerations, such as reducing the number of test animals and so on, the consequences reach even wider and further. Modeling can help in the design of experiments in several ways. We can use it to try to maximize or minimize any of the above factors. Furthermore, we can use it to help suggest the best experiment to perform for discriminating between rival hypothesis. Mathematical modeling can also equally well help in deciding where and when to measure to reduce uncertainty about specific parameters or model-based biomarkers [9].

7.2.4 Data analysis

Modeling is a useful tool as part of the initial data analysis. In Paper II, we showed how a simple mathematical model was able to fit to data it at first glance intuitively should not have been able to explain. This reveals the power of mathematical modeling to lay bare assumptions made in interpreting relevant data, and highlight what conclusions, if any, can be drawn directly from the data, and what conclusions that are unsubstantiated, or need corroborative evidence to be supported.

7.2.5 Healthcare

One of the brightest future prospects of modeling will be its application to personalized medicine. In Paper III, we showed how you can use the model to acquire relevant biomarkers, in this case parameter values, for individual specimens. In *Forsgren et. al 2014* [9], a similar approach is applied to retrieve highly informative clinical parameters from patients with liver damage. In this way, modeling will be able to play an integral role as a diagnostic tool. Also for treatments in the clinic there are several applications of modeling. One example is from the intensive care unit (ICU), where modeling has potential to help in glucose control, i.e. how much glucose and insulin to give to patients, with potential to reduce mortality rate during a patient's hospital stay.

Modeling of course also underlies all of the pharmaceutical industry, where Pharmacokinetic and Pharmacodynamic (PKPD) modeling is crucial for the development of new drugs, and to assess the optimal dosage and treatment levels. Recent developments also include more types of mechanistic modeling that has the potential to revolutionize the industry. As better and better models become available of the various systems in the body, these models will start to substitute part of the certification process of new medical devices and pharmaceutical drugs. As an example, the American Food and Drug Administration (FDA) accepted a couple of years ago the replacement of animal trials in the preclinical testing of insulin pump algorithms for closed-loop control strategies with testing in *in-silico* [142]. These kinds of advancements have significantly reduced the prize, as measured in money, time, and animal lives, for developing new pumps and strategies, and is a likely trajectory also for pharmaceutical industry.

7.3 Outlook

“We live on an island surrounded by a sea of ignorance. As our island of knowledge grows, so does the shore of our ignorance.”

John Archibald Wheeler. Scientific American Vol. 267, 1992.

Today, more and more scientists strengthen their experimental findings within biology and medicine with mathematical modeling and other analysis. This leads to scientists from different domains working together within multi-disciplinary fields such as systems biology, bioinformatics, biotechnology, biomedical engineering, and so on. These multi-disciplinary

groups and institutions are a veritable melting pot of new ideas, and new combinations of old ideas, where the different competences can complement each other and lead to both new answers, but also to new questions. It is my prediction that this trend will continue long into the future and continue to be a successful strategy for tackling an expanding array of problems in the modern world.

ACKNOWLEDGEMENTS

Many people have contributed to this thesis, in one way or another. I cannot possibly thank all of you, but I can try...

Gunnar Cedersund, my main supervisor. We are very alike in certain aspects, and completely different in others. It has been a fun ride working in your group. I have so many memories from conferences, trips, students and projects. We could probably reminisce about them for hours, but in particular, our first trip to San Francisco and into the blizzard in the German mountains stand out. Hopefully we can continue to collaborate in the future!

My co-supervisors **Tomas Strömberg** and **Peter Strålfors**. Thank you, Tomas, for helping me through the harder times and giving me the support I needed. Thank you, Peter, for the scientific discussions we have had over the years.

Thank you, all the people, in the systems biology group. **Elin Nyman**, you are a rock! We have had so much fun together. Good luck with your continued research, I know you will do great things! **Karin Lundengård**, thank you for allowing me to bounce my frustration on you from time to time. I am sure you will nail your thesis, because after all, who's awesome? **William Lövfors**, always there with unbridled enthusiasm. Thank you for many interesting discussions on solutions and tips and tricks, and for letting me win in all the trivia games we played. Thank you also to **Mikael Forsgren**, **Markus Karlsson**, and **Sebastian Sten** for all the fun times over the years. I would also like to thank all the wonderful students that has come and gone in our group over the years: **Robert Palmér**, **Eva-Maria Ekstrand**, **Oscar Samuelsson**, **Amanda Jonsson**, **David "Julle" Bergenholm**, **Linnea Bergenholm**, **David Janzén**, **Pelle Lundberg**, **Mattias Köpsén**, **Rasmus Magnusson**, **Andrea Hjelm**, **Hao Li**, and **Tilda Herrgårdh**, just to name a few.

My PhD-student colleagues at IMT. **Peter Rejmstad**, for clearing a safe path for me to follow in. Thank you for all advice over the years, and good luck with your new job. **Maria Ewerlöf** for your dedication to improving the work situation for all us PhD-students. You take on too many responsibilities! Thank you also for all the gaming nights and social activities. A big thank you also goes out to **Tobias Romu**, **Anette Karlsson**, **Olivier Cros**, **Malcolm Latorre**, **Martin Hultman**, and all other PhD-students at IMT with whom I crossed paths.

Thank you, all other colleagues, at IMT for a very happy and inviting work place. Thank you **Göran Salerud** for supporting me, and for insisting I use your coffee cup. Thank you, **Christine Ericsson**, for all the help. I always know when you are in the fika-room cause your laughter echoes down the halls. Thank **Bengt Ragnemalm** for all the IT-related help, and for the nice gaming nights we've had. Thank you also **Anna Holm**, **Jenny Eilertsen**, **Ann-Marie Simonsson**, **Karoline Waltelius**, **Martin Eneling**, **Therese Götberg**, and **Jörgen Jonsson** for helping me whenever I needed to wrestle with the bureaucracy or other problems. Thank you **Marcus Larsson** and **Daniel Karlsson** for your support and follow up during my studies.

Thank you, all old colleagues, back at floor 12 at IKE. **Cecilia Brännmark**, you were a pillar of support to lean on during my time at IKE! **Siri Fagerholm**, it's a shame our PKC project never bore any fruit. Thank you **Meenu Rajan**, **Åsa Jufvas**, **Cecilia Jönsson**, **Anita Öst**, **Tobias Strid**, **Sebastian Shultz**, **Mats Söderström**, and **Martin Öst**.

To all my friends near and far, a big thank you. **Mikael Mide**, still going strong after all these years. Hold the fort! **Oscar Danielsson**, always a fighter to the *bitter* end. Thank you also everyone else who has allowed me to take my mind off work when I have needed it the most.

To my family. **Mamma Lilian** and **Pappa Roger**: even though home is many hours away, it is always great to come back there. Thank you for all the support over the years, and help with real life issues. **Storebror Björn**, always there to put me in my place. To **Hannah** and to **Lo**, the latest addition to the family: Welcome! Thank you **Faster Ingela** for all the help during my stay in Linköping, and for helping me plan the big day. Thank you, all other members of my extended family. I have it on good scientific authority (me), that you are the best family! Christmas is always a blast!

Lastly, a big thank you to everyone joining me on the big day. Tonight, we celebrate!

*Do not go gentle into that good night,
Old age should burn and rave at close of day;
Rage, rage against the dying of the light.*
- Dylan Tomas

Rikard Johansson

Linköping, 2017

REFERENCES

1. Oerter R. *The Theory of Almost Everything: The Standard Model, the Unsung Triumph of Modern Physics*. Penguin Publishing Group; 2006.
2. Kholodenko B, Yaffe MB, Kolch W. Computational approaches for analyzing information flow in biological networks. *Sci Signal*. 2012 Apr 17;5(220):re1.
3. Nerlich B, Dingwall R, Clarke DD. The Book of Life: How the Completion of the Human Genome Project was Revealed to the Public. *Health*. 2002;6(4):445–69.
4. Winkler H. *Verbreitung und Ursache der Parthenogenesis im Pflanzen- und Tierreiche / von Dr. Hans Winkler*. Jena : G. Fischer,; 1920. 1-248 p.
5. Yadav SP. The wholeness in suffix -omics, -omes, and the word om. *J Biomol Tech*. 2007 Dec;18(5):277.
6. Ganapathiraju MK, Thahir M, Handen A, Sarkar SN, Sweet RA, Nimgaonkar VL, et al. Schizophrenia interactome with 504 novel protein–protein interactions. *Nat Publ Gr*. 2016;212.
7. Lu Y-F, Goldstein DB, Angrist M, Cavalleri G. Personalized medicine and human genetic diversity. *Cold Spring Harb Perspect Med*. 2014 Jul 24;4(9):a008581.
8. The Academy of Medical Sciences. Stratified, personalised or P4 medicine: a new direction for placing the patient at the centre of healthcare and health education. *Meet Rep*. 2015;(May).
9. Forsgren MF, Dahlqvist Leinhard O, Dahlström N, Cedersund G, Lundberg P. Physiologically realistic and validated mathematical liver model reveals hepatobiliary transfer rates for Gd-EOB-DTPA using human DCE-MRI data. *PLoS One*. 2014;9(4):e95700.
10. Kitano H. Computational systems biology. *Nature*. 2002 Nov 14;420(6912):206–10.
11. Chalmers AF (Alan F. What is this thing called science? 3rd ed. University of Queensland; 1999. 266 p.
12. Popper K. *The Logic of Scientific Discovery*. Taylor & Francis; 2005. (Routledge Classics).
13. Will CM. The Confrontation between General Relativity and Experiment. *Living Rev Relativ*. 2006 Dec 27;9(1):3.
14. Kuhn TS, Hacking I. *The Structure of Scientific Revolutions: 50th Anniversary Edition*. University of Chicago Press; 2012.
15. Feyerabend P. *Against Method*. Verso; 1993.
16. Einstein A. Physics and reality. *J Franklin Inst*. 1936 Mar;221(3):349–82.
17. Wigner EP. The unreasonable effectiveness of mathematics in the natural sciences. Richard courant lecture in mathematical sciences delivered at New York University, May 11, 1959. *Commun Pure Appl Math*. 1960;13(1):1–14.
18. Kaplan A. *The conduct of inquiry*. Transaction Publishers; 1973.

19. Maslow AH. The psychology of science: a reconnaissance. Harper & Row; 1966. (The John Dewey Society lectureship series).
20. Atkins PW. Creation Revisited. W.H. Freeman; 1992.
21. Tegmark M. The Mathematical Universe. *Found Phys.* 2008 Feb 8;38(2):101–50.
22. Gow M. Measuring the Earth: Eratosthenes and His Celestial Geometry. Enslow; 2009. (Great Minds of Ancient Science and Math).
23. Hughes DW, Fowler PH, Lovell B, Lynden-Bell D, Message PJ, Wilkinson JE. The History of Halley’s Comet [and Discussion]. *Philos Trans R Soc London A Math Phys Eng Sci.* 1987;323(1572).
24. Box GEP, Draper NR. Empirical model-building and response surfaces. New York : Wiley, cop. 1987; 1987. (Wiley series in probability and mathematical statistics).
25. Klipp E. Systems biology : a textbook. Weinheim : Wiley-VCH ; Chichester : John Wiley [distributor], 2009; 2009.
26. Morris MK, Saez-Rodriguez J, Sorger PK, Lauffenburger DA. Logic-Based Models for the Analysis of Cell Signaling Networks. *Biochemistry.* 2010 Apr 20;49(15):3216–24.
27. Wang R-S, Saadatpour A, Albert R. Boolean modeling in systems biology: an overview of methodology and applications. *Phys Biol.* 2012 Oct 1;9(5):55001.
28. de Jong H. Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. *J Comput Biol.* 2002 Jan;9(1):67–103.
29. Hübner K, Sahle S, Kummer U. Applications and trends in systems biology in biochemistry. *FEBS J.* 2011 Aug;278(16):2767–857.
30. Verhulst PF. Notice sur la loi que la population poursuit dans son accroissement. Impr. d’H. Vandekerckhove; 1837. 113-121 p. (Correspondance mathématique et physique; vol. 10).
31. Hodgkin AL, Huxley AF. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol.* 1952 Aug;117(4):500–44.
32. Heinrich R, Schuster S. The Regulation of Cellular Systems. Springer US; 2012.
33. Maier C, Loos C, Hasenauer J. Robust parameter estimation for dynamical systems from outlier-corrupted data. *Bioinformatics.* 2017 Jan 6;9(5):btw703.
34. Cedersund G, Roll J, Ulfhielm E, Danielsson A, Tidefelt H, Strålfors P. Model-Based Hypothesis Testing of Key Mechanisms in Initial Phase of Insulin Signaling. Hunter P, editor. *PLoS Comput Biol.* 2008 Jun 20;4(6):e1000096.
35. Johansson R, Strålfors P, Cedersund G. Combining test statistics and models in bootstrapped model rejection: it is a balancing act. *BMC Syst Biol.* 2014;8.
36. Mangan S, Zaslaver A, Alon U. The Coherent Feedforward Loop Serves as a Sign-sensitive Delay Element in Transcription Networks. *J Mol Biol.* 2003;334(2):197–204.
37. Wilkinson GN, Rogers CE. Symbolic Description of Factorial Models for Analysis of Variance. *J R Stat Soc Ser C (Applied Stat.* 1973;22(3):392–9.
38. Naegle KM, Welsch RE, Yaffe MB, White FM, Lauffenburger DA. MCAM: Multiple

-
- Clustering Analysis Methodology for Deriving Hypotheses and Insights from High-Throughput Proteomic Datasets. Papin JA, editor. PLoS Comput Biol. 2011 Jul 21;7(7):e1002119.
39. Janes KA, Yaffe MB. Data-driven modelling of signal-transduction networks. *Nat Rev Mol Cell Biol.* 2006 Nov;7(11):820–8.
 40. Halasz M, Kholodenko BN, Kolch W, Santra T. Integrating network reconstruction with mechanistic modeling to predict cancer therapies. *Sci Signal.* 2016;9(455):ra114.
 41. Choi S. Systems biology for signaling networks. Springer; 2010.
 42. Rangamani P, Iyengar R. Modelling spatio-temporal interactions within the cell. *J Biosci.* 2007 Jan;32(1):157–67.
 43. Nielsen J, Hohmann S. Systems biology. Wiley; 2017.
 44. Vanegas-Acosta JC, Garzón-Alvarado DA. Biological modelling and computational implementation using the finite elements method. *Comput Appl Math.* 2014 Oct 16;33(3):621–40.
 45. Wood DT, Kojouharov H V., Dimitrov DT. Universal approaches to approximate biological systems with nonstandard finite difference methods. *Math Comput Simul.* 2017;133:337–50.
 46. Székely T, Burrage K. Stochastic simulation in systems biology. *Comput Struct Biotechnol J.* 2014;12(20):14–25.
 47. Raj A, van Oudenaarden A, Ren X, Lao K, Xie XS, Kingsman AJ, et al. Nature, nurture, or chance: stochastic gene expression and its consequences. *Cell.* 2008 Oct 17;135(2):216–26.
 48. Gillespie DT. Exact stochastic simulation of coupled chemical reactions. *J Phys Chem.* 1977 Dec;81(25):2340–61.
 49. Wilkinson DJ. Stochastic modelling for quantitative description of heterogeneous biological systems. *Nat Rev Genet.* 2009 Feb;10(2):122–33.
 50. Covert MW. Fundamentals of Systems Biology: From Synthetic Circuits to Whole-cell Models. CRC Press; 2014.
 51. Julleson D, Johansson R, Rajan MR, Strålfors P, Cedersund G. Dominant negative inhibition data should be analyzed using mathematical modeling - Re-interpreting data from insulin signaling. *FEBS J.* 2015;282(4):788–802.
 52. Cedersund G, Roll J. Systems biology: model based evaluation and comparison of potential explanations for given biological data. *FEBS J.* 2009 Feb;276(4):903–22.
 53. Brännmark C, Palmér R, Glad ST, Cedersund G, Strålfors P. Mass and information feedbacks through receptor endocytosis govern insulin signaling as revealed using a parameter-free modeling framework. *J Biol Chem.* 2010 Jun 25;285(26):20171–9.
 54. Kiselyov V V, Verstehey S, Gauguin L, De Meyts P. Harmonic oscillator model of the insulin and IGF1 receptors' allosteric binding and activation. *Mol Syst Biol.* 2009;5:243.
 55. Weigert C, Kron M, Kalbacher H, Pohl AK, Runge H, Häring H-U, et al. Interplay and Effects of Temporal Changes in the Phosphorylation State of Serine-302, -307, and -318

- of Insulin Receptor Substrate-1 on Insulin Action in Skeletal Muscle Cells. *Mol Endocrinol*. 2008;22(12):2729–40.
56. Raue A, Schilling M, Bachmann J, Matteson A, Schelke M, Kaschek D, et al. Lessons Learned from Quantitative Dynamical Modeling in Systems Biology. Hernandez-Lemus E, editor. *PLoS One*. 2013 Sep 30;8(9):e74335.
 57. Bluman G, Anco S. Symmetry and Integration Methods for Differential Equations. Springer New York; 2008. (Applied Mathematical Sciences).
 58. Hairer E, Nørsett SP, Wanner G. Solving Ordinary Differential Equations I: Nonstiff Problems. Springer Berlin Heidelberg; 2008. (Springer Series in Computational Mathematics).
 59. Fehlberg E. Klassische Runge-Kutta-Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Wärmeleitungsprobleme. *Computing*. 1970 Mar;6(1–2):61–71.
 60. Shampine LF. Numerical Solution of Ordinary Differential Equations. Taylor & Francis; 1994. (Chapman & Hall mathematics).
 61. Ghosh S, Matsuoka Y, Asai Y, Hsin K-Y, Kitano H. Software for systems biology: from tools to integrated platforms. *Nat Rev Genet*. 2011 Dec;12(12):821–32.
 62. Schmidt H, Jirstrand M. Systems Biology Toolbox for MATLAB: a computational platform for research in systems biology. *Bioinformatics*. 2006 Feb 15;22(4):514–5.
 63. Schmidt H. SBaddon: high performance simulation for the Systems Biology Toolbox for MATLAB. *Bioinformatics*. 2007 Mar 1;23(5):646–7.
 64. Nocedal J, Wright S. Numerical Optimization. Springer New York; 2006. (Springer Series in Operations Research and Financial Engineering).
 65. Leis JR, Kramer MA. The simultaneous solution and sensitivity analysis of systems described by ordinary differential equations. *ACM Trans Math Softw*. 1988 Mar 1;14(1):45–60.
 66. Nelder JA, Mead R. A Simplex Method for Function Minimization. *Comput J*. 1965 Jan 1;7(4):308–13.
 67. Bianchi L, Dorigo M, Gambardella LM, Gutjahr WJ. A survey on metaheuristics for stochastic combinatorial optimization. *Nat Comput*. 2009 Jun 19;8(2):239–87.
 68. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by Simulated Annealing. *Science* (80-). 1983;220(4598).
 69. Glover F. Heuristics for Integer Programming Using Surrogate Constraints. *Decis Sci*. 1977 Jan;8(1):156–66.
 70. Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of ICNN'95 - International Conference on Neural Networks*. IEEE; 1995. p. 1942–8.
 71. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of State Calculations by Fast Computing Machines. *J Chem Phys*. 1953 Jun;21(6):1087–92.
 72. Hastings WK. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*. 1970;57(1):97–109.

-
73. Gelman A, Carlin JB, Stern HS, Rubin DB. Bayesian Data Analysis, Second Edition. Taylor & Francis; 2003. (Chapman & Hall/CRC Texts in Statistical Science).
 74. Sharma AV. Review of Evolutionary Optimization Algorithms for Test Case Minimization. *Int J Eng Comput Sci*. 2015;4(4).
 75. Sarker R, Kamruzzaman J, Newton C. Evolutionary Optimization (EvOpt): A Brief Review and Analysis. *Int J Comput Intell Appl*. 2003 Dec;3(4):311–30.
 76. Prasada Tripathy S, Kanhar D. Optimization of Software Testing for Discrete Testsuite using Genetic Algorithm and Sampling Technique. *Int J Comput Appl*. 2013 Feb 15;63(7):1–5.
 77. Martí R, Laguna M, Glover F. Principles of scatter search. *Eur J Oper Res*. 2006 Mar;169(2):359–72.
 78. Bonyadi MR, Michalewicz Z. Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review. *Evol Comput*. 2017 Mar;25(1):1–54.
 79. Shi Y, Eberhart R. A modified particle swarm optimizer. In: 1998 IEEE International Conference on Evolutionary Computation Proceedings IEEE World Congress on Computational Intelligence (Cat No98TH8360). IEEE; p. 69–73.
 80. Box GEP, Hunter JS, Hunter WG. Statistics for experimenters: design, innovation, and discovery. Wiley-Interscience; 2005. (Wiley series in probability and statistics).
 81. Ljung L. System Identification: Theory for the User. Pearson Education; 1998.
 82. Seber GAF, Wild CJ. Nonlinear Regression. Wiley; 2003. (Wiley Series in Probability and Statistics).
 83. Maiwald T, Timmer J, Kreutz C, Klingmüller U, Raue A. Addressing parameter identifiability by model-based experimentation. *IET Syst Biol*. 2011 Mar 1;5(2):120–30.
 84. Raue A, Kreutz C, Maiwald T, Bachmann J, Schilling M, Klingmüller U, et al. Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*. 2009 Aug 1;25(15):1923–9.
 85. Kreutz C, Raue A, Kaschek D, Timmer J. Profile likelihood in systems biology. *FEBS J*. 2013 Jun;280(11):2564–71.
 86. Nyman E, Lindgren I, Lövfors W, Lundengård K, Cervin I, Sjöström TA, et al. Mathematical modeling improves EC 50 estimations from classical dose-response curves. *FEBS J*. 2015 Mar;282(5):951–62.
 87. Kreutz C, Raue A, Timmer J. Likelihood based observability analysis and confidence intervals for predictions of dynamic models. *BMC Syst Biol*. 2012 Sep 5;6:120.
 88. Cedersund G. Conclusions via unique predictions obtained despite unidentifiability - New definitions and a general method. *FEBS J*. 2012;279(18):3513–27.
 89. Mélykúti B, August E, Papachristodoulou A, El-Samad H. Discriminating between rival biochemical network models: three approaches to optimal experiment design. *BMC Syst Biol*. 2010 Apr 1;4(1):38.
 90. Roberts MAJ, August E, Hamadeh A, Maini PK, McSharry PE, Armitage JP, et al. A model invalidation-based approach for elucidating biological signalling pathways,

- applied to the chemotaxis pathway in *R. sphaeroides*. *BMC Syst Biol*. 2009 Oct 31;3(1):105.
91. Apgar JF, Toettcher JE, Endy D, White FM, Tidor B. Stimulus Design for Model Selection and Validation in Cell Signaling. *PLoS Comput Biol*. 2008 Feb;4(2):e30.
 92. Akaike H. A new look at the statistical model identification. *IEEE Trans Automat Contr*. 1974 Dec;19(6):716–23.
 93. Akaike H. Modern development of statistical methods. In: *Trends and Progress in System Identification* (P Eykoff, Ed) Pergamon. New York: Press; 1981.
 94. Schwarz G. Estimating the Dimension of a Model. *Ann Stat*. 1978;6(2):461–4.
 95. Neyman J, Pearson ES. On the Use and Interpretation of Certain Test Criteria for Purposes of Statistical Inference: Part I. *Biometrika*. 1928;20A(1–2):175–240.
 96. Vuong QH, Vuong, H Q. Likelihood Ratio Tests for Model Selection and Non-nested Hypotheses. *Econometrica*. 1989;57(2):307–33.
 97. Wilks SS. The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses. *Ann Math Stat*. 1938 Mar;9(1):60–2.
 98. Cox DR. Tests of separate families of hypotheses. *Proc 4th Berkeley Symp Math Stat Probab* 1, 105–123 (1961). 1961;
 99. Cox DR. Further Results on Tests of Separate Families of Hypotheses. *J R Stat Soc Ser B*. 1962;24(2):406–24.
 100. Sheskin DJ. *Handbook of Parametric and Nonparametric Statistical Procedures*, Fifth Edition. London: Chapman & Hall/CRC; 2011. (A Chapman & Hall book).
 101. Chernoff H. On the Distribution of the Likelihood Ratio. *Ann Math Stat*. 1954 Sep;25(3):573–87.
 102. Chant D. On asymptotic tests of composite hypotheses in nonstandard conditions. *Biometrika*. 1974;61(2):291–8.
 103. Miller JJ. Asymptotic Properties of Maximum Likelihood Estimates in the Mixed Model of the Analysis of Variance. *Ann Stat*. 1977;5(4):746–62.
 104. Shapiro A. Asymptotic distribution of test statistics in the analysis of moment structures under inequality constraints. *Biometrika*. 1985;72(1):133–44.
 105. Self SG, Liang K-Y. Asymptotic Properties of Maximum Likelihood Estimators and Likelihood Ratio Tests under Nonstandard Conditions. *J Am Stat Assoc*. 1987;82(398):605–10.
 106. Kanji GK. *100 Statistical Tests*. Thousand Oaks, California, US: SAGE Publications; 2006. (100 Statistical Tests).
 107. Williams DA. Discrimination between regression models to determine the pattern of enzyme synthesis in synchronous cell cultures. *Biometrics*. 1970 Mar;26:23–32.
 108. Efron B. Bootstrap Methods: Another Look at the Jackknife. *Ann Stat*. 1979;7(1):1–26.
 109. Efron B. *The Jackknife, the Bootstrap, and Other Resampling Plans* (CBMS-NSF Regional Conference Series in Applied Mathematics). Montpelier, Vermont, USA:

-
- Society for Industrial Mathematics; 1987.
110. Müller TG, Faller D, Timmer J, Swameye I, Sandra O, Klingmüller U. Tests for cycling in a signalling pathway. *Appl Stat.* 2004;53(4):557–8.
 111. Kerr MK, Churchill GA. Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments. *Proc Natl Acad Sci USA.* 2001 Jul;98(16):8961–5.
 112. Kirk PD, Stumpf MP. Gaussian process regression bootstrapping: exploring the effects of uncertainty in time course data. *Bioinformatics.* 2009;25(10):1300–6.
 113. Felsenstein J. Confidence Limits on Phylogenies: An Approach Using the Bootstrap. *Evolution (N Y).* 1985;39(4):783–91.
 114. Efron B, Halloran E, Holmes S. Bootstrap confidence levels for phylogenetic trees. *Proc Natl Acad Sci USA.* 1996 Jul;93(14):7085–90.
 115. Lanfear R, Bromham L. Statistical tests between competing hypotheses of Hox cluster evolution. *Syst Biol.* 2008;57(5):708–18.
 116. Wagenmakers EJ, Ratcliff R, Gomez P, Iverson GJ. Assessing model mimicry using the parametric bootstrap. *J Math Psychol.* 2004 Feb;48:28–50.
 117. Timmer J, Müller TG, Swameye I, Sandra O, Klingmüller U. Modeling the nonlinear dynamics of cellular signal transduction. *Int J Bifurc Chaos.* 2004 Jun;14(6):2069–79.
 118. Hinde J. Choosing between nonnested models: a simulation approach. In: *Advances in GLIM and Statistical Modelling Proceedings of the Glim92 Conference.* Munich, Germany: (Fahrmeir L et al., eds). Springer-Verlag; 1992.
 119. Scully T. Diabetes in numbers. *Nature.* 2012 May 16;485(7398):S2–3.
 120. Herman MA, Kahn BB. Glucose transport and sensing in the maintenance of glucose homeostasis and metabolic harmony. *J Clin Invest.* 2006 Jul 3;116(7):1767–75.
 121. Smith U. Impaired ('diabetic') insulin signaling and action occur in fat cells long before glucose intolerance--is insulin resistance initiated in the adipose tissue? *Int J Obes Relat Metab Disord.* 2002 Jul;26(7):897–904.
 122. Magnusson R, Gustafsson M, Cedersund G, Strålfors P, Nyman E. Cross-talks via mTORC2 can explain enhanced activation in response to insulin in diabetic patients. *Biosci Rep.* 2017;37(1).
 123. Nyman E, Brännmark C, Palmér R, Brugård J, Nyström FH, Strålfors P, et al. A Hierarchical Whole-body Modeling Approach Elucidates the Link between *in Vitro* Insulin Signaling and *in Vivo* Glucose Homeostasis. *J Biol Chem.* 2011 Jul 22;286(29):26028–41.
 124. Nyman E, Cedersund G, Strålfors P. Insulin signaling – mathematical modeling comes of age. *Trends Endocrinol Metab.* 2012 Mar;23(3):107–15.
 125. Brännmark C, Nyman E, Fagerholm S, Bergenholm L, Ekstrand E-M, Cedersund G, et al. Insulin signaling in type 2 diabetes: experimental and modeling analyses reveal mechanisms of insulin resistance in human adipocytes. *J Biol Chem.* 2013 Apr 5;288(14):9867–80.

126. Nyman E, Rajan MR, Fagerholm S, Brännmark C, Cedersund G, Strålfors P. A Single Mechanism Can Explain Network-wide Insulin Resistance in Adipocytes from Obese Patients with Type 2 Diabetes. *J Biol Chem*. 2014 Nov 28;289(48):33215–30.
127. Rajan MR, Fagerholm S, Jönsson C, Kjølhede P, Turkina M V., Strålfors P. Phosphorylation of IRS1 at Serine 307 in Response to Insulin in Human Adipocytes Is Not Likely to be Catalyzed by p70 Ribosomal S6 Kinase. Kanzaki M, editor. *PLoS One*. 2013 Apr 2;8(4):e59725.
128. Goffeau A, Barrell BG, Bussey H, Davis RW, Dujon B, Feldmann H, et al. Life with 6000 Genes. *Science* (80-). 1996;274(5287).
129. Pan B, Zucker RS. A General Model of Synaptic Transmission and Short-Term Plasticity. *Neuron*. 2009;62(4):539–54.
130. Sherman SM, Guillery RW. On the actions that one nerve cell can have on another: distinguishing “drivers” from “modulators.” *Proc Natl Acad Sci U S A*. 1998 Jun 9;95(12):7121–6.
131. Mongillo G, Barak O, Tsodyks M. Synaptic Theory of Working Memory. *Science* (80-). 2008;319(5869).
132. Wróbel A, Hedström A, Lindström S. Synaptic excitation of principal cells in the cat’s lateral geniculate nucleus during focal epileptic seizures in the visual cortex. *Acta Neurobiol Exp (Wars)*. 1998;58(4):271–6.
133. Granseth B, Ahlstrand E, Lindström S. Paired pulse facilitation of corticogeniculate EPSCs in the dorsal lateral geniculate nucleus of the rat investigated in vitro. *J Physiol*. 2002 Oct 15;544(Pt 2):477–86.
134. Matveev V, Bertram R, Sherman A. Residual bound Ca²⁺ can account for the effects of Ca²⁺ buffers on synaptic facilitation. *J Neurophysiol*. 2006;96(6):3389–97.
135. Rozov A, Burnashev N, Sakmann B, Neher E. Transmitter release modulation by intracellular Ca²⁺ buffers in facilitating and depressing nerve terminals of pyramidal cells in layer 2/3 of the rat neocortex indicates a target cell-specific difference in presynaptic calcium dynamics. *J Physiol*. 2001 Mar 15;531(Pt 3):807–26.
136. Jackman SL, Turecek J, Belinsky JE, Regehr WG. The calcium sensor synaptotagmin 7 is required for synaptic facilitation. *Nature*. 2016 Jan 6;529(7584):88–91.
137. Armour SM, Baur JA, Hsieh SN, Land-Bracha A, Thomas SM, Sinclair DA. Inhibition of mammalian S6 kinase by resveratrol suppresses autophagy. *Aging (Albany NY)*. 2009 Jun 3;1(6):515–28.
138. Bae EJ, Yang YM, Kim SG. Abrogation of hyperosmotic impairment of insulin signaling by a novel class of 1,2-dithiole-3-thiones through the inhibition of S6K1 activation. *Mol Pharmacol*. 2008 May 1;73(5):1502–12.
139. Poon IKH, Jans DA. Regulation of Nuclear Transport: Central Role in Development and Transformation? *Traffic*. 2005 Mar;6(3):173–86.
140. Chu CT, Plowey ED, Wang Y, Patel V, Jordan-Sciutto KL. Location, Location, Location. *J Neuropathol Exp Neurol*. 2007 Oct;66(10):873–83.
141. Silverman BW. Density Estimation for Statistics and Data Analysis. London: Chapman and Hall; 1986. (Monographs on applied probability and statistics).

142. Kovatchev BP, Breton M, Man CD, Cobelli C. In silico preclinical trials: a proof of concept in closed-loop control of type 1 diabetes. *J Diabetes Sci Technol*. 2009 Jan;3(1):44–55.

ENDNOTES

Below follow some details on endnotes used throughout this work. These are mostly additional copyright details, links to non-peer-reviewed sources, and other miscellaneous items.

-
- i Figure 1 (top): Chakazul. Redistributed under [CC BY-SA 4.0](#). Available at: https://upload.wikimedia.org/wikipedia/commons/6/6e/Metabolic_Metro_Map.svg
- ii Figure 1 (bottom): Narayanese, WikiUserPedia, YassineMrabet, TotoBaggins. [CC BY-SA 3.0](#). Available At: https://upload.wikimedia.org/wikipedia/commons/0/0b/Citric_acid_cycle_with_aconitate_2.svg
- iii Figure 2: Public domain. Available at: <https://upload.wikimedia.org/wikipedia/commons/e/eb/Molbio-Header-2.svg>
- iv Figure 3: Madhavi K Ganapathiraju. Redistributed under [CC BY-SA 4.0](#). Available at: https://upload.wikimedia.org/wikipedia/commons/1/1d/Schizophrenia_PPI.jpg
- v Figure 4: Wetterstrand KA. DNA Sequencing Costs:
Data from the NHGRI Genome Sequencing Program (GSP)
Available at: www.genome.gov/sequencingcostsdata. Accessed November 28th, 2016.
- vi p. 9: De revolutionist orbium coelestium (On the Revolutions of the Celestial Spheres), 1543.
- vii p. 9: Philosophie Naturalis Principia Mathematica (Mathematical Principles of Natural Philosophy), 1687.
- viii p. 38: The Systems Biology Toolbox 2. Accessed on June 29th, 2017.
Available at: <http://www.sbtoolbox2.org>.
- ix p. 38: Shampine LF, Reichelt MW. THE MATLAB ODE SUITE. Available at: https://www.mathworks.com/help/pdf_doc/otherdocs/ode_suite.pdf. Accessed June 29th, 2017.
- x p. 38: CVODE. SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers. Available at: <https://computation.llnl.gov/projects/sundials/cvode>. Accessed on July 5th, 2017.
- xi Figure 21: Public domain. Available at: https://upload.wikimedia.org/wikipedia/commons/d/da/Newton_optimization_vs_grad_descent.svg
- xii p. 48: Called *simannealingSBAO* in Systems Biology Toolbox 1.
- xiii Figure 35: Thomas Splettstoesser. Redistributed under [CC BY-SA 4.0](#). Available at: https://upload.wikimedia.org/wikipedia/commons/d/d7/SynapseSchematic_lines.svg