Parameter and Topology Uncertainty for Optimal Experimental Design

by

David Robert Hagen

B.S. in Biochemistry, B.A. in Economics, and Minor in Chemistry,
Case Western Reserve University (2008)

Submitted to the Department of Biological Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

Author . . . .
Department of Biological Engineering
March 31, 2014

Certified by . . . .
Bruce Tidor
Professor of Biological Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Forest M. White
Chairman, Department Committee on Graduate Theses

Thesis Committee

Accepted by . . . .

Forest M. White
Associate Professor of Biological Engineering
Chairman of Thesis Committee

Accepted by . . . .

Bruce Tidor
Professor of Biological Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .

Jacob K. White
Professor of Electrical Engineering and Computer Science
Thesis Committee Member

Parameter and Topology Uncertainty for Optimal Experimental Design

by

David Robert Hagen

Submitted to the Department of Biological Engineering
on March 31, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

A major effort of systems biology is the building of accurate and detailed models of biological systems. Because biological models are large, complex, and highly nonlinear, building accurate models requires large quantities of data and algorithms appropriate to translate this data into a model of the underlying system. This thesis describes the development and application of several algorithms for simulation, quantification of uncertainty, and optimal experimental design for reducing uncertainty. We applied a previously described algorithm for choosing optimal experiments for reducing parameter uncertainty as estimated by the Fisher information matrix. We found, using a computational scenario where the true parameters were unknown, that the parameters of the model could be recovered from noisy data in a small number of experiments if the experiments were chosen well. We developed a method for quickly and accurately approximating the probability distribution over a set of topologies given a particular data set. The method was based on a linearization applied at the maximum *a posteriori* parameters. This method was found to be about as fast as existing heuristics but much closer to the true probability distribution as computed by an expensive Monte Carlo routine. We developed a method for optimal experimental design to reduce topology uncertainty based on the linear method for topology probability. This method was a Monte Carlo method that used the linear method to quickly evaluate the topology uncertainty that would result from possible data sets of each candidate experiment. We applied the method to a model of ErbB signaling. Finally, we developed a method for reducing the size of models defined as rule-based models. Unlike existing methods, this method handles compartments of models and allows for cycles between monomers. The methods developed here generally improve the detail at which models can be built, as well as quantify how well they have been built and suggest experiments to build them even better.

Thesis Supervisor: Bruce Tidor
Title: Professor of Biological Engineering and Computer Science

Acknowledgments

Foremost, I acknowledge the unequivocal and unequaled support of my wife, Patricia. Having joined me near the beginning of my graduate work at MIT, she has been my primary source of encouragement through the frustrations and exasperations of graduate research. As I move on in my career, I look forward to continuing my life with her.

My Mom, Dad, and my brothers, John and Mark, were the first supporters of my work in Cambridge. Even though I moved away from Ohio, they always felt close.

This thesis owes its existence to my advisor Bruce Tidor, who has provided the guidance and resources necessary to complete this work. He is particularly responsible for encouraging the biological applications of the methods I developed. He brought together my colleagues in the Tidor lab, including Josh Apgar, Jared Toettcher, Bracken King, Nate Silver, Yang Shen, Felipe Gracio, Yuanyuan Cui, Nirmala Paudel, Ishan Patel, Gil Kwak, Tina Toni, Andrew Horning, Brian Bonk, Raja Srinivas, David Flowers, Kevin Shi, David Witmer, Gironimo Mirano, and Sally Guthrie. These people made coming into the lab fun and deserve to be thanked for putting up with my occasional speculations on grand problems and frequent rants on everything that is wrong with science today.

My closest collaborator was Tim Curran from the Forest White lab, a classmate and superb mass-spectrometrist who provided the only real data analyzed in this thesis. More important were our discussions into designing and refining both his and my methods.

Jacob White provided much helpful guidance in refining my methods, especially in the areas of optimization, simulation, linear algebra, and Matlab early in my graduate work.

I finally acknowledge Ryan Dorow and Sean O'Reilly. The weekly card games we have together are just an excuse to discuss science, economics, policy, etc. They have helped keep me sharp and sane while I completed by graduate work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Upon the publication of first draft of the human genome in the year 2000, the director of the NIH Francis Collins remarked, “I would be willing to make a prediction that within 10 years, we will have the potential of offering any of you the opportunity to find out what particular genetic conditions you may be at increased risk for, based upon the discovery of genes involved in common illnesses like diabetes, hypertension, heart disease, and so on [1].” For the most important diseases fourteen years later, their genetic basis remains incomplete with at most partial correlations between particular genes and disease outcome having been uncovered [2, 3, 4]. The enthusiasm surrounding the human genome project was warranted given the knowledge of the day. Some of the most destructive diseases in the West, heart disease, cancer, stroke, Alzheimer’s disease, diabetes, and kidney disease [5], appeared and still appear to originate not from external agents, such as bacteria and viruses, but from internal pathological processes. As the genome is the program for making and maintaining a human being, where else would we look for answers to those diseases except in the genome? Of the inborn diseases whose genetic basis had been solved, each was caused by a single mutated gene. Sickle-cell disease was found by Linus Pauling and colleagues in 1949 to be caused by a mutation in the hemoglobin molecule [6]. The first mutation that caused cystic fibrosis was discovered by Francis Collins and colleagues in 1988 [7]. Using technology ultimately used for the human genome project, the U. S.–Venezuela Collaborative Research Project located the gene responsible for

Huntingtin's disease in 1993 [8]. With so many diseases remaining to be understood, it was reasonable to just sequence the human genome, vacuuming up the remaining unsolved diseases, rather than search for each one individually. It turned out that many of the unsolved diseases were outside the paradigm of one gene, one protein, one disease. The human body is an extraordinarily complex machine, and some of the defects are born not from a single broken device, but from the miscooperation of slightly misaligned parts. Thus, large swaths of biology and medicine cannot be reduced to a single gene. These swaths, which seem to encompass almost all of human biology, can only be understood as large networks of interacting components. Out of these limitations arose systems biology, a study of biology not through the removal and study of a single part but through the reassembly of many parts. Biological systems can exhibit behavior that is more complex than can be easily predicted from the behavior of the individual parts. Integrating the parts to discover the emergent behavior requires at least mathematics and, for all but the simplest collection of parts, a computer.

A computational model is one of the most important tools available in systems biology. A computational model is a means to simulate the behavior of the system under different conditions, assumptions, parameters, and other differences that may be of interest to us. A running model allows scientists to study parts of the system that are experimentally unobservable with current technology. The detail to which a model can be trusted depends on how it was constructed. There are a number of different frameworks in which to construct a model, each with its own advantages and disadvantages [9].

At the most abstract level are the machine learning techniques. These methods analyze a set of measurements and return a relationship between the parts. The advantage of this technique is that few assumptions need to be made about the data. It does not need to be a particular type of data or even the same type of data. The disadvantage is that no mechanistic or causal relationships are determined, only correlative ones. Because these methods are not constrained by mechanistic assumptions, they typically use a simple and efficient framework, so that very large data sets can be examined with low computational cost. A famous example of a method of this type in the biological sciences is hierarchical clustering [10], which takes particular measurements under different conditions and clusters the behavior of the measurements according to how similarly the measurements change in response to the conditions. In terms of understanding the system, measurements that behave similarly may measure components closely associated in the real system. In terms of prediction, such models cannot predict the results of an experiment whose conditions are dissimilar to conditions already studied. Other important methods at this abstract level are principle component analysis [11] and partial-least squares [12], which extract correlations between measured variables.

At the next level are non-mechanistic network models. These models have nodes, representing species in the model, and edges between the nodes, representing causal relationships between the nodes. The advantage of these modeling techniques is that, by disregarding the mechanism of molecular interaction, they can use a mathematical structure that is amenable to automatic construction from data. The disadvantage is that disregarding the mechanism results in models that necessarily deviate from the behavior of biological systems. Some well-known frameworks include Bayesian networks [13], Petri nets [14], and Boolean networks [15]. Some techniques can assemble these models automatically from data in a computationally efficient way [16, 17, 18].

At the most detailed level are the mechanistic models. These models are built out of chemical reactions between molecular species. The advantage is that there are few assumptions required because, fundamentally, much of biology actually is networks of chemical reactions.

1.0.1 Mechanistic Modeling

The reaction-diffusion master equation (RDME) is a general formula for describing chemical reactions [19]. The RDME models a system as a collection of particles each with a state and a position. The particles can be produced from nothing via a zeroth-order reaction, which typically represents particles being introduced from outside the conceptual boundary of the model, since nothing can truly be produced in this way.

A single particle can transform into one or more particles via a first-order reaction, and two particles can collide to form one or more particles. As a representation of chemical kinetics, the RDME makes the following assumptions:

1. The state retains no memory of the past (Markov property)
2. The position changes over time according to molecular diffusion (Brownian motion)

Concerning the state, it would be more accurate to describe each particle with numerous internal bond lengths and angles. Its propensity to react either alone or in a collision with another particle is dependent on the state of those bonds. The probability that a particle will or will not be internally configured to react in a particular way is abstracted in the rate constant for that reaction. Once sufficient time has passed since the particle was created for the internal configuration to reach equilibrium, the probability of the particle being in a particular configuration remains constant and the Markov property holds. We can justify making the assumption broadly across biology by recognizing that the time it takes for atoms to move around within a molecule is usually much shorter than the time it takes from molecules to move around the cell. Whenever the assumption is not valid, additional states may be added to the model to represent these metastable internal configurations.

Concerning diffusion, it is more accurate to describe the motion of the particles as traveling in straight lines until they collide with another molecule. The approximation of Brownian motion is appropriate as long as the distance that the molecules travel between reactions is much greater than the distance they travel between collisions [20]. Given that the cell is made mostly of water, which is largely a non-reactive solvent for the purposes of biology, the simplifying assumption is well justified.

The reaction-diffusion master equation describes the evolution of the spatial probability distribution of each state over time. In general, solving the partial differential equation to obtain the probability distribution is impossible analytically and intractable numerically. There are various spatial-stochastic algorithms to generate a simulation of the system—one particular draw from the probability distribution

[21, 22, 23]. The advantage of spatial stochastic algorithms is that few assumptions need to be made. Each simulation tracks the movement and change in each particle of the system. Needless to say, this can be very computationally expensive. Even if there are many copies of each state, each particle must be tracked individually, it must diffuse individually, and its propensity to react with each other particle must be computed as well. Furthermore, because each simulation is stochastic, it is necessary to run the simulation many times in order obtain a representative sample of the system's behavior.

One way to speed up the simulation is to make an assumption that allows for lumping together particles with identical states. The assumption, which underlies the use of partial differential equation modeling, is as follows:

3. Each state is a continuous density

This assumption is valid when the number of particles in each state is very large such that the reaction of a single particle has a negligible impact on the local concentration of that state. The density can be evolved over time with a numeric partial differential equation solver. With this assumption, the stochastic nature of the chemical system has been averaged out. Simulation produces a deterministic result, which obviates the need for running many simulations in order to obtain the average behavior. The caveat is that this assumption is not true for many biological systems. Models that include DNA, for example, will have about two copies of each gene, which is certainly not large enough to satisfy a continuous density assumption.

Disregarding the assumption made for partial differential equation modeling, a different assumption can be made instead to arrive at non-spatial stochastic modeling, which is as follows:

4. Each state is well-mixed within the reaction volume

This assumption is valid if the time it takes for a reaction to occur is much longer than the time it takes each state to spread throughout the reaction compartment. Every particle of a particular state is the same no matter where it was produced.

This allows the spatial part of the simulation to be removed entirely, keeping only a count of the number of particles in each state. Simulation of this model can be done using various stochastic simulation methods, the most famous of which is the Gillespie algorithm [24]. As long as the number of particles is small, the simulation is computationally inexpensive, but if the number of particles is large, then simulating each molecule change one at a time will be very slow.

Combining both additional assumptions leads to the kinetic aspect of the law of mass action, the primary modeling framework of this thesis. By assuming that the system is both well-mixed and uses a large count of particles, it can be modeled using ordinary differential equations (ODEs).

1.0.2 KroneckerBio Modeling Toolbox

There are many features of biological systems, such as compartments, outputs, and higher-level approximations, that may or may not be supported in a specific modeling framework. To assist me in completing the work described in this thesis, I developed a software toolbox in Matlab called KroneckerBio, a project that began with Joshua F. Apgar, Jared E. Toettcher, Jacob K. White, and Bruce Tidor. This toolbox is free and open-source software. A model in KroneckerBio is composed of the following components: compartments, states, inputs, parameters, seeds, reactions, and outputs.

Compartments map to cellular compartments like the cytoplasm, nucleus, cytoplasmic membrane, or DNA. They have a specific dimensionality, 3 for volumes, 2 for membranes, 1 for fibers, and 0 for points. The rates of bimolecular reactions, being dependent on the frequency of collisions, are inversely proportional to the size of the compartment in which the reaction takes place. If the reaction is between two species in different compartment, such as a free ligand binding to a membrane-bound receptor, then the compartment with the highest dimensionality is the compartment for the reaction. The volume of the compartments is represented by v a vector of length n_v.

KroneckerBio uses the formalism that a reaction rate is the change in the *amount* of the species, not the change in the *concentration*. When dealing with a reaction in a single compartment, using the concentration is somewhat simpler because the volume of the compartment does not need to be considered, except indirectly through the concentration. But when molecules can move between the compartments, conversion factors need to be used when concentrations are tracked. For example, a molecule moved from one compartment to another twice as small will increase the concentration in the new compartment twice as much as it decreased the concentration in the old compartment, even though the amount gained is equal to the amount lost. Computing all rates and tracking all states as amounts allows the simulation to be simplified.

States are the main species of the model. Each state has a value that represents the number of molecules of a particular type that exists. This value evolves over time according to the simulation of the reactions of the model. Each state is confined to a particular compartment, so that chemically identical states in different compartments are tracked separately. The initial condition of the state may be a constant defined in the model or a function of the seed parameters. The amount of the states as a function of time are represented by $x\left(t\right)$ a vector of length n_x.

Inputs are also species of the model, but their amount as a function of time is defined before the simulation and is not affected by the reactions, even if the inputs are reactants or products of a reaction. Inputs are useful in representing species that are at the boundary of the model—species whose behavior we already know or can control. For the models used in this thesis that have inputs, most represent external ligands whose conditions are under the control of the experimenter. For any simulation in KroneckerBio, a flag determines whether the inputs are defined as part of the model (the same for all experiments) or part of the experiment (may be different for each experiment). The amount of the inputs are represented by $u\left(t,q\right)$ a vector of length n_u as an arbitrary function of time and input control parameters q a vector of length n_q.

Parameters are the rate parameters for the reactions. Each reaction names exactly one kinetic parameter to use. Each reaction may also have a scaling factor for its parameter, which is only useful when simulating rule-based models. The rate parameters are represented by k a vector of length n_k.

Seeds are parameters that determine the initial amounts of states. Each state is a linear combination of seed parameters, and states may share seed parameters. For any simulation, a flag determines whether the seeds are defined as part of the model or as part of the experiment. The initial amounts of the states are computed according to the following formula:

$$x\left(0\right) = x_0 = \frac{dx_0}{ds} \cdot s + x_c \tag{1.1}$$

where s is a vector of length n_s representing the seed values, $\frac{dx_0}{ds}$ is a matrix n_x by n_s representing the mapping of the seeds onto the states they influence, and x_c is the constant initial amount of the state.

Reactions determine how the states evolve over time. All reactions are assumed to follow the well-established law of mass action [25]. Each reaction has at most two reactants and usually at most two products. The rate of the reaction is equal to the rate parameter times the amount of the first reactant times the amount of the second reactant divided by the volume of the reaction compartment. Mass-action reactions can be represented in a compact matrix notation:

$$\begin{aligned} r\left(t,x,u\right) &= D_1 \cdot x + D_2 \cdot \left(x \otimes \left(x/v_x\right)\right) + D_3 \cdot \left(u \otimes \left(x/v_x\right)\right) \\ &\quad + D_4 \cdot \left(x \otimes \left(u/v_u\right)\right) + D_5 \cdot \left(u \otimes \left(u/v_u\right)\right) + D_6 \cdot u + d \end{aligned} \tag{1.2}$$

where $a \otimes b$ represents the Kronecker product of vectors a and b, and here a/b represents elementwise division of vector a by vector b. Matrices D_1 through D_6 and vector d (collectively called the D matrices) have a number of rows equal to the number of reactions n_r in the model and a number of columns appropriate to the number of species or Kronecker product of species. These matrices are exceedingly sparse. For each row across all the matrices there is exactly one non-zero entry; it is the rate parameter for the reaction put into the correct column that represents the reactant or reactants of that reaction.

The system of ODEs is generated by applying the stoichiometry matrix to the rate:

$$\dot{x} = \frac{dx}{dt} = f\left(t, x, u\right) = S \cdot r\left(t, x, u\right) \tag{1.3}$$

where S is an integer matrix n_x by n_r, mapping how much each state, either reactant or product, is changed by the reaction. Because S and the D matrices are constant over the course of a simulation, we distribute S over the D matrices when constructing the model. This results in the KroneckerBio formulation of mass-action models, a compact and computationally efficient representation assuming that fast sparse matrix multiplication and addition algorithms are available:

$$\begin{aligned} f\left(t, x, u\right) \quad = \quad & A_1 \cdot x + A_2 \cdot \left(x \otimes \left(x / v_x\right)\right) + A_3 \cdot \left(u \otimes \left(x / v_x\right)\right) \\ & + A_4 \cdot \left(x \otimes \left(u / v_u\right)\right) + A_5 \cdot \left(u \otimes \left(u / v_u\right)\right) + A_6 \cdot u + a \end{aligned} \tag{1.4}$$

where matrices A_1 through A_6 and vector a (collectively called the A matrices) map the effect that each species or Kronecker product of species has on the state variables.

Outputs are a linear combination of the states and inputs and are the observables of the model. They are a convenience for biological models which usually have many complexes or aggregates of monomeric species. Most measurement techniques cannot measure the amounts of all complexes in the system; they can only measure something like the total amount of phosphorylation of a protein. There is no species in the model that represents this observable because the phosphorylated protein is bound up in many complexes, possibly in multiple copies in some complexes. Outputs provide a way to define observable quantities without having to sift through the state data externally after every simulation. It also allows for competing topologies with different numbers of states to describe the same data. The outputs are computed via the following formula:

$$y\left(t\right) = C_1 \cdot x\left(t\right) + C_2 \cdot u\left(t\right) + c \tag{1.5}$$

where C_1 is a matrix n_y by n_x representing the contribution of each state to each output, C_2 is a matrix n_y by n_u representing the contribution of each input to each output, and c is a vector of length n_y representing the constant contribution to each output. For most models, only the states contribute to the outputs.

1.0.3 Statistical Modeling

We can conceptually separate the model into two main components: the topology and the parameters. The topology is the structure of the model—here it is the equations underlying $f(t, x, u, k)$. For a mechanistic model, the topology is the set of chemical reactions/interactions that occur. The parameters are values within the topology that can be varied to produce different simulations—here it is k, s, and q. Both the topology and the parameters can have uncertainty. Topology uncertainty would question whether or not a particular reaction takes place, while parameter uncertainty would question how fast that reaction takes place.

For a given problem, not all the kinetic parameters and seed parameters need to be free, unknown parameters of the model. Some parameters of the model may be known, with the rest being variable. In this thesis, the variable kinetic parameters θ_k, the variable seed parameters θ_s, and the variable input control parameters $\theta_q$will be concatenated into a single vector θ of length n_θ and be simply referred to as "the parameters".

When gathering data on a system, some measurements are more likely to be observed under one model than another. Because of this, data provides evidence in favor of some models at the expense of others. The likelihood $p_{\hat{y}|m,\theta}(\hat{y}, m, \theta)$ is the quantification of the probability that measurements $\hat{y}$ will be observed if the true model consists of topology m and parameters θ. The likelihood function is specific to a set of experimental conditions and measurement scheme.

In a Bayesian framework, the probability distribution of the parameters given the data is described by the parameter posterior, which is proportional to the product of the likelihood and parameter prior:

$$p_{\theta|\hat{y},m}(\theta, \hat{y}, m) = \frac{p_{\hat{y}|\theta,m}(\hat{y}, \theta, m) \cdot p_{\theta|m}(\theta, m)}{p_{\hat{y}|m}(\hat{y}, m)} \tag{1.6}$$

where $p_{\theta|m}(\theta, m)$ is the prior for the parameters θ in topology m and $p_{\hat{y}|m}(\hat{y}, m)$ is the marginal likelihood defined by:

$$p_{\hat{y}|m}(\hat{y}, m) = \int_{\theta} p_{\hat{y}|m,\theta}(\hat{y}, m, \theta) \cdot p_{\theta|m}(\theta, m) \tag{1.7}$$

Similarly, the probability distribution of the topologies given the data is given by the topology posterior, which is proportional to the product of the marginal likelihood and the topology prior:

$$p_{m|\hat{y}}\left(m, \hat{y}\right) = \frac{p_m\left(m\right) \cdot p_{\hat{y}|m}\left(\hat{y}, m\right)}{\sum_i p_m\left(i\right) \cdot p_{\hat{y}|m}\left(\hat{y}, i\right)} \tag{1.8}$$

where $p_m\left(m\right)$ is the prior for the topologies.

For all but the simplest biological models, the posterior distributions of the parameters and topologies do not have analytical solutions. They can only be computed with Monte Carlo methods—in particular, methods like the Metropolis-Hastings algorithm to sample the nonlinear parameter space [26, 27]. Because the number of parameters in a biological model can be very large, the volume of parameter space that needs to be sampled grows exponentially, making it computationally expensive to accurately evaluate the parameter or topology probability. This is particularly problematic when it is not just the one probability distribution for the current data that is desired but when the probability calculation is a subproblem in a larger problem, such as estimating the probability distribution for each experiment in a large set of candidate experiments in order to evaluate which ones are likely to be best at reducing the uncertainty.

If fast but accurate approximations can be developed, then the high cost of the Monte Carlo methods can be circumvented. The main theme of this thesis is the development and use of approximations to arrive at a good estimate faster than a Monte Carlo method for the purpose of optimal experimental design. In particular, we found that a linear approximation of the model, and the analytical solution it allowed, was an excellent approximation for several important problems in systems biology.

1.0.4 Project Summaries

In Chapter 2, we use the Fisher information matrix, a well-established linear approximation, to predict what the parameter uncertainty would be after performing an experiment from a large set of candidate experiments. We synthetically perform the best experiment according to the expected Fisher information matrix by simulating a "true" model and generating noisy data. We found that the approximation is effective for selecting experiments that reduced parameter uncertainty, which also reduced parameter error of the best-fit parameters compared to the "true" parameters, and reduced prediction error of the best-fit model compared to the behavior of the "true" model. This method was demonstrated using a model of the EGF-NGF pathway.

In Chapter 3, we develop a method using linearization to compute the topology probability. This is a new method in system biology, which solves a problem that is uniquely difficult to solve with a Monte Carlo method. This chapter also introduces a new Monte Carlo method, which was necessary to act as a gold standard to which to compare the performance of the linearization method. The computational cost of linearization was similar to the cost of several popular heuristics, but produced a probability distribution much closer to the gold standard. The method was tested on a set of four topologies of the one-step MAPK cascade.

In Chapter 4, we use the linearization method to develop an algorithm for evaluating experiments' ability to reduce topology uncertainty. The linearization method is a subfunction of the algorithm. This optimal experimental design algorithm generates random potential data sets from each candidate experiment and uses the linearization method to evaluate how much the topology uncertainty was reduced by the synthetic data. The method was tested on a mass-action model of ErbB signaling.

In Chapter 5, we develop a different type of algorithm, not based on uncertainty or optimal experimental design, but one to improve the speed of simulating rule-based models. If some proteins in the model have many modification sites, then there are an exponentially large number of possible species of that protein. However, if the sites behave independently and are measured independently, then it is possible to simulate the system using fewer states than the number of species. Existing methods for this problem do not take into account the importance of compartments in biology nor the problem of proteins being able to form a cycle. Our method introduces compartments and allows for proteins to form cycles. The method was used to construct a model of the ErbB pathway, with a level of detail at the top of the pathway that would be impossible using conventional model building techniques.

Chapter 2

Convergence in Parameters and Predictions using Optimal Experimental Design

2.1 Introduction

A goal of systems biology is to construct models that incorporate known mechanisms and reflect existing data under laboratory conditions. The notion is that mechanistic mathematical models not only recapitulate existing measurements but can ultimately predict the behavior of modeled systems under novel conditions not previously tested and be the basis of design work as is done in more mature fields of engineering [28, 29, 30, 31, 32, 33]. In addition, high-quality, mechanistically accurate models can also lead to novel insights into systems operations. Biological systems are sufficiently complex that mechanistic models will contain large numbers of parameters and thus will require correspondingly large quantities of data for training. Recent and future advances in the development of high-throughput measurement techniques (e.g., mass spectrometry [34] and flow cytometry [35]) continue to increase the quantity and quality of data collected, and bring nearer the promise of meeting the needs of true mechanistic understanding of biological complexity, as reflected in the ability to determine the topology and parameters of corresponding models. Important research areas include the development of experimental design strategies to efficiently deploy experiments to probe new aspects of their operation, computational framing of the space of relevant models, and probabilistic treatments of model uncertainty. Here we focus on the first of these areas.

Recent work by Gutenkunst *et al.* [36] has suggested that it is difficult, if not impossible, to accurately estimate the parameters of a typical biological model, regardless of how accurately the data is collected, how many species of the model are simultaneously measured, or how finely the species are measured in time. It was found that, for typical biological models under typical experimental conditions, there were some directions in parameter space that had so little effect on the measured quantities that the resulting uncertainty in many of the parameters was too vast to be overcome by higher quality measurements. In later work by Apgar *et al.* [37, 38, 39], however, our group showed that the seemingly vast parameter uncertainty could be dramatically reduced with a relatively small number of carefully selected perturbation experiments. We demonstrated that sets of experiments could be found that together exercised the epidermal growth factor (EGF) and nerve growth factor (NGF) pathways in sufficiently complementary ways so as to allow all parameters to be determined within 10% uncertainty. This proof-of-concept study highlighted a potential role for computational design of experimental conditions to efficiently reduce parameter uncertainty.

Our previous work effectively demonstrated the existence in principle of a sequence of experiments that progressively reduce parameter uncertainty to manageable levels; it did not, however, investigate whether the sequence of experiments might be discoverable in a practical setting. In an effort to address the challenge of parameter error reduction issued by Gutenkunst *et al.* [36], most aspects of our study paralleled theirs, and these choices precluded drawing conclusions regarding the practicality of parameter error reduction through our scheme. These limitations included (1) the actual model parameters were known and used at each stage of the experimental design progression to select the next experiment in the sequence, but in any real applica-

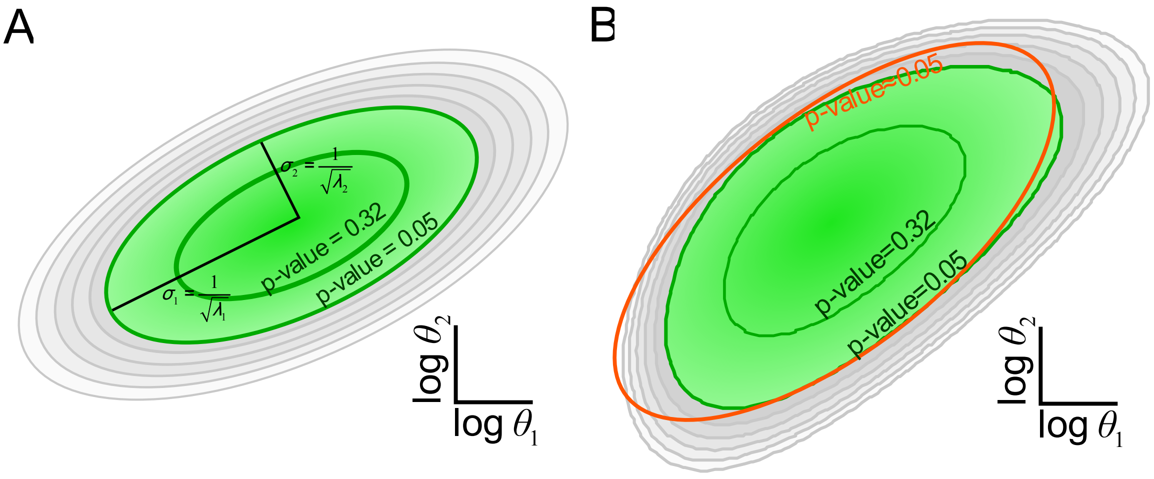

Figure 2-1: **Nonlinearity.** (A) In a linear model, the Fisher information matrix exactly describes the likelihood of the parameter sets in the neighborhood of the most likely parameters. This likelihood is a Gaussian, which has contours that are ellipsoids in parameter space. (B) The likelihood of the parameters of biological models is not exactly Gaussian. For two parameters of the EGF–NGF model fit to a nominal experiment, it can be seen that the true contour for the likelihood of the association and dissociation parameters (green line) are only approximated by the linearization (orange line). All contours in both plots represent parameter sets of equal likelihood.

tion the actual model parameters would be unknown; (2) the data measurements in each experiment provided the average information that could be obtained from any species at any time, but in practical situations each data point provides information from a single species at a discrete time; and (3) the model was assumed linear in the sense that the Fisher information matrix was assumed to accurately represent the parameter uncertainty, whereas in practice the Fisher information matrix is just the first (linear) term in an expansion of that error (Figure 2-1). This work addresses the practicality of setting up and solving as an optimization problem the task of selecting experiments to progressively reduce parameter uncertainty in biological models by removing these limitations and seeking convergence to the true, unknown parameters. In particular, the performance of the approach could degrade significantly because best-fit parameters with their inherent errors, rather than perfect parameters, are used in the experimental design phase. A major result of the work presented here is that fit parameters do, indeed, perform well in this role.

For comparison to previous work from our group and that of Sethna, we carried out this study with the same model of the EGF and NGF receptor kinase cascades, which are an important pair of interlaced signaling networks in mammalian cells [40]. The EGF receptor pathway, in particular, has become one of the best-studied signaling pathways in biology [41, 42, 43, 44]. Constitutive activation of this pathway is associated with cancers of the breast, bladder, cervix, kidney, ovary, lung, and other tissues. Despite nearly half a century of investigation, much remains unknown about this pathway [45, 46]. A number of models has been developed, differing in the species included, the connections among them, and the data with which they were parameterized. The diversity of the available models of this system reflects the underlying uncertainty concerning the true nature of these pathways, in terms of both topology and parameters [40, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59]. The EGF–NGF model used in the current work is an ordinary differential equation (ODE) model developed by Brown *et al.* in which the enzymatic reactions of the network are modeled with Michaelis–Menten kinetics [40].

We used synthetic data sets generated according to the published model and showed that the uncertainty in all 48 parameters can be effectively reduced below 10% using the discrete data generated from a small set of complementary experiments chosen according to a greedy optimization algorithm. The parameters estimated by fitting to the data of these chosen experiments converged to their true values with a residual error consistent with 10% uncertainty. Furthermore, the error in the predictions made according to these parameters was consistent with 10% parameter uncertainty.

2.2 Methods

2.2.1 Scenario Overview

In our scenario, we treated the published model as the true system. To perform a synthetic experiment, this model was simulated according to the defined experimental conditions, and noisy data were generated according to the measurement scheme of that experiment by adding Gaussian random noise corresponding to 10% measurement error.

A nominal experiment was performed and a starting model was fit to the resulting data. A nominal Fisher information matrix was computed. Using the fitted model, the expected information matrices for a large set of candidate experiments were computed. The nominal information matrix was added to each of the expected information matrices to predict the combined information matrix after doing each of the candidate experiments. The utility of each sum was quantified using a goal function, and the highest-ranked experiment was selected.

The selected experiment was performed using the true model to generate noisy measurements in accordance with the experiment's design. The model was fit to the union of the nominal data set and the new data set from the best experiment. This fitting returned a new parameter set from which the expected information matrices were recomputed and the subsequent best experiment was selected. This procedure of computing, selecting, performing, and fitting was repeated iteratively until all the parameter directions had uncertainties below 10%.

2.2.2 The Model

The model describes signaling from the EGF and NGF receptors in rat PC12 cells and was developed by Brown *et al.* [40]. It has 32 species and 48 parameters for 24 reactions in 2 compartments. We obtained a version encoded in the Systems Biology Markup Language (SBML) from the BioModels Database [60]. The model includes two extracellular species, EGF and NGF, which each bind to the corresponding receptor to form two complexes. The remaining species are divided between 11 enzymes than can exist as either an active or inactive species and four enzymes that are constitutively active. The parameters are divided into three classes: (1) four rate constants for ligand–receptor association and dissociation, (2) 22 k_{cat} values, and (2) 22 K_m values of the Michaelis–Menten enzymatic reactions. The species, reactions, and rate parameters were retained from the original model. An illustration of the model topol-

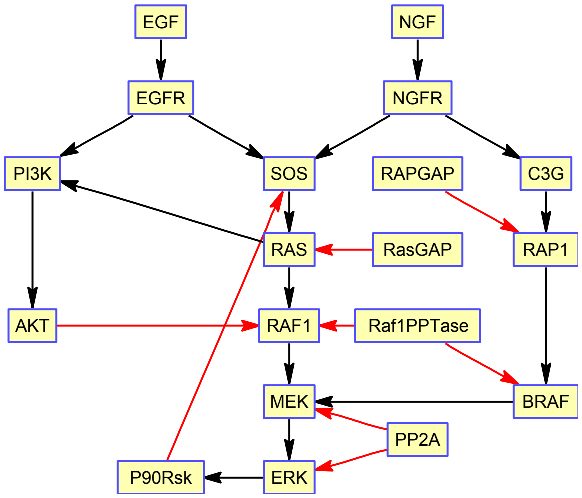

Figure 2-2: **Illustration of EGF-NGF model topology.** Each node (except EGF and NGF) exists in an active and inactive species. Black arrows indicate Michaelis-Menten reactions that activate the target node, while red arrows inactivate. EGF and NGF, are exceptions, and bind in a mass-action fashion to their respective receptors to form an active complex.

ogy is provided in Figure 2-2 and a list of parameters and their values is available in Table A.1. The extracellular compartment was given a volume of 1000 times that of the intracellular compartment to reflect the modification made by Apgar *et al.* [37]. The starting model had the topology of the true model, but each parameter was set to the geometric mean of the class of parameters to which it belonged.

2.2.3 The Experiments

We defined a battery of candidate experiments that served as a collection of different conditions and perturbations to the system, a selection of which could potentially drive a sufficiently wide variety of system behavior to allow definition of most or all of the parameters with small uncertainty. Each experiment included stimulation with one of five discrete values of EGF (1.0×10^7, 1.0×10^5, 1.0×10^3, 1.0×10^1, and 0.0 molecules/cell) and five discrete values of NGF (4.56×10^7, 4.56×10^5, 4.56×10^3, 4.56×10^7, and 0.0 molecules/cell). In this model, 1 ng/ml is equal to 1000 molecules/cell of EGF and 4560 molecules/cell of NGF. In addition, up to three proteins in the network could have their concentrations changed through knock-down or over-expression. The species that could be changed were the two receptors, the eleven inactive enzymes, and the four constitutively active enzymes, all of which started with non-zero concentrations as their initial condition. To represent knock-down or over-expression, each of these species had its initial concentration decreased or increased to 100-fold its nominal value, respectively. Considering all combinations of EGF and NGF concentrations and knock-downs and over-expressions, there were 150 475 possible experiments. The nominal experiment was the one with initial EGF concentration equal to 1000 molecules/cell (1 ng/ml) and initial NGF concentration equal to 4560 molecules/cell (1 ng/ml) and no knock-downs or over-expressions.

All experiments were performed synthetically by simulation of the system for 120 minutes using the numerical ODE solver `ode15s`, with analytical Jacobian supplied, in Matlab® (2008b, The MathWorks, Natick, MA). Each experiment called for measuring all 32 species at 100 evenly spaced time points, and all data points were subjected to random Gaussian noise with a standard deviation of 10% or 1 molecule, whichever was larger. The experimental conditions and measurement scheme are comparable to those used by Apgar *et al.* [10].

2.2.4 Data Fitting

The fit of the model to data for any particular parameterization was quantified using generalized least squares:

$$\chi^2(\theta) = e^T(\theta) \cdot V_{\bar{y}}^{-1} \cdot e(\theta) \tag{2.1}$$

$$e(\theta) = \bar{y}(\theta) - \hat{y} \tag{2.2}$$

where $V_{\bar{y}}$ is the variance–covariance matrix of the measurements, θ is the vector of parameters of length n_θ, and $e(\theta)$ is the difference between the model predictions $\bar{y}(\theta)$ and the data points $\hat{y}$. $V_{\bar{y}}$ is a square symmetric positive semi-definite matrix the size of the number of measurements $n_{\bar{y}}$, and $e(\theta)$, $y(\theta)$, and $\hat{y}$ are all vectors of length $n_{\bar{y}}$. Our procedure assumed that $V_{\bar{y}}$ was a constant for the data set. For the example in this paper, there was no covariance proper, so $V_{\bar{y}}$ was diagonal, and we estimated the uncertainty as 10% of the value of each data point or 1 molecule, whichever was larger.

The best-fit parameters were defined as the vector θ that minimized $\chi^2(\theta)$. This nonlinear optimization was accomplished using the active-set algorithm implemented within `fmincon` in Matlab. The lower bound of the search space was 1000-fold less than the smallest value in each class of parameters in the published model; the upper bound was 1000-fold greater than the largest value in each class.

2.2.5 Information

The Fisher information matrix was used to quantify the knowledge of the parameters. The Fisher information matrix $F(\theta)$ for a set of normalized parameters that affect the means of a multivariate normal distribution is given by:

$$F(\theta) = \left(\frac{\partial \bar{y}(\theta)}{\partial \log \theta}\right)^T \cdot V_{\bar{y}}^{-1} \cdot \frac{\partial \bar{y}(\theta)}{\partial \log \theta} \tag{2.3}$$

where $\frac{\partial \bar{y}(\theta)}{\partial \log \theta}$ is the sensitivity of the values of the data points to the normalized parameters (an $n_{\bar{y}}$ by n_θ matrix calculated by integrating the forward sensitivities together with the system during simulation using the numerical ODE solver `ode15s` in Matlab).

To compute the information matrix for candidate experiments, the model was simulated for each candidate experiment using the fitted parameters. The uncertainty of each measurement was computed according to the measurement scheme of the candidate experiment. The sensitivity of each concentration to the parameters was also integrated. Using Equation 2.3, the measurement uncertainty and sensitivity were used to compute the information matrix from the expected results of a candidate experiment. When the information matrix was computed in this way, it was called an *expected* information matrix.

2.2.6 Parameter Goal

Three different goal functions were used to evaluate the efficiency of candidate experiments. Each of the goal functions was based on the eigenvalues resulting from an eigendecomposition of the information matrix. The inverse square roots of these eigenvalues correspond to the uncertainties in the eigendirections of parameter space [61, 62]. The first goal function maximized the number of eigendirections whose uncertainties were less than 10% and used the remaining directions to break ties:

$$G_1 = -\sum_{i=1}^{n_\theta} g_i \tag{2.4}$$

$$g_i = \begin{cases} 1, & \lambda_i \geq \frac{1}{\sigma_{\text{cut}}^2} \\ \frac{\lambda_i}{\frac{1}{\sigma_{\text{cut}}^2} \cdot n_\theta}, & \text{otherwise} \end{cases} \tag{2.5}$$

where λ_i is the ith eigenvalue of the information matrix and σ_{cut} is 0.1, the uncertainty cutoff value. This goal function is equivalent to that used by Apgar *et al.* [10].

The second goal function minimized the sum of the natural logarithm of the ellipsoid axes. It is equivalent to minimizing the volume of the uncertainty ellipsoid, as well as minimizing the entropy of the parameter probability distribution, as well as maximizing the determinant of the Fisher information matrix:

$$G_2 = \sum_{i=1}^{n_\theta} \log \lambda_i^{-1/2} \tag{2.6}$$

The third goal function was identical to the second except that no additional advantage was given to directions that were tighter than 10%. In other words, directions that had errors lower than 10% contributed to the goal function the same as if they were exactly 10%:

$$G_3 = \sum_{i=1}^{n_\theta} \max(\log \lambda_i^{-1/2}, \log \sigma_{\text{cut}}) \tag{2.7}$$

2.3 Results

We began our parameter estimation experiments with no knowledge of the actual model parameters. Instead, the process was begun with simulated experimental data with 10% added noise from a nominal experiment (intact network stimulated with 1000 molecules (1 ng/ml) of EGF and 4560 molecules (1 ng/ml) of NGF per cell). Initial parameters were estimated by fitting to the data according to weighted least squares, and the corresponding Fisher information matrix was computed. The spectra of inverse-square-root eigenvalues $\lambda_i^{-1/2}$, also referred to as the uncertainties in parameter eigendirections, is given on the far left column of Figure 2-3 (marked Nominal). The three panels of the figure correspond to the three goal functions used. The nominal spectra show some parameter directions to be well determined (uncertainties below 10%) but others to be poorly determined (uncertainties greater than 1000-fold). The actual errors in each parameter are shown in the left of Figure 2-4 (marked Nominal) and are consistent with the uncertainties.

A large collection of 150 475 candidate experiments that included a variety of levels of stimulation with EGF and NGF acting on versions of the network modified with up to three expression changes (100-fold over-expression or under-expression each) was evaluated to determine which would lead to the largest reduction in parameter uncertainty. The selected experiment was simulated and 10% noise was added to the resulting concentrations to produce simulated experimental data. New parameters were fit using cumulative data from the simulated experiments, the corresponding updated Fisher information matrices were computed, and the process was repeated iteratively. The resulting uncertainties and parameter errors are given in Figure 2-3 and Figure 2-4. The selection and implementation of experiments proceeded until all parameter uncertainties were below 10%. The entire procedure was carried out three times in parallel for the three different goal functions used, with similar results.

The sequential addition of experiments selected in the greedy optimization progressively reduced parameter uncertainty. Uncertainty was very low after just three experiments beyond the nominal experiment, and all parameters were determined to

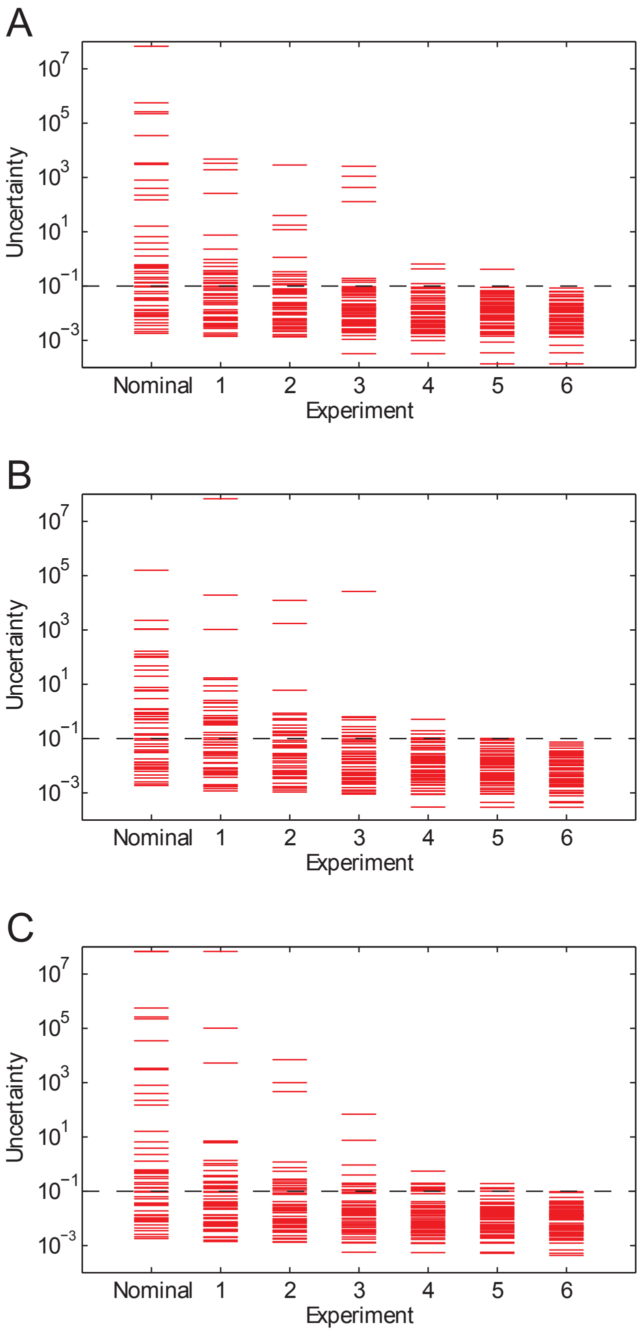

Figure 2-3: **Parameter uncertainty.** Progressively choosing the optimal experiment eventually led to all 48 parameter direction uncertainties being less than 10%. In all three cases, it took 6 additional experiments. It does not appear that the rate of convergence was strongly influenced by the particular goal function used.

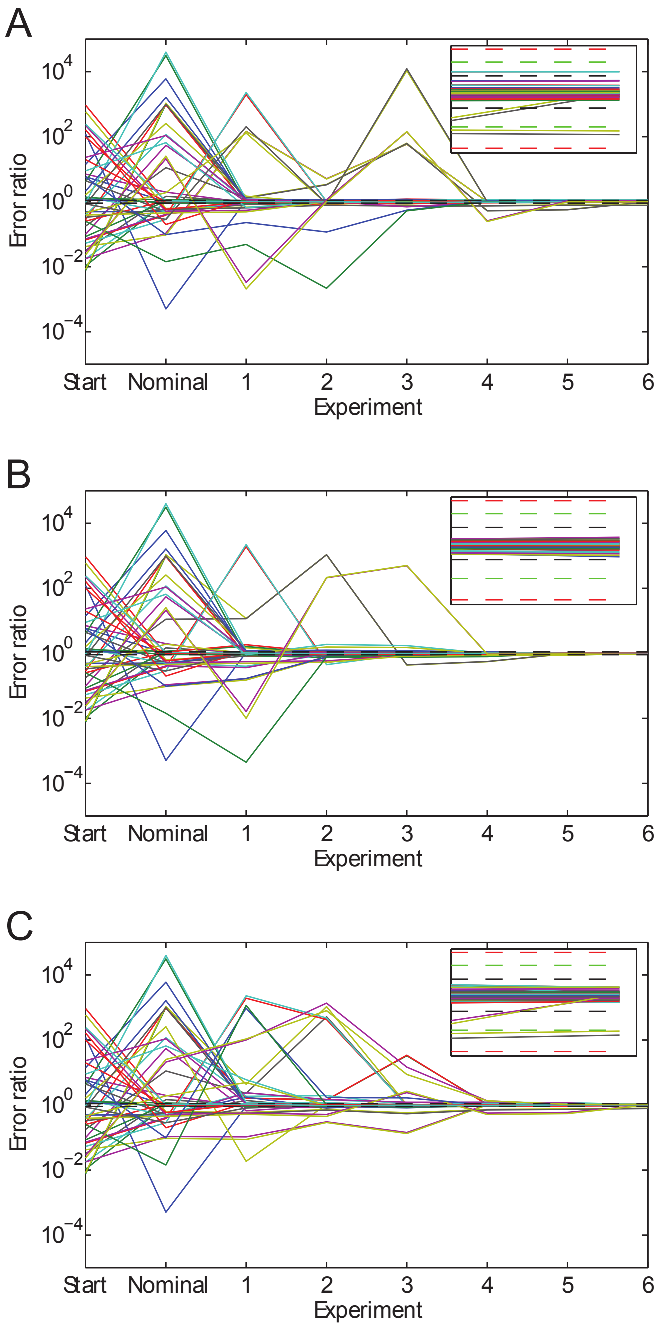

Figure 2-4: **Parameter error.** Here is shown the ratio of the fitted parameters to the true parameters after each iteration of optimally chosen experiments according to the three goal functions. “Start” is the scrambled parameter set before the nominal experiment is fit. Each inset shows a magnified view of the final parameter set error. The black, green, and red dashed lines are 10%, 20%, and 30% error, respectively. By the final experiment, all parameter errors are less than 30% and most parameter errors are less than 10%. This suggests that the linear approximation of the uncertainties is an appropriate estimate of the underlying parameter error when the data are plentiful and the uncertainties are small.

within 10% with six experiments beyond the nominal.

An important aspect of the work is that the approximate parameters fit after each new experiment were used to compute the Fisher information matrices leading to the selection of the following experiment. Figure 2-4 shows that, early in the procedure, many of the estimated parameters were significantly in error, yet they still led to the selection of experiments that efficiently reduced parameter uncertainty. This is significant in light of the fact that the theory strictly applies only to linear models. Even though there are substantial nonlinearities in the EGF–NGF model, the behavior of the parameter uncertainties is similar to that expected from the linear theory. After the final experiment, nearly all 48 parameters were correct to within 10% error. On average, 2 parameters were outside 10%, roughly 1 was outside 20%, and none were outside 30%. This is consistent with the linearized uncertainties from which it is expected that no more than 33% and 5% of parameters should be outside 1 and 2 standard deviations, respectively (keeping in mind that many of the parameters had less than 10% uncertainty by the end of the procedure). Taken together, the results show that the estimated parameters converged to the true parameters and the residual parameter errors were consistent with the calculated uncertainties. This correspondence lends confidence to the usefulness of the method for cases where the actual parameters will be unknown.

The experiments selected are shown in Table 2.1. Three genetic perturbations were used for each experiment except for the final experiment for goal functions 1 and 3. Because the average information provided by one experiment alone is the same for 0, 1, 2, or 3 perturbations (Figure A-1), the apparent preference for 3 perturbations is probably due not to an inherently greater sensitivity to the parameters but instead to the greater diversity (and thus, perhaps complementarity) that 3 perturbations allow as well as the fact that three-perturbation experiments make up 90% of the candidate experiment pool. One anonymous reviewer noted that, for the first goal function, there was never chosen an EGF-dominated stimulation, suggesting reasonable parameter estimation may be achievable without broad exploration of the input concentration space. It remains to be seen how much of the input space could

Goal Function 1. Count of uncertainties above 10%

Exp.	[EGF] (molec./cell)	[NGF] (molec./cell)	Knocked-down	Over-expressed
Nom.	1000	4560		
1	0	4560	RasGap, Erk	Rap1
2	10	4.56E+07	RapGap	Erk, Akt
3	0	4560	Raf1PPtase	Mek, C3G
4	1.00E+05	4.56E+07		Sos, Raf1, Braf
5	0	4.56E+07		Braf, C3G, Rap1
6	1.00E+07	4.56E+07		Sos, Ras

Goal Function 2. Sum of log of uncertainties

Exp.	[EGF] (molec./cell)	[NGF] (molec./cell)	Knocked-down	Over-expressed
Nom.	1000	4560		
1	1.00E+07	4560	EGFR, P90Rsk	Rap1
2	10	4.56E+05	RasGap, Raf1PPtase	Erk
3	1.00E+05	45.6		Sos, Mek, PI3K
4	0	4560	RapGap	Braf, Rap1
5	1.00E+07	4560	RasGap	Ras, Raf1
6	1000	4.56E+07	Rap1	PI3K, Akt

Goal Function 3. Sum of log of uncertainties floored at 10%

Exp.	[EGF] (molec./cell)	[NGF] (molec./cell)	Knocked-down	Over-expressed
Nom.	1000	4560		
1	0	4.56E+07	RasGap	Raf1, Rap1
2	1.00E+05	45.6	Erk, RapGap	Braf
3	10	4.56E+05	RasGap	Ras, Mek
4	1.00E+07	4.56E+07	Raf1PPtase	Sos, Braf
5	10	4560		Braf, C3G, Rap1
6	10	4560		Erk, Akt

Table 2.1: **Optimal Experiments.** The experiments chosen according each of the three goal functions are shown here. The scenarios began with fitting to a nominal experiment. In each case, it required 6 additional experiments to constrain the parameter uncertainty below 10%. All optimal experiments knocked-down or over-expressed the maximum of three proteins in the network, with the exception of the final experiments in the cases of goal functions 1 and 3.

be removed while still retaining the ability to determine the parameters well within a small number of experiments. Table 2.1 also shows that different sets of experiments were used in the three parallel runs using different goal functions. Based on our previous work we hypothesize that what is important about the sets of experiments is their internal complementarity [37]. Multiple sets of experiments can efficiently inform about all 48 parameters, but a set must still be self-complementary to be informative about all parameters. For example, sets of six experiments constructed by mixing two experiments from each of the three complementary sets in Table 2.1 were significantly less informative about the parameters than the optimized sets in the table (Figure A-2).

In a linear system, the outputs change linearly in response to a change in the parameters. Therefore, when a system has nonnegative outputs for any nonnegative parameters, the percent error in any output is bounded by the worst percent error in the parameters. With a nonlinear system, this guarantee disappears. To examine the effect that the optimal experiments had on the prediction error of the EGF–NGF model, we compared the predictions of the three final fitted models to the predictions of the true model. We simulated the model under all 150 475 candidate experimental conditions using both the true parameters and the final estimated parameters and sampled the species according to the standard scheme. No noise was added to the measurements, so that we quantified only the component of the error that came from incorrect parameterization. We computed the relative error in each prediction:

$$\text{relerr} = |\log \bar{y}_{\text{pred}} - \log \bar{y}_{\text{true}}| = \left| \log \frac{\bar{y}_{\text{pred}}}{\bar{y}_{\text{true}}} \right| \tag{2.8}$$

where $\bar{y}_{\text{pred}}$ is the predicted species concentration and $\bar{y}_{\text{true}}$ is the actual species concentration according to the true model. Predicted or actual concentrations less than 1 molecule/cell were not considered.

Because there were 3200 data points in each experiment (32 species, 100 data points each), we summarized the prediction error in each experiment in three ways: (1) the largest prediction error of all the data points in a given experiment, (2) the

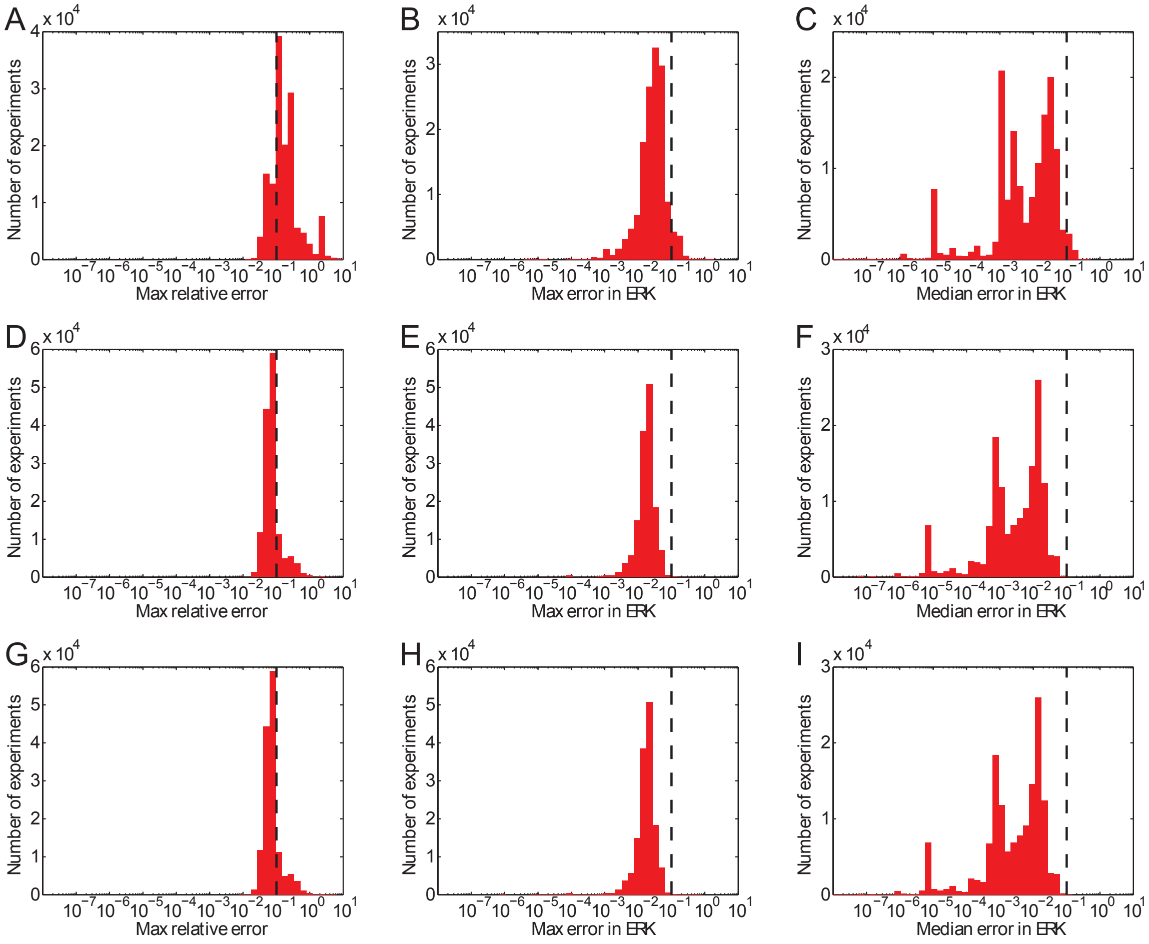

Figure 2-5: **Final prediction errors.** For all 150 475 experiments, we computed the relative error of all data points, by comparing the predictions of the final fitted model to the true model. The error in each experiment was summarized in three ways: the maximum relative error in any species at any time, the maximum relative error in active ERK at any time, and the median relative error in active ERK over all time. Here each column of plots is a different summary and each row is a different goal function. A, B, and C are for goal function 1; C, D, and E are for goal function 2; and G, H, and I are for goal function 3. A, D, and G are the overall maximum relative error results; B, E, and H are the maximum ERK relative error results; and C, F, and I are the median ERK relative error results. As expected, the worst error in all species clusters around 10% (dashed black line), consistent with parameter uncertainty of about 10%. Some errors are much worse (∼100%), though this is dominated by transient intermediate species, as the worst error in active ERK, the output of the system, is much smaller and almost exclusively confined below 10%.

largest of all ERK prediction errors, and (3) the median of all ERK prediction errors. The first summary is the most conservative, but the second and third may be the most informative, because it is often the prediction error of a specified output, not of all intermediate species, that is of greatest interest. A histogram of the prediction errors according to these three summaries from the 150 475 candidate experiments can be seen in Figure 2-5. The worst prediction errors cluster around 10%, which is consistent with the parameter uncertainties and parameter errors. Taken together, the results show that the parameters converged to their true values as data from optimal experiments were added (Figure 2-4) and the residual prediction error in yet-to-be-done experiments was consistent with the calculated parameter uncertainty (Figure 2-5). While entirely expected for linear models, it is gratifying to see similar results in this nonlinear system representative of the types of signaling models currently used in systems biology.

In addition to the prediction error after the final iteration, we also examined the prediction error from the fitted models at intermediate iterations. The predictions of many experiments made by the models only fit to three optimal experiments were worse than 10% (Figure A-3). For each of the three error summary types at each iteration, we further summarized the histograms two ways: (1) the fraction of experiments whose errors were below 10% and (2) the median error over all experiments. Figure 2-6 shows the summarized prediction errors of the models after each experiment was added. As expected, the predictions by all measures improved nearly monotonically with increasing data. The improvements in the predictions tended to taper off with the last few experiments; that is, the experiments chosen to determine the most difficult parameters did not as greatly improve the global predictions. This is consistent with the findings of Apgar *et al.* [37], who showed that the few final parameters determined by the greedy search algorithm had only a few experiments that were sensitive enough to these parameters to determine them. The converse, that there are only a few conditions whose outcome is strongly affected by finally determined parameters, is shown here.

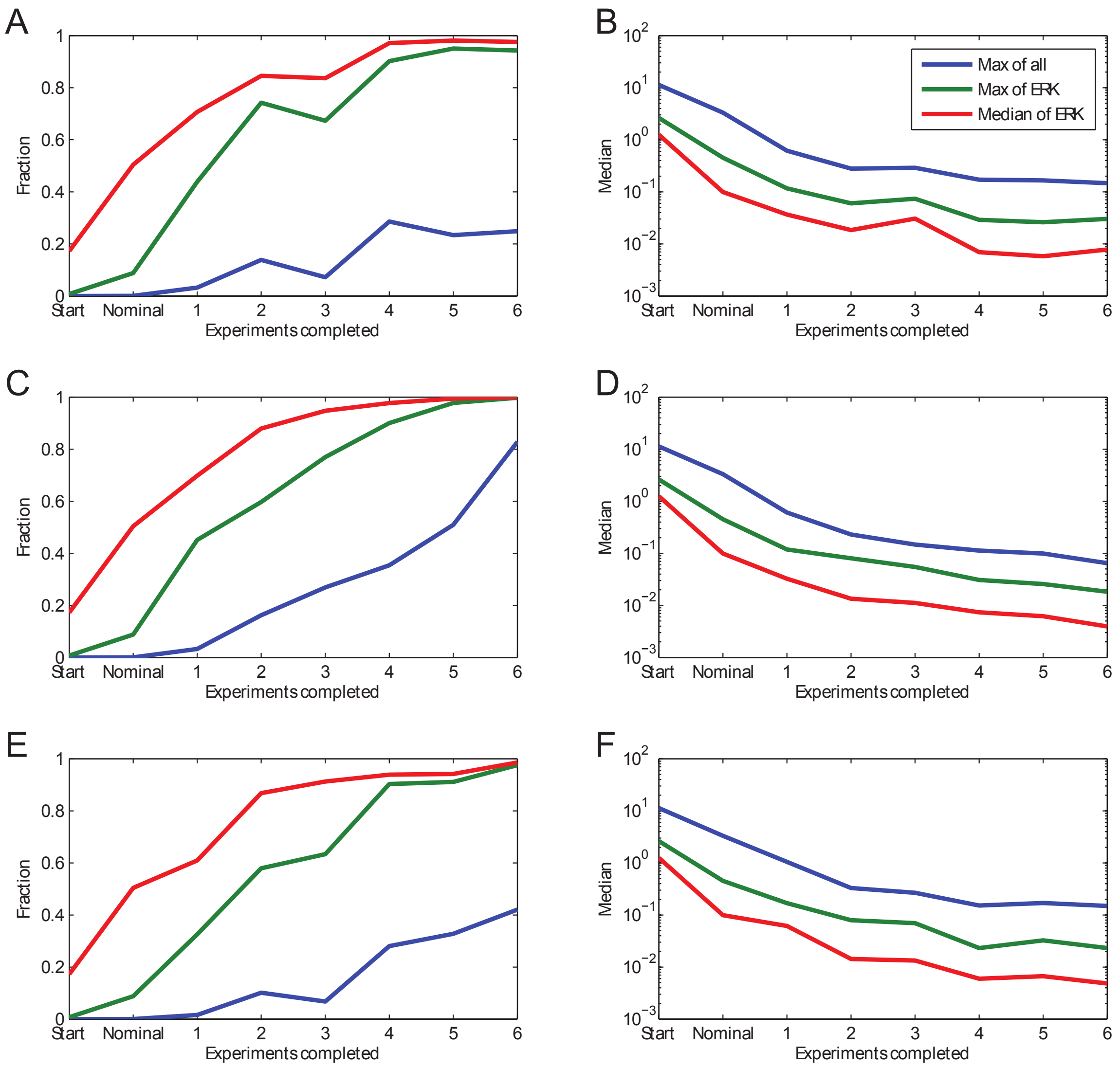

Figure 2-6: **Prediction errors.** As the model was fit to an increasing number of experiments directed by the three goal functions, we summarized the prediction errors in the set of candidate experiments compared to the true model according to the three summaries described in Figure 4: the worst error in all species (blue), the worst error in active ERK (green), and the median error in active ERK (red). A and B are for goal function 1; C and D are for goal function 2; and E and F are for goal function 3. These plots show the fraction of the experiments whose error was less than 10% (left plots: A, C, and E) and the median error over all experiments (right plots: B, D, and F). The prediction errors generally improve with each experiment regardless of the goal function used or the technique used to summarize the error. The early optimally chosen experiments appear to improve the prediction more than final experiments.

2.4 Conclusion

Our previous work demonstrated that optimal experimental design could select experiments that sequentially reduced parameter uncertainty, but it was a theoretical demonstration that made use of the ideal model parameters, which in practice are unknown [37]. Here we extend those findings by demonstrating convergence to the correct parameters through iterative estimation and experimental design cycles, without knowledge of the actual parameters. The parameter uncertainties converged to below 10% for all parameter directions, and the actual parameter errors of the final models were consistent with this uncertainty level. Moreover, the prediction errors on experiments not used in the parameterization were also small and consistent with the parameter errors. This is an important demonstration because, at each stage, the scenario essentially needs to identify measurements that are sensitive to the poorly known parameters based on a linearization about a set of parameters that could still have large errors. If the model were purely linear, this would not be a concern; for nonlinear models, the parameter error means the linearization could frequently be taken about a point significantly different from the true parameter set, and so the linearization would be grossly inaccurate. A major result of the current work is that these inaccuracies do not spoil the rapid convergence of the method. We note that the increase from 5 experiments in our previous work to the current 6 is due to the use of discrete data here rather than continuous data as used previously; we determined this from the observation that even using perfect parameters with discretely drawn data requires 6 experiments to provide all parameters with 10% uncertainty (data not shown). Using 6 randomly chosen experiments or a single experiment with all the perturbations of a set of optimal experiments did not reduce parameter uncertainty as effectively as the set of experiments chosen by our optimization scheme (Figures A-6, A-7, and A-8).

Our method can be applied to any model defined as a system of ODEs, including those that model multistable systems and oscillating systems, as well as those with multiple minima when parameter fitting. In models with these properties, the nonlinearity is even more apparent than in the EGF–NGF model. Linearization was an appropriate approximation for our test system, but further study will be required to determine if this continues to hold for other biological systems with more complex behaviors.

As close as we tried to make our synthetic procedure mirror a real example, there are a number of real system traits that were neglected. Most obviously, a real system is not a set of ODEs. Real biomolecular systems are confined within cells and subject to stochastic effects. Because the number of molecules in our test system was mostly in the hundreds of thousands, stochasticity would be expected to be only a small perturbation. But many biological systems operate with only a few molecules of some species per cell, notably if DNA is involved, which usually exists as one or two copies per cell. Stochasticity could mitigate our conclusions in three ways: (1) even a perfectly parameterized ODE may not accurately recreate the behavior of a stochastic system, (2) aggregate measurements over many cells may not reflect the behavior of any single cell and, thus, fitting to such data could be meaningless or inappropriate, and (3) predicting the next best experiment requires integrating the sensitivities of the species to the parameters, and it is unclear how meaningful these calculations would be in systems where stochasticity was dominant. One way to deal with systems in which stochasticity is important is to use a deterministic approximation to the stochastic noise, such as the linear noise approximation [63] or mass fluctuation kinetics [64]. The Fisher information matrix has been derived for the linear noise approximation and has been used for optimal experimental design [65].

We also did not consider any uncertainty in the inputs to the model. For example, the knock-downs were assumed to flawlessly reduce the initial concentrations to one-hundredth of their prior value. This cannot be done in reality, but a straightforward extension of the method would propagate the uncertainty in the inputs to the uncertainty in the outputs. In lieu of propagating input uncertainty, we repeated the procedure for the first goal function where, instead of the knock-downs and over-expressions being perfectly effective at making 100-fold changes, the effect of the knock-down and over-expression in the true experiments was determined randomly between 1 and 1000-fold on a logarithmic scale. Despite the fact that some knock-downs and over-expressions could have nearly no effect and that the optimal experimental design algorithm had no knowledge of this possibility when selecting the next best experiment to do, this only increased the number of needed experiments to 10 (Figures A-4 and A-5). The experiments chosen and the actual perturbations used are available in Table A.2.

Alternative methods for minimizing the uncertainty in biological models through optimal experimental design have been previously described. Some methods adopt a rigorous Bayesian approach [66, 67]. Other methods adopt a different control mechanism, such as time point selection [68] and dynamic input control [69, 70]. The method investigated here, which selects experiments from a discrete set, is complementary to these alternative control mechanisms. One could imagine combining methods to first select the optimal general conditions from a discrete set and then using time-point selection to remove non-informative measurements and altering the dynamic input profile to further enhance the information gained from a possible experiment. Existing methods also vary in terms of their goal, such as the various optimality functions of the covariance matrix [71] and minimizing prediction uncertainty [72]. All three of our goal functions operated on the parameter covariance matrix and were designed to minimize uncertainty in parameter eigendirections. In fact, goal function 2 is equivalent to the popular D-optimal criterion of maximizing the determinant of the Fisher information matrix. Seeking to maximize the trace of the Fisher information matrix (T-optimality), or minimize the trace of the covariance matrix (A-optimality) are popular goals that we did not test. Operating on prediction uncertainty instead may be preferable if knowledge of the system is a secondary goal to using the model for a specialized task.

It should be noted that having a diverse set of candidate experiments is critical to the successful outcome of this procedure. This method selects the best experiment but does not design new experiments should the candidate set be insufficient to find all the parameters. As indicated by the computation of the Fisher information matrix, good experiments are those that make measurements that are sensitive to the yet unknown parameters. If there are portions of the network that cannot be differentially perturbed with existing techniques, it may not be possible to discover the values of the parameters important there. If the number of time points per experiment is reduced from 100 to 80, 40, 20, 10, or 5, the parameters can still be determined, though it takes a few more experiments to do so (up to 12 experiments; Figure A-9). Furthermore, if the measurement uncertainty is increased from 10% to 20%, it still takes only 6 additional experiments to determine all parameters better than 10%, reinforcing the importance of complimentary. But when the measurement uncertainty is increased to 2-fold, it now takes 22 experiments, although the number of needed experiments can be brought back down by reducing the desired parameter uncertainty to 2-fold as well (Table A.3).

Finally, our method assumes that the topology is already known. This can be far from true. Even with an incorrect topology fit to data, it is possible to use our approach and predict the best experiment to minimize parameter uncertainty. Yet it is unclear what the ultimate result of that would be. Would the experiments actually be effective at minimizing parameter uncertainty? Would it choose experiments that eliminated it as a possible topology? Would the experiments it chose still be best, or nearly best, once the correct topology was found?

Akt is known to also downregulate B-Raf [73], a reaction that is not described in this model. We added a reaction to the original model in which Akt downregulated B-Raf with Akt's normal K_m and a k_{cat} of 1/100 of the strength by which Akt influences Raf-1. We used this modified model as the true model to generate data, while using the published model to actually fit to the data. This was intended to represent one case where the model topology does not match the real system. The model was able to fit the data of the nominal experiment, and two optimally chosen experiments according to goal function 1. However, the model failed to fit the data after the third experiment according to a chi-square test between the data and the best-fit model (data not shown). This suggests that computational experimental design can not only lead to well-determined parameters for appropriate topologies but can also lead to indications that topologies are insufficient to explain observed data.

Chapter 3

Efficient Bayesian Estimates for Discrimination among Topologically Different Systems Biology Models

3.1 Introduction

In systems biology, mechanistic models of biochemical networks can be seen as a combination of two main components, a topology that defines the set of elementary reactions that occur and a parameter set that defines the rate constants of those interactions and perhaps initial concentrations. By mapping components of the model to components of the system, one can computationally ask what role individual parts of the system play with respect to a particular behavior—what behavior would result if a particular part of the system were altered or what part of the system would have to be altered to effect a desired behavior.

Determining the topology of a biological network from data is a difficult and widely studied problem [74, 75, 76, 77]. The space of possible topologies is a discrete one. For a finite number of chemical species, there is a finite, though exponentially large, number of possible ways to connect those species in a network of reactions. There is currently a tradeoff between greater freedom in the mathematical formulation of the topologies and an ability to consider a larger space of topologies, since only some structures have algorithms that can define good topologies without enumerating all possibilities. One can consider three main classes of topology determination methods along this spectrum.

At the most abstract level are the statistical clustering algorithms [78, 79, 80, 81, 82, 83]. In hierarchical clustering [10], well-known for its use in analyzing microarrays, each state is organized as a leaf on a tree where the distance along the branches indicates the amount of dissimilarity in the behavior of the states either in response to a set of perturbations or over time in response to a single perturbation. If a previously unknown state is clustered closely with several known states, this suggests that the unknown state may be involved in the same role as the known states. However, a specific function or mechanism is not elucidated for any state. Another popular method is principal component analysis, which finds the relationships between the states that explain the most variance under the conditions studied [11]. The resulting relationships may reveal the states that are most closely associated with the process that is perturbed between the conditions as well as group the conditions with similar responses. Like hierarchical clustering, such groupings only suggest a coarse organization of the topology, leaving out individual interactions. Methods at this level are widely used because they provide testable hypotheses from very data large sets, even if the studied system is poorly understood.

At the next level are algorithms that reverse engineer causal networks. These algorithms use data to generate *de novo* interaction networks between states of the system [84, 85, 86, 87, 88]. These methods exploit a useful mathematical relation between a specific formulation of the model and a specific type of data. An algorithm by Sachs *et al.* generates an acyclic Bayesian network using single-cell measurements [16]. This method exploits the fact that the short-term stochastic fluctuations in one state would be most strongly correlated with the short-term fluctuations of the nearest states. Thus, a causal graph can be build, not by finding the strongest correlations in the states, but by finding the strongest correlations in the stochastic fluctuations of the states about their mean value. Another algorithm by Yeung *et al.* generates a system of linear ODEs using concentrations of states near a gently perturbed steady state [89]. The method exploits the fact that a linear approximation is good near a steady state, allowing a sparse SVD to be used to solve for the topology. By requiring little *a priori* information, methods at this level bridge the gap between the exponentially large number of possible topologies and a smaller number of topologies supported by the data.

At the most specific level are algorithms that compare the evidence for an enumerated set of topologies. Because one cannot actually enumerate all possible networks for even a small number of states, the set must be shrunk either by assembling topologies based on prior knowledge or by collecting the most favorable topologies generated by a higher-level method like one mentioned in the previous paragraph. These algorithms make use of the likelihood that a topology generated the data to compute the probability that the topology is correct. Several of these methods are used in this work and are described below. Because these methods only require the likelihood of the data, they can be used on a broad range of mathematical modeling techniques such as dynamic nonlinear ODE modeling, which is used in this work.

We phrase the problem of topology probability in a Bayesian framework. Bayes theorem provides the basic identity for computing the posterior topology probability:

$$p_{m|\hat{y}}\left(m,\hat{y}\right) = \frac{p_{m}\left(m\right)\cdot p_{\hat{y}|m}\left(\hat{y},m\right)}{\sum_{i} p_{m}\left(i\right)\cdot p_{\hat{y}|m}\left(\hat{y},i\right)} \tag{3.1}$$

where $p_{m|\hat{y}}\left(m,\hat{y}\right)$ is the posterior probability that the topology with index m is correct given that data $\hat{y}$ (a vector of length $n_{\bar{y}}$) has been observed, $p_{m}\left(m\right)$ is the topology prior of model m, and $p_{\hat{y}|m}\left(\hat{y},m\right)$ is the marginal likelihood of data $\hat{y}$ given model m.

The marginal likelihood is the probability that a set of data would be observed under a particular topology. Because topologies alone do not generate data (parameterized topologies do) the average probability over all parameters weighted by the prior on the parameters is computed by an integral over parameter space:

$$p_{\hat{y}|m}\left(\hat{y},m\right)=\int_{\theta}p_{\hat{y}|m,\theta}\left(\hat{y},m,\theta\right)\cdot p_{\theta|m}\left(\theta,m\right) \tag{3.2}$$

where $p_{\hat{y}|m,\theta}\left(\hat{y},m,\theta\right)$ is the likelihood of data $\hat{y}$ being produced by model topology m parameterized with values θ and $p_{\theta|m}\left(\theta,m\right)$ is the parameter prior for parameter values θ in model topology m.

It is difficult and computationally expensive to evaluate the Bayesian result because of the multidimensional integral required to compute the marginal likelihood in Equation 3.2. This integral does not have an analytical solution for many interesting problems, including mass-action models, and the possibly large number of dimensions of the integral precludes the use of standard quadrature methods such as the trapezoidal rule for numerical integration.

A number of methods have been developed to solve this integral for biological problems [90]. All are Monte Carlo methods that compare a known distribution to the unknown posterior distribution and currently require prohibitive computational resources even for simple topologies. To be a known distribution means that its normalization factor, the integral over all space, is known. The simplest methods compare the prior distribution to the posterior distribution while either sampling from the prior (Prior Arithmetic Mean Estimator [91]) or from the posterior (Posterior Harmonic Mean Estimator [92]). Unfortunately, these methods are inefficient [90, 93, 94] and cannot be used effectively for any biological system because the difference between the prior and posterior is always large for a topology with more than a few parameters and a few data points, and the size of this difference determines the uncertainty in the estimators [93]. Bridge sampling improves on these methods by having one distribution “in between” the prior and posterior to which the prior and posterior are compared, rather than to each other, so that the differences between the compared distributions (and, thus, the variances) are smaller resulting in faster convergence [95]. Other methods, such as Thermodynamic Integration [94, 96, 97], Path Sampling [98], Annealed Importance Sampling [99], and more [100, 101], use even more distributions between the prior and the posterior, so that each comparison is between two quite similar distributions resulting in a variance that is low enough to converge for simple biological topologies [102]. We tried several of these methods but were unable to find one that would converge in a reasonable time for the system we investigated.

Because of this, we developed our own Monte Carlo method for use here. Our method is similar to the one used by Neal [99]. Like almost all methods of this type, ours integrates the marginal likelihood by stepping through a sequence of distributions between the unknown marginal likelihood and a known distribution. Our method uses the linear approximation as the known starting distribution, and the step size from one distribution to the next is generated dynamically to minimize the variance in the answer. A detailed description of our linearized approximation and full Monte Carlo method is provided in the Methods section. The full method was used as the gold standard to which our linearization and other methods were compared.

Because of the computational costs of Monte Carlo methods, approximations to the topology probability are often used instead. The simplest method is to fit each topology to the data and compare the likelihoods of obtaining the data from each topology parameterized by the best-fit parameters [103, 104]. According to this method, a topology that has a higher likelihood has more evidence in its favor. The method is problematic for one main reason: because topologies have different numbers of parameters, and topologies with more parameters can typically fit data better whether or not they are true, this leads to a bias in favor of more complex topologies and an inability to rule out complex topologies if a simpler topology is true.

To compensate for the shortcomings of a simple comparison of likelihoods, several methods have been developed to appropriately penalize topologies with more parameters. The two most popular are the Akaike Information Criterion (AIC) [105] and the Schwarz (or Bayes) Information Criterion (SIC) [106], each justified by a different derivation. These heuristics are no more expensive to compute than the likelihood. One assumption of both heuristics is that sufficient data has been collected to make the parameter uncertainty small [107]. This is not the case for typical biological models fit to typical data, as our work and that of others has found [36, 37, 38, 39, 108]. As a result, the heuristics can be quite inaccurate [109, 110], which is also the case in the current work.

Unsatisfied with the accuracy of the existing heuristics and computational cost of the Monte Carlo methods, we created an approximation to the topology probability problem that provides an accurate answer but at a lower computational cost. We noticed that, if the model has a linear relationship between the parameters and outputs and the measurements have Gaussian noise, the topology probability has an analytical solution. We wondered if there was a way to linearize the nonlinear model such that it provided an effective approximation to the nonlinear answer. In this work, we derive a method to compute the topology probability for a model linearized at the maximum *a posteriori* parameters (the best-fit parameters considering the data and prior).

A detailed derivation is provided in the Methods section of this work, but the key insight in developing this method, visualized in Figure 3-1, is that the marginal likelihood (Equation 3.2) of a linear Gaussian model can be written as:

$$p_{\hat{y}|m}\left(\hat{y},m\right) = p_{\hat{y}|\theta,m}\left(\hat{y},\tilde{\theta}\left(\hat{y},m\right),m\right) \cdot p_{\theta|m}\left(\tilde{\theta}\left(\hat{y},m\right),m\right) \cdot \left\|\tau \cdot V_{\tilde{\theta}}\left(\hat{y},m\right)\right\|^{\frac{1}{2}} \tag{3.3}$$

where $\|X\|$ is the determinant of matrix X, τ is the circle constant equal to $2 \cdot \pi$, $\tilde{\theta}\left(\hat{y},m\right)$ is the maximum *a posteriori* parameter set, and $V_{\tilde{\theta}}\left(\hat{y},m\right)$ is the posterior variance of the parameters. These are terms that can be calculated for a nonlinear model as well; thus, using this equation to compute the marginal likelihood provides a linear approximation of the topology probability.

We demonstrated this method on a set of four candidate topologies of MAPK signaling by Ferrell *et al.* [111]. We generated random data sets by selecting a random topology from the set of four according to a prior, a random parameter set according to a prior, and a random data set by simulating the model and adding noise. We then asked the various methods (Monte Carlo, linearization, likelihood comparison, AIC, and SIC), to determine which topology had generated the data set

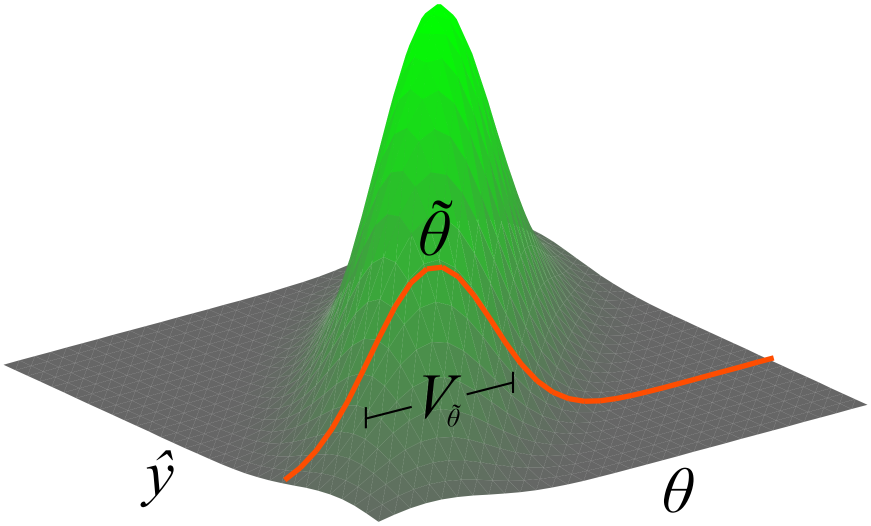

Figure 3-1: **Illustration of linear topology probability.** Here is a plot of the joint probability distribution between the parameter and data point of a one-parameter, one-data-point model. The orange curve has the same shape as the posterior, the probability distribution over the parameters given that a particular data point was observed, but does not have an integral equal to 1, which a true distribution must have. The integral of that curve is the marginal likelihood and the critical component to determining the topology probability. For a linear Gaussian model, the curve has the shape of a Gaussian with a mean at the maximum *a posteriori* parameter set and a variance equal to the posterior variance. Such an expression has an analytical solution to the integral. If the model is nonlinear, then a linearization at the maximum *a posteriori* parameter set will provide a linear approximation to the marginal likelihood.

and compared the accuracy and speed of the methods. The Monte Carlo method gave the most accurate answer, but took significantly more time, while the heuristics took only the time needed to fit the data, but performed only slightly better than random. The linearization method performed almost as well as Monte Carlo but took no longer than the heuristics. These results suggest that this method is an effective tool for topology discrimination for systems biology.

3.2 Methods

3.2.1 Linearization

Important to the linearization method is not just having an analytical solution to the linear model, but writing that solution with terms that can be calculated for the nonlinear model. In this section, we outline the derivation of the analytical solution to the marginal likelihood (Equation 3.2) for a model that has a linear relationship between the parameters and the outputs, which are measured with Gaussian noise superimposed. The likelihood function of a topology with this form is defined by:

$$p_{\hat{y}|\theta,m}\left(\hat{y},\theta,m\right) = N\left(\hat{y},\bar{y}\left(\theta,m\right),V_{\bar{y}}\right) \tag{3.4}$$

where $N\left(\hat{y},\bar{y},V_{\bar{y}}\right)$ is the probability density function of the normal distribution over the data $\hat{y}$ with a mean of $\bar{y}$ (a vector of length $n_{\bar{y}}$) and a variance of $V_{\bar{y}}$ (an $n_{\bar{y}}$ by $n_{\bar{y}}$ matrix). The mean, which can be interpreted as the true value underneath a noisy measurement, is a function of the topology and parameters and, in a linear model, is defined in the following way:

$$\bar{y}\left(\theta,m\right) = A\left(m\right)\cdot\theta + b\left(m\right) \tag{3.5}$$

where $A(m)$ is a matrix $n_{\bar{y}}$ by $n_\theta(m)$ and $b(m)$ is a column vector of length $n_{\bar{y}}$. Together, $A(m)$ and $b(m)$ define the linear topology m. The length of the parameter vector θ depends on the topology. Combining Equations 3.4 and 3.5, we arrive at the likelihood of a linear Gaussian model:

$$p_{\hat{y}|\theta,m}\left(\hat{y},\theta,m\right) = N\left(\hat{y}, A\left(m\right)\cdot\theta + b\left(m\right), V_{\bar{y}}\right) \tag{3.6}$$

We also assume that the prior on the parameters is a Gaussian as well:

$$p_{\theta|m}\left(\theta,m\right) = N\left(\theta,\bar{\theta}\left(m\right),V_{\bar{\theta}}\left(m\right)\right) \tag{3.7}$$

where $\bar{\theta}\left(m\right)$ is the mean of the prior on the parameters for topology m (a vector of length $n_\theta(m)$) and $V_{\bar{\theta}}\left(m\right)$ is the variance (an $n_\theta(m)$ by $n_\theta(m)$ symmetric positive definite matrix).

Substituting the Gaussian definitions for the likelihood and prior into Equation 3.2, we get:

$$p_{\hat{y}|m}\left(\hat{y},m\right) = \int_\theta N\left(\hat{y}, A\left(m\right)\cdot\theta + b\left(m\right), V_{\bar{y}}\right)\cdot N\left(\theta,\bar{\theta}\left(m\right),V_{\bar{\theta}}\left(m\right)\right) \tag{3.8}$$

This integral, the marginal likelihood of a linear Gaussian model, has a well-known analytical solution:

$$p_{\hat{y}|m}\left(\hat{y},m\right) = N\left(\hat{y}, A\left(m\right)\cdot\bar{\theta}\left(m\right) - b\left(m\right), V_{\bar{y}} + A\left(m\right)\cdot V_{\bar{\theta}}\left(m\right)\cdot A\left(m\right)^T\right) \tag{3.9}$$

Nonlinear models are not defined using the $A\left(m\right)$ and $b\left(m\right)$ matrices, so this form is not directly applicable as a linear approximation of nonlinear models. As shown in Appendix 1, this can be rearranged into a convenient form that is the product of the likelihood and prior evaluated at the maximum *a posteriori* parameter set and a term involving the determinant of the posterior variance:

$$p_{\hat{y}|m}\left(\hat{y},m\right) = p_{\hat{y}|\theta,m}\left(\hat{y},\tilde{\theta}\left(\hat{y},m\right),m\right)\cdot p_{\theta|m}\left(\tilde{\theta}\left(\hat{y},m\right),m\right)\cdot\left\|\tau\cdot V_{\tilde{\theta}}\left(\hat{y},m\right)\right\|^{\frac{1}{2}} \tag{3.3}$$

where $\tilde{\theta}\left(\hat{y},m\right)$ is the maximum *a posteriori* parameter set, the best-fit parameters of topology m for data $\hat{y}$, and $V_{\tilde{\theta}}\left(\hat{y},m\right)$ is the inverse of the Fisher information matrix evaluated at the maximum *a posteriori* parameter set. This representation of

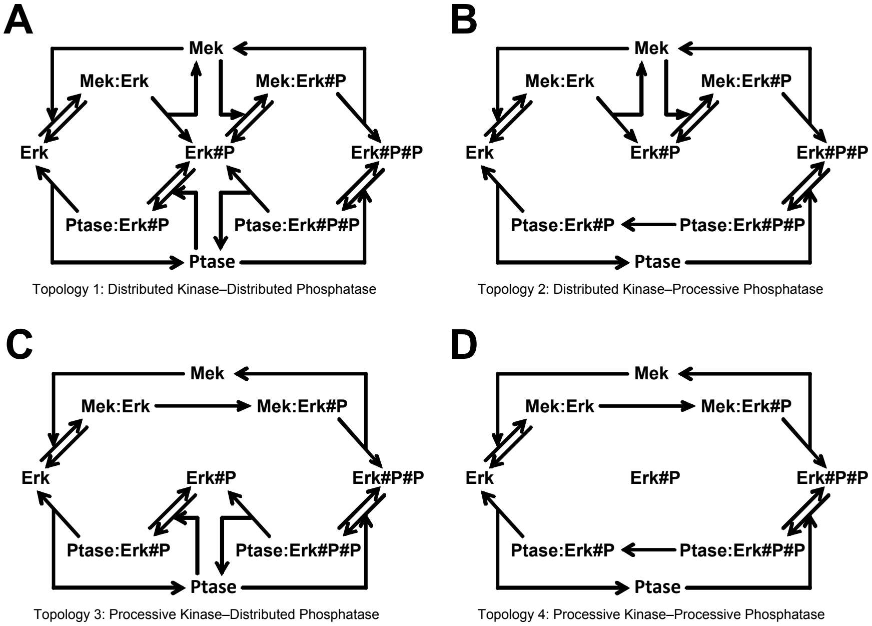

Figure 3-2: **MAPK topologies.** These are the four topologies used in the scenario to generate synthetic data, which was then presented to several topology discrimination methods to determine what the probability was that each topology had generated the particular data set. The suffix "#P" indicates a phosphorylated species.

the marginal likelihood is the central formula to our method. While it is an exact representation for linear models, it is composed of terms that are also well defined for nonlinear models. Since all terms are calculated at the maximum *a posteriori* parameter set, this formula can be interpreted as a linearization at that point. As we show in Results, this turns out to be a powerfully effective approximation for ODE models of biological systems.

3.2.2 Topologies

As our test case, we used four mass-action ODE topologies of MAPK activation [111]. A set of reaction diagrams illustrating these topologies is provided in Figure 3-2. The topologies model the double phosphorylation of Erk by Mek. Each topology has a pair of phosphorylation reactions in which the kinase either binds, phosphorylates once, and falls off before rebinding and phosphorylating a second time (distributive mechanism) or, after initial binding, the kinase phosphorylates once and remains bound until a second phosphorylation occurs (processive mechanism). Each topology also has a pair of phosphatase reactions that follow either the distributive or processive mechanisms like the kinase, falling off or remaining bound between reactions. The four possible combinations of these two mechanisms for these two enzymes constitute the four topologies used in this work.

The four model topologies have 12, 10, 10, and 8 parameters in the respective order they will be listed throughout this work and shown in Figure 3-2. Each distributive mechanism has two additional parameters for the on and off rates of enzyme rebinding that don't exist for the corresponding distributive topology. Each topology has 8 species, although in topology 4 (processive/processive) the free singly phosphorylated state is not populated. Each topology has 1 input, the amount of kinase, which has a constant value of 1 μM. The initial amount of substrate is 2 μM, the initial amount of phosphatase is 1 μM, and all other initial amounts are 0 μM. These values are comparable to experimental procedures of Ferrell *et al.* [45].

There are three outputs, the amounts of unphosphorylated substrate, singly phosphorylated substrate, and doubly phosphorylated substrate. The outputs include the amounts of that species that are free or are bound in a complex with the kinase or phosphatase.

3.2.3 Scenario

We set up a computational scenario to generate many data sets from the topologies so that we could interrogate several methods of topology discrimination to determine how well each performed. To generate each data set, a topology was chosen randomly from a uniform distribution (all four topologies were equally likely to be chosen) and the topology was parameterized with random parameters chosen from a multivariate log-normal distribution with a geometric mean of 0.1 and an independent geometric variance such that the 95% confidence intervals stretched 100-fold above and below the geometric mean. This meant that each parameter was essentially chosen over a range of four orders of magnitude.

Each randomly drawn model was simulated for 100 minutes and the three outputs were measured at 12.5, 25.0, 37.5, 50.0, 62.5, 75.0, 87.5, and 100.0 min. Each measurement had Gaussian error added to it with a standard deviation equal to 10% plus 0.01 μM. The resulting noisy measurements were floored at 0 (negative values were moved to zero). By measuring the sum of phosphorylation sites across the complexes in which they appear and by only measuring at 8 time points, we intended to represent the modern measurement capabilities of mass spectrometry [112].

This scenario was repeated 1000 times to generate that many random models with that many random data sets.

3.2.4 Monte Carlo

The various Monte Carlo methods used to solve this problem are all similar in that they compare the unknown likelihood function to a known likelihood function by sampling from one and comparing the sample in some way to the other [91, 92, 94, 98, 99]. To be a known likelihood function means that its normalization factor, the integral over all space, is known. The method we use in this work has some conceptual similarity to the Annealed Importance Sampling method [99], but is procedurally very different.

To use importance sampling to determine the normalization constant z_1 of a distribution determined by likelihood function l_1, we sample from a distribution determined by likelihood l_0 with known normalization constant z_0 and use the following formula to estimate the ratio of the normalization constants:

$$\frac{z_1}{z_0} \approx \hat{w} = \frac{1}{n} \sum_i \frac{l_1\left(\theta_i\right)}{l_0\left(\theta_i\right)} \tag{3.10}$$

where each θ_i is one of n random parameter sets drawn from the distribution represented by l_0. The uncertainty in this estimator is:

$$\sigma_{\hat{w}} = \sqrt{\frac{1}{n-1} \sum_i \left(\frac{l_1\left(\theta_i\right)}{l_0\left(\theta_i\right)} - \hat{w} \right)^2} \tag{3.11}$$

The convergence of this estimator is dependent on the amount of overlap between the known and unknown distributions. If the distributions are similar, the estimator will converge quickly. If the distributions are very different, the estimator will converge slowly. To ensure that the distributions are similar enough, we used a sequence of distributions between the known and unknown distribution defined by the formula:

$$l(\theta, \beta) = l_0(\theta)^{1-\beta} \cdot l_1(\theta)^{\beta} \tag{3.12}$$

which, by tuning β, gradually transforms the known distribution at $\beta = 0$ into the unknown distribution at $\beta = 1$.

For the known distribution, we used a linear Gaussian approximation of the posterior by using a nonlinear fitting algorithm to find the maximum *a posteriori* parameter set (the best-fit parameters) and the Fisher information matrix evaluated at the best-fit parameter. The best-fit parameters became the mean and the inverse of the Fisher information matrix plus the inverse of the prior variance became the variance of a log-normal distribution in parameter space that served as the known, starting distribution of the Monte Carlo procedure.

The final piece of the transformation process is the schedule on β to transform the known distribution into a sequence of unknown distributions culminating in the final unknown distribution. Again, there are many ways to select the points between 0 and 1. The most basic method, a uniform spacing did not allow the Monte Carlo method to converge because the distribution changed far more near the ends than near the middle (data not shown). For example, a change from 0% to 1% or 99% to 100% unknown distribution was a far greater change than going from 49% to 50%. As a result, the importance sampling estimates near the ends had very large uncertainties, but making the steps fine enough to reduce the uncertainty resulted in many wasteful estimates being made of the low-uncertainty middle region. To ensure that each step had a reasonably low variance, we started from $\beta = 0$ and determined the next value of β by generating a small sample from the distribution defined by the current value of β and finding, via Matlab's numerical root finder `fzero`, the value of the next β that would result in a desired sample standard deviation. We chose 0.2, or 20%, as the desired sample standard deviation of each step.

The importance sampling at each span provides an estimate to the change in the integral across that span and an uncertainty in that estimate. The estimates are combined by a simple product:

$$\hat{w}_{final} = \prod_{j} \hat{w}_j \tag{3.13}$$

where j is an index over each bridge point. (Because of the limitations of floating point arithmetic, these calculations were actually performed in log space and exponentiated to get the final answer.) The uncertainty in this estimate can be computed by the linear propagation of uncertainty, but in working with this system we found that this dramatically overestimated the uncertainty (data not shown). So we used bootstrap resampling instead. We resampled with replacement each bridge point and recomputed the estimate of the integral. This resampling was repeated 100 times, the sample standard deviation of the recomputed integrals was used as the uncertainty in the integral.

The sampling of the posterior was done using the Metropolis-Hastings algorithm [26, 27]. At each bridge point, the sampling was started at the maximum *a posteriori* parameter set. The proposal distribution of the algorithm was a log-normal distribution with a geometric mean of the current point and a geometric variance equal to the inverse of the Fisher information matrix plus the inverse of the prior variance computed at the starting point of the sampling multiplied by 5.66 divided by the number of dimensions [113]. The log-normal distribution was truncated below 1×10^{-10} and above 1×10^{8} to reduce the chance of drawing an extreme parameter set that could destabilize the integrator. The sampling was thinned by saving only every fifth point, and the sampling was restarted every 100 samples after thinning using an updated proposal variance. The autocorrelation in each parameter was computed with Matlab's `autocorr` function. The sampling was thinned further using the smallest step size such that the estimated autocorrelation in every parameter was less than 0.05. To ensure that the estimate of the autocorrelation was itself accurate, the autocorrelation step size was not trusted until the total length of the sample used to compute the autocorrelation was 20 times longer than the step size.

3.2.5 Akaike Information Criterion

The Akaike Information Criterion (AIC) [105] is a popular heuristic for topology discrimination:

$$AIC\left(m,\hat{y}\right)=2\cdot n_{\theta}\left(m\right)-2\cdot\log p_{\hat{y}|\theta,m}\left(\hat{y},\hat{\theta},m\right) \tag{3.14}$$

which evaluates the log likelihood at the best-fit parameters and adds a penalty proportional to the number of parameters. To plot the relative evidence, we return the AIC to probability space:

$$p_{AIC}\left(m,\hat{y}\right)=\frac{\frac{p_{\hat{y}|\theta,m}\left(\hat{y},\hat{\theta},m\right)}{\exp(n_{\theta}(m))}}{\sum_{i}\frac{p_{\hat{y}|\theta,m}\left(\hat{y},\hat{\theta},i\right)}{\exp(n_{\theta}(i))}} \tag{3.15}$$

The ranking of topologies under this metric is the same, but makes the values comparable to the Monte Carlo and linear methods.

3.2.6 Schwarz Information Criterion

The Schwarz (or Bayes) Information Criterion (SIC) [106] is another popular heuristic for topology discrimination:

$$SIC\left(m,\hat{y}\right)=n_{\theta}\left(m\right)\cdot\log\left(n_{\bar{y}}\right)-2\cdot\log p_{\hat{y}|\theta,m}\left(\hat{y},\hat{\theta},m\right) \tag{3.16}$$

which differs from the AIC only by the size of the penalty. Both use the log likelihood of the best-fit parameters, but the SIC penalizes the topologies with more parameters more strongly. This metric can be transformed into parameter space in a similar way to the AIC:

$$p_{SIC}\left(m,\hat{y}\right)=\frac{\frac{p_{\hat{y}|\theta,m}\left(\hat{y},\hat{\theta},m\right)}{n_{\bar{y}}\cdot\exp(n_{\theta}(m))}}{\sum_{i}\frac{p_{\hat{y}|\theta,m}\left(\hat{y},\hat{\theta},i\right)}{n_{\bar{y}}\cdot\exp(n_{\theta}(i))}} \tag{3.17}$$

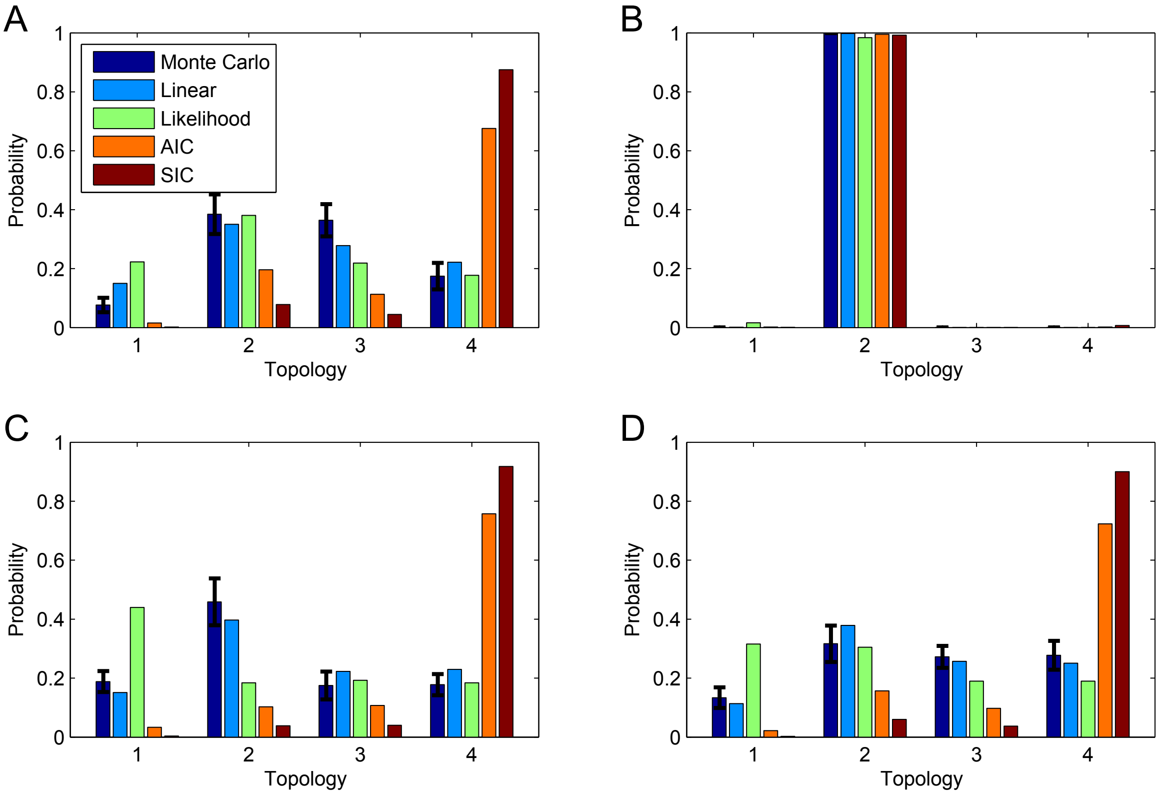

Figure 3-3: **Typical results.** The topology probability according to each of the five methods is shown for four example data sets. The synthetic data underlying A, B, C, and D were generated by topologies 1, 2, 3, and 4, respectively. The error bars on the Monte Carlo method are the standard error on the mean as computed by bootstrap resampling.

3.3 Results

We generated 1000 data sets from 1000 random parameterized topologies and asked each of the methods to determine the relative evidence that each topology had generated the data, quantified as a probability distribution over the four candidate topologies. These probability distributions were compared to each other and, in particular, to the Monte Carlo result, which should have converged to the correct probability distribution.

We show four of the thousand runs in Figure 3-3 to illustrate typical results seen. The true topologies underlying Figures 3-3A, 3-3B, 3-3C, and 3-3D were topologies 1, 2, 3, and 4, respectively. The results for our scenario can be classified into two main cases. The less common case, represented by Figure 3-3B, is the case where the data unambiguously indicate the true topology; in this case, it was topology 2. When only one topology can fit the data, with the ability to fit the data indicated by the "Likelihood" bars, then all methods agree that the topology that fits is the correct topology. The more common case is represented in Figures 3-3A, 3-3C, and 3-3D. Here, all the topologies can the fit the data to some degree and the different methods give different probability distributions on the data. In these cases, one can see that the likelihood method tends to overstate the true probability, given by the "Monte Carlo" bars, for topology 1, which has the greatest number of parameters. Interestingly, the AIC and SIC methods show a strong bias in favor of topology 4, which has the fewest parameters. However, it can be seen that the linearization method is quite close to the Monte Carlo method in each case, suggesting that it is a good approximation. If one were to look at just one result, for instance Figure 3-3D, it may appear that the AIC and SIC are the superior methods because they are the only ones that put the highest probability on the true topology, topology 4. However, this would be misleading, because they frequently put a high probability on topology 4, even when it is not the topology that generated the data (Figures 3-3A and 3-3C). In fact, even in Figure 3-3D, they are overstating the evidence that topology 4 is true, for the actual probability is provided by the Monte Carlo.

For each of the 1000 runs, we calculated the Jensen-Shannon (JS) divergence between the probability distribution given by each method and the Monte Carlo probability distribution. The JS divergence is one standard measure of how different two probability distributions are, which in this case provides a single quantification for how far each method's answer is from the correct answer. The JS divergence returns a value between 0 (identical distributions) and 1 (non-overlapping distributions). The divergence values for each method over all runs were binned and plotted as a histogram in Figure 3-4. Of the other methods, the linearization method is closest to the Monte Carlo. The likelihood comparison was the next closest, followed by the AIC and the SIC.

While the JS divergence is one measure of how different one probability distribution is from a reference distribution, it does not report numbers that can easily be

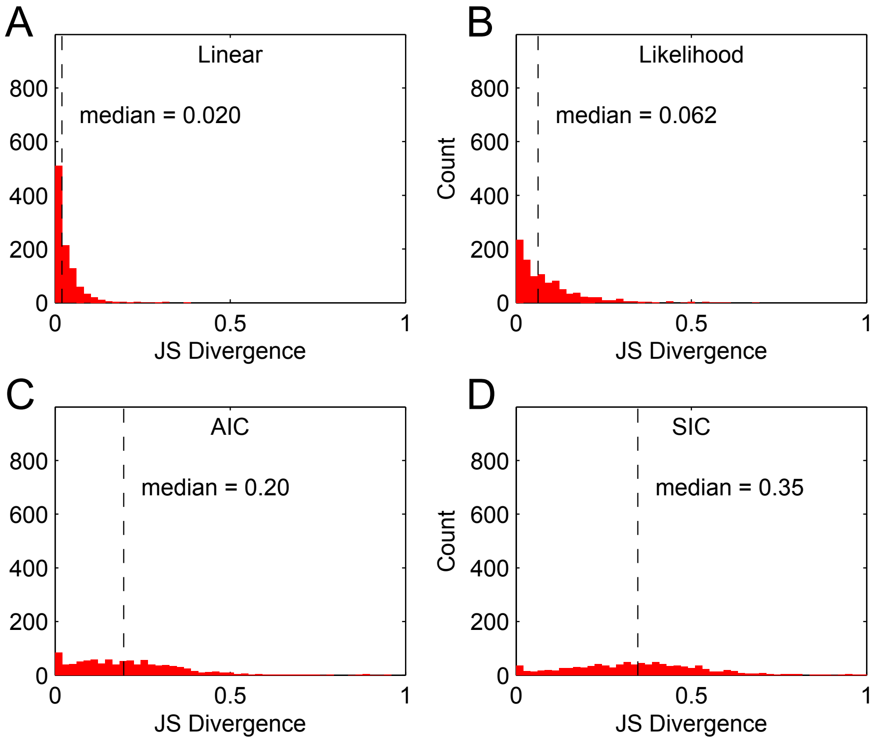

Figure 3-4: **Divergence of each method from the gold standard.** The Jensen-Shannon (JS) divergence measures the difference between two distributions on a scale of 0 to 1, which ranges from identical to no overlap, respectively. The divergence between the topology probability supplied by each method and the gold standard Monte Carlo were computed for all 1000 data sets, sorted into 50 evenly spaced bins, and plotted as histograms. For reference, the median residual divergence in the Monte Carlo from the true probability distribution was estimated with bootstrap resampling to be 0.0061.

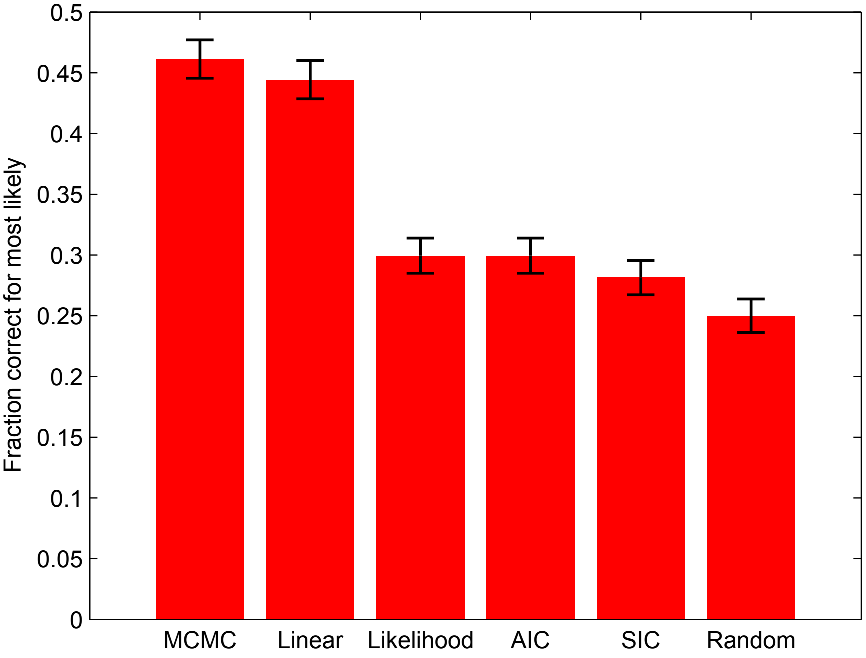

Figure 3-5: **Accuracy of the most probable topology.** For all 1000 data sets, the most likely topology according to each method was compared to the actual topology that generated the data. The fraction that each method found correct is plotted here. The error bars are the standard error of the mean.

used to understand if the error in each approximation is large enough to matter. To aggregate the results in a way that was easier to interpret, we took the most likely topology according to each method and compared it to the topology that actually generated the data. In the real world, we would not be able to do this test because the true topology would be unknown, but this computational scenario allows us to investigate whether the methods actually do what they are intended to do—tell us which topology is correct according to the data. We computed the fraction of top hits that were correct for each method (Figure 3-5). As expected, the Monte Carlo was correct most often; the most likely topology according to this method was the true topology 46% of the time. Because Monte Carlo provides the correct probability, it is impossible to do better than this provided that the Monte Carlo has converged and a sufficiently large number of runs are done to approach statistical averages. No method could pick the correct topology 100% of the time because that information was not contained in the data. The linearization method did almost as well as Monte Carlo, finding the correct topology 44% of the time. The likelihood comparison, the

AIC, and the SIC were correct 30%, 30%, and 28% of the time, respectively. Surprisingly, these heuristics only do slightly better than randomly guessing one of the four topologies, which would be correct 25% of the time.

We analyzed the bias in each method by plotting the mean probability each method returned for each topology (Figure 3-6). An unbiased method will return a mean of 0.25 for each topology because that is the probability by which each topology was drawn. The bias that the likelihood comparison has for the topology with the most parameters can be seen though it is not particularly large. Interestingly, AIC and SIC are strongly biased toward the topology with the fewest parameters. The Monte Carlo method has no bias, as expected, but neither does the linearization, which is a satisfying result.

Despite the improved accuracy of linearization, the method does not take substantially greater computational resources than the heuristics, which take radically less time to compute than the full Monte Carlo. While the Monte Carlo method took a median of 13 days to complete, the linearization method, likelihood comparison, AIC, and SIC all took a median of 4.2 minutes to complete. The fast methods took the same amount of time to complete because the time of each was dominated by the time it took to simply fit parameters for each of the topologies to the data. The computation of the likelihood (needed for all methods) and the Fisher information matrix (needed for the linearization method) took about as much time as a single iteration of the gradient descent fitting algorithm. Computing the Fisher information matrix requires computing the sensitivities of the outputs to the parameters, which is not needed to compute a likelihood comparison, the AIC, or the SIC and is more expensive that simply simulating the system to compute the likelihood of the data. If the time to fit the topologies to the data is ignored, it took a median of 0.80 seconds to compute the likelihood comparison, AIC, and SIC and 3.4 seconds to compute the linearization method. Thus, the linearization was slightly more time consuming than the other fast methods, but insignificantly so.

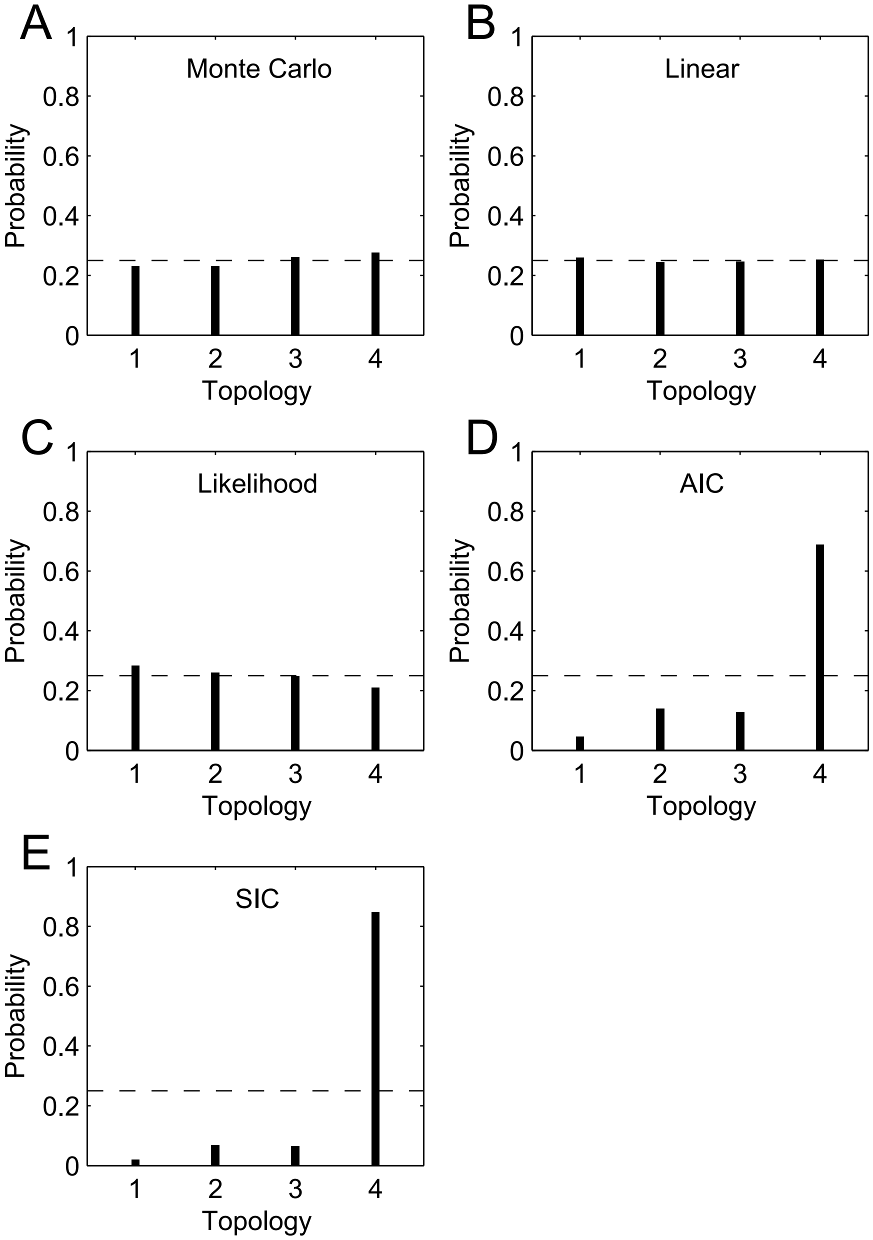

Figure 3-6: **Bias in methods.** The mean topology probability distribution was taken over all 1000 runs. Because all topologies were drawn with equal probability, the mean probability distribution should be uniform if the method is unbiased (dashed line). The linearization method shows this lack of bias as does the Monte Carlo method. The likelihood method is expected to have a bias toward the topology with the most parameters (topology 1) and against the topology with the fewest parameters (topology 4), which is visible but slight. A strong bias in favor of topologies with fewer parameters can be seen with the AIC and SIC methods. The number of parameters in topologies 1, 2, 3, and 4 are 12, 10, 10, and 8, respectively. (A) Monte Carlo method, (B) the linearization method developed here, (C) likelihood method, (D) Akaike Information Criterion method, and (E) Schwarz Information Criterion method.

3.4 Conclusion

The quantification of parameter uncertainty in ODE models of biological systems has a number of successful and computationally feasible methods [101, 36, 37, 108, 114]. However, doing the same for the other half of the model, the topology, has not been as successful. The existing methods are either expensive (Monte Carlo methods) or inaccurate (various heuristics). We have proposed one method, our linearized Bayesian approach, which may fill this gap. It returns an answer that is similar to the Monte Carlo gold standard, but does so at a computational cost no more than fitting the topologies to data.

There are several ways to interpret what the method is doing. The simplest one and the one we have used throughout this work is that it is a linearization at the maximum *a posteriori* parameter set, because we arrive at this parameter set with nonlinear fitting and then evaluate the likelihood, the prior, and the Fisher information matrix with these parameters. These values are then plugged into a formula that is exactly true only for linear Gaussian topologies. Another interpretation is that the integrand of the marginal likelihood equation has been replaced by a Laplace approximation. A Laplace approximation is like a second-order Taylor approximation except that an exponential of a polynomial is used rather than a polynomial itself [115]. A Laplace approximation generates a single Gaussian at a point to approximate the rest of the function. This interpretation has one additional caveat: instead of the second-order derivative of the log likelihood with respect to the parameters (also known as the Hessian), we use the Fisher information matrix, which is only exactly equal to the Hessian if the model is linear. Computing the Hessian takes greater computational resources, yet has little impact on the result (data not shown). The use of the Hessian and Fisher information matrix in the Laplace approximation of marginal likelihoods even has some use in other fields [116].

The number of possible topologies grows exponentially with the number of states. The linearization method would not be very effective at reverse engineering the topology from scratch because the method considers each topology individually. However, the method could work effectively as a subsequent step to other methods that efficiently pare down the vast topological space to a manageable number of topologies. As long as the number of topologies is small enough such that each can be fit to data, possibly in parallel, then the linearization method would efficiently quantify the uncertainty in the remaining set.

Because the problem is phrased in a Bayesian fashion, the probability distribution returned by the linearization method sums to 1. This means that, like all Bayesian methods, it is implicitly assumed that the true topology is in the set of possibilities. The possibility that no topology is a good fit for the data can be mitigated by checking after the fact that there is one at least one topology that fits the data by using a frequentist statistic, such as a chi-square p-value.

In this work we have demonstrated the effectiveness of the approximation only on a single set of simple biological topologies. Testing on more systems, especially more complex systems, is warranted. The main limitation with our testing scenario in evaluating the method on more complex topologies was that the Monte Carlo method already took 13 days to complete. A noticeably more complex set of topologies would not finish in a reasonable amount of time, so that there would be no gold standard with which to compare. Perhaps this illustrates why a good approximation of the topology probability is so important: most of the models that biologists care about are too large to compute the topology probability with a Monte Carlo method.

The approximation is dependent on the "area" under the hyperdimensional Gaussian being similar to the "area" under the product of the likelihood and the prior, which has the shape of the parameter posterior distribution. If the region of probable parameters is substantially larger or smaller than the approximation, the approximation will fail unless the difference is similar for all topologies. It may be interesting to note that the linear Gaussian approximation does not have to be very similar to the true distribution; it only has to have a similar integral. This may be an important property because the posterior parameter uncertainty is typically very large for biological models. When the uncertainty is large, there will be regions of likely parameter sets that a linear approximation will not recognize as likely parameters because the linear approximation is only valid for a short range. Fortunately, the linear approximation does not actually have to overlay the underlying posterior distribution in order to be a good approximation for the purpose of topology probability; it only has to have a similar integral.

Chapter 4

An Optimal Experimental Design Method for Efficiently Reducing Topology Uncertainty

4.1 Introduction

The building of accurate models is one of the goals of systems biology. Scientists desire accurate models to understand the behavior of the system, make predictions under new conditions, and aggregate knowledge of the components. A thorough understanding of a biological system requires knowing both the topology and the parameters. Gathering data provides evidence for some models at the expense of others, but because of uncertainty in the measurements, there will be uncertainty in the parameters and topology, which can be statistically quantified. Once the uncertainty is known, the goal is to reduce that uncertainty to an acceptable level. The only way to reduce the uncertainty is to collect data from additional experiments. Some experiments will reduce the uncertainty more than other experiments. Optimal experimental design is the process of determining which experiments are likely to reduce the uncertainty the most before the experiments are actually performed.

We consider ordinary differential equation (ODE) models of biological systems.

In this framework, the topology is the set of equations. If it is unknown which reactions take place, then there will be multiple possible sets of ODEs to describe the system. If the experimental data only supports one of the possible topologies, then one can consider the topology uncertainty to be low. However, it is usually the case that the existing data is consistent with multiple topologies. There is a diversity of methods to quantify the evidence in favor of one topology over another. The most common methods include the Akaike Information Criterion (AIC) and the Schwarz (or Bayes) Information Criterion. Also used are Bayesian Monte Carlo methods, such as Thermodynamic Integration [94, 96, 97], Path Sampling [98], and Annealed Importance Sampling [99]. Recently, we developed a method based on linearization that approximates the Bayesian topology probability, which was successful in giving a result close to the Monte Carlo result but at a computational cost comparable to the common heuristics [117]. Once the uncertainty in the topology is quantified, additional experiments are necessary to reduce the uncertainty. Different experiments will have different utility in distinguishing between topologies. Being able to predict the quality of experiments before doing them is useful and fundamentally centered on exercising the system under conditions that will give behaviors that are different depending on which topology is true.

The earliest experimental design methods for topology discrimination in chemical dynamics, such as the one by Hunter and Riener [118], simply fit the topologies to the data, simulate the models under different experiments, and rank the experiments according to which had the greatest sum of square difference between the simulations at the measurement points. This criterion was later generalized to allow for different weights on each measurement. These weights could either be proportional to the magnitude of the measurement [119], be proportional the magnitude of the uncertainty in the measurement, or give similar weight to all types of measurements [120]. Previous work by our group found that different topologies often required different time-varying inputs to drive the outputs to particular values and suggested that these input profiles may be good at distinguishing between the topologies [121].

Because of measurement noise, the exact measurement value cannot be predicted even assuming that a particular model is true. Instead, each measurement has a different probability distribution depending on which topology is true. More recent methods measure the overlap in these distributions, and rank experiments favorably if they have many measurements with little overlap. The method by Skanda and Lebiedz, which sums the KL-divergence over all measurement points, is one such technique [122]. The methods that use only the sum of square distance between the measurements can be interpreted as a measurement of overlap if one assumes only Gaussian measurement noise and a weight that is proportional to the uncertainty in those measurements.

The best-fit parameters of each topology are not the only parameters to consider, and some existing methods take the parameter uncertainty into account. The method by Chen and Asprey [123], which extends the method by Buzzi-Ferraris and Forzatti [124], computes the ratio of the distance between predicted measurements to the uncertainty in those measurements, ranking experiments with greater separation more highly. This method includes the contribution of parameter uncertainty to the measurement uncertainty, ensuring that the best experiments would separate the topologies regardless of their parameters.

The common goal of these methods is to find experiments where the topologies give the greatest separation in the predicted measurement values. Some may take the measurement noise into account, or the parameter uncertainty, but the framing is ultimately a frequentist one. The evidence for a topology is defined as being how well it fits the data, and a good experiment is defined by how many topologies cannot fit the data. There are several well-known problems with using a frequentist approach to distinguish between topologies. Most notable is that topologies with more parameters tend to fit the data better than topologies with fewer parameters regardless of whether or not the topology with more parameters is correct. In the case of nested topologies, where a simpler topology is a special case of a more complex topology, the more complex topology will always have at least as much evidence for it as the simpler one because there is no data generated by the simpler topology that the more complex cannot fit equally well. Working in a Bayesian framework provides a way to compensate for this and works by computing the probability distribution over the set of possible topologies according to the data.

The difficult step in computing the topology probability is the marginal likelihood, which requires integrating the likelihood over all parameter space. This integral does not have an analytical solution for mass-action models, so it must be computed by an expensive Monte Carlo method or approximated by some other method. A method by Busetto *et al.* maximizes the expected divergence between the initial probability distribution and the probability distribution after collecting data from a candidate experiment [125]. The method uses Sequential Monte Carlo or unscented Kalman filtering to perform the integration over parameter space.

Our group recently developed a method for approximating the topology probability in a biological system by linearizing the system at the maximum *a posteriori* parameter set of each topology [117]. Because this method provides the topology probability at minimal computational cost, we wanted to see if it was effective for optimal experimental design for topology uncertainty. The optimal experimental design method outlined in this work minimizes the expected entropy in the probability distribution after performing the experiment. Entropy is one way to quantify the "peakiness" of a distribution; if the distribution is flat with all values being equally likely, then the entropy and topology uncertainty are high, while a peaked distribution with some values being likely and others being very unlikely, then the entropy and topology uncertainty are low. Naturally, the posterior probability distribution over the set of topologies is dependent on what data is obtained from the experiment. But what data is obtained depends on the topology, which is unknown, and the parameters, which are also unknown, and the measurement noise. In order to be an effective experimental design method, it needs to predict the resulting entropy without having to first run the experiment and collect the data. The method described in this work finds the expected entropy by repeatedly drawing a random topology, a random parameter set, and simulating the candidate experimental conditions to provide a random data set. The topologies are fit to each of these data sets and the topology probability is computed via the linearization method. The entropy of each probability distribution is computed, and the sequence is averaged, providing an expected entropy in a Monte Carlo fashion.

We tested this method on a set of topologies based on the model by Chen *et al.* describing the ErbB pathway [58]. This mass-action model includes the main features of the ErbB system. The extracellular ligand EGF binds to receptor tyrosine kinase ErbB1, and extracellular heregulin binds to ErbB3 and ErbB4. When ligand is bound, these receptors and ErbB2 form various homodimers and heterodimers. Through dimerization, the receptors phosphorylate each other. Adaptor proteins bind to the phosphorylated sites, which then activate various downstream proteins. Not all possible dimers were allowed to form in the model for various reasons. One interesting case was that homodimerization of ErbB3 was not modeled because ErbB3 has an inactive kinase so that it cannot phosphorylate itself, and so contributes nothing to the downstream effects. While the complete inactivity of the ErbB3 kinase has been recently questioned, it undoubtedly has a far weaker kinase than the other receptors [126, 127]. But even if its kinase is inactive, dimerization of ErbB3 could still affect signaling by sequestering ErbB3 away from other receptors, preventing it from becoming phosphorylated and allowing the other receptors to find more active partners. We were interested in understanding how experimental perturbations could be used to unambiguously determine topological features of biological networks.

We applied our optimal experimental design algorithm for reducing topology uncertainty to this system using synthetically generated data, examining a set of candidate experiments and determining the expected entropy in the topology probability distribution. We found that our simple set of candidate experiments which observed downstream effects had a mild ability to distinguish between the topologies—some experiments could rule out a couple of topologies, but no single experiment could conclusively determine the topology. We observed that heregulin, the ligand which activates ErbB3 dimerization, was necessary to provide any distinguishing power between the topologies about ErbB3 dimerization.

We studied the terms of the linearized Bayesian topology probability in order to understand what part of the model was important for distinguishing between the topologies and found that one such part was the parameters to which the measurements were sensitive in every topology. The parameters that had this property in this scenario tended to be those that controlled a broad portion of the network, not necessarily those parameters closest to the point in the network where the topologies differed.

4.2 Methods

4.2.1 Scenario Overview

We tested our method by assuming that the model with dimerization of ErbB3 when heregulin was bound was the true model, which we used to generate synthetic data. Starting data was generated using a nominal experiment and all topologies were fit to the data. This data did not distinguish between the topologies, so we queried the optimal experimental design algorithm to predict the ability of member of a set of 408 candidate experiments to reduce the uncertainty in the topology.

4.2.2 Topologies

The base topology was a model of ErbB signaling by Chen *et al.* [58]. It is a mass-action ODE model with 3 inputs, 500 states, 167 kinetic parameters, and 27 seed parameters. At the top of the pathway are all four of the ErbB receptor tyrosine kinases, ErbB1, ErbB2, ErbB3, and ErbB4. The first input, EGF, binds to ErbB1, activating it through inducing homodimerization and heterodimerization with other activated receptors. The second input, heregulin, binds ErbB3 and ErbB4 and activates them in the same way. The third input is a competitive inhibitor for ATP binding to the receptor kinases. A bound inhibitor prevents the kinases from performing their primary function, which is to phosphorylate the other receptor to which they are dimerized. Several adaptor proteins, such as Grb2, Shc, Sos, and Gab1, bind the phosphorylated receptors leading to cascades of activation to downstream proteins, such as Ras, Raf, Mek, Erk, PI3K, and Akt.

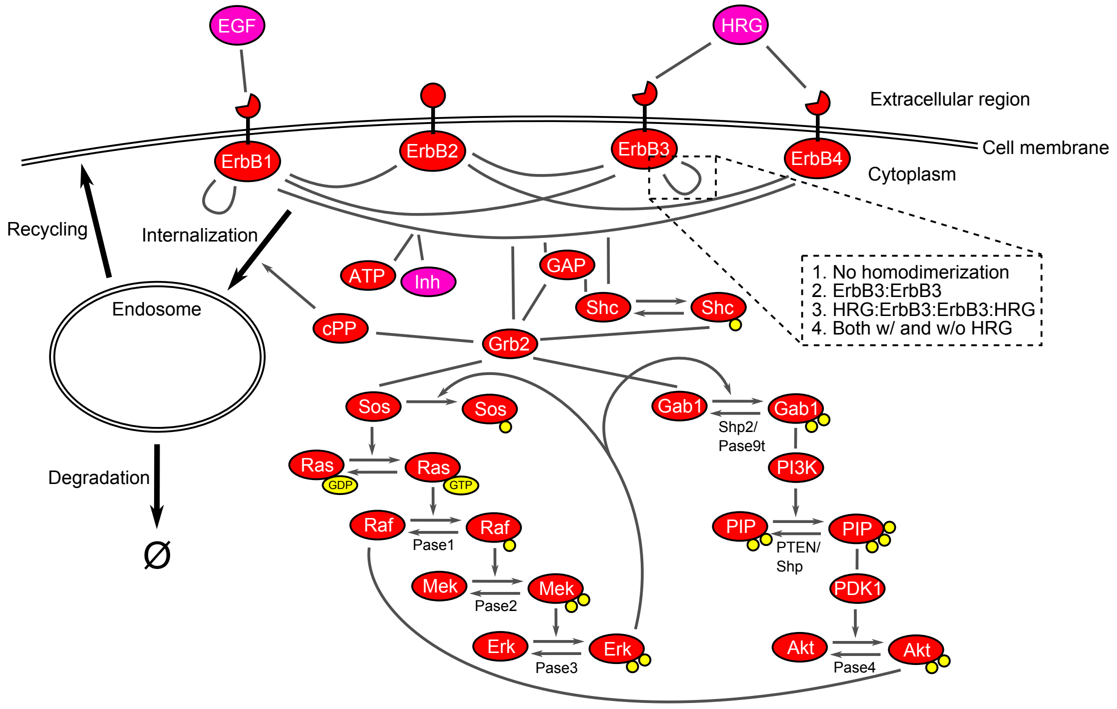

Figure 4-1: **The topologies.** This is a simplified representation of the ErbB model used in this work. The four topologies in question differ at the point of ErbB3 dimerization, highlighted by the dashed-line box. Grey lines indicate bonding between two monomers. Grey arrows from a monomer to a monomer indicate a conversion, while grey arrows from a monomer to another arrow indicate a stimulation of that conversion. Pink monomers are inputs to the model. Not shown is phosphorylation of the receptors when dimerized with ErbB1, ErbB2, or ErbB4. ErbB3 dimers are unique (when present in a topology) in that no phosphorylation occurs. The top adaptors Grb2, Shc, and GAP bind to various phosphorylated dimers, even though only one line is shown for each. All dimers can be internalized and degraded.

The topologies differed in the circumstances under which ErbB3 could dimerize. A simplified diagram of the topologies is provided in Figure 4-1. The first topology was the original topology, with ErbB3 being unable to homodimerize under any conditions. In the second topology, ErbB3 could dimerize without heregulin bound. This is not believed to occur in nature but was included as an alternative hypothesis for our method to detect. Because ErbB3 has an inactive kinase, this led to no downstream signaling by itself. The third topology had ErbB3 dimerizing upon binding heregulin. As this is the mechanism believed to actually occur, this topology served as the "true" topology, generating the synthetic data. The fourth topology allowed dimerization between ErbB3 both with and without heregulin binding—a combination of the second and third topologies.

4.2.3 Experiments

Synthetic data was generated by simulating the true topology under given experimental conditions and generating noisy data. Each experiment was run for 120 minutes. For the candidate experiments, the concentration of EGF, heregulin, and inhibitor could be either 0 or 1 nM. The nominal experiment used 5 nM of EGF and 0 nM of the others. The model has 42 outputs which are the unphosphorylated and phosphorylated forms of every protein monomer in the network. Each output was the sum of all species containing that form of the monomer. All outputs were measured at 8 time points: 2.5, 5.0, 7.5, 10, 15, 30, 60, and 120 minutes. The system was run to equilibrium with no stimulation before the experiment began.

The model had 25 protein monomers with non-zero initial conditions, which were the seed parameters of the model. Each candidate experiment could have the published seed parameters, or it could have one protein knocked-down, represented by a 10-fold reduction in the appropriate initial condition, or it could have one protein over-expressed, represented by a 10-fold increase in the initial condition. All possible combinations of the knock-downs, over-expressions, and input values resulted in 408 candidate experiments.

4.2.4 Priors

The prior on the topology probability was a uniform density, giving 0.25 to each of the four topologies.

The prior on the parameters was derived from the parameters of the published model. The parameters were divided into three categories based on the type of reaction they were involved in: unimolecular reactions, intracellular bimolecular reactions, and extracellular bimolecular reactions. The parameters of the unimolecular reactions had a geometric mean of 8.70×10^{-2} and a geometric standard deviation of 3.00×10^{-1}. The parameters of the intracellular bimolecular reactions had a geometric mean of 4.32×10^{-6} and a geometric standard deviation of 1.52×10^{-5}. The parameters of the extracellular bimolecular reactions had a geometric mean of 1.22×10^{4} and a geometric standard deviation of 7.47×10^{4}. Here, the geometric standard deviation is defined as the square root of the product of the variance and the geometric mean. The distribution of the parameter values in each group was well represented by a log-normal distribution (data not shown). A multivariate log normal distribution with these means and standard deviations was used for the parameter prior.

4.2.5 Topology Probability

The probability distribution over the set of topologies according to a data set was computed using linearized topology probability at the maximum *a posteriori* parameters [117]. This method is used to solve the topology probability according to Bayes rule:

$$p_{m|\hat{y}}\left(m, \hat{y}\right) = \frac{p_m\left(m\right) \cdot p_{\hat{y}|m}\left(\hat{y}, m\right)}{\sum_i p_m\left(i\right) \cdot p_{\hat{y}|m}\left(\hat{y}, i\right)} \tag{4.1}$$

where $p_{m|\hat{y}}\left(m, \hat{y}\right)$ is the posterior probability for topology m given that data $\hat{y}$ has been observed, $p_m\left(m\right)$ is the topology prior of topology m, and $p_{\hat{y}|m}\left(\hat{y}, m\right)$ is the marginal likelihood of data $\hat{y}$ given model m.

The marginal likelihood is the probability that a set of data would be observed under a particular topology. Because topologies alone do not generate data (parameterized topologies do) the average probability over all parameters weighted by the

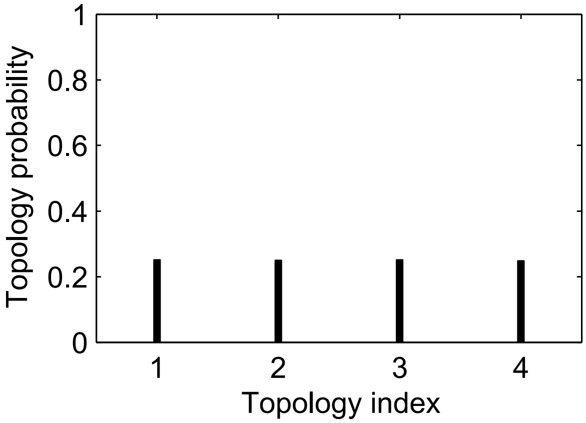

Figure 4-2: **Topology probability after nominal experiment.** After fitting to the nominal experiment, which was treatment with EGF and no knock-downs or over-expressions, the probability of each topology was roughly equal.

prior on the parameters is computed by an integral over parameter space:

$$p_{\hat{y}|m}\left(\hat{y},m\right)=\int_{\theta}p_{\hat{y}|m,\theta}\left(\hat{y},m,\theta\right)\cdot p_{\theta|m}\left(\theta,m\right) \tag{4.2}$$

where $p_{\hat{y}|m,\theta}\left(\hat{y},m,\theta\right)$ is the likelihood of data $\hat{y}$ being produced by model topology m parameterized with values θ and $p_{\theta|m}\left(\theta,m\right)$ is the parameter prior for parameter values θ in model topology m.

For a linear Gaussian topology with a Gaussian prior, the marginal likelihood can be computed analytically:

$$p_{\hat{y}|m}\left(\hat{y},m\right)=p_{\hat{y}|\theta,m}\left(\hat{y},\tilde{\theta}\left(\hat{y},m\right),m\right)\cdot p_{\theta|m}\left(\tilde{\theta}\left(\hat{y},m\right),m\right)\cdot\left\|\tau\cdot V_{\tilde{\theta}}\left(\hat{y},m\right)\right\|^{\frac{1}{2}} \tag{4.3}$$

where $\tilde{\theta}\left(\hat{y},m\right)$ is the maximum a posterior parameter set and $V_{\tilde{\theta}}\left(\hat{y},m\right)$ is the posterior variance of the parameters. This method was applied to the nonlinear topologies as a linear approximation. The starting probability distribution was computed first. The nominal experiment had little distinguishing power and all topologies were roughly equally probable (Figure 4-2).

4.2.6 Optimal Experimental Design Algorithm

To compute the expected entropy of each candidate experiment, a set of random data sets was generated according the experiment by a Monte Carlo procedure. First, a random topology was chosen according to the nominal topology probability.

A random parameter set was chosen for the topology using the Metropolis-Hastings algorithm. The proposal distribution for the parameter sampling was a normal distribution in log-space with a mean of the current parameter set and a variance equal to inverse of the Fisher information matrix multiplied by 5.66 divided by the number of parameters [113]. The Fisher information matrix used in the proposal was recomputed every 500 steps and parameter directions of very large uncertainties reduced so that acceptance ratio was between 0.15 and 0.5. The autocorrelation was computed for each parameter and the samples were thinned so that the autocorrelation in any parameter was less than 0.05. The autocorrelation was not trusted until the sample used to compute the autocorrelation was 20 times longer than the step size.

The drawn model was simulated according to the candidate experiment, and a noisy data set was generated according to the measurement scheme and measurement noise of that experiment.

All the topologies were fit to the combined set of the original data and the Monte Carlo data. The linearized topology probability was computed according to this combined data set. The entropy of this topology probability distribution was saved.

This was repeated many times, drawing a different topology, parameter set, and data set each time. An average over the entropy provided a Monte Carlo estimate for the expected entropy of performing that experiment given the current uncertainties.

4.3 Results

We studied four topologies of the ErbB network, each differing in the conditions under which ErbB3 could dimerize. We fit the topologies to a nominal data set generated by simulating the topology in which ErbB3 could dimerize when heregulin was bound. All four topologies were roughly equally probable according to the nominal data set.

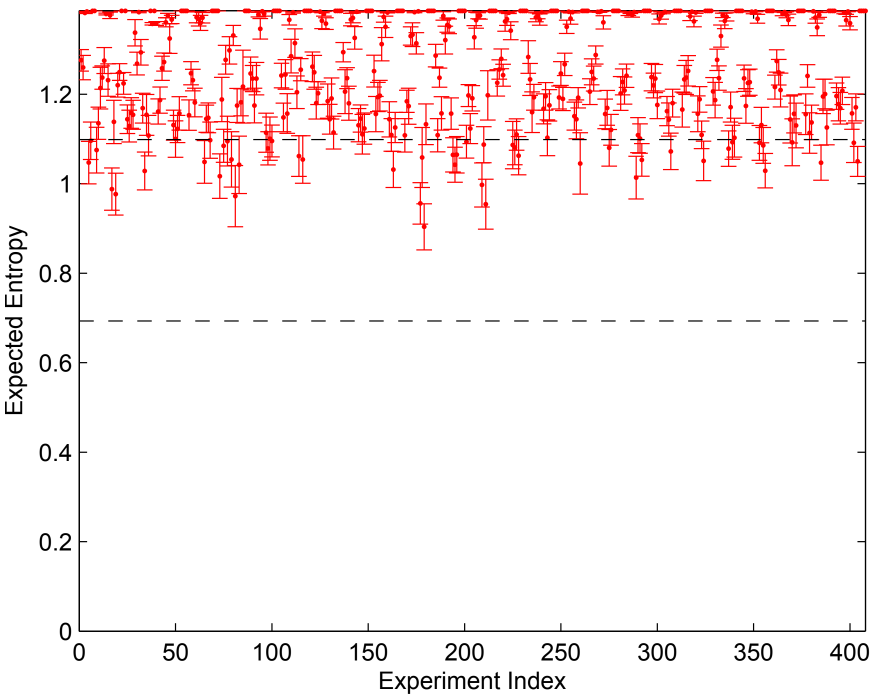

Figure 4-3: **Expected entropies of candidate experiments.** The expected entropy from each experiment is plotted according to its index. The error bars are the standard error on the mean of the Monte Carlo sample. From top to bottom, the dashed lines indicate (1) a uniform distribution on the topologies (complete uncertainty), (2) the equivalent entropy of 1 topology eliminated, and (3) the equivalent of 2 topologies eliminated.

Using our optimal experimental design procedure, we computed the expected entropy of each of 408 candidate experiments, which differed in the concentration of the inputs to the system and up to one protein in the network that was knocked-down or overexpressed.

There was a diverse range of expected entropies from the set of candidate experiments (Figure 4-3). Only 47 experiments were expected to have lower entropy than the elimination of one topology. No experiments were expected to eliminate two topologies, let alone three. The best experiment according to the target function was #179, with an expected entropy of 0.90. This experiment was not a unique outlier and other experiments have similar expected entropy. For example, the tenth best experiment, #34, had an entropy of 1.0.

The properties of the top ten experiments are listed in Table 4.1. The most noticeable trend is all of the top ten experiments apply heregulin as an input (the

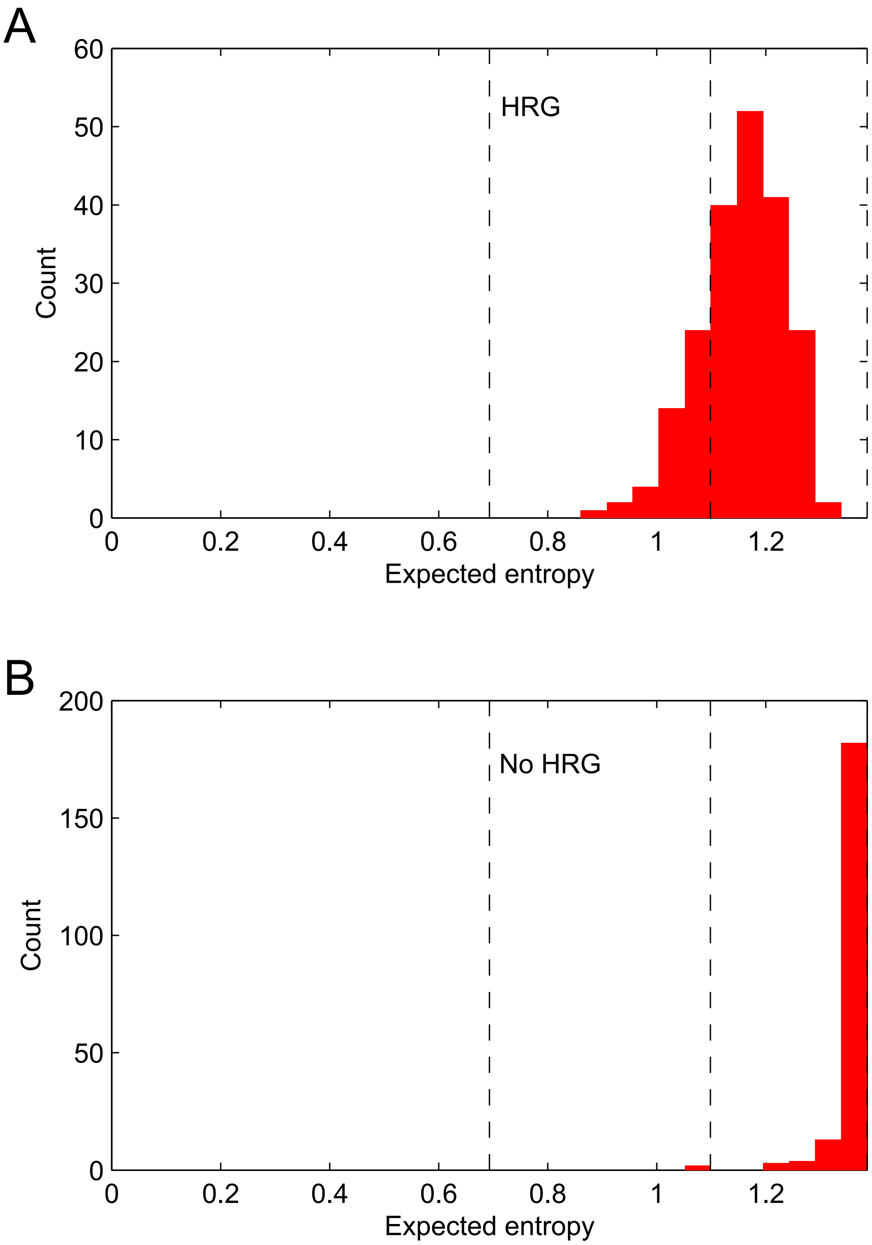

Figure 4-4: **Effect of heregulin on experiment quality.** A histogram of the expected entropies (A) with and (B) without treatment using heregulin. Vertical lines are the same as the horizontal lines of Figure 4-3, indicating entropy equivalent to 0, 1, and 2 topologies eliminated.

Rank	Index	EGF	HRG	Inh	Perturbation	Expected Entropy	Actual Entropy
1	179	0	1 nM	0	Pase1−	0.903	1.3
2	211	0	1 nM	0	Pase2−	0.954	0.76
3	177	0	1 nM	1 nM	Pase1−	0.956	1.3
4	81	0	1 nM	1 nM	GAP−	0.972	1.3
5	19	0	1 nM	0	ErbB2−	0.977	0.70
6	17	0	1 nM	1 nM	ErbB2−	0.988	0.57
7	209	0	1 nM	1 nM	Pase2−	0.998	1.4
8	289	0	1 nM	1 nM	PDK1−	1.01	1.4
9	73	1 nM	1 nM	1 nM	GAP−	1.02	1.3
10	34	0	1 nM	1 nM	ErbB3+	1.03	1.2

Table 4.1: **Top 10 experiments.** A description of each of the top experiments is provided here, including its index in the list of 408 experiments, the concentration of EGF supplied (EGF), the concentration of heregulin supplied (HRG), the concentration of ATP analog inhibitor supplied (Inh), the protein that was knocked-down (−) or over-expressed (+), the expected entropy of the experiment according to the optimal experimental design algorithm, and the actual entropy from computationally performing the experiment according to the "true" model.

trend is not broken until the 22-best experiment). This underlies the fact, illustrated in Figure 4-4, that heregulin clearly divides the experiments into two populations: those that use heregulin that tend to be useful and those that do not use heregulin that tend to be useless for the purpose of determining if ErbB3 dimerizes. This is not particularly surprising considering that heregulin is the ligand which stimulates the dimerization of ErbB3 and that the nominal experiment already applied the other ligand EGF.

Also notable is that 9 out the top 10 experiments are knock-down experiments. Only 1 is an over-expression experiment and 0 are without perturbation to the network. Our earlier work on optimal experimental design for reducing parameter uncertainty also found that perturbations were absolutely favored over experiments without perturbations [108]. Like in that case, it is remains unclear if favoring perturbation experiments is due to the fact that perturbation experiments greatly outnumber non-perturbation experiments, which only comprise 2% of the total experiments, or the fact that the nominal experiment itself is a non-perturbation experiment means that another non-perturbation experiment is unlikely to provide new information on the

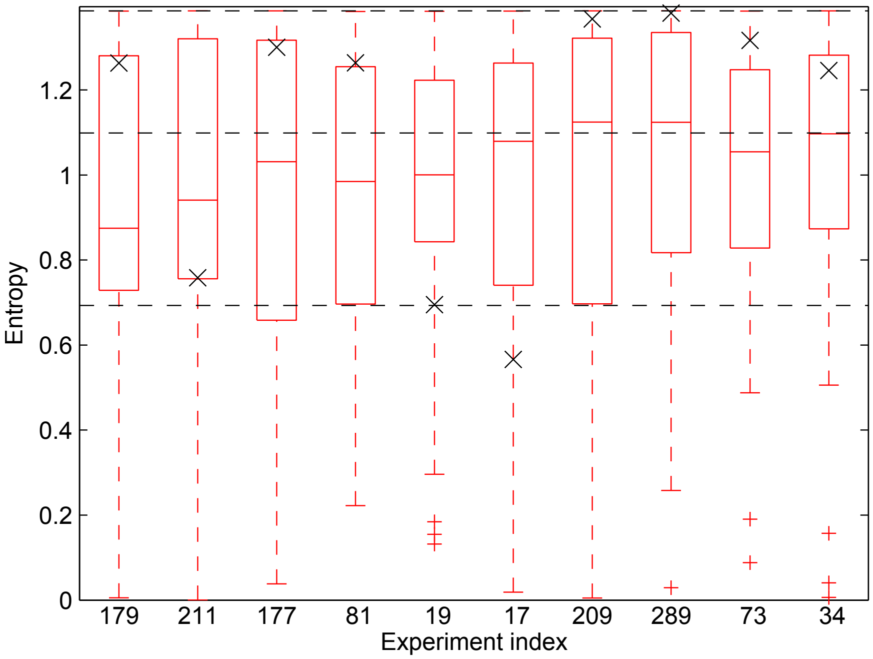

Figure 4-5: **Distribution of results from top experiments.** Standard box plots of the entropies obtained by Monte Carlo sampling indicate the range of expected results. Red boxes contain the first and third quartiles, line within boxes are the median, the whiskers extend to the nearest point within 1.5 times the height of the box, plusses mark outliers beyond the whiskers, and black crosses mark the actual entropy resulting from simulating the experiment using the "true" model. Horizontal dashed lines are same as Figure 4-3, indicating 0, 1, or 2 topologies eliminated. For each experiment, there are some data sets that provide no additional evidence for the true topology (near top) and some data sets that provide almost complete certainty for the true topology (near bottom).

system.

Also notable is that the top 10 experiments only utilize 6 of the 54 possible perturbations. The knock-down of Pase1, which is the perturbation of the first best experiment, appears again in the third best. The knock-down of Pase2 from the second best experiments appears again in seventh best. The knock-down of GAP appears in fourth and ninth best experiments. The knock-down of ErbB2 appears in the fifth and sixth experiments. The repetition of similar experiments near the top of the list indicates that there are some important features being revealed by our optimal experimental design procedure.

It is tempting to expect that the entropy one will get after doing the experiment will be equal to the expected entropy. However, the expected entropy is an expected value over the entropies that result from all possible data sets from that experiment. For the top 10 experiments, the spread of possible entropies is very large (Figure 4-5). Some data sets result in little change in the topology uncertainty, while other data sets result in the topology being determined almost entirely.

To look deeper at the properties of good experiments, each of the top 10 experiments was performed using the true model to generate the data. The resulting topology probabilities are listed in Table 4.1 and the actual entropies are marked on the box plots in Figure 4-5. Depending on which experiment was done, the actual entropy was either more or less than the expected entropy but, in all cases, lay within the expected distribution for the entropy.

We looked closer at experiments 179 and 211 because they were the first and second best expected experiments, though each gave different results when the true topology was simulated. When the topology prior is uniform, as it is in our scenario, the topology probability is dependent only on the marginal likelihood, which in the linear approximation is equal to Equation 4.3. When the prior is a Gaussian, as it is in our scenario, the prior has the form:

$$p_{\theta|m}\left(\tilde{\theta}\left(\hat{y},m\right),m\right) = \left\|\tau\cdot V_{\bar{\theta}}\left(m\right)\right\|^{-\frac{1}{2}}\cdot\exp\begin{pmatrix}\left(\tilde{\theta}\left(\hat{y},m\right)-\bar{\theta}\left(m\right)\right)^{T}\\ \cdot V_{\bar{\theta}}^{-1}\left(m\right)\\ \cdot\left(\tilde{\theta}\left(\hat{y},m\right)-\bar{\theta}\left(m\right)\right)\end{pmatrix} \tag{4.4}$$

where $\bar{\theta}(m)$ is the prior mean, $V_{\bar{\theta}}(m)$ is the prior variance, and $\tilde{\theta}(\hat{y},m)$ is the maximum *a posteriori* parameter set. We can rewrite the marginal likelihood into three terms:

$$p_{\hat{y}|m}\left(\hat{y},m\right) = \underbrace{p_{\hat{y}|\theta,m}\left(\hat{y},\tilde{\theta}\left(\hat{y},m\right),m\right)}_{\text{likelihood}}\cdot\underbrace{\exp\begin{pmatrix}\left(\tilde{\theta}\left(\hat{y},m\right)-\bar{\theta}\left(m\right)\right)^{T}\\ \cdot V_{\bar{\theta}}^{-1}\left(m\right)\\ \cdot\left(\tilde{\theta}\left(\hat{y},m\right)-\bar{\theta}\left(m\right)\right)\end{pmatrix}}_{\text{prior}}\cdot\underbrace{\left\|V_{\tilde{\theta}}\left(\hat{y},m\right)\right\|^{\frac{1}{2}}\cdot\left\|V_{\bar{\theta}}\left(m\right)\right\|^{-\frac{1}{2}}}_{\text{robustness}} \tag{4.5}$$

Topology Probability	Likelihood	Prior	Robustness
.16	.02	.44	.30
.11	.56	.07	.03
.30	.40	.11	.08
.43	.02	.38	.59

Table 4.2: **Deconstruction of experiment 179.** The topology probability of computationally performing experiment 179 according to the published model was decomposed into three components: the likelihood the data would have come from the best-fit model of each topology, the prior probability that the parameters of the best-fit model would stray this far from the mean, and the ratio of the volume of parameter space of the best-fit model to the prior parameter space volume. Each column is normalized to a sum of 1 to allow comparison of the relative contributions of each term.

The likelihood is the easiest term to understand; it is merely the distance the data is from the best fit parameters. It is well-known that topologies with more parameters tend to be able to fit the data better even when they are not the true topology. The remaining two terms appear in the Bayesian answer, which compensates for this bias that exists in favor of more complex topologies. The prior term penalizes topologies for each parameter that, in the best-fit parameter set, deviates from the prior mean in order to provide the fit. The robustness term measures the ratio of the final volume of parameter space to the prior volume of parameter space, penalizing topologies for having a relatively small parameter space that fits the data as well as the best-fit parameters. These terms compensate for the likelihood term because topologies with more parameters have more ways that they can be penalized in the prior and robustness term.

The values of the three terms of the topology probability for experiment 179 are shown in Table 4.2. The final topology probabilities are not very different across the topologies, but the contributions from the different terms show some structure. The simplest topology, topology 1, has the lowest likelihood, indicating that it had the worst fit. If only using likelihood, it would have a probability of about 1.5%. However, the prior and robustness terms partially compensate, leaving it ultimately with a 16% of being correct according to the data.

Topology Probability	Likelihood	Prior	Robustness
.46	.0003	.91	.97
.0053	.60	.0019	.0031
.52	.24	.067	.022
.0073	.16	.021	.0014

Table 4.3: **Deconstruction of experiment 211.** The topology probability of computationally performing experiment 211 according to the published model was decomposed in the same way as experiment 179 in Table 4.2.

The values of the three terms of topology probability for experiment 211 are shown in Table 4.3. The topology probability indicates that 2 of the topologies were nearly eliminated by the addition of the new data set from experiment 211. The topologies eliminated were topologies 2 and 4. When looking at the components of the topology probability, it is clear that they were not eliminated because they could not fit the data. In fact, topology 1 is the worst fitting topology, yet it was ultimately rescued by the other two terms. The prior and robustness terms were highly unfavorable to topologies 2 and 4.

We looked at the contribution of each parameter to the parameter prior term. The effect of each parameter is shown in Figure 4-6. A few parameters are mostly responsible for making topology 1 seem more favorable relative to the other three topologies. The top ten most influential parameters are listed in Table 4.4. The overall effect on topologies 2, 3, and 4 is negative, though a few parameters are favorable. The dominant theme of the important parameters is reactions related to PIP3. This is particularly interesting because PIP3 is not a measured output. We only measured proteins of the network, and PIP3 is a small molecule. The strongest effect was with parameter 91, whose value was about 20 times less likely in topologies 2 and 3 than in topology 1. The fitting step takes this probability into account, so that, in order to use this value rather than the one that was optimal for topology 1, some other aspect of the fitting had to be improved. We hypothesize that PIP3 shows up here because of a quirk in the model: when any complex forms in this model, only the last monomer can fall off; the intermediate associations are frozen.

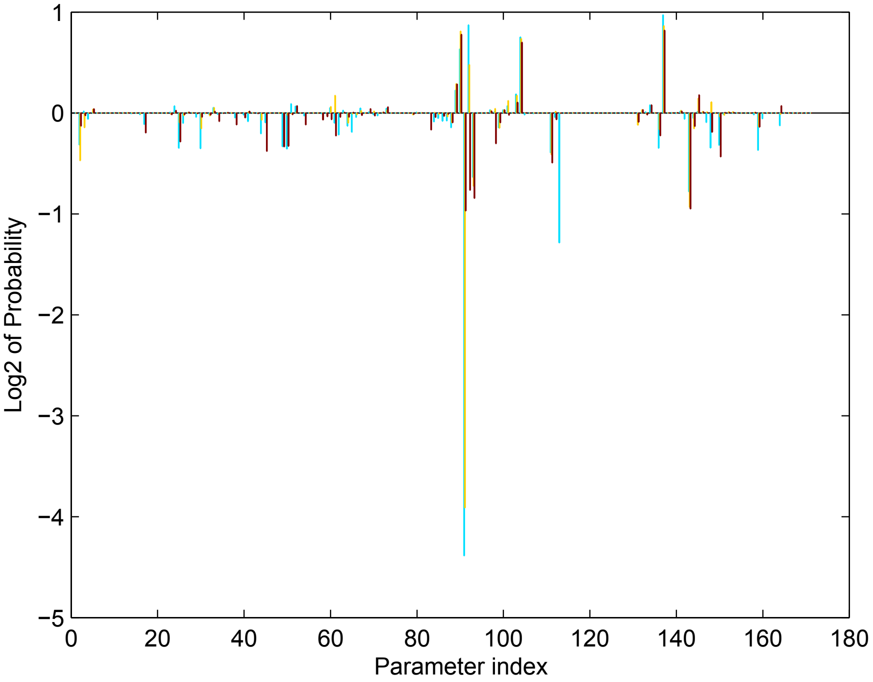

Figure 4-6: **Impact of parameter prior in experiment 211.** Each parameter contributes to the probability of each topology depending on how far its best-fit value is from the mean. Using the parameters of the topologies fit to the nominal experiment and experiment 211, the log2 of the contribution by each parameter is plotted relative to topology 1, which was the worst-fit topology. The dominant peaks, which on the net reduce the probability of topologies 2, 3, and 4, are discussed in the text. The teal, yellow, and red bars correspond to topologies 2, 3, and 4, respectively.

Index	Name	Reaction	Description	Contribution
89	k67	(R:R)#P:GAP:Grb2:Gab1#P + PI3K -> (R:R)#P:GAP:Grb2:Gab1#P:PI3K	Association of PI3K	[1.2, 1.2, 1.2]
90	kd67	(R:R)#P:GAP:Grb2:Gab1#P:PI3K -> (R:R)#P:GAP:Grb2:Gab1#P + PI3K	Dissociation of PI3K	[1.6, 1.8, 1.7]
91	kd68	(R:R)#P:GAP:Grb2:Gab1#P:PI3K:PIP2 -> (R:R)#P:GAP:Grb2:Gab1#P:PI3K + PIP3	Phosphorylation of PIP2 to PIP3	[0.05, 0.07, 0.5]
92	kd68b	(R:R)#P:GAP:Grb2:Gab1#P:PI3K:(PIP2)2 -> (R:R)#P:GAP:Grb2:Gab1#P:PI3K:PIP2 + PIP3	Phosphorylation of PIP2 to PIP3	[1.8, 1.4, 0.6]
93	k69	PIP3 + AKT -> PIP3:AKT	Activation of AKT	[0.6, 0.6, 0.6]
104	kd76	PIP3 + PDK1 -> PIP3:PDK1	Activation of PDK1	[1.7, 1.7, 1.7]
111	kd96	ErbB2#P + ErbB2#P -> ErbB2#P:ErbB2#P	ErbB2 dimerization	[0.8, 0.8, 0.7]
113	kd7	(R:R)#P@endosomal_membrane -> (R:R)#P@cell_membrane	Phosphorylated receptor recycling	[0.4, 1.0, 1.0]
137	kd106	(R:R)#P:GAP:Grb2:Gab1#P:PI3K:PIP2 -> (R:R)#P:GAP:Grb2:Gab1#P:PI3K + PIP2	Dissociation of PIP2	[2.0, 1.8, 1.8]
143	k109	PIP3 + PTEN -> PIP3:PTEN PIP3 + Shp -> PIP3:Shp	Binding of PIP3 to its phosphatase	[0.6, 0.5, 0.5]

Table 4.4: **Influential parameters according to the parameter prior.** Each of the parameters contributed independently to the parameter prior term. The ten most important parameters are listed, including their index in the model, their name, the reaction they are associated with, and a short description. Their contribution was dependent on how far the best-fit value of that parameter was from the prior mean. The contribution was measured relative to the first topology, which was the worst-fit topology, so that the relative values of topologies 2, 3, and 4 are shown respectively. A number greater than one indicated that the parameter gave its associated topology a higher probability, while a number less than 1 means that it gave a lower probability relative to topology 1. Some parameters were used in multiple similar reactions, which were summarized here. "R" is any receptor.

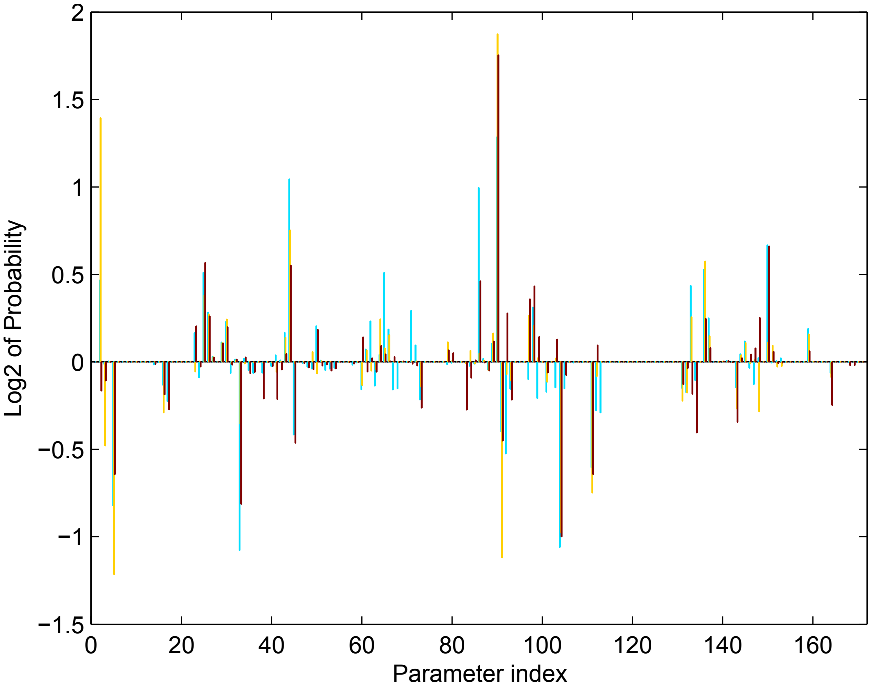

Figure 4-7: **Impact of parameter robustness in experiment 211.** The parameter uncertainty as a whole contributes to the probability of each topology. To examine the impact from each parameter, the marginal uncertainty of each parameter was computed. The ratio of the uncertainty to the prior uncertainty of that parameter was computed. This ratio roughly approximates the impact on the probability by that parameters uncertainty. The impact on the topology probability is plotted relative to topology 1. The dominant peaks, which on the net reduce the probability of topologies 2, 3, and 4, are discussed in the text. Topologies 2, 3, and 4 were colored teal, yellow, and red, respectively.

As a consequence, PIP3 binding sequesters a portion of the receptors, GAP, Grb2, Gab1, and PI3K. Because these are important proteins in the network, controlling their release may give PIP3 a mild control over many outputs and acts as a way to give the model a slightly better fit to the data. Biologically, this is not an ability that PIP3 has because the intermediate associations are not actually influenced by it, but it reveals a possible property of Bayesian topology discrimination. Namely, parameters that affect many measurements are those that will be driven to extreme values to improve the fit because only a strong influence on the measurements can overcome the restraint that the prior provides to keep the parameter from going to extreme values. Parameter 113, being the recycling parameter for every dimer, may be another example of this, though the strong effect is seen only in topology 2.

Index	Name	Reaction	Description	Contribution
2	k1c	EGF + ErbB2:ErbB3 -> (ErbB2:ErbB3)#P	Activation of ErbB2:ErbB3	[1.4, 2.6, 0.90]
5	k1d	EGF + ErbB2:ErbB4 -> (ErbB2:ErbB4)#P	Activation of ErbB2:ErbB4	[0.57, 0.43, 0.64]
33	kd21	(R:R):GAP:Shc#P?:Grb2:Sos:Ras:GDP -> (R:R):GAP:Shc#P?:Grb2:Sos + Ras:GDP	Dissociation of Ras	[0.47, 0.78, 0.57]
44	kd29	Ras:GTP:Raf -> Ras_activated:GTP + Raf#P	Phosphorylation of Raf by Ras	[2.1, 1.7, 1.5]
86	kd65	*:Sos:Erk#P#P -> *:Sos#P + Erk#P#P	Phosphorylation of Sos by Erk	[2.0, 1.0, 1.4]
90	kd67	(R:R)#P:GAP:Grb2:Gab1#P:PI3K -> (R:R)#P:GAP:Grb2:Gab1#P + PI3K	Dissociation of PI3K	[2.5, 3.7, 3.4]
91	kd68	(R:R)#P:GAP:Grb2:Gab1#P:PI3K:PIP2 -> (R:R)#P:GAP:Grb2:Gab1#P:PI3K + PIP3	Phosphorylation of PIP2 to PIP3	[0.76, 0.46, 0.73]
104	kd76	PIP3 + PDK1 -> PIP3:PDK1	Activation of PDK1	[0.48, 0.51, 0.50]
111	kd96	ErbB2#P + ErbB2#P -> ErbB2#P:ErbB2#P	ErbB2 dimerization	[0.66, 0.59, 0.64]
150	kd113	(R:R):GAP:Grb2:Gab1#P:PI3K:Ras:GDP -> (R:R):GAP:Grb2:Gab1#P:PI3K + Ras:GTP	Activation of Ras by PI3K	[1.6, 1.1, 1.6]

Table 4.5: **Influential parameters according to the posterior volume.** We approximated the contribution of each parameter to the robustness term by dividing the marginal standard deviation of each best-fit parameter by the prior standard deviation. Each parameter has the same information associated as Table 4.4. Also like Table 4.4, the contribution was measured relative to the worst-fit topology, topology 1, so that the relative values of topologies 2, 3, and 4 are shown respectively. A number greater than one indicated that the parameter gave its associated topology a higher probability, while a number less than 1 means that it gave a lower probability relative to topology 1.

The robustness term was dissected in a similar way. We computed the ratio of the standard deviation of each parameter to the standard deviation of the prior. Because the robustness term takes into account covariance in the parameters, comparing the absolute variance of the parameters will give an incomplete picture, but a picture that is easier to visualize and understand than one obtained by comparing eigenvectors of parameters. The size of the contribution from each parameter relative to topology 1 is shown in Figure 4-7. The top ten most important parameters of this term are listed in Table 4.5. There is some overlap between the two lists—notably, parameter 91, which had the strongest impact in the prior term. The actual fraction of the prior that was available for this parameter in topologies 1, 2, 3, and 4 was 1.00, 0.76, 0.46, and 0.73, respectively. This means that the robustness term itself only reduced the probability of the topologies by about 25% to 50%. But this smaller range of parameter values not only affected the topology probability directly, but also constrained the topology to a more extreme value of this parameter via the prior term, which had as much as a 16-fold effect.

The box plots in from Figure 4-5 were separated by topology to reveal if the entropy distribution was different depending on which topology was true (Figure 4-8). For most experiments, the difference appeared to be small. One notable distribution difference is experiment 211. In experiment 211, the expected entropy distribution has a smaller variance when topology 3 is true than when another topology is true. The actual true topology was, in fact, topology 3 and the final value of the entropy after performing experiment 211 is consistent with the Monte Carlo distribution. In a realistic scenario, the true topology is not known beforehand, so evaluating a single experiment based on the entropy distributions for each topology may not be that useful in isolation. While experiment 211 appears to have low variance for topology 3, the other three topologies ensure a rather large variance in the expected entropy of the experiment. This type of analysis may be useful in considering several experiments at once. If one is able to perform several additional experiments and wishes to guarantee that at least one of the experiments in the set of experiments will be good, one could choose one experiment that is good for each topology considered, which would ensure

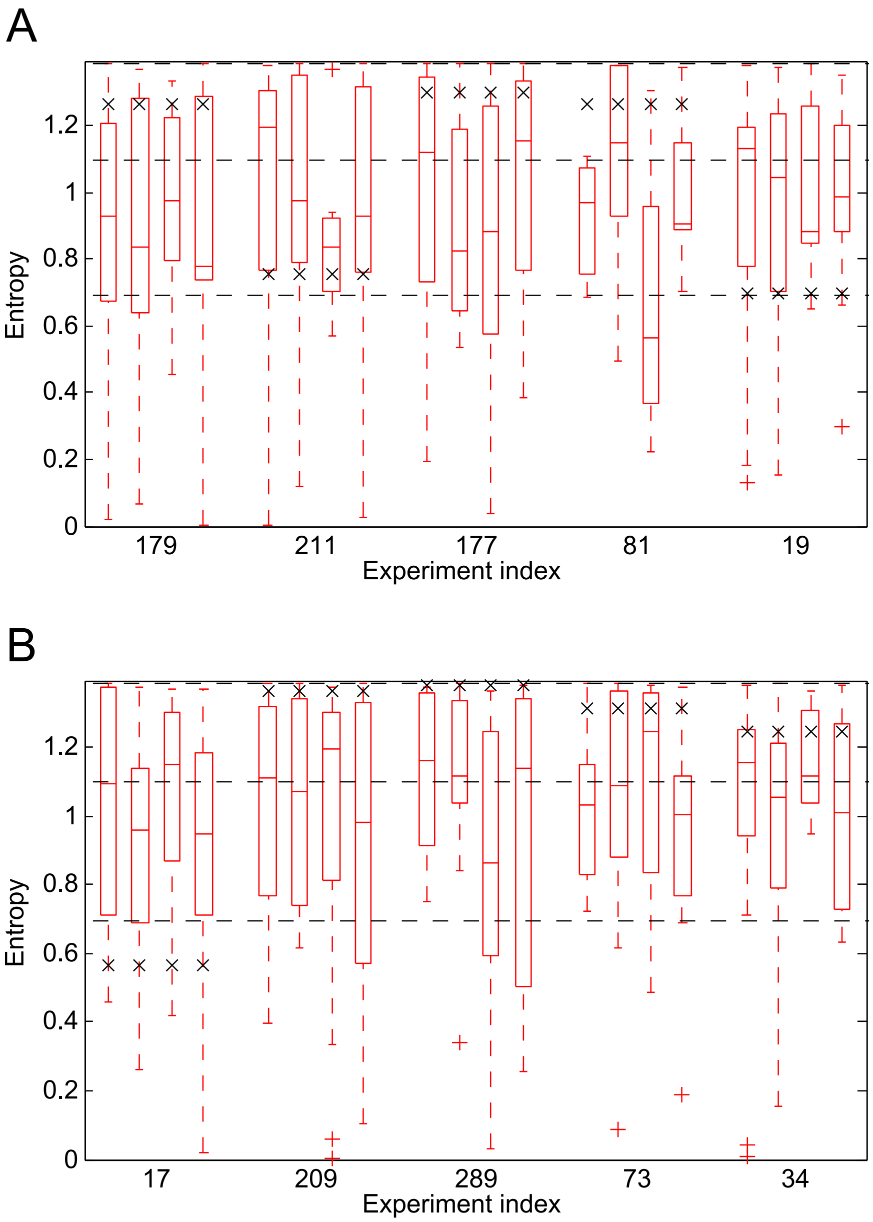

Figure 4-8: **Distribution of results by topology.** During the Monte Carlo procedure, the topology used to generate the sample data set was recorded. The entropies from the sample data sets were plotted in a standard box plot for each topology for each experiment. The style of the box plots are the same as in Figure 4-5. Red boxes contain the first and third quartiles, line within boxes are the median, the whiskers extend to the nearest point within 1.5 times the height of the box, plusses mark outliers beyond the whiskers, and black crosses mark the actual entropy resulting from simulating the experiment using the "true" model. Horizontal dashed lines indicate 0, 1, or 2 topologies eliminated.

that at least one good experiment is performed. This assumes that one does not have the computational power to directly compute the expected entropy of each possible set of experiments, which would give a more complete analysis of the quality of each set.

Ultimately, there was no experiment in the set of 408 candidate experiments that could decisively distinguish between the topologies, assuming that the experimental design procedure would have located it if it existed. We suspected that this was because none of the experiments had the ability to directly measure the amount of ErbB3 dimerization. We designed a synthetic experiment that would represent the direct measurement of the ErbB3 dimer. This experiment used 1 nM of heregulin and over-expressed ErbB3 by 10-fold and measured only the total amount of dimer at the 0 min time point and the eight time points defined for the rest of the experiments. The measurement uncertainty was the same as the rest of the experiments. This experiment was simulated with the true topology and synthetic measurements were collected. According to the topology probability, the probability of each topology was 0.0, 0.0, 1.0, and 0.0 for topologies 1, 2, 3, and 4, respectively, indicating that this type of experiment could reveal topology 3 to be the true topology.

4.4 Conclusion

We developed a method of optimal experimental design for efficiently reducing topology uncertainty. The method used a Monte Carlo procedure to generate a sample of data sets that would be expected from each candidate experiment and used a linearized method for topology probability to quickly approximate the resulting topology probability. The expected entropy of the topology probability was used to judge how effective each experiment was expected to be.

We tested the method on a published model of ErbB signaling by generating several competing models that added a few variations on how receptor ErbB3 could dimerize. The topologies were fit to the synthetic data from a nominal experiment, which did not distinguish between the topologies at all.

The candidate experiments differed in whether or not they treated the system with EGF, heregulin, or an inhibitory ATP analog. The most striking trend in the expected effectiveness of the experiments was the separation between heregulin treatment, which was necessary for relatively good experiments and was expected to eliminate about 1 of the topologies, and no heregulin treatment, which was almost universally expected to provide no information on which topology was correct. Because heregulin is the ligand that activates ErbB3 for dimerization, it seems reasonable that this ligand would be important for understanding the behavior of this receptor. It may also be relevant that the nominal experiment already treated the system with EGF, thus an experiment that treats the system with heregulin exercises a different mode of the systems behavior. Previous research has shown that experiments that are complimentary to existing experiments are important for efficiently reducing parameter uncertainty [108, 37]. It would be interesting if this property held for reducing topology uncertainty as well, though to conclusively show this would require studying a larger set of candidate experiments.

A closer look at one of the experiments that was effective at eliminating two of the topologies revealed a complex interaction between the quality of the fit, the prior on the parameters, and the robustness term of the topology probability. The topology with the fewest parameters (topology 1) fit the worst, a well-known phenomenon, but parameter prior and robustness terms compensated, resulting in it being equally probable with the true topology (topology 3). Topologies 2 and 4 were punished worse for their parameter and robustness terms, resulting in very low probability. It is interesting to note that none of the terms ruled out a topology in a frequentist sense. While the first topology fit the data worse, it was still a good fit by a chi-squared goodness-of-fit sense (data not shown). Furthermore, in the parameter prior term, no parameter was very far away from the mean in terms of standard deviations, but the cumulative effect of many parameters needing to be some distance from the mean in order to get the better fit was meaningful. Only by using a Bayesian approach to aggregate the evidence could it be concluded that topologies 2 and 4 were unlikely given the data from the experiment.

One of the most curious observations is that the distribution of possible entropies from each experiment was very large. For each experiment, there were some possible data sets that would provide no additional information on the topology, but other data sets that would conclusively eliminate all but one topology. Most of the data sets were somewhere in the middle, eliminated 1 or 2 topologies, which is where the actual results of the true model fell, as well. This is different from our previous work on optimal experimental design for parameter uncertainty, in which the expected uncertainty from an experiment and the actual uncertainty from a realized synthetic data set were very similar [108]. The nature of parameter distributions and topology distributions are very different, with the former being a continuous function in a hyperdimensional space and the latter being a discrete function. For a linear model, the entropy of the parameter distribution is perfectly known before the experiment is done, while the entropy of the topology distribution before the experiment is done is not known. This may pose a fundamental limitation on our ability to predict how good an experiment will be for reducing topology uncertainty. If this is true, then this increases the utility of our Monte Carlo optimal experimental design method. We looked closely at the expected entropy to rank experiments, but one could also rank experiments based one other factors besides the mean entropy. One could select experiments that also had a low variance in the entropy and, therefore, were guaranteed to be good. Or one could select experiments that had a high variance and, therefore, a chance to be a killer experiment. It is never required that the entropy be the target function minimized. A reasonable alternative would be to maximize the probability that at least one topology is effectively eliminated by having its probability being below some threshold.

We have made several observations about the role of the different components of topology probability equation to distinguish between topologies. We have also made several observations on the properties of experiments that may be good for distinguishing between topologies of biological systems. Further study is warranted to ensure that these observations generalize to broad classes of models in systems biology. The computational cost of the method is tractable for a medium sized model, so it may be useful for experimenters to apply in order to ensure that they are getting the most out of their experiments. But we expect that this method will be more useful as a means of studying what kinds of experiments are good for distinguishing between topologies of biological systems and for comparing computational approximations to the optimal experimental design problem for topology uncertainty.

Chapter 5

Exact Coarse-grained Reduction of Rule-based Models Featuring Compartments and Cyclization

5.1 Introduction

Increasingly detailed models are being used to aggregate knowledge of biological systems. These models act as a framework to understand the higher-level behavior as a function of the underlying components, to constrain the parameters of the underlying reactions as a function of the higher-level measurements, to predict the behavior of the system under new conditions, and to design interventions to alter the behavior of the system to follow design specifications. The paucity of molecular data and the difficulty of simulating a model with hundreds or thousands of states are the main limitations of systems biology modeling.

In this work, we address a particular problem that arises when simulating models that have proteins with many modification sites. If such a protein is present in the model, this poses a problem for simulating the model because the number of possible species in which that monomer can exist is exponential with respect to the number of modification sites. For example, consider ErbB1 with five phosphorylation sites

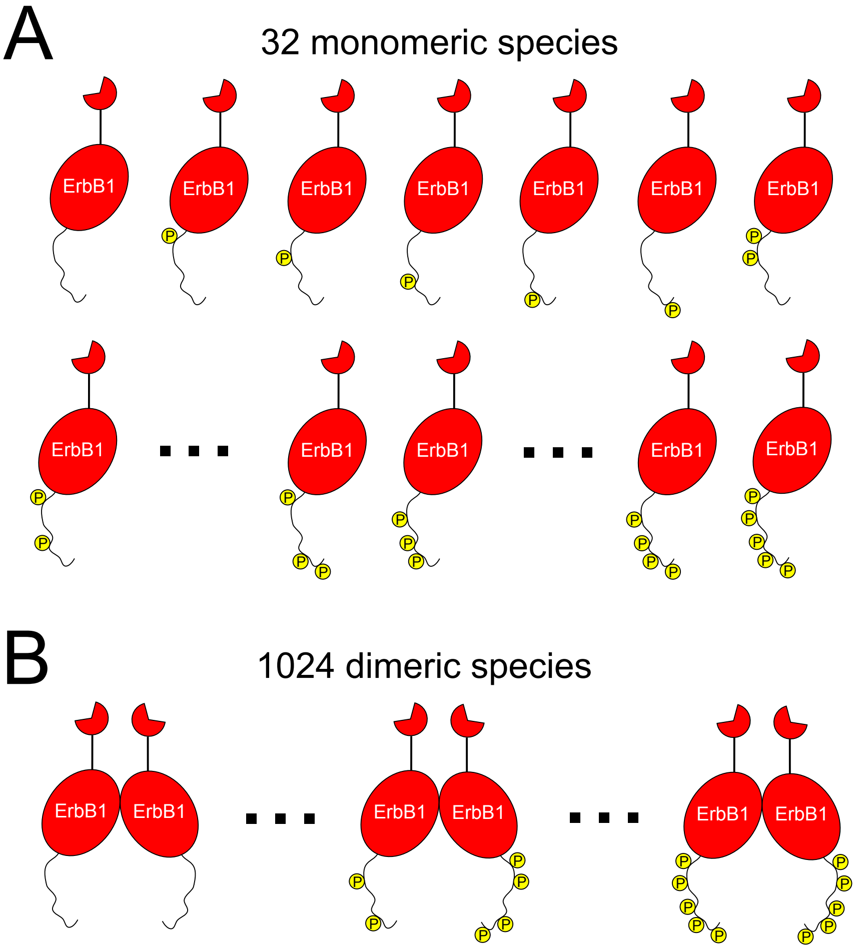

Figure 5-1: **Exploding species count.** As the number of modifiable sites on a monomer increases, the number of possible states the monomer can be in grows exponentially. For example, a monomer of ErbB1 with five sites has 32 states, as in panel (A). If the monomers form protein complexes, the number of possible complexes can be enormous. For example, a homodimer of ErbB1 has 1024 states, as in panel (B).

(Figure 5-1). If each site can be in either a phosphorylated or unphosphorylated state, then there are 2^5 or 32 possible species of a single monomer of ErbB1. If ErbB1 can dimerize in the model, then there are 2^{10} or 1024 possible species of an ErbB1 dimer. If each site can bind just one protein, then there are 3^{10} or 59 049 possible states. Having many modification sites on a single monomer is a common motif in biology, but to define the possible species and their reactions in a model quickly becomes infeasible as more sites are included in the model. Not only can such a model be too large to write, it can also be too large to simulate if each species is represented by a state in the model.

The principle effort to circumvent the problem of exploding states without making additional assumptions or approximations is rule-based modeling (also called agent-based modeling) [128, 129, 130]. Rule-based modeling in systems biology is based on the observation that many of the sites on a monomer behave independently of each other, or at least, our models treat them independently. For example, the phosphorylation rate at one site will be the same whether or not a site on the opposite side of the protein is phosphorylated. Of course, there are instances where one site does affect another. Rule-based modeling allows a modeler to specify the dependent instances without specifying all the independent instances. If a program can generate the species and reactions described by those rules, this eliminates considerable work on the part of the modeler, assuming that the number of dependent rules is substantially less than the number of independent reactions. A few software packages implement rule-based modeling of this kind, including Kappa [59], BioNetGen [131], little b [132], and Biocham [133].

If the model assumes that some sites behave and are measured independently, then it is often possible to simulate the system with fewer states than the full set of species. Solving this problem by hand is extremely difficult and error-prone for all but the most trivial systems. A few algorithms to construct the reduced-state model automatically have been designed. They all operate on one basic principle: it is exactly correct in a differential equation model to simulate the state of a particular site alone and ignore the states of any other sites when that other state has no influence on rate of any reaction affecting the particular state. Early investigations found various transformations that, when applied to the full system, resulted in smaller systems that gave exact results on some observables and inexact results on other variables [134, 135, 136, 137, 138]. Eventually, these findings were codified in several algorithms that automatically built coarse-grained systems of ODEs from a set of user-defined rules [139, 140, 141, 142, 143].

The existing methods of rule-based modeling lacked several features that we felt were important for building a detailed model of ErbB signaling. The broadest feature missing was a means to use compartments in the model. There are several special needs for compartments in rule-based models. Rule-based models allow for reactions to be specified with some ambiguity in the species that are participating. For example, the binding of Sos to Ras in the ErbB pathway is one step in the mechanism by which Ras is activated in the pathway. Sos becomes able to activate Ras when Sos is recruited to the membrane, increasing the local concentration of Sos around Ras, which is constitutively associated with the membrane. Sos is confined to the membrane when it binds via one or more adaptor proteins to ErbB receptors, which are constitutively associated with the membrane. To describe this process with rules that do not have compartments, one would have to write a rule for each possible way that Sos could be associated with the membrane, such as `erbb1(Y1068~P-1).grb2(sh2-1,sh3N-2).sos(pp-2)`, `erbb1(Y992~P-1).shc(sh2-1,Y317~P-2).grb2(sh2-2,sh3C-3).sos(pp-3)`, and many more. Writing all possible internal representations is the kind of time-consuming and error-prone model writing that rule-based modeling is intended to prevent. It would be preferable to be able to express this common motif in a rule like `sos(r).@cell_membrane + ras(s) -> sos(r-1).ras(s-1)`, where `.@name` indicates that the entire species is localized to the compartment given by name.

If both reactants of a bimolecular rule can be localized to two different compartments, then with the above syntax there are four rules that need to be written, one for each possible combination of compartments. For example, cytosolic proteins B-Raf and C-Raf may heterodimerize when either is bound or not bound to Ras on the membrane.

1. `braf(d).@cell + craf(d).@cell -> braf(d-1).craf(d-1)`

2. `braf(d).@cell + craf(d).@cell_membrane -> braf(d-1).craf(d-1)`

3. `braf(d).@cell_membrane + craf(d).@cell -> braf(d-1).craf(d-1)`

4. `braf(d).@cell_membrane + craf(d).@cell_membrane -> braf(d-1).craf(d-1)`

It may be reasonable to give reactions 1, 2, and 3 the same rate constant. The probability of a reactive collision between B-Raf and C-Raf is theoretically the same as long as one of them is free-floating in solution. However, it is definitely wrong to give the same kinetic parameter to a rule where both reactants are confined to a lower dimensional compartment—the units are different! Rather than have to write all four rules, it would be preferable to write one rule and assign a different parameter to each compartment, which would be applied according to the highest-dimensional compartment of each reactant.

A second feature missing from existing solutions was the ability to make models in which cycles could form. An example of a cycle would be three proteins where each had a specialized binding site for the others. A species in such a system could either be a cyclic trimer `A(b-1,c-3).B(a-1,c-2).C(b-2,a-3)` or a polymer `A(b-1).B(a-1,c-2).C(b-2,a-3).A(c-3,b-4)...`. Existing methods fail because of the possibility of polymerization, which leads to an infinite number of possible states. If the rules allow for polymerization, then making a finite ODE model is impossible for all but a few special cases. Even if the rules were written in a way that made it impossible for polymerization to happen, existing methods will fail because they look at all the possible bonds of the system at once. We introduce notation that makes it easier to specify rules where cyclization can happen, but polymerization cannot—for example, by saying that a particular monomer will bind only when it is not already present in the species.

We demonstrate the method by building a model of ErbB signaling and fitting it to absolutely quantified measurements of phosphorylated peptides. This model compiled into 620 states. It represented the mechanism of the ErbB pathway in greater detail than conventional models would permit.

5.2 Methods

A rule-based model is composed of compartments, agents, seeds, observables, parameters, and rules. The notation used in a model file of the software implementation will be used to describe the model components here.

A compartment is a container for species. It has a name by which it will be identified elsewhere. It has a dimensionality, which is an integer between 0 and 3 and determines whether the compartment is volume, a membrane, a fiber, or a point. The compartment has a size, which is a positive real number with units appropriate to the dimensionality. Finally, each compartment may have a containing compartment in which this compartment rests. No compartment may be contained by a compartment that in turns contains the first compartment either directly or indirectly. Also, a compartment can only be contained by a compartment that is either of higher dimensionality or exactly one lower dimensionality. This can be understood in the layout of a biological system. The cytoplasm of a cell, which is a 3-dimensional compartment, can only be contained by a 2-dimensional membrane. A 2-dimensional compartment may be contained by a 3-dimensional one, such as the cytoplasmic membrane in the extracellular region, or it may be contained by a 1-dimensional compartment, such as a lipid raft contained by the boundary of a lipid raft. A 1-dimensional compartment may be contained by a 3-dimensional compartment, such as DNA in a nucleus, or it may be contained by a 2-dimensional compartment, such as the boundary of a lipid raft, ignoring 0-dimensional compartments, which do not have much utility.

An agent is a description of an indivisible particle in the system, usually a protein or small molecule. The agent has a name and a list of sites. Each site has a name and a list of possible states. Each state can either be the unmodified state or a name of a particular modification, such as "P" to represent a phosphorylated state. The rule-based model will use the agents to build any multimeric complexes in the model, and each monomer in the complex must be an instance of one of the agents.

An essential concept of rule-based modeling is the graph. Each graph has a set of monomers. Each monomer is an instance of an agent and may list some of the sites on that agent and may have a list of possible compartments. Each site has a list of possible states. Each site is either bound to another specific site in the graph or has a list of possible sites on agents. By convention, sites that are not listed are totally ambiguous; they are equivalent to sites that may be in any state connected to any site. Of all the species in the model, the graph represents the subset that it matches. A species can be represented as a completely unambiguous graph. A graph matches a species if a map can be made from each monomer in the graph to a monomer in the species such that each monomer to which it is mapped is different, the possible compartments on each graph monomer contains the compartment on the species monomer, every bound edge in the graph is present in the species and, for every site, the possible states and edges of the graph contain the actual state and edge of the species. The number of matches is equal to the number of ways that the graph can be aligned with the species divided by the number of ways the graph can be aligned with itself.

Similar to a graph is a pattern. A pattern is also a description of a set of species. It has a graph as its core definition, but it has additional constraints. It has a list of possible compartments for the species as a whole. The species compartment is the compartment that the species as a whole is confined to. It is the lowest dimensional compartment of all the monomer compartments. A pattern also has a Boolean expression of graphs called the disjoint context. For a species to be matched by the pattern, each graph in the disjoint is tested to see if the species contains it. Then the Boolean expression is evaluated by checking whether or not each graph in the expression matches the species and then evaluating those results through the AND, OR, and NOT operators of the Boolean expression. The species matches if the expression evaluates to true and does not match if the expression evaluates to false. The number of matches per graph is not meaningful—the disjoint is merely a filter on the species.

The current implementation limits each graph in the disjoint to a single monomer. This is all that was needed at the time because it is enough to represent compartment constraints, though this limitation could be lifted by changing the algorithm for resolving disjoint context.

A seed describes an amount of a species with which the model will be initialized. It is a completely unambiguous graph—no state, edge, or compartment may be ambiguous.

An observable is defines the states that are measurable for this system. It is a name associated with a list of patterns and a real number indicating how much each pattern contributes to the value of the observable.

A parameter is a kinetic rate constant of the model. It has a name and a nonnegative real value. The units of the parameter must be appropriate for the dimensionality of the rules it will be used in.

A rule is a description of how fast a particular type of reaction occurs. Like a reaction, it has a rate constant, at most two reactants, and any number of products, usually at most two. Instead having specific species for reactants, it has a pattern for each reactant. Any species that matches the pattern undergoes the reaction. In this work, the product is an array of graphs, and the transformation of the rule is discovered by mapping each monomer of the reactant graphs to a monomer of the product. The monomers may be labeled so that the mapping is explicit, but by convention, each unlabeled monomer in the reactants is mapped to the first unlabeled monomer in the products of the same type that has not yet been labeled. Monomers on the reactant side that are not mapped to product monomers are consumed by the rule. Similarly, monomers on the product side not mapped to reactant monomers are produced by the rule. Monomers that are produced must be completely unambiguous. Once the monomers are mapped from reactant to product, the differences in the states and edges define the transformation. If the state of a site changes, it must change to a specific state. Similarly, if the compartment changes, it must change to a specific compartment. Finally, if the edge of a site changes, it must change to an unbound edge or to a bound edge with another site in the product. The rule has a list of parameters and a list of compartments for each parameter. Because of the units on the parameters, bimolecular reactions must take place at a dimensionality consistent with its units. Bimolecular rules must have different parameters when the dimensionality of the reaction is different. By convention, if only one parameter is given, the rule is assumed to apply only to the highest possible dimensionality given the possible compartments of the reactants based on the possible compartments of the monomers considering the seeds and monomer production.

5.2.1 Notation

We designed a file format for defining rule-based models in way that is human and machine readable and writable. Much of the notation is similar to that of BioNetGen files and KroneckerBio files and, to a lesser extent, Kappa files. The format is line-based, but otherwise whitespace invariant. Blank lines are ignored, as are lines beginning with a `#`. The file is divided into sections, of which there are six types: compartments, agents, seeds, observables, parameters, and rules. The beginning of a new section is specified with a section header beginning with a `%` character followed by a name of a section, such as `% Compartments`, `% Agents`, `% Seeds`, `% Observables`, `% Parameters`, or `% Rules`.

The compartment section header has an optional model name that may appear at the end of the line, such as `% Compartments Hagen_ErbB`, which can be interpreted as saying that the following lines are compartments of the Hagen_ErbB model. Each line in a compartment section corresponds to one compartment. The line is simply the name of the compartment, the dimensionality of the compartment, the volume of the compartment, and optionally, the container of the compartment. For example, the compartment section of a model may be:

```
% Compartments Hagen_ErbB
extracellular 3 1
cell_membrane 2 1 extracellular
cell 3 1 cell_membrane
```

Each line in an agents section corresponds to one agent. Each line begins with the name of the agent, optionally followed by parentheses containing a list of sites separated by commas, where each site is optionally followed by a ~ and braces containing list of possible states separated by commas. Each site has a state 0, even if it is not listed, and the braces may be omitted for a site that has a single state listed. Here are some example agents from an ErbB model:

```
% Agents
egf(l)
erbb1(l,d,Y992~P,Y1045~P,Y1068~P,Y1086~P,S1142~P,Y1148~P,Y1173~P,ub~ub)
erbb2(d,Y877~P,Y1023~P,Y1139~P,Y1196~P,Y1221~P,Y1222~P,Y1248~P)
grb2(sh2,sh3N,sh3C)
shc(sh2,ptb,Y239~P,Y240~P,Y317~P)
sos(pp,r)
hras(s,nuc~{GDP,GTP})
```

The seeds section header has an optional compartment name that may appear at the end of the line. Each monomer must be localized to a specific compartment, but it can become burdensome to list the compartment for every monomer. This compartment name acts as the compartment for all monomers in its section that do not have a compartment specified. Each line is the graph of the seed, followed by its initial amount. A graph is a list of monomers separated by periods. Each monomer is the same as an agent except that each site listed must also specify if it is unbound, by appending -0 which can be elided, or if it is bound, by appending a hyphen followed by some other number. Two sites with the same number are bound to each other. A monomer on the compartment is specified by appending to the monomer an at-sign followed by a compartment name. Some example seeds are:

```
% Seeds
egf(l)@extracellular              1
hras(s,nuc~GDP)@cell_membrane     1
% Seeds cell
grb2(sh2,sh3N,sh3C-1).sos(pp-1,r)  1
```

Each line of the observable section has the name of an observable followed by a list of patterns separated by commas with each pattern optionally followed by an equal sign followed by a number that is the contribution to the observable by this pattern. If no contribution is listed, it is assumed to be 1. A pattern is a graph optionally followed by a disjoint and optionally followed by a species compartment. Unlike seed graphs, these graphs can specify ambiguity. Ambiguity in the state or edge is specified by listing multiple items between braces separated by commas. Sites not specified on monomers are completely ambiguous in their states and edges. States or edges marked with a question mark (`~?` or `-?`) are completely ambiguous. States or edges not mentioned on a site are the unmodified state or the unbound edge, respectively. A partial edge may be specified by the name of a monomer followed by an at-sign followed by the name of one of that monomer's sites. While there may be multiple partial edges and unbound edges on a site, a bound edge must be unambiguously connected to a specific site in the graph. The disjoint is a period followed by braces enclosing graphs arraigned in a boolean expression with `|` indicating OR, `&` indicating AND, `^` indicating XOR, and `!` indicating NOT. Parentheses group subexpressions. The species compartment is specified by a period followed by an at-sign followed by braces enclosing compartment names. An integer may be specified for the compartment to indicate all compartments of that dimensionality. The braces may be elided if there is only one compartment listed. An example observable section is:

```
% Observables
ErbB1#pY1068        erbb1(Y1068~P-?)
ErbB1#S1142#pY1148  erbb1(S1142-?,Y1148~P-?)
TotalErbB1          erbb1()
RecruitedSos        sos.@cell_membrane
SosNotBoundToErbB   sos.{!(erbb1|erbb2|erbb3|erbb4)}
ErbB1Dimers         erbb1(d-1).erbb1(d-1)
```

Each line in a parameters section is simply the name followed by its value. An example parameter section is:

```
% Parameters
```

```
egf_erbb1_on      1

egf_erbb1_off     1

erbb1_erbb1_on    1

erbb1_erbb1_off   1
```

Each line in the rules section is one rule. There are at most two reactants, which are patterns exactly as specified in the observables, separated by a plus sign. There are an arbitrary number of products, which are graphs, separated by plus signs. The reactants and products are separated by -> for a forward rule, <- for a reverse rule, and <-> for a reversible rule. The rule is then followed by one rule parameter for the forward and reverse rules and two rule parameters for the reversible rule. Each rule parameter is a pair of parentheses enclosing a list of parameter names optionally followed by an at-sign followed by braces enclosing a list of compartments names with the braces being optional with only one compartment. The compartments are the rule compartments of that parameter. Some example rules are as follows:

```
% Rules
egf(l) + erbb1(l,d) <-> egf(l-1).erbb1(l-1,d) egf_erbb1_on egf_erbb1_off

erbb1(l-!0,d) + erbb1(l-!0,d) -> erbb1(l-!0,d-1).erbb1(l-!0,d-1) erbb1_dimer

shc(Y239~P) + ptpe -> shc(Y239) + ptpe (deP_shc_Y239_3@3, deP_shc_Y239_2@2)
```

5.2.2 Model Assembly

The first set of steps in building the executable model is to simplify the components by removing the many different ways that a user can specify the model and redefining everything in its most general form.

In order refine graphs, it is necessary to know all the possible monomers, compartments of the monomer, sites of the monomers, states of the sites of monomers, and edges of the sites of monomers. We assemble the list of possible agents by collecting the name of every monomer that is introduced via a seed or via production in a rule. On the site of each agent we assemble the list of possible states by collecting the states that are present on a seed monomer or produced via a rule. Also on each site, the list of other sites this site can form an edge with is assembled by collecting the bonds that are present in the seeds or produced in a rule. In the same way, we also assemble the possible compartments of each agent. It is assumed that an agent can only be in a compartment if it is listed in that compartment by a seed or moved to or created in that compartment by a rule. At the same time, we also assemble for each site state and site edge the possible site states and site edges that may coexist with it on the same monomer. For example, some sites may only be bound if they are phosphorylated. In theory, this could be extended to even higher levels of coupling, though at greater computational cost and lower usefulness. It may be desirable to allow transport reactions representing movement from one membrane to other that implicitly moves monomers from one side of the membrane to the equivalent compartment after transport, but this capability was not implemented. This list of agents is similar to the annotated contact map of Feret *et al.*, but ours both contains more information and is less central to our method. While their contact map was the central object in defining the coarse-grained model, ours is only used a reference for the possible values that the sites may have.

The seeds are unique in that they do not need substantial simplification because they are already completely unambiguous graphs.

The observables are simplified first by converting all pattern compartments into disjoint context. To say that a species is located in a specific compartment means that it contains at least one monomer in that compartment and no monomers in a lower dimensional compartment. The agent list tells what monomers could possibly be present in each compartment. These constraints are translated into a Boolean expression and combined so that the disjoint context now must match the old disjoint and one monomer of a correct compartment and none of the monomers with a lower dimensionality.

To process the observables further, we use the process of refinement, which is a technique that will be used in several later steps as well. Refinement takes a graph and creates two new graphs, each of which is less ambiguous than original, but together, the union of the species that each represents is equal to the species that the original represents. An example of this would be a graph that was ambiguous with respect to the state of a particular site, and this graph was split into one graph that had only one of the states and another graph that has all states except the one in the first graph. A special step in refinement is the conversion of a site with one partial edge into a site bound to that specific monomer and site. One of the resulting graphs is with a newly added monomer. However, the refinement also creates the graphs where the refined site binds to each of the sites already present in the graph that is compatible with binding to the site being refined.

The observables are further simplified by converting the disjoint context into explicit context on the graph. This process begins by comparing each graph in the disjoint context to the main graph of the pattern. Any disjoint graph that matches the main graph entirely is replaced with True in the Boolean expression. Any disjoint graph that is completely incompatible with the main graph is replaced with False. A complete incompatibility is typically more difficult to prove than matching. There has to be no way for any of the ambiguous edges of the main graph to connect to a monomer of the disjoint graph in order to declare it incompatible. One determines this by walking through the possible sites in the agent list. Once the True and False disjoint graphs have been evaluated, the Boolean expression is simplified. If the entire disjoint simplifies to True, then the main graph alone describes the species represented by the pattern; the disjoint is already represented and can be discarded. If it simplifies to False, then this pattern represents no species because there is no species that can be matched by the main graph without disobeying the disjoint context; this pattern is discarded entirely because it will not contribute to the observable. If the Boolean expression does not completely simplify, then refinement is needed. The algorithm searches the agent list for a site on the main graph that can connect to a monomer on the disjoint. The site on the main graph is then refined by splitting the graph into one graph that is not bound to that monomer and another graph that is bound to that monomer. The graphs resulting from the refinement are used to create two new patterns that have the same disjoint but different main graphs. The process is then restarted with each of these patterns until all disjoint context has been resolved.

The rules are simplified by converting the parameter compartments into reactant compartments. This is only relevant for bimolecular rules, because zeroth-order and first-order rules do not have rule compartments. Because the reactants of the bimolecular rules must physically collide in order to take part in the reaction, the rule in each compartment must be tracked separately. Each rule with several possible compartments is split to create a new rule for each possible compartment. In order for a rule compartment to be satisfied, one of the reactants must be in that compartment and the other must either be in that compartment also or in an adjacent compartment, one that contains the rule compartment or is contained by it. A new rule is generated for every possible ordered pair of compartments that meets this criterion by intersecting the pattern compartment of the first reactant with the first member of the ordered pair and doing the same with the second reactant and the second compartment. If any resulting rule has a reactant with an empty pattern compartment, that rule is deleted. The rule compartment is now codified in the reactants' pattern compartments.

The rules are further simplified by converting the pattern compartments of the reactants into disjoint context on the reactants. This is done exactly the same way as it is done on the observables patterns. The reactants are simplified further by refining away the disjoint context of the reactants. This is also done in the same way as the observable patterns.

5.2.3 Fragmentation

We build the ordinary differential equation system using a set of graphs called fragments. The following is true for every fragment and for every rule:

1. No partial consumption. If a reactant consumes any instance of a fragment, then it must consume all instances of that fragment. Procedurally, this is determined by aligning the reactant and the fragment. If the alignment is not a complete match with respect to the reactant, then the alignment is partial. If the same alignment does not partially match in the product, then the fragment

is consumed. If the alignment is both partial and consumed, then this is not a proper fragment of the system and must be refined.

2. No partial production. If a product produces any instance of a fragment, it must always produce an instance of a fragment. Procedurally, this is done by doing a partial alignment of the product and fragment. If the alignment is not a complete match with respect to the fragment, then the alignment is partial. If the same alignment does not partially match the reactant, then the fragment is produced. It is not a proper fragment if the alignment is both partial and produced.

The fragmentation algorithm begins by initializing a set of fragment clusters with the graphs of the observables. A mapping is retained from each observable contributor to each fragment cluster. The fragment clusters are unique, so any observable contributors that have identical graphs map to the same fragment cluster. An iteration of the fragmentation algorithm takes each of the new fragment clusters and initializes with the main graph a mutable set of fragments that will become associated with the fragment cluster. The set of fragments is compared to the reactants. If any fragment partially matches the reactant, the fragment is refined toward the reactant so that the resulting fragments are either not consumed or completely consumed by the reactant. The original fragment is replaced by the refined fragments in the set of fragments associated with the fragment cluster. In the set on the fragment clusters, each fragment has a modifier that changes when the fragment is refined. The modifier on the fragments resulting from refinement is equal to the original modifier times the automorphism of the new fragment divided by the automorphism of the original fragment. If comparison to the reactants resulted in any refinement, the set of fragments is compared to the reactants again. Once no further refinement has taken place, which means that there is no partial consumption of these fragments, the list of rules is searched for each fragment and a list of rules that consume each fragment is associated with the fragment. For each bimolecular rule with a reactant that consumes a fragment, the other reactant initializes a new fragment cluster. The rules are searched again to find rules that produce the fragments. If any are found to partially produce the fragment, the products are refined toward the fragment. The refined rules that no longer match the fragment are discarded. The refined version of the rule that fully produces the fragment is associated with the fragment and the reactants of that rule are added as new fragment clusters. The fragmentation algorithm then repeats with all the new fragment clusters created from the various reactants of the rules.

A mass-action model is built using the fragments as species and the rules associated with each fragments are reactions. The compartments of the mass-action model are the same as the rule-based model. The hierarchy of the compartments is not needed to construct a mass-action model. However, one additional default compartment may need to be created for fragments that are not associated with a particular compartment, which is possible for some fragments that never participate in bimolecular rules. The states of the mass-action model are the fragments. For each unimolecular consumption rule associated with the fragment, a reaction is generated with the fragment as a reactant and nothing as the products. For each bimolecular consumption rule associated with the fragment and for each fragment in the fragment cluster that was generated from the other reactant, a reaction is generated with the fragment as one reaction reactant and the fragment from the rule reactant as both the other reaction reactant and the sole reaction product, so that it is not actually consumed in this reaction, it only plays a part in controlling the rate of the reaction. For each zeroth-order production rule associated with the fragment, a reaction is generated with the fragment as a product. For each unimolecular production rule associated with the fragment and for each fragment in the fragment cluster that was the reactant of that rule, a reaction is generated with the fragment from the rule reactant as both the reaction reactant and the product and the fragment as another product. For each bimolecular production rule associated with the fragment, a reaction is generated for every possible combination of fragments in the reactant fragment clusters. The rule reactant fragments are the reaction reactants and products with the fragment being a third product.

The fragmentation algorithm can viewed as a process that started off with the "things we need to know", namely the observables that initialize the set of fragment clusters. The rules change the amount of the observables over time. If the rules only consume part of the things we need to know, then we need to split the things we need to know into groups, namely fragments, that are either fully consumed or fully not consumed. If the rule is unimolecular, then knowing the state of the fragment and the rate constant is all that is needed to compute the reaction rate of that fragment according to that rule. If the rule is bimolecular, the rate at which a fragment is consumed depends on the concentration of the other reactant, so that reactant becomes a new thing we need to know. If a rule produces a fragment, it needs to be made more specific with respect to that fragment, and then its reactants become things we need to know. This is repeated until everything that we need to know is already being tracked. This fragmentation algorithm guarantees that everything we need to know is being tracked and nothing that we do not need to know is being tracked. This by itself results in simulations that are far more efficient than simulating the amounts of all the species in the model. However, the algorithm does not guarantee that everything we need to know is only included once. Exact copies are already handled, but a fragment can be represented by the sum of several other fragments.

A post-processing algorithm is run to ensure that the states needed are included exactly once. For each fragment the algorithm searches for a set of other fragments whose sum is equal to the original fragment. This means that there is a series of refinements that can be made to the original fragment that results in the equivalent set of fragments. A corollary of these requirements is that the original fragment is a subfragment of each member of the set of the equivalent fragments. This means that no fragment can be in each other's equivalent sets unless they are exactly equal, which is prevented when the list of fragments is initially constructed. The algorithm searches for fragments that can represent another fragment by filtering the list for fragments of which the fragment in question is a subset. The fragments that pass through the filter are the candidates to replace this fragment. The fragment is aligned in every possible way with each candidate. Each alignment is refined toward its associated candidate with one random refinement to generate a set of refined graphs. The duplicates in the refinements are removed. If any refined graph is not a subgraph of any candidate, this refinement fails to lead toward finding representative fragments. If the refined graph is found exactly in the list of candidates, it is saved as a possible member of the representative set. If the refined graph is not found exactly but is a subgraph of any of the candidates, it is recursively checked to see if there is a set of representative fragments in the candidates. If all subgraphs produced by a refinement are represented in the set of candidates, either in this iteration or from a recursive iteration, then a representative set of fragments has been found and this is returned. Otherwise, no refinement finds a representative set of fragments. All fragments with a representative set are replaced in the fragment clusters with their representatives.

5.2.4 Model

Using the rule-based modeling software described here, we constructed a model of ErbB signaling. The scope of the model was to explain the behavior of the ErbB pathway in MCF-10A cells [144] in the time scale of 30 seconds to 30 minutes. The concentration of ErbB3 and ErbB4 is very low in this system, so the model only considered the ErbB1 and ErbB2 receptors and their downstream effectors until Erk.

In the model, EGF binds to ErbB1, which allows it to dimerize with other ErbB1 monomers and ErbB2. Each member of the dimer tyrsosine phosphorylates the other. ErbB1 becomes phosphorylated at Y992, Y1045, Y1068, Y1086, Y1148, and Y1173. The evidence is against direct phosphorylation of Y1045 by either receptor, but because the actual kinase for the site is unknown and the phosphorylation does occur indirectly after dimerization, we made the approximation that it is directly phosphorylated by the receptors. The sites undergo non-mechanistic dephosphorylation. Site S1142 is also phosphorylated on ErbB1, but by the serine kinase CamK2. This phosphorylation is modeled as a two-reactant/two-product rule, rather than the mechanistic binding of the kinase followed by phosphorylation within the kinase. This "drive-by phosphorylation" is used here and elsewhere to limit the size of the model, because forming complexes dramatically increases the number of states that need to be tracked.

ErbB1 is degraded by C-Cbl when Y1045 is phosphorylated. This dramatically telescopes the real behavior of the system in which C-Cbl ubiquitinates ErbB1 leading to its internalization, which can lead to either recycling or degradation. ErbB1 is replenished by synthesis.

Shc binds to phosphorylated Y992, Y1148, and Y1173 on ErbB1. Shc itself becomes phosphorylated on Y239 by ErbB1 when bound to it. Grb2 binds to phosphorylated Y1068 and Y1086 on ErbB1 and also on phosphorylated Y239 on Shc. Sos binds to Grb2. When Sos is localized to the membrane, via indirect bonding to the receptors, it catalyzes the conversion of Ras-GDP to Ras-GTP.

Raf binds to Ras-GTP if it is not in a closed conformation with 14-3-3. If Raf is in an open conformation, PP2A dephosphorylates S365, preventing 14-3-3 from returning Raf to a closed conformation. In the same way, if Ksr is in an open conformation, PP2A dephosphorylates S392, preventing 14-3-3 from returning it to a closed conformation. PKA rephosphorylates Raf on S365 and Ksr on S392, allowing both to be reset. Raf can dimerize with Ksr or with itself when both are an open conformation. If it dimerizes with itself, it is an active kinase for Mek on S218 and S222. Finally, Mek is kinase for Erk on T202 and Y204 when Mek is phosphorylated on either S218 or S222.

The ODE model was simulated using the KroneckerBio toolbox in Matlab.

5.2.5 Data

The data was provided by Tim Curran of the Forest White lab. MCF-10A cells were treated with 20 nM EGF. The amount of various phosphopeptides were measured by the inclusion of an internal standard for each peptide. The amounts of all peptides were measured before application of EGF (referred to as 0 seconds) and at 30 s, 1 min, 2 min, 3 min, 5 min, 10 min, and 30 min after the addition of EGF. Depending on the number of times each peptide was detected, there were up to five replicates of each data point.

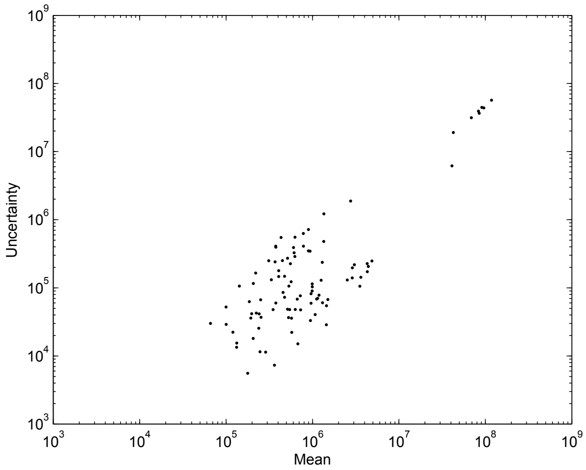

Figure 5-2: **Measurement uncertainty relationship.** For most of the measurements of a particular phosphopeptide at a particular time, there were multiple biological replicates of the measurement. For each set of replicates, the mean and standard deviation was estimated and plotted. A roughly linear relationship between the mean and standard deviation can be seen—a property that was used to estimate the percent measurement uncertainty and the noise floor.

5.2.6 Measurement Uncertainty

The replicates in each point were used to estimate the measurement uncertainty. The mean and standard deviation of all samples of each peptide-time pair was computed. A scatter plot of the mean vs. standard deviation for all points is shown in Figure 5-2. The relationship appears to be roughly linear. A line was fit to the data with weighted least squares to estimate the measurement uncertainty. The weights were the means of each point so that fit weighted a percent error the same for all points. The slope of the fit was 0.26, which indicated a 26% measurement uncertainty. The intercept was 12 000 molecule per cell, which indicated the noise floor.

5.2.7 Fitting

The model was fit to the data using sequential quadratic programming as implemented in Matlab's `fmincon` using an analytical gradient computed using the adjoint method.

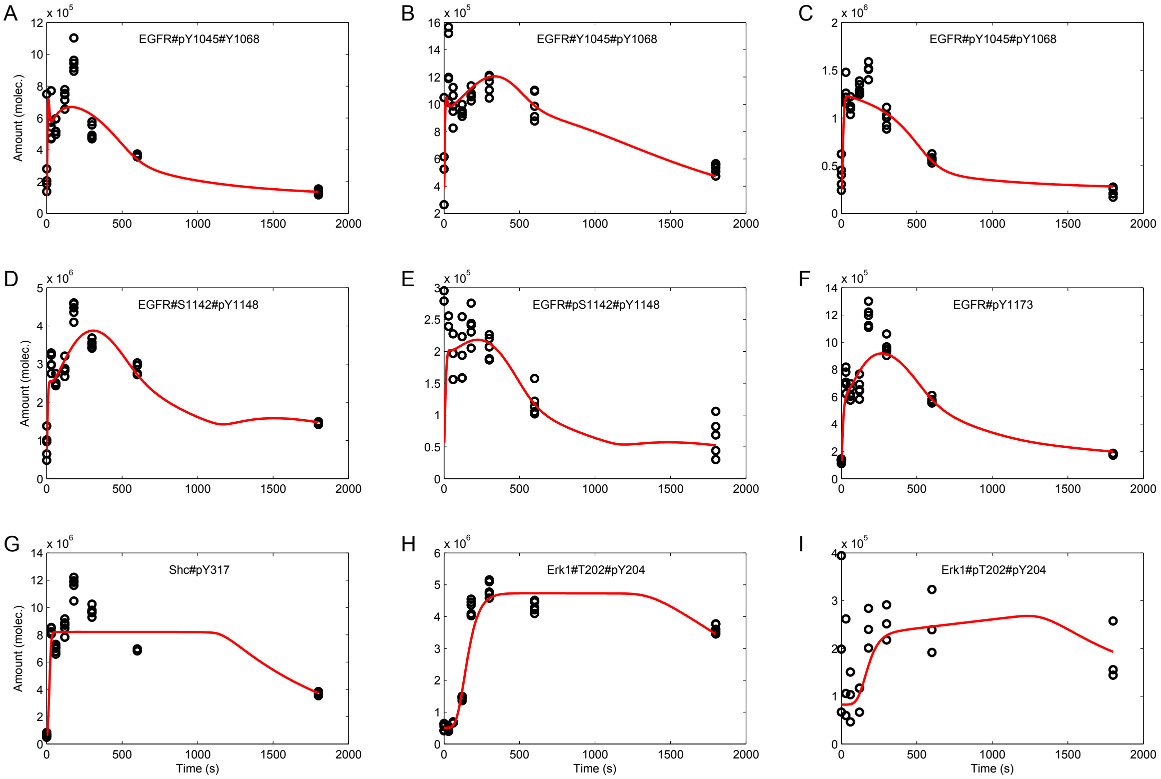

Figure 5-3: **Behavior of best-fit model.** The black circles are the data collected on this system. The red line is the behavior of the model parameterized with the best-fit parameters. Each peptide is named with the protein it comes from followed by the list of phosphorylatable sites on the peptide. If the site begins with a "p", then it is phosphorylated at that site and unmodified otherwise.

5.3 Results

We created a model of the ErbB system using rule-based modeling and fit it to absolutely quantified mass spectrometry data. The best-fit traces to the data are shown in Figure 5-3. Some outputs do not fit particularly well. It remains unknown whether this is due to limitations of the model or due to limitations in the fitting algorithm.

5.4 Conclusion

We built a model of the ErbB system with greater detail in the mechanism of the top of the network than had been achieved with previous ODE models. The phosphorylation and dephosphorylation of each site was modeled individually. The complex binding of Raf, Ksr, and Mek was modeling mechanistically, a portion of the pathway that is usually modeled as a simple kinase cascade. Some mechanisms that we wanted to include had to be simplified due to the model becoming too large to simulate. The internalization and degradation of ErbB1 had to be simplified as just Cbl reacting with Y1045 to eliminate ErbB1. Because the amount of all sites on ErbB1 depended on degradation and most of the states in the model were some form of ErbB1, every addition to this aspect of the model would nearly double the size of the model, whether it was binding of Cbl, ubiquitination itself, binding to clatherin-coating pit protein, or the entirely separate set of states in the internalized compartment.

We had expected to be able to build a larger model. However, we unexpectedly ran into a limitation inherent in our use of Matlab sparse matrices to store and evaluate the ODEs. Matlab uses compressed column storage (CSC) to store sparse matrices. This normally allows us to simulate models very efficiently. However, when the number of states becomes very large, several matrices become exceedingly sparse—far fewer than one entry per row or column. The CSC matrix format must still store one entry for every column. This allows for some algorithms to perform quickly when there is at least one entry per row and column, but this leads to extremely inefficient storage of matrices in which most columns are empty. The matrices in the model required much memory to store and much time to operate over. Theoretically, a different sparse matrix representation could be much more efficient, but an implementation of such a matrix and the necessary matrix operations was not available. In the end, the model had to be limited to about 700 states.

It is unclear how large of a model could be simulated or, more importantly, fit to data, with current computational resources if a more efficient means of evaluating the ODEs were developed. Upon recognition of the immediate limitation of sparse matrix multiplication, we did not aggressively optimize the rule-based modeling code to see how large of models it could construct. The current implementation would construct models with thousands of states in a few hours.

Despite the limitations of the current downstream implementation, the current model was still able to provide insight into the behavior of the ErbB system. This model would have been difficult to construct without the compartmental rule-based modeling features introduced by the algorithm described here.

Chapter 6

Conclusion

If we consider that biological models are made of a topology and a parameter set, then being able to compute the uncertainty in both the parameters and topology represents a complete package in quantifying the uncertainty in biological models. Similarly, having algorithms to design optimal experiments for reducing the uncertainty in both is another complete package.

The use of linearization in the form of the Fisher information matrix to approximate the uncertainty in nonlinear systems has a well-established history. For this reason, it is the only problem in the realm of quantifying uncertainty and optimal experimental design for parameters and topology that did not get a chapter in this thesis. Whether or not an approximation is good depends on the purpose for which it is used. I showed that it is a good approximation for the purpose of predicting how good an experiment is at reducing uncertainty. I also showed that it is a good approximation for quantifying how far the fitted parameters are from their true values when the linear uncertainty is small. The approximation is not good for drawing parameters that are consistent with the data, which is why the Monte Carlo algorithm for optimal experimental design for topology uncertainty uses the Metropolis–Hastings algorithm to sample parameter space.

Previous research had found that optimally chosen experiments could reduce the linearized uncertainty to very low levels. My work found that the nonlinear situation was just as good. Not knowing the parameters beforehand, and therefore, using the wrong parameters to predict the next best experiment still allowed for experiments to be found that were good enough to determine the parameters to very high accuracy. Part of the reason this method was successful was that it was computationally inexpensive. There was no Monte Carlo step needed to test an individual experiment. The evaluation was purely linear. This allowed hundreds of thousands of experiments to be evaluated in a few hours when spread over a few hundred processors. While this problem is called optimal experimental design, this algorithm was more like optimal experimental selection. The experiments had to be enumerated. This means that a continuous search space, such as optimizing the time to take a measurement, cannot be explored with this method without discretizing the space. It would be possible to use gradient descent on the goal function. This would require computing the gradient of the Fisher information matrix. This is an expensive calculation, and it is unlikely that the marginally better experiments would be worth the effort. The findings of this work showed conclusively that it is not a specific, narrowly-defined experiment that is needed to reduce the uncertainty in the model. Instead, it is a set of experiments that are collectively complementary. There is not a specific measurement that needs to be made, but a broad set of measurements under experimental conditions that exercise the system into different modes that is needed to fully understand the system.

The linearization method for topology probability is probably the most substantial contribution of this thesis. Seeing the success of linearization in quantifying parameter uncertainty, I wondered if a similar approach would work for topology uncertainty. There is an analytical solution for the topology probability of linear models. There are different ways of expressing this solution, which give the same answer for linear models, but of the ones I tested, only the one described in this thesis gave a robustly accurate answer for nonlinear models. The existing heuristics were not good approximations for biological models. This is probably because they all assume that the number of measurements is so large that the parameter uncertainty is small. That would only be true for someone who ran the experimental design algorithm for parameter uncertainty. In all other cases, the uncertainty in the models is enormous. It is necessary in systems biology to be able to quantify the topology uncertainty before all the parameters have been effectively determined.

I used to believe that a straightforward and efficient Monte Carlo algorithm existed for any statistical question. The topology probability problem is a counter-example. Sampling from the posterior parameter distribution is not sufficient to compute the marginal likelihood. It is necessary to sample from a large carefully designed sequence of distributions. The extraordinary computational expense of computing the gold-standard answer for topology probability reinforces the need for a fast and accurate approximation. The linearization method clearly filled this role for the MAPK topologies. It remains to be seen if there are scenarios where it is not a good approximation. I suspect that when the parameter uncertainty is extremely large, the algorithm will begin to break down as the robustness term swamps all other terms. I never got to fully investigate this. Comparing to a gold standard for larger models is probably impossible as it took weeks to compute for four topologies with only a dozen parameters.

Interestingly, there is no analytical solution for optimal experimental design for linear topologies, though this is only when the target function is the entropy. This was disappointing because the entropy of the parameter distribution could be minimized analytically with the goal function to minimize the determinant of the variance matrix. There are some functions with reasonable properties of the topology probability that have analytical solutions for their expected value. These methods were never fully investigated, though I suspect that some would work nicely for evaluating more experiments than can be currently tested with the Monte Carlo method.

As larger models are needed, several computational bottlenecks will arise. The simplest living organism has a few hundred genes, which in on the same order of magnitude as the Chen model from Chapter 4 and ErbB model from Chapter 5. Some portions of the ErbB model had be removed in order to get it down to this size because, above about 1000 states, it was no longer possible to build the model. The immediate limitation is the sparse matrix representation in Matlab. By using the CSC format for sparse matrices, Matlab requires at least one entry for every matrix column. Some matrices in a large model have an enormous number of columns, usually because they represent a matrix derivative, which is naturally a higher-dimensional array, but has been flattened to two dimensions because a sparse representation is only available in two dimensions. After flattening, almost all columns themselves are empty, but still need to be stored in memory. None of the popular scientific computing environments have higher-dimensional sparse arrays that support dot products. Despite the substantial computational improvement that the KronckerBio formulation of the model gives, the sparse matrix implementation will limit the size of models that it can build.

Another looming limitation is that the best stiff ODE solvers, namely `ode15s`, take the sparse LU decomposition of the jacobian. In order to compute the Fisher information matrix, which is needed for all the higher-level algorithms except for fitting, it is necessary to integrate the sensitivities of all the states with respect to all the parameters. If the number of states and parameters each grow linearly with the size of the model, then this integration grows with the square of the size of the model and the jacobian grows with the fourth power of the size of the model. The LU decomposition already dominates the cost of computing the Fisher information matrix for the largest models I have used. It will be intractable once models get much larger than 1000 states. It is possible that a stiff ODE integrator will be invented that scales better with the number of states, but given that stiff ODE integrators have been an active area of research for years, it seems unlikely that major advances will be forthcoming. A possible avenue of advancement would be to find a way to compute the Fisher information matrix without computing the sensitivities. This would take advantage of the highly regular structure in the sensitivities calculations in order to compute the Fisher information matrix through the integration of a smaller alternative system much like the gradient can be calculated through the adjoint system rather than naively through the sensitivities.

For models of proteins with multiple sites of binding or modification, the limitations are immediate. While a fragmentation algorithm with a rule-based model can astronomically reduce the size of ODE models of these systems, it cannot reduce them enough to accommodate the biology that we would like to include. The fundamental problem is that any ODE model must, for any site, track every possible combination of sites that affect that site either directly or indirectly. This grows exponentially with the number of sites that interact. I found that I could not include all the sites that I wanted to interact and ultimately had to greatly simplify the model.

One way around the problem of exploding ODE states is to not track all possible states, but track the individual particles. This is the main feature of the BioNetGen and Kappa packages. Such systems can use the same rule-based models, but simulate the system stochastically by calculating the probability of each rule applying to each species. Here the cost of the algorithm is dependent on the number of particles and not the number of possible species. Thus, if there are a large number of molecules in the model, then the simulation is very costly. Even if the assumptions of mass action are valid, some model may have to be simulated stochastically because the number of states is so large. The greatest disadvantage of stochastic simulation is that one cannot compute a gradient with which to inform parameter fitting. Without an accurate gradient to follow, then it is unlikely that one will be able to find the best-fit parameters of a model.

It remains to be seen if stitching together a model from multiple types of simulation, ODE, PDE, stochastic, and spatial stochastic, can make an excellent platform for systems biology. Ideally, the correct algorithm could be chosen automatically based on whether or not the species met the assumptions of each method. Alternatively, it may be simpler to simulate everything at the greatest common denominator, which is spatial stochastic simulation. Possibly this could be done with specialized hardware. In any case, any element of stochasticity prevents the model was being optimized by gradient descent. It may be possible to use an ODE model in optimization, even if the assumptions are not met, and transfer the parameters to a stochastic simulation for an actual look at the system. It is unknown what relation, if any, the best-fit parameters of an ODE model have to the parameters of a model that is truly a stochastic model. Most of them would probably be close, but some could be very wrong.

In the end, switching to stochastic algorithms because the ODE algorithms became too computationally expensive is like moving to the North Pole because one cannot stand the cold winters in Boston. For all models built to date, simulating the system with an ODE is radically faster than simulating it stochastically. Eventually, computer hardware will be constructed that can simulate a large spatial stochastic model. It is difficult to conceive of this happening in the near future without some specialized hardware invented.

Until then, ODEs are the fastest method for simulating mechanistic biological models. This thesis presented several new algorithms and demonstrated several old algorithms for quantifying the uncertainty in these models and finding experiments that most efficiently reduce this uncertainty. Linearization is the central technique. Given the highly nonlinear behavior for which biological models are famous, its generality is somewhat surprising. Perhaps this can be credited to the universality of Taylor's theorem and the central limit theorem, which together say, in laymans terms, everything eventually becomes a linear Gaussian.

Appendix A

Material for Chapter 2

Parameter	Value	Parameter	Value
krbEGF	2.1850E-05	kdErk	8.8912
kruEGF	0.0121	KmdErk	3496490
krbNGF	1.3821E-07	kpP90Rsk	0.0214
kruNGF	0.0072	KmpP90Rsk	763523
kEGF	694.7310	kPI3K	10.6737
KmEGF	6086070	KmPI3K	184912
kNGF	389.4280	kPI3KRas	0.0771
KmNGF	2112.7	KmPI3KRas	272056
kdSos	1612.0	kAkt	0.0566
KmdSos	896896	KmAkt	653951
kSos	32.3440	kdRaf1ByAkt	15.1212
KmSos	35954	KmRaf1ByAkt	119355
kRasGap	1509.4	kC3GNGF	146.9120
KmRasGap	1432410	KmC3GNGF	12876
kRasToRaf1	0.8841	kC3G	1.4015
KmRasToRaf1	62465	KmC3G	10966
kpRaf1	185.7590	kRapGap	27.2650
KmpRaf1	4768350	KmRapGap	295990
kpBRaf	125.0890	kRap1ToBRaf	2.2100
KmpBRaf	157948	KmRap1ToBRaf	1025460
kdMek	2.8324	kdRaf1	0.1263
KmdMek	518753	KmdRaf1	1061.7
kpMekCytoplasmic	9.8537	kdBRaf	441.2870
KmpMekCytoplasmic	1007340	KmdBRaf	10879500

Table A.1: **True parameters.** The parameters of the EGF-NGF network as published by Brown *et al.* and used as the "true" model in our simulations.

Exp.	[EGF] (molec./cell)	[NGF] (molec./cell)	Knocked-down	Over-expressed
Nom.	1000	4560		
1	0	4560	RasGap×1.2, Erk×760	Rap1×67
2	1000	4.56E+05		Ras×7.3, Erk×4.9, PI3K×510
3	1000	4560		Sos×750, Ras×220, Braf×2.3
4	1000	45.6	RasGap×2.2, RapGap×2.7	Mek×32
5	10	4.56E+05	RasGap×780, RapGap×160	Ras×100
6	10	4.56E+07	PI3K×4.6, Rap1×17, Raf1×13	
7	0	4.56E+05	Raf1×7.0	Braf×61, C3G×16
8	1000	4.56E+07	Raf1×13	Braf×3.3, C3G×36
9	1.00E+07	4.56E+07	RapGap×85, Raf1PPtase×580	Braf×590
10	1.00E+07	4.56E+07		Braf×163, C3G×230, Rap1×63

Table A.2: **Experiments chosen for randomly effective experiments.** We tested how effective the experiments would still be at reducing parameter uncertainty if the knock-down and over-expression effectiveness was randomly chosen on a log scale between 1-fold and 1000-fold rather than the 100-fold change under which the experiments were chosen. The number of experiments needed to reduce all parameter uncertianties below 10% increased from 6 to 10 for the first goal function. The number next to each protein indicates the amount by which it was actually changed rounded to two significant digits.

	10% uncertainty	20% uncertainty	100% uncertainty
10% threshold	6	6	22
20% threshold	4	5	12
100% threshold	3	4	5

Table A.3: **Effect of measurement uncertainty on parameter convergence.** Using the true parameters, the Fisher information matrices for the candidate experiments were computed. Optimal experiments were repeatedly chosen using the first goal function with different levels of target parameter uncertainty. The parameters were not refit between experiments and table reports the number of experiments needed to bring the uncertianties in all parameters below the desired threshold. The elements on the diagonal would be the same if the system was exactly linear with Gaussian noise.

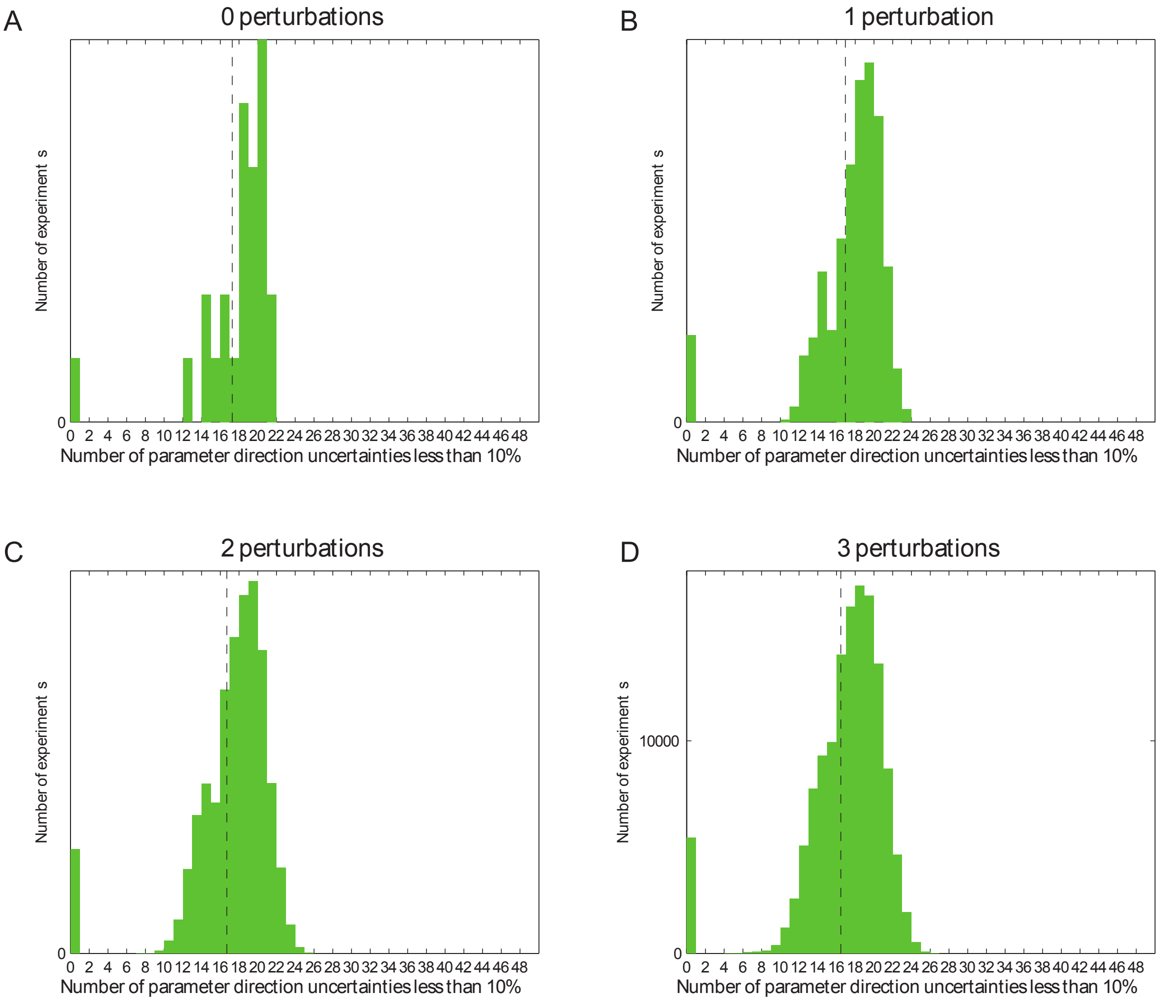

Figure A-1: **Experiment effectiveness by number of perturbations.** All 150 475 experiments were grouped according to whether they represented 0, 1, 2, or 3 knockdowns or over-expressions to the network. The Fisher information matrix was computed for each experiment according to the true model. For each group, the experiments were binned according to how many parameter eigendirections were known better than 10%. The histograms for the groups are shown with 0, 1, 2, and 3 perturbations corresponding to A, B, C, and D, respectively. The vertical dashed line shows the mean of each histogram. That the means and distributions are about the same suggests that some numbers of perturbations are not inherently more informative than others for this model.

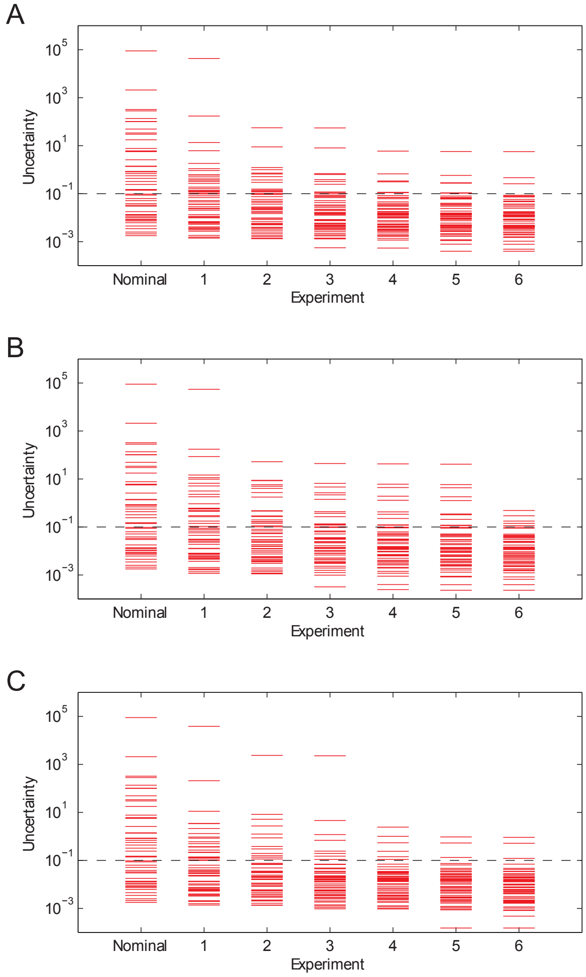

Figure A-2: **Mixing experiments between goal functions.** The three sets of six experiments generated by each of the three goal functions were mixed into three new sets. In subplot A, the mixed set of experiments was experiments 1, 2, 3, 4, 5, and 6 from goal functions 1, 2, 3, 1, 2, and 3, respectively. In subplot B, the mixed set of experiments was experiments 1, 2, 3, 4, 5, and 6 from goal functions 2, 3, 1, 2, 3, and 1, respectively. In subplot C, the mixed set of experiments was experiments 1, 2, 3, 4, 5, and 6 from goal functions 3, 1, 2, 3, 1, and 2, respectively. The uncertainty in parameter eigendirections according to the cumulative information from these experiments as calculated by the true model is shown. Several directions remain worse than 10%, unlike the unmixed sets that the experiments came from (Figure 2-3). This suggests that while each goal function finds a good self-complementary set of experiments, the order in which the complementarity is chosen differs so that complementary experiments in one set are not complementary experiments in a different set.

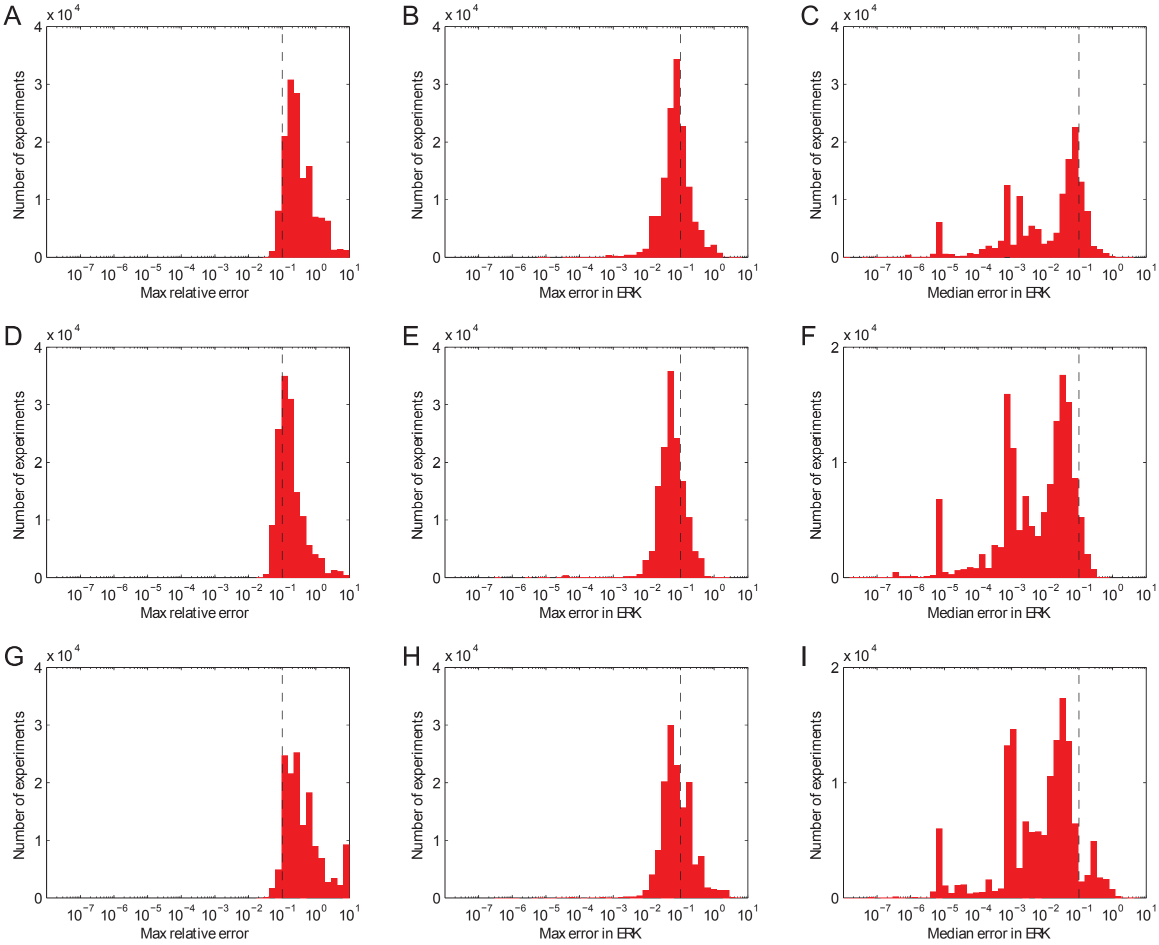

Figure A-3: **Prediction errors after 3 experiments.** All 150 475 candidate experiments were simulated with the model fit to only the first three optimal experiments chosen from the greedy optimization. The relative error of the predictions made by the fitted model were summarized in the same way as Figure 4. Here each column of plots is a different summary and each row is a different goal function. A, B, and C are for goal function 1; D, E, and F are for goal function 2; and G, H, and I are for goal function 3. A, D, and G are the overall maximum relative error results; B, E, and H are the maximum ERK relative error results; and C, F, and I are the median ERK relative error results. Because many of the parameter errors at this stage are still greater than 10% (Figure 2-3), the prediction errors tend to also be greater than 10% (dashed black line). The prediction errors in ERK tend to cluster around 10% error, but this is worse than the model fitted to all 6 experiments, where the errors tended to be confined below 10% (Figure 2-5).

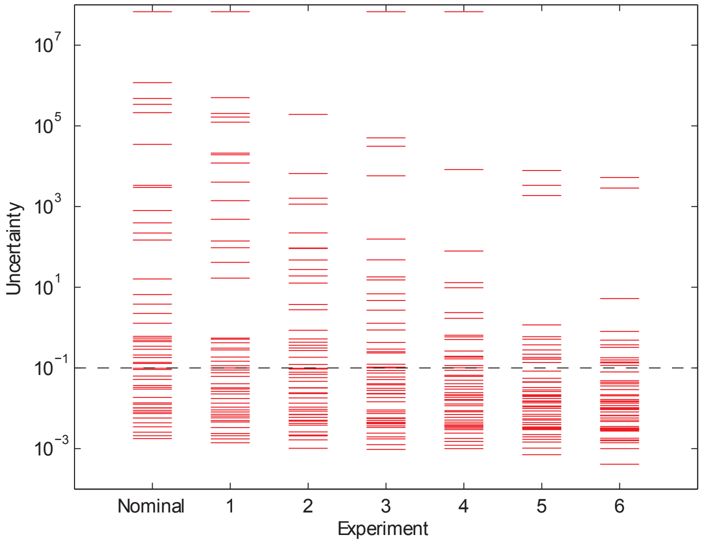

Figure A-4: **Uncertainty after random experiments.** By choosing a set of random experiments, we showed that a diversity of experiments alone was ineffective at reducing all parameter uncertainties below the desired threshold of 10%, unlike the choosing of complementary experiments (Figure 2-3). Six random experiments total were chosen. As each random experiment was chosen, the model was fit to the cumulative data and the information matrix was computed. The uncertainty in parameter eigendirections is shown.

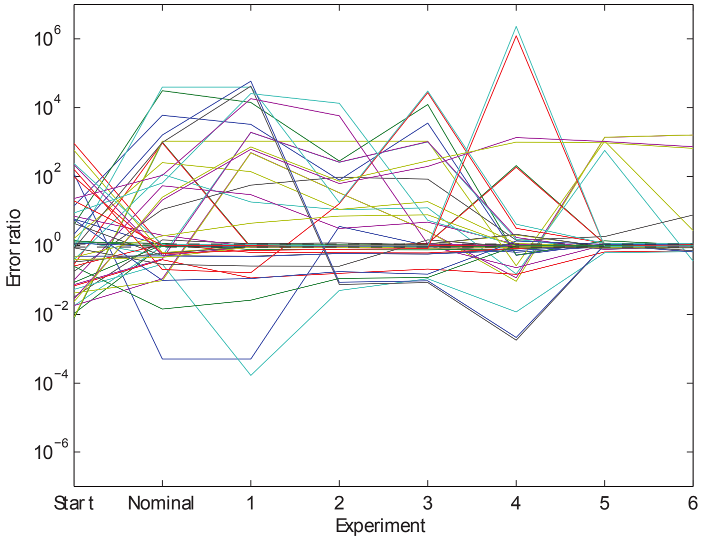

Figure A-5: **Uncertainty after random experiments.** The difference between the true parameters and the fitted parameters was computed after each randomly added experiment. The relative error in the parameters at each step is shown. The parameters do not converge to their true values after 6 experiments, which is consistent with the uncertainties at this point (Figure A-4) and worse than choosing complementary experiments according to any of the three goal functions (Figure 2-4).

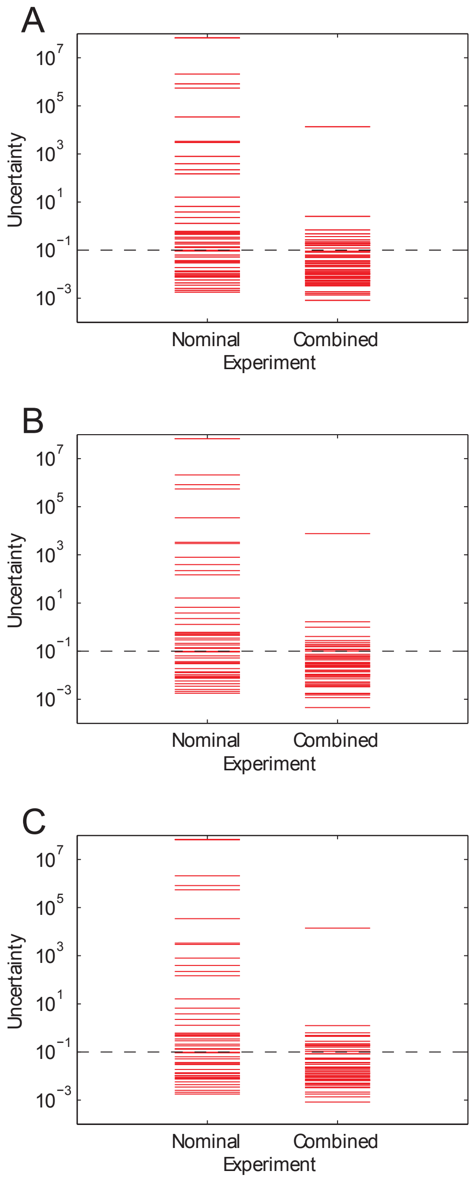

Figure A-6: **Uncertainty after single experiment with all perturbations.** A single experiment was created which implemented all the perturbations recommended by each set of optimal experiments. If a single protein was both knocked-down and over-expressed by separate experiments, the knock-down was used. This tested if the advantage from perturbations can be obtained independently in a single experiment. The uncertainty in parameter eigendirections is shown in the "Combined" column with "Nominal" for reference. The three plots refer to the three goal functions. High uncertainty compared to doing each perturbation separately (Figure 2-3) suggests that information is lost when the perturbations of multiple experiments are combined.

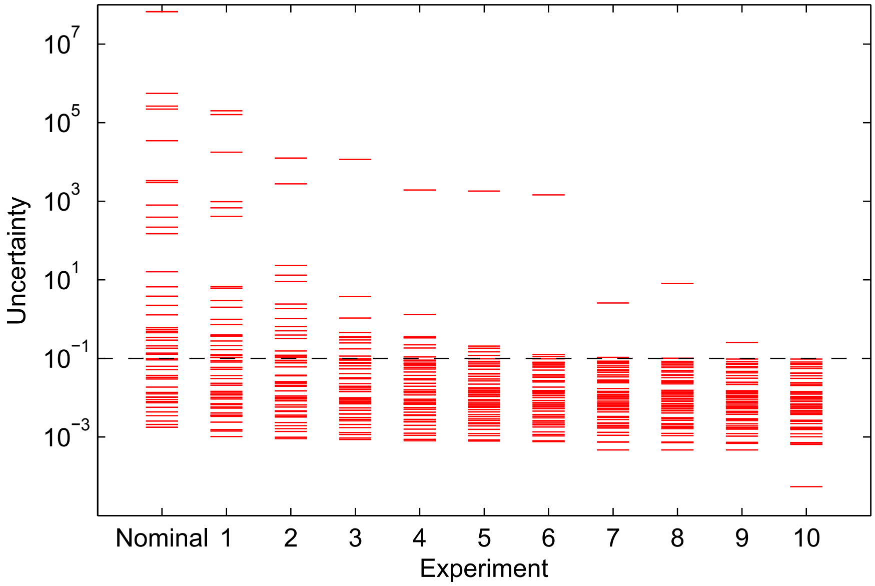

Figure A-7: **Uncertainty with randomly effective experiments.** Rather than the knock-down and over-expression effectiveness performing at exactly 100-fold when generating the data, the actual data was generated using a perturbation between 1 and 1000 on a log scale. The model was refit between experiments.

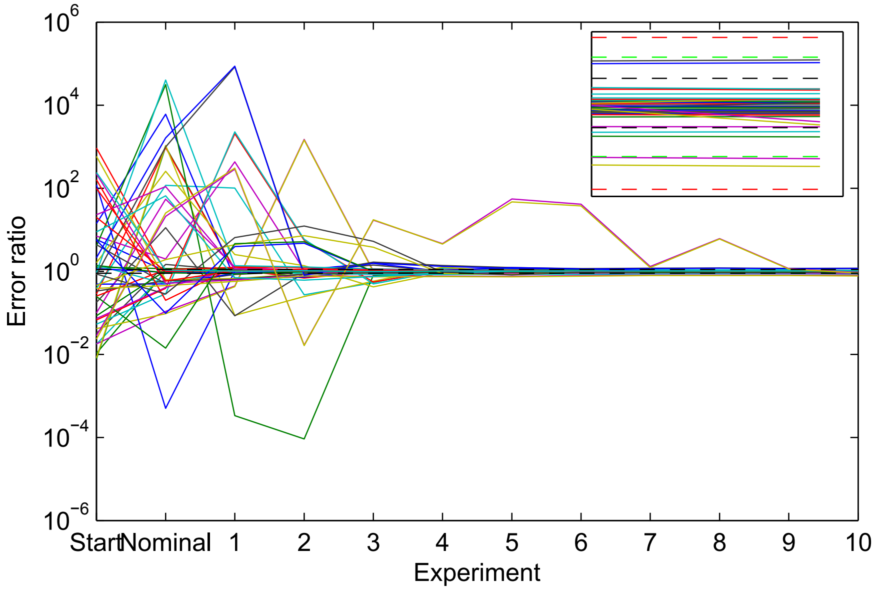

Figure A-8: **Convergence with randomly effective experiments.** The convergence of parameters to their true values is consistent with the uncertainty spectra in Figure A-7. It appears that, occasionally, the precision of the parameters falls short of the target, but eventually all parameters are known with lower error.

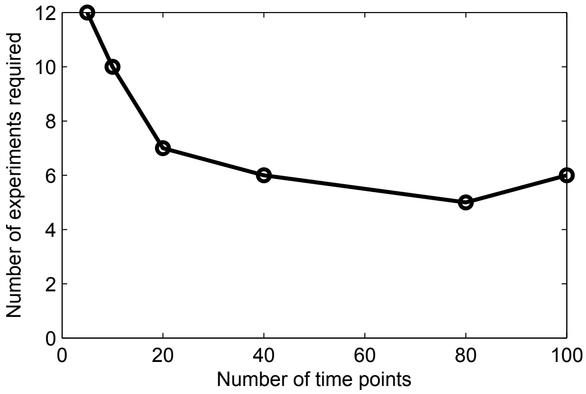

Figure A-9: **Experiments needed versus time points taken.** The number of time points at which measurements were taken was varied. The plot shows the number of additional experiments beyond the nominal experiment needed to reduce all parameter uncertainties to less than 10%. The time points were still taken linearly spaced over the time course of each experiment. We see an expected general trend of requiring fewer experiments as the amount of data per experiments and reduction in the marginal benefit to more data when much data has already been collected under those conditions. The occasional increase in needed experiments when the amount of data is increased probably reflects the loss of efficiency when using a greedy search algorithm.

Appendix B

Material for Chapter 3

Below is the derivation that the standard form of the marginal likelihood expressed as a normal distribution

$$p_{\hat{y}|m}\left(\hat{y}, m\right) = N\left(\hat{y}, A\left(m\right) \cdot \bar{\theta}\left(m\right) - b\left(m\right), V_{\bar{y}} + A\left(m\right) \cdot V_{\bar{\theta}}\left(m\right) \cdot A\left(m\right)^{T}\right) \tag{3.9}$$

can be expressed as a product of the likelihood evaluated at the best-fit parameter set, the prior evaluated at the best-fit parameter set, and an expression containing the determinant of the posterior variance

$$p_{\hat{y}|m}\left(\hat{y}, m\right) = p_{\hat{y}|\theta,m}\left(\hat{y}, \tilde{\theta}\left(\hat{y}, m\right), m\right) \cdot p_{\theta|m}\left(\tilde{\theta}\left(\hat{y}, m\right), m\right) \cdot \left\|\tau \cdot V_{\tilde{\theta}}\left(\hat{y}, m\right)\right\|^{\frac{1}{2}} \tag{3.3}$$

For brevity, the arguments for $A(m)$, $\bar{\theta}(m)$, $\tilde{\theta}(\hat{y}, m)$, $V_{\bar{\theta}}\left(m\right)$, and $V_{\tilde{\theta}}\left(\hat{y}, m\right)$ will be elided. Furthermore, $\bar{\theta}(m)$ will be elided because $\hat{y}$ can always be redefined as the difference between the data and the intercept.

Expanding Equation 3.3 with the equivalent expressions for the likelihood (Equation 3.4) and prior (Equation 3.7) gives:

$$\begin{aligned} p_{\hat{y}|m}(\hat{y}, m) \quad = \quad & \left\|\tau \cdot V_{\bar{y}}\right\|^{-\frac{1}{2}} \cdot \exp\left(-\frac{1}{2} \cdot \left(\hat{y} - A \cdot \tilde{\theta}\right)^{T} \cdot V_{\bar{y}}^{-1} \cdot \left(\hat{y} - A \cdot \tilde{\theta}\right)\right) \\ & \cdot \left\|\tau \cdot V_{\bar{\theta}}\right\|^{-\frac{1}{2}} \cdot \exp\left(-\frac{1}{2} \cdot \left(\tilde{\theta} - \bar{\theta}\right)^{T} \cdot V_{\bar{\theta}}^{-1} \cdot \left(\tilde{\theta} - \bar{\theta}\right)\right) \cdot \left\|\tau \cdot V_{\tilde{\theta}}\right\|^{\frac{1}{2}} \end{aligned} \tag{B.1}$$

Combining the exponential terms under one exponential gives:

$$\begin{aligned} p_{\hat{y}|m}(\hat{y},m) \quad = \quad & \left\|\tau \cdot V_{\bar{y}}\right\|^{-\frac{1}{2}} \cdot \left\|\tau \cdot V_{\bar{\theta}}\right\|^{-\frac{1}{2}} \cdot \left\|\tau \cdot V_{\tilde{\theta}}\right\|^{\frac{1}{2}} \\ & \cdot \exp\left(-\frac{1}{2} \cdot \left(\left(\hat{y} - A \cdot \tilde{\theta}\right)^{T} \cdot V_{\bar{y}}^{-1} \cdot \left(\hat{y} - A \cdot \tilde{\theta}\right) + \left(\tilde{\theta} - \bar{\theta}\right)^{T} \cdot V_{\bar{\theta}}^{-1} \cdot \left(\tilde{\theta} - \bar{\theta}\right)\right)\right) \end{aligned} \tag{B.2}$$

This equation will be considered in two parts: the product of determinants in the front:

$$\left\|\tau \cdot V_{\bar{y}}\right\|^{-\frac{1}{2}} \cdot \left\|\tau \cdot V_{\bar{\theta}}\right\|^{-\frac{1}{2}} \cdot \left\|\tau \cdot V_{\tilde{\theta}}\right\|^{\frac{1}{2}} \tag{B.3}$$

and the sum in the exponential:

$$\left(\hat{y} - A \cdot \tilde{\theta}\right)^{T} \cdot V_{\bar{y}}^{-1} \cdot \left(\hat{y} - A \cdot \tilde{\theta}\right) + \left(\tilde{\theta} - \bar{\theta}\right)^{T} \cdot V_{\bar{\theta}}^{-1} \cdot \left(\tilde{\theta} - \bar{\theta}\right) \tag{B.4}$$

The maximum *a posteriori* parameter set has the following definition for linear Gaussian models:

$$\tilde{\theta}\left(\hat{y},m\right) = V_{\tilde{\theta}} \cdot \left(A^{T} \cdot V_{\bar{y}}^{-1} \cdot \left(\hat{y} - b\right) + V_{\bar{\theta}}^{-1} \cdot \bar{\theta}\right) \tag{B.5}$$

Replacing $\tilde{\theta}$ in Equation B.4 with its definition (Equation B.5) gives:

$$\begin{aligned} & \left(\hat{y} - A \cdot V_{\tilde{\theta}} \cdot \left(A^{T} \cdot V_{\bar{y}}^{-1} \cdot \left(\hat{y} - b\right) + V_{\bar{\theta}}^{-1} \cdot \bar{\theta}\right)\right)^{T} \\ & \cdot V_{\bar{y}}^{-1} \\ & \cdot \left(\hat{y} - A \cdot V_{\tilde{\theta}} \cdot \left(A^{T} \cdot V_{\bar{y}}^{-1} \cdot \left(\hat{y} - b\right) + V_{\bar{\theta}}^{-1} \cdot \bar{\theta}\right)\right) \\ & + \left(V_{\tilde{\theta}} \cdot \left(A^{T} \cdot V_{\bar{y}}^{-1} \cdot \left(\hat{y} - b\right) + V_{\bar{\theta}}^{-1} \cdot \bar{\theta}\right) - \bar{\theta}\right)^{T} \\ & \cdot V_{\bar{\theta}}^{-1} \\ & \cdot \left(V_{\tilde{\theta}} \cdot \left(A^{T} \cdot V_{\bar{y}}^{-1} \cdot \left(\hat{y} - b\right) + V_{\bar{\theta}}^{-1} \cdot \bar{\theta}\right) - \bar{\theta}\right) \end{aligned} \tag{B.6}$$

Expanding those terms gives:

$$
\begin{aligned}
&\hat{y}^T \cdot V_{\bar{y}}^{-1} \cdot \hat{y} \\
&-2 \cdot \hat{y}^T \cdot V_{\bar{y}}^{-1} \cdot A \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot \hat{y} + V_{\bar{\theta}}^{-1} \cdot \bar{\theta}\right) \\
&+ \left(A^T \cdot V_{\bar{y}}^{-1} \cdot \hat{y} + V_{\bar{\theta}}^{-1} \cdot \bar{\theta}\right)^T \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot A^T \cdot V_{\bar{y}}^{-1} \cdot A \\
&\cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot \hat{y} + V_{\bar{\theta}}^{-1} \cdot \bar{\theta}\right) \\
&+\bar{\theta}^T \cdot V_{\bar{\theta}}^{-1} \cdot \bar{\theta} \\
&-2 \cdot \bar{\theta}^T \cdot V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot \hat{y} + V_{\bar{\theta}}^{-1} \cdot \bar{\theta}\right) \\
&+ \left(A^T \cdot V_{\bar{y}}^{-1} \cdot \hat{y} + V_{\bar{\theta}}^{-1} \cdot \bar{\theta}\right)^T \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot V_{\bar{\theta}}^{-1} \\
&\cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot \hat{y} + V_{\bar{\theta}}^{-1} \cdot \bar{\theta}\right)
\end{aligned}
\tag{B.7}
$$

Expanding those terms further gives:

$$
\begin{aligned}
&\hat{y}^T \cdot V_{\bar{y}}^{-1} \cdot \hat{y} \\
&-2 \cdot \hat{y}^T \cdot V_{\bar{y}}^{-1} \cdot A \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot A^T \cdot V_{\bar{y}}^{-1} \cdot \hat{y} \\
&-2 \cdot \hat{y}^T \cdot V_{\bar{y}}^{-1} \cdot A \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot V_{\bar{\theta}}^{-1} \cdot \bar{\theta} \\
&+\hat{y}^T \cdot V_{\bar{y}}^{-1} \cdot A \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot A^T \cdot V_{\bar{y}}^{-1} \cdot A \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot A^T \cdot V_{\bar{y}}^{-1} \cdot \hat{y} \\
&+2 \cdot \hat{y}^T \cdot V_{\bar{y}}^{-1} \cdot A \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot A^T \cdot V_{\bar{y}}^{-1} \cdot A \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot V_{\bar{\theta}}^{-1} \cdot \bar{\theta} \\
&+\bar{\theta}^T \cdot V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot A^T \cdot V_{\bar{y}}^{-1} \cdot A \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot V_{\bar{\theta}}^{-1} \cdot \bar{\theta} \\
&+\bar{\theta}^T \cdot V_{\bar{\theta}}^{-1} \cdot \bar{\theta} \\
&-2 \cdot \bar{\theta}^T \cdot V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot A^T \cdot V_{\bar{y}}^{-1} \cdot \hat{y} \\
&-2 \cdot \bar{\theta}^T \cdot V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot V_{\bar{\theta}}^{-1} \cdot \bar{\theta} \\
&+\hat{y}^T \cdot V_{\bar{y}}^{-1} \cdot A \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot A^T \cdot V_{\bar{y}}^{-1} \cdot \hat{y} \\
&+2 \cdot \hat{y}^T \cdot V_{\bar{y}}^{-1} \cdot A \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot V_{\bar{\theta}}^{-1} \cdot \bar{\theta} \\
&+\bar{\theta}^T \cdot V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot V_{\bar{\theta}}^{-1} \cdot \bar{\theta}
\end{aligned}
\tag{B.8}
$$

We will group each term according to whether it is quadratic, linear, or constant with respect to $\hat{y}$.

B.0.1 Quadratic Terms

Collecting the four terms from Equation B.8 that are quadratic with respect to $\hat{y}$ and factoring $\hat{y}$ from the front and back gives:

$$\hat{y}\cdot\left(\begin{array}{l} V_{\bar{y}}^{-1} \\ -2\cdot V_{\bar{y}}^{-1}\cdot A\cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1}\cdot A^T\cdot V_{\bar{y}}^{-1} \\ +V_{\bar{y}}^{-1}\cdot A\cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1}\cdot A^T\cdot V_{\bar{y}}^{-1}\cdot A\cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1}\cdot A^T\cdot V_{\bar{y}}^{-1} \\ +V_{\bar{y}}^{-1}\cdot A\cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1}\cdot V_{\bar{\theta}}^{-1}\cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1}\cdot A^T\cdot V_{\bar{y}}^{-1} \end{array}\right)\cdot\hat{y} \tag{B.9}$$

Factoring the $-A^T\cdot V_{\bar{y}}^{-1}$ term from the front and $A^T\cdot V_{\bar{y}}^{-1}$ from the back of the second, third, and fourth terms of the sum gives:

$$\hat{y}\cdot\left(\begin{array}{l} V_{\bar{y}}^{-1} \\ -V_{\bar{y}}^{-1}\cdot A\cdot\left(\begin{array}{l} 2\cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1} \\ -\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1}\cdot A^T\cdot V_{\bar{y}}^{-1}\cdot A \\ \cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1} \\ -\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1}\cdot V_{\bar{\theta}}^{-1} \\ \cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1} \end{array}\right)\cdot A^T\cdot V_{\bar{y}}^{-1} \end{array}\right)\cdot\hat{y} \tag{B.10}$$

Factoring $\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1}$ from the front and back of the third and fourth terms to collapse the third and fourth terms together gives:

$$\hat{y}\cdot\left(\begin{array}{l} V_{\bar{y}}^{-1} \\ -V_{\bar{y}}^{-1}\cdot A\cdot\left(\begin{array}{l} 2\cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1} \\ -\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1} \\ \cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right) \\ \cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1} \end{array}\right)\cdot A^T\cdot V_{\bar{y}}^{-1} \end{array}\right)\cdot\hat{y} \tag{B.11}$$

Cancelling $\left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)$ in the third term gives:

$$\hat{y} \cdot \left(\begin{array}{l} V_{\bar{y}}^{-1} \\ -V_{\bar{y}}^{-1} \cdot A \cdot \left(\begin{array}{l} 2 \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \\ -\left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \end{array} \right) \cdot A^T \cdot V_{\bar{y}}^{-1} \end{array} \right) \cdot \hat{y} \tag{B.12}$$

Collapsing the second and third terms which now have the same form gives:

$$\hat{y} \cdot \left(V_{\bar{y}}^{-1} - V_{\bar{y}}^{-1} \cdot A \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot A^T \cdot V_{\bar{y}}^{-1}\right) \cdot \hat{y} \tag{B.13}$$

Finally, applying the Woodbury identity [145] gives:

$$\hat{y} \cdot \left(V_{\bar{y}} + A \cdot V_{\bar{\theta}} \cdot A^T\right)^{-1} \cdot \hat{y} \tag{B.14}$$

B.0.2 Linear Terms

Collecting from Equation B.8 the terms that are linear with respect to $\hat{y}$ and factoring $-2 \cdot \hat{y}^T \cdot V_{\bar{y}}^{-1} \cdot A$ from the front and $V_{\bar{\theta}}^{-1} \cdot \bar{\theta}$ from the back gives:

$$-2 \cdot \hat{y}^T \cdot V_{\bar{y}}^{-1} \cdot A \cdot \left(\begin{array}{l} \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \\ -\left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot A^T \cdot V_{\bar{y}}^{-1} \cdot A \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \\ +\left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \\ -\left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \end{array} \right) \cdot V_{\bar{\theta}}^{-1} \cdot \bar{\theta} \tag{B.15}$$

Collapsing the first and third terms which have the same form gives:

$$-2\cdot\hat{y}^T\cdot V_{\bar{y}}^{-1}\cdot A\cdot\left(\begin{array}{l} 2\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1} \\ -\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1}\cdot A^T\cdot V_{\bar{y}}^{-1}\cdot A\cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1} \\ -\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1}\cdot V_{\bar{\theta}}^{-1}\cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1} \end{array}\right)\cdot V_{\bar{\theta}}^{-1}\cdot\bar{\theta} \tag{B.16}$$

Factoring $\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1}$ from the second and third terms gives:

$$-2\cdot\hat{y}^T\cdot V_{\bar{y}}^{-1}\cdot A\cdot\left(\begin{array}{l} 2\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1} \\ -\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1}\cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right) \\ \cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1} \end{array}\right)\cdot V_{\bar{\theta}}^{-1}\cdot\bar{\theta} \tag{B.17}$$

Cancelling $\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1}\cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)$ in the second term gives:

$$-2\cdot\hat{y}^T\cdot V_{\bar{y}}^{-1}\cdot A\cdot\left(\begin{array}{c} 2\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1} \\ -\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1} \end{array}\right)\cdot V_{\bar{\theta}}^{-1}\cdot\bar{\theta} \tag{B.18}$$

Collapsing the first and second terms which now have the same form gives:

$$-2\cdot\hat{y}^T\cdot V_{\bar{y}}^{-1}\cdot A\cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1}\cdot V_{\bar{\theta}}^{-1}\cdot\bar{\theta} \tag{B.19}$$

Transforming $V_{\bar{y}}^{-1}\cdot A\cdot\left(A^T\cdot V_{\bar{y}}^{-1}\cdot A+V_{\bar{\theta}}^{-1}\right)^{-1}$ with the positive definite variant of the Woodbury identity [145] gives:

$$-2\cdot\hat{y}^T\cdot\left(V_{\bar{y}}+A\cdot V_{\bar{\theta}}\cdot A^T\right)^{-1}\cdot A\cdot V_{\bar{\theta}}\cdot V_{\bar{\theta}}^{-1}\cdot\bar{\theta} \tag{B.20}$$

Cancelling $V_{\bar{\theta}}\cdot V_{\bar{\theta}}^{-1}$ gives:

$$-2\cdot\hat{y}^T\cdot\left(V_{\bar{y}}+A\cdot V_{\bar{\theta}}\cdot A^T\right)^{-1}\cdot A\cdot\bar{\theta} \tag{B.21}$$

B.0.3 Constant Terms

Collecting from Equation B.8 the terms that are not functions of $\hat{y}$ and factoring $\bar{\theta}$ from the front and back gives:

$$\bar{\theta}^T \cdot \left(\begin{array}{l} V_{\bar{\theta}}^{-1} \\ -2 \cdot V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1} \right)^{-1} \cdot V_{\bar{\theta}}^{-1} \\ +V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1} \right)^{-1} \cdot A^T \cdot V_{\bar{y}}^{-1} \cdot A \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1} \right)^{-1} \cdot V_{\bar{\theta}}^{-1} \\ +V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1} \right)^{-1} \cdot V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1} \right)^{-1} \cdot V_{\bar{\theta}}^{-1} \end{array} \right) \cdot \bar{\theta} \tag{B.22}$$

Factoring $-V_{\bar{\theta}}^{-1}$ from the front and $V_{\bar{\theta}}^{-1}$ from the back of the second, third, and fourth terms gives:

$$\bar{\theta}^T \cdot \left(\begin{array}{l} V_{\bar{\theta}}^{-1} \\ -V_{\bar{\theta}}^{-1} \cdot \left(\begin{array}{l} 2 \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1} \right)^{-1} \\ -\left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1} \right)^{-1} \cdot A^T \cdot V_{\bar{y}}^{-1} \cdot A \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1} \right)^{-1} \\ -\left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1} \right)^{-1} \cdot V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1} \right)^{-1} \end{array} \right) \cdot V_{\bar{\theta}}^{-1} \end{array} \right) \cdot \bar{\theta} \tag{B.23}$$

Factoring $\left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1} \right)^{-1}$ from the front and back of the third and fourth terms gives:

$$\bar{\theta}^T \cdot \left(\begin{array}{l} V_{\bar{\theta}}^{-1} \\ -V_{\bar{\theta}}^{-1} \cdot \left(\begin{array}{l} 2 \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1} \right)^{-1} \\ -\left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1} \right)^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1} \right) \\ \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1} \right)^{-1} \end{array} \right) \cdot V_{\bar{\theta}}^{-1} \end{array} \right) \cdot \bar{\theta} \tag{B.24}$$

Canceling $\left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)$ in the third term gives:

$$\bar{\theta}^T \cdot \begin{pmatrix} V_{\bar{\theta}}^{-1} \\ -V_{\bar{\theta}}^{-1} \cdot \begin{pmatrix} 2 \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \\ -\left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \end{pmatrix} \cdot V_{\bar{\theta}}^{-1} \end{pmatrix} \cdot \bar{\theta} \tag{B.25}$$

Collapsing the second and third terms which now have the same form gives:

$$\bar{\theta}^T \cdot \left(V_{\bar{\theta}}^{-1} - V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot V_{\bar{\theta}}^{-1}\right) \cdot \bar{\theta} \tag{B.26}$$

Applying the Woodbury identity [145] to $V_{\bar{\theta}}^{-1} - V_{\bar{\theta}}^{-1} \cdot \left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1} \cdot V_{\bar{\theta}}^{-1}$ gives:

$$\bar{\theta}^T \cdot A^T \cdot \left(V_{\bar{y}} + A V_{\bar{\theta}} A^T\right)^{-1} \cdot A \cdot \bar{\theta} \tag{B.27}$$

B.0.4 Determinant Terms

Combining all determinant terms from Equation B.3 under a single exponent gives:

$$\left(\|\tau \cdot V_{\bar{y}}\| \cdot \|\tau \cdot V_{\bar{\theta}}\| \cdot \left\|\tau \cdot V_{\tilde{\theta}}\right\|^{-1}\right)^{-\frac{1}{2}} \tag{B.28}$$

Pulling out $\tau$from each determinant gives:

$$\left(\tau^{n_{\bar{y}}} \cdot \|V_{\bar{y}}\| \cdot \tau^{n_\theta} \cdot \|V_{\bar{\theta}}\| \cdot \tau^{-n_\theta} \cdot \left\|V_{\tilde{\theta}}\right\|^{-1}\right)^{-\frac{1}{2}} \tag{B.29}$$

Cancelling $\tau^{n_\theta} \cdot \tau^{-n_\theta}$ gives:

$$\left(\tau^{n_{\bar{y}}} \cdot \|V_{\bar{y}}\| \cdot \|V_{\bar{\theta}}\| \cdot \left\|V_{\tilde{\theta}}\right\|^{-1}\right)^{-\frac{1}{2}} \tag{B.30}$$

Replacing $V_{\tilde{\theta}}$ with its linear definition $\left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1}$ gives:

$$\left(\tau^{n_{\bar{y}}} \cdot \|V_{\bar{y}}\| \cdot \|V_{\bar{\theta}}\| \cdot \left\|\left(A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right)^{-1}\right\|^{-1}\right)^{-\frac{1}{2}} \tag{B.31}$$

Cancelling the stacked negative exponents gives:

$$\left(\tau^{n_{\bar{y}}} \cdot \|V_{\bar{y}}\| \cdot \|V_{\bar{\theta}}\| \cdot \left\|A^T \cdot V_{\bar{y}}^{-1} \cdot A + V_{\bar{\theta}}^{-1}\right\|\right)^{-\frac{1}{2}} \tag{B.32}$$

Applying a determinant identity gives:

$$\left(\tau^{n_{\bar{y}}} \cdot \left\|V_{\bar{y}} + A \cdot V_{\bar{\theta}} \cdot A^T\right\|\right)^{-\frac{1}{2}} \tag{B.33}$$

Finally, bringing the τ back into the determinant gives:

$$\left\|\tau \left(V_{\bar{y}} + A \cdot V_{\bar{\theta}} \cdot A^T\right)\right\|^{-\frac{1}{2}} \tag{B.34}$$

B.0.5 Recombination

Recombining the simplified quadratic (Equation B.14), linear (Equation B.21), and constant (Equation B.27) expressions, which together equal Equation B.8, gives:

$$\hat{y} \cdot \left(V_{\bar{y}} + A \cdot V_{\bar{\theta}} \cdot A^T\right)^{-1} \cdot \hat{y} - 2 \cdot \hat{y}^T \cdot \left(V_{\bar{y}} + A \cdot V_{\bar{\theta}} \cdot A^T\right)^{-1} \cdot A \cdot \bar{\theta} + \bar{\theta}^T \cdot A^T \cdot \left(V_{\bar{y}} + A \cdot V_{\bar{\theta}} A^T\right)^{-1} \cdot A \cdot \bar{\theta} \tag{B.35}$$

Factoring the quadratic equation gives:

$$\left(\hat{y} - A \cdot \bar{\theta}\right)^T \cdot \left(V_{\bar{y}} + A \cdot V_{\bar{\theta}} \cdot A^T\right)^{-1} \cdot \left(\hat{y} - A \cdot \bar{\theta}\right) \tag{B.36}$$

Replacing the simplified determinant (Equation B.34) and the simplified quadratic equation (Equation B.36) in Equation B.2 gives:

$$p_{\hat{y}|m}(\hat{y}, m) = \left\|\tau \cdot \left(V_{\bar{y}} + A \cdot V_{\bar{\theta}} \cdot A^T\right)\right\|^{-\frac{1}{2}} \cdot \exp\left(-\frac{1}{2} \cdot \left(\hat{y} - A \cdot \bar{\theta}\right)^T \cdot \left(V_{\bar{y}} + A \cdot V_{\bar{\theta}} \cdot A^T\right)^{-1} \cdot \left(\hat{y} - A \cdot \bar{\theta}\right)\right) \tag{B.37}$$

Given that the probability density function of a normal distribution is:

$$N\left(x, \mu, V\right) = \left\|\tau \cdot V\right\|^{-\frac{1}{2}} \cdot \exp\left(-\frac{1}{2} \cdot \left(x - \mu\right)^T \cdot V^{-1} \cdot \left(x - \mu\right)\right) \tag{B.38}$$

it is clear that the linearization formula (Equation 3.3) is equivalent to a normal distribution of variable $\hat{y}$ with a mean of $A \cdot \bar{\theta}$ and a variance of $V_{\bar{y}} + A \cdot V_{\bar{\theta}} \cdot A^T$, which is the standard marginal likelihood given in Equation 3.9.

Bibliography

[1] Francis S. Collins, June 2000.

[2] Christopher B. Newgard and Alan D. Attie. Getting biological about the genetics of diabetes. *Nature Medicine*, 16(4):388–391, April 2010.

[3] Norihiro Kato. Candidate genes revisited in the genetics of hypertension and blood pressure. *Hypertension Research*, 36(12):1032–1034, December 2013.

[4] Aldi T. Kraja, Steven C. Hunt, D. C. Rao, Victor G. Dvila-Romn, Donna K. Arnett, and Michael A. Province. Genetics of hypertension and cardiovascular disease and their interconnected pathways: Lessons from large studies. *Current Hypertension Reports*, 13(1):46–54, February 2011.

[5] Donna L. Hoyert and Jiaquan Xu. Deaths: Preliminary data for 2011. *National Vital Statistics Report*, 61(6), October 2012.

[6] Linus Pauling, Harvey A. Itano, S. J. Singer, and Ibert C. Wells. Sickle cell anemia, a molecular disease. *Science*, 110(2865):543–548, November 1949.

[7] Wanda K. Lemna, Gerald L. Feldman, Bat-sheva Kerem, Susan D. Fernbach, Elaine P. Zevkovich, William E. O'Brien, John R. Riordan, Francis S. Collins, Lap-Chee Tsui, and Arthur L. Beaudet. Mutation analysis for heterozygote detection and the prenatal diagnosis of cystic fibrosis. *New England Journal of Medicine*, 322(5):291–296, February 1990.

[8] James F. Gusella, Nancy S. Wexler, P. Michael Conneally, Susan L. Naylor, Mary Anne Anderson, Rudolph E. Tanzi, Paul C. Watkins, Kathleen Ottina, Margaret R. Wallace, Alan Y. Sakaguchi, Anne B. Young, Ira Shoulson, Ernesto Bonilla, and Joseph B. Martin. A polymorphic DNA marker genetically linked to huntington's disease. *Nature*, 306(5940):234–238, November 1983.

[9] Wayne Materi and David S. Wishart. Computational systems biology in drug discovery and development: Methods and applications. *Drug Discovery Today*, 12(78):295–303, April 2007.

[10] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, December 1998.

[11] Soumya Raychaudhuri, Joshua M. Stuart, and Russ B. Altman. Principal components analysis to summarize microarray experiments: Application to sporulation time series. *Pacific Symposium on Biocomputing*, pages 455–466, 2000.

[12] Paul Geladi and Bruce R. Kowalski. Partial least-squares regression: a tutorial. *Analytica Chimica Acta*, 185:1–17, 1986.

[13] Nir Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805, February 2004.

[14] Peter J. E. Goss and Jean Peccoud. Quantitative modeling of stochastic systems in molecular biology by using stochastic petri nets. *Proceedings of the National Academy of Sciences*, 95(12):6750–6755, June 1998.

[15] Sui Huang. Gene expression profiling, genetic networks, and cellular states: An integrating concept for tumorigenesis and drug discovery. *Journal of Molecular Medicine*, 77(6):469–480, June 1999.

[16] Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, April 2005.

[17] Jing Yu, V. Anne Smith, Paul P. Wang, Alexander J. Hartemink, and Erich D. Jarvis. Advances to bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, 20(18):3594–3603, December 2004.

[18] Seiya Imoto, Tomoyuki Higuchi, Takao Goto, Kousuke Tashiro, Satoru Kuhara, and Satoru Miyano. Combining microarrays and biological knowledge for estimating gene networks via bayesian networks. *Journal of Bioinformatics and Computational Biology*, 02(01):77–98, March 2004.

[19] Steven S. Andrews, Tuan Dinh, and Adam P. Arkin. Stochastic models of biological processes. In Robert A. Meyers Ph.D, editor, *Encyclopedia of Complexity and Systems Science*, pages 8730–8749. Springer New York, January 2009.

[20] Albert Einstein. ber die von der molekularkinetischen theorie der wrme geforderte bewegung von in ruhenden flssigkeiten suspendierten teilchen. *Annalen der Physik*, 322(8):549–560, 1905.

[21] M.-Y. Hsieh, S. Yang, M. A. Raymond-Stinz, S. Steinberg, D. G. Vlachos, W. Shu, B. Wilson, and J. S. Edwards. Stochastic simulations of ErbB homo and heterodimerisation: Potential impacts of receptor conformational state and spatial segregation. *IET Systems Biology*, 2(5):256–272, September 2008.

[22] Michelle N. Costa, Krishnan Radhakrishnan, Bridget S. Wilson, Dionisios G. Vlachos, and Jeremy S. Edwards. Coupled stochastic spatial and non-spatial simulations of ErbB1 signaling pathways demonstrate the importance of spatial organization in signal transduction. *PLoS ONE*, 4(7):e6316, July 2009.

[23] Tatiana T. Marquez-Lago and Kevin Burrage. Binomial tau-leap spatial stochastic simulation algorithm for applications in chemical kinetics. *The Journal of Chemical Physics*, 127(10):104101, September 2007.

[24] Daniel T. Gillespie. Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, 58(1):35–55, May 2007.

[25] Cato Maximilian Guldberg and Peter Waage. Studies concering affinity. *Forhandlinger: Videnskabs-Selskabet i Christiana*, 35, 1864.

[26] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, June 1953.

[27] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

[28] Drew Endy. Foundations for engineering biology. *Nature*, 438(7067):449–453, November 2005.

[29] Matthew R. Bennett and Jeff Hasty. Overpowering the component problem. *Nature Biotechnology*, 27(5):450–451, May 2009.

[30] Kevin Clancy and Christopher A. Voigt. Programming cells: Towards an automated 'genetic compiler'. *Current Opinion in Biotechnology*, 21(4):572–581, August 2010.

[31] Ahmad S. Khalil and James J. Collins. Synthetic biology: Applications come of age. *Nature Reviews Genetics*, 11(5):367–379, May 2010.

[32] Nagarajan Nandagopal and Michael B. Elowitz. Synthetic biology: Integrated gene circuits. *Science*, 333(6047):1244–1248, September 2011.

[33] Priscilla E. M. Purnick and Ron Weiss. The second wave of synthetic biology: From modules to systems. *Nature Reviews Molecular Cell Biology*, 10(6):410–422, June 2009.

[34] Xiaojun Feng, Xin Liu, Qingming Luo, and Bi-Feng Liu. Mass spectrometry in systems biology: An overview. *Mass Spectrometry Reviews*, 27(6):635–660, 2008.

[35] Sebastian J. Maerkl. Integration column: Microfluidic high-throughput screening. *Integrative Biology*, 1(1):19–29, January 2009.

[36] Ryan N. Gutenkunst, Joshua J. Waterfall, Fergal P. Casey, Kevin S. Brown, Christopher R. Myers, and James P. Sethna. Universally sloppy parameter sensitivities in systems biology models. *PLoS Computational Biology*, 3(10):e189, October 2007.

[37] Joshua F. Apgar, David K. Witmer, Forest M. White, and Bruce Tidor. Sloppy models, parameter uncertainty, and the role of experimental design. *Molecular BioSystems*, 6(10):1890–1900, October 2010.

[38] Ricky Chachra, Mark K. Transtrum, and James P. Sethna. Comment on Sloppy models, parameter uncertainty, and the role of experimental design. *Molecular BioSystems*, 7(8):2522–2522, August 2011.

[39] David R. Hagen, Joshua F. Apgar, David K. Witmer, Forest M. White, and Bruce Tidor. Reply to comment on Sloppy models, parameter uncertainty, and the role of experimental design. *Molecular BioSystems*, 7(8):2523–2524, August 2011.

[40] K. S. Brown, C. C. Hill, G. A. Calero, C. R. Myers, K. H. Lee, J. P. Sethna, and R. A. Cerione. The statistical mechanics of complex signaling networks: Nerve growth factor signaling. *Physical Biology*, 1(3):184–195, October 2004.

[41] Roi Avraham and Yosef Yarden. Feedback regulation of EGFR signalling: Decision making by early and delayed loops. *Nature Reviews Molecular Cell Biology*, 12(2):104–117, February 2011.

[42] H-W Lo and M-C Hung. Nuclear EGFR signalling network in cancers: Linking EGFR pathway to cell cycle progression, nitric oxide pathway and patient survival. *British Journal of Cancer*, 94(2):184–188, January 2006.

[43] Anna Bauer-Mehren, Laura I Furlong, and Ferran Sanz. Pathway databases and tools for their exploitation: Benefits, current limitations and challenges. *Molecular Systems Biology*, 5(1):290, July 2009.

[44] Tianhui Hu and Cunxi Li. Convergence between wnt-β-catenin and EGFR signaling in cancer. *Molecular Cancer*, 9(1):236, September 2010.

[45] Sandra Morandell, Taras Stasyk, Sergej Skvortsov, Stefan Ascher, and Lukas A. Huber. Quantitative proteomics and phosphoproteomics reveal novel insights into complexity and dynamics of the EGFR signaling network. *PROTEOMICS*, 8(21):4383–4401, November 2008.

[46] H. Steven Wiley, Stanislav Y. Shvartsman, and Douglas A. Lauffenburger. Computational modeling of the EGF-receptor system: a paradigm for systems biology. *Trends in Cell Biology*, 13(1):43–50, January 2003.

[47] R. J. Orton, O. E. Sturm, A. Gormand, W. Kolch, and D. R. Gilbert. Computational modelling reveals feedback redundancy within the epidermal growth factor receptor/extracellular-signal regulated kinase signalling pathway. *IET Systems Biology*, 2(4):173–183, July 2008.

[48] Richard Orton, Michiel Adriaens, Amelie Gormand, Oliver Sturm, Walter Kolch, and David Gilbert. Computational modelling of cancerous mutations

in the EGFR/ERK signalling pathway. *BMC Systems Biology*, 3(1):100, October 2009.

[49] Jorrit J. Hornberg, Bernd Binder, Frank J. Bruggeman, Birgit Schoeberl, Reinhart Heinrich, and Hans V. Westerhoff. Control of MAPK signalling: from complexity to what really matters. *Oncogene*, 24(36):5533–5542, June 2005.

[50] Satoru Sasagawa, Yu-ichi Ozaki, Kazuhiro Fujita, and Shinya Kuroda. Prediction and validation of the distinct dynamics of transient and sustained ERK activation. *Nature Cell Biology*, 7(4):365–373, April 2005.

[51] Frances A Brightman and David A Fell. Differential feedback regulation of the MAPK cascade underlies the quantitative differences in EGF and NGF signalling in PC12 cells. *FEBS Letters*, 482(3):169–174, October 2000.

[52] Dongru Qiu, Likai Mao, Shinichi Kikuchi, and Masaru Tomita. Sustained MAPK activation is dependent on continual NGF receptor regeneration. *Development, Growth & Differentiation*, 46(5):393–403, October 2004.

[53] Satoshi Yamada, Takaharu Taketomi, and Akihiko Yoshimura. Model analysis of difference between EGF pathway and FGF pathway. *Biochemical and Biophysical Research Communications*, 314(4):1113–1120, February 2004.

[54] Stephen Chapman and Anand R. Asthagiri. Resistance to signal activation governs design features of the MAP kinase signaling module. *Biotechnology and Bioengineering*, 85(3):311–322, February 2004.

[55] Birgit Schoeberl, Claudia Eichler-Jonsson, Ernst Dieter Gilles, and Gertraud Muller. Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. *Nature Biotechnology*, 20(4):370–375, April 2002.

[56] Melody K. Morris, Julio Saez-Rodriguez, Peter K. Sorger, and Douglas A. Lauffenburger. Logic-based models for the analysis of cell signaling networks. *Biochemistry*, 49(15):3216–3224, April 2010.

[57] William S. Hlavacek. How to deal with large models? *Molecular Systems Biology*, 5(1):240, January 2009.

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

[59] Vincent Danos, Jerome Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-based modelling of cellular signalling. In *CONCUR 2007 – Concurrency Theory*, pages 17–41. 2007.

[60] Chen Li, Marco Donizelli, Nicolas Rodriguez, Harish Dharuri, Lukas Endler, Vijayalakshmi Chelliah, Lu Li, Enuo He, Arnaud Henry, Melanie I Stefan, Jacky L Snoep, Michael Hucka, Nicolas Le Novre, and Camille Laibe. BioModels database: An enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Systems Biology*, 4:92, June 2010.

[61] Eric Walter and Luc Pronzato. *Identification of Parametric Models: From Experimental Data*. Springer, first edition edition, January 1997.

[62] I. T. Jolliffe. Mathematical and statistical properties of population principal components. In *Principal Component Analysis*. John Wiley & Sons, Ltd, 2002.

[63] N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry*. Elsevier, November 1992.

[64] Carlos A. Gomez-Uribe and George C. Verghese. Mass fluctuation kinetics: Capturing stochastic effects in systems of chemical reactions through coupled mean-variance computations. *Journal of Chemical Physics*, 126(2):024109–024109–12, January 2007.

[65] Micha Komorowski, Maria J. Costa, David A. Rand, and Michael P. H. Stumpf. Sensitivity, robustness, and identifiability in stochastic chemical kinetics models. *Proceedings of the National Academy of Sciences*, 108(21):8645–8650, May 2011.

[66] Xun Huan and Youssef M. Marzouk. Gradient-based stochastic optimization methods in bayesian experimental design. *arXiv:1212.2228*, December 2012.

[67] Juliane Liepe, Sarah Filippi, Micha Komorowski, and Michael P. H. Stumpf. Maximizing the information content of experiments in systems biology. *PLoS Comput Biol*, 9(1):e1002888, January 2013.

[68] Zoltn Kutalik, Kwang-Hyun Cho, and Olaf Wolkenhauer. Optimal sampling time selection for parameter estimation in dynamic pathway modeling. *Biosystems*, 75(1-3):43–55, July 2004.

[69] R. J. Flassig and K. Sundmacher. Optimal design of stimulus experiments for robust discrimination of biochemical reaction networks. *Bioinformatics*, 28(23):3089–3096, December 2012.

[70] S.P. Asprey and S. Macchietto. Designing robust optimal dynamic experiments. *Journal of Process Control*, 12(4):545–556, June 2002.

[71] D. Faller, U. Klingmller, and J. Timmer. Simulation methods for optimal experimental design in systems biology. *SIMULATION*, 79(12):717–725, December 2003.

[72] F.P. Casey, D. Baird, Q. Feng, R.N. Gutenkunst, J.J. Waterfall, C.R. Myers, K.S. Brown, R.A. Cerione, and J.P. Sethna. Optimal experimental design in an epidermal growth factor receptor signalling and down-regulation model. *IET Systems Biology*, 1(3):190–202, May 2007.

[73] Kun-Liang Guan, Claudia Figueroa, Teresa R. Brtva, Tianquan Zhu, Jennifer Taylor, Theodore D. Barber, and Anne B. Vojtek. Negative regulation of the serine/threonine kinase B-Raf by Akt. *Journal of Biological Chemistry*, 275(35):27354–27359, September 2000.

[74] Pedro Mendes, Wei Sha, and Keying Ye. Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, 19(suppl 2):ii122–ii129, September 2003.

[75] Jason A. Papin, Tony Hunter, Bernhard O. Palsson, and Shankar Subramaniam. Reconstruction of cellular signalling networks and analysis of their properties. *Nature Reviews Molecular Cell Biology*, 6(2):99–111, February 2005.

[76] Jrg Stelling. Mathematical models in microbial systems biology. *Current Opinion in Microbiology*, 7(5):513–518, October 2004.

[77] Ariel Bensimon, Albert J.R. Heck, and Ruedi Aebersold. Mass spectrometrybased proteomics and network biology. *Annual Review of Biochemistry*, 81(1):379–405, July 2012.

[78] Neal S. Holter, Madhusmita Mitra, Amos Maritan, Marek Cieplak, Jayanth R. Banavar, and Nina V. Fedoroff. Fundamental patterns underlying gene expression profiles: Simplicity from complexity. *Proceedings of the National Academy of Sciences*, 97(15):8409–8414, July 2000.

[79] Neal S. Holter, Amos Maritan, Marek Cieplak, Nina V. Fedoroff, and Jayanth R. Banavar. Dynamic modeling of gene expression data. *Proceedings of the National Academy of Sciences*, 98(4):1693–1698, February 2001.

[80] Wolfram Liebermeister. Linear modes of gene expression determined by independent component analysis. *Bioinformatics*, 18(1):51–60, January 2002.

[81] James C. Liao, Riccardo Boscolo, Young-Lyeol Yang, Linh My Tran, Chiara Sabatti, and Vwani P. Roychowdhury. Network component analysis: Reconstruction of regulatory signals in biological systems. *Proceedings of the National Academy of Sciences*, 100(26):15522–15527, December 2003.

[82] Erwin P. Gianchandani, Jason A. Papin, Nathan D. Price, Andrew R. Joyce, and Bernhard O. Palsson. Matrix formalism to describe functional states of transcriptional regulatory systems. *PLoS Compututational Biology*, 2(8):e101, August 2006.

[83] Iman Famili and Bernhard O. Palsson. Systemic metabolic reactions are obtained by singular value decomposition of genome-scale stoichiometric matrices. *Journal of Theoretical Biology*, 224(1):87–96, September 2003.

[84] Karen Sachs, Solomon Itani, Jennifer Carlisle, Garry P. Nolan, Dana Pe'er, and Douglas A. Lauffenburger. Learning signaling network structures with sparsely distributed data. *Journal of Computational Biology*, 16(2):201–212, February 2009.

[85] Jesper Tegnr, M. K. Stephen Yeung, Jeff Hasty, and James J. Collins. Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *Proceedings of the National Academy of Sciences*, 100(10):5944–5949, May 2003.

[86] A. Julius, M. Zavlanos, S. Boyd, and G.J. Pappas. Genetic network identification using convex programming. *IET Systems Biology*, 3(3):155–166, May 2009.

[87] Kumar Selvarajoo and Masa Tsuchiya. Systematic determination of biological network topology: Nonintegral connectivity method (NICM). In Sangdun Choi, editor, *Introduction to Systems Biology*, pages 449–471. Humana Press, January 2007.

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

[89] M. K. Stephen Yeung, Jesper Tegnr, and James J. Collins. Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences*, 99(9):6163–6168, April 2002.

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

[91] Robert E. McCulloch and Peter E. Rossi. Bayes factors for nonlinear hypotheses and likelihood distributions. *Biometrika*, 79(4):663–676, December 1992.

[92] Michael A. Newton and Adrian E. Raftery. Approximate bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(1):3–48, January 1994.

[93] Radford Neal. The harmonic mean of the likelihood: Worst monte carlo method ever, August 2008.

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

[95] Xiao-Li Meng and Wing Hung Wong. Simulating ratios of normalization constants via a simple identity: A theoretical exploration. *Statistica Sinica*, 6:831–860, 1996.

[96] Nicolas Lartillot and Herv Philippe. Computing bayes factors using thermodynamic integration. *Systematic Biology*, 55(2):195–207, April 2006.

[97] N. Friel and A. N. Pettitt. Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3):589–607, July 2008.

[98] Andrew Gelman and Xiao-Li Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 13(2):163–185, May 1998.

[99] Radford Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, April 2001.

[100] Peter J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, December 1995.

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

[102] Tian-Rui Xu, Vladislav Vyshemirsky, Amelie Gormand, Alex von Kriegsheim, Mark Girolami, George S. Baillie, Dominic Ketley, Allan J. Dunlop, Graeme Milligan, Miles D. Houslay, and Walter Kolch. Inferring signaling pathway topologies from multiple perturbation measurements of specific biochemical species. *Science Signaling*, 3(113):ra20, March 2010.

[103] David Posada and Thomas R. Buckley. Model selection and model averaging in phylogenetics: Advantages of akaike information criterion and bayesian approaches over likelihood ratio tests. *Systematic Biology*, 53(5):793–808, October 2004.

[104] Adrian Raftery. Choosing models for cross-classifications. *American Sociological Review*, 51(1):145–146, February 1986.

[105] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, December 1974.

[106] Gideon Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, March 1978.

[107] Kenneth P. Burnham and David R. Anderson. *Model selection and multi-model inference: a practical information-theoretic approach.* Springer, July 2002.

[108] David R. Hagen, Jacob K. White, and Bruce Tidor. Convergence in parameters and predictions using computational experimental design. *Interface Focus*, 3(4), August 2013.

[109] Jouni Kuha. AIC and BIC: comparisons of assumptions and performance. *Sociological Methods & Research*, 33(2):188–229, November 2004.

[110] David L. Weakliem. A critique of the bayesian information criterion for model selection. *Sociological Methods & Research*, 27(3):359–397, February 1999.

[111] James E. Ferrell and Ramesh R. Bhatt. Mechanistic studies of the dual phosphorylation of mitogen-activated protein kinase. *Journal of Biological Chemistry*, 272(30):19008–19016, July 1997.

[112] Caroline Evans, Josselin Noirel, Saw Yen Ow, Malinda Salim, Ana G. Pereira-Medrano, Narciso Couto, Jagroop Pandhal, Duncan Smith, Trong Khoa Pham, Esther Karunakaran, Xin Zou, Catherine A. Biggs, and Phillip C. Wright. An insight into iTRAQ: where do we stand now? *Analytical and Bioanalytical Chemistry*, 404(4):1011–1027, September 2012.

[113] Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling for various metropolis-hastings algorithms. *Statistical Science*, 16(4):351–367, November 2001.

[114] Clemens Kreutz and Jens Timmer. Systems biology: Esxperimental design. *FEBS Journal*, 276(4):923–942, February 2009.

[115] Robert E. Kass, Luke Tierney, and Joseph B. Kadane. Laplace's method in bayesian analysis. *Contemporary Mathematics*, 115:89–99, 1991.

[116] Adrian E. Raftery. Approximate bayes factors and accounting for model uncertainty in generalised linear models. *Biometrika*, 83(2):251–266, June 1996.

[117] David R. Hagen and Bruce Tidor. Efficient bayesian estimates for discrimination among topologically different systems biology models. 2014.

[118] William G. Hunter and Albey M. Reiner. Designs for discriminating between two rival models. *Technometrics*, 7(3):307–323, 1965.

[119] D. Espie and S. Macchietto. The optimal design of dynamic experiments. *AIChE Journal*, 35(2):223–229, 1989.

[120] M. J. Cooney and K. A. McDonald. Optimal dynamic experiments for bioreactor model discrimination. *Applied Microbiology and Biotechnology*, 43(5):826–837, October 1995.

[121] Joshua F. Apgar, Jared E. Toettcher, Drew Endy, Forest M. White, and Bruce Tidor. Stimulus design for model selection and validation in cell signaling. *PLoS Computational Biology*, 4(2):e30, February 2008.

[122] Dominik Skanda and Dirk Lebiedz. An optimal experimental design approach to model discrimination in dynamic biochemical systems. *Bioinformatics*, 26(7):939–945, April 2010.

[123] Bing H. Chen and Steven P. Asprey. On the design of optimally informative dynamic experiments for model discrimination in multiresponse nonlinear situations. *Industrial & Engineering Chemistry Research*, 42(7):1379–1390, April 2003.

[124] Guido Buzzi-Ferraris and Pio Forzatti. A new sequential experimental design procedure for discriminating among rival models. *Chemical Engineering Science*, 38(2):225–232, 1983.

[125] Alberto Giovanni Busetto, Alain Hauser, Gabriel Krummenacher, Mikael Sunnker, Sotiris Dimopoulos, Cheng Soon Ong, Jrg Stelling, and Joachim M. Buhmann. Near-optimal experimental design for model selection in systems biology. *Bioinformatics*, page btt436, July 2013.

[126] Fumin Shi, Shannon E. Telesco, Yingting Liu, Ravi Radhakrishnan, and Mark A. Lemmon. ErbB3/HER3 intracellular domain is competent to bind ATP and catalyze autophosphorylation. *Proceedings of the National Academy of Sciences*, 107(17):7692–7697, April 2010.

[127] Mara P. Steinkamp, Shalini T. Low-Nam, Shujie Yang, Keith A. Lidke, Diane S. Lidke, and Bridget S. Wilson. erbB3 is an active tyrosine kinase capable of homo- and heterointeractions. *Molecular and Cellular Biology*, 34(6):965–977, March 2014.

[128] Russ Harmer, Vincent Danos, Jrme Feret, Jean Krivine, and Walter Fontana. Intrinsic information carriers in combinatorial dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(3):037108, September 2010.

[129] James R. Faeder. Toward a comprehensive language for biological systems. *BMC Biology*, 9(1):68, October 2011.

[130] Lily A. Chylek, Edward C. Stites, Richard G. Posner, and William S. Hlavacek. Innovations of the rule-based modeling approach. In Ale Prokop and Bla Csuks, editors, *Systems Biology*, pages 273–300. Springer Netherlands, January 2013.

[131] Michael L. Blinov, James R. Faeder, Byron Goldstein, and William S. Hlavacek. BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, 20(17):3289–3291, November 2004.

[132] Aneil Mallavarapu, Matthew Thomson, Benjamin Ullian, and Jeremy Gunawardena. Programming with models: Modularity and abstraction provide powerful capabilities for systems biology. *Journal of the Royal Society Interface*, 6(32):257–270, March 2009.

[133] Laurence Calzone, Franois Fages, and Sylvain Soliman. BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807, July 2006.

[134] Nikolay M. Borisov, Nick I. Markevich, Jan B. Hoek, and Boris N. Kholodenko. Signaling through receptors and scaffolds: Independent interactions reduce combinatorial complexity. *Biophysical Journal*, 89(2):951–966, August 2005.

[135] Nikolay M Borisov, Nick I Markevich, Jan B Hoek, and Boris N Kholodenko. Trading the micro-world of combinatorial complexity for the macro-world of protein interaction domains. *Biosystems*, 83(2-3):152–166, March 2006.

[136] Holger Conzelmann, Julio Saez-Rodriguez, Thomas Sauter, Boris N Kholodenko, and Ernst D Gilles. A domain-oriented approach to the reduction of combinatorial complexity in signal transduction networks. *BMC Bioinformatics*, 7(1):34, January 2006.

[137] Markus Koschorreck, Holger Conzelmann, Sybille Ebert, Michael Ederer, and Ernst Dieter Gilles. Reduced modeling of signal transduction - a modular approach. *BMC Bioinformatics*, 8:336, September 2007.

[138] Holger Conzelmann, Dirk Fey, and Ernst D Gilles. Exact model reduction of combinatorial reaction networks. *BMC Systems Biology*, 2(1):78, August 2008.

[139] N. M. Borisov, A. S. Chistopolsky, J. R. Faeder, and B. N. Kholodenko. Domain-oriented reduction of rule-based network models. *IET Systems Biology*, 2(5):342–351, September 2008.

[140] Markus Koschorreck and Ernst D. Gilles. ALC: automated reduction of rule-based models. *BMC Systems Biology*, 2(1):91, October 2008.

[141] Jrme Feret, Vincent Danos, Jean Krivine, Russ Harmer, and Walter Fontana. Internal coarse-graining of molecular systems. *Proceedings of the National Academy of Sciences*, 106(16):6453–6458, April 2009.

[142] V. Danos, J. Feret, W. Fontana, R. Harmer, and J. Krivine. Abstracting the differential semantics of rule-based models: Exact and automated model reduction. In *2010 25th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 362–381. IEEE, July 2010.

[143] Ferdinanda Camporesi and Jerome Feret. Formal reduction for rule-based models. *Electronic Notes in Theoretical Computer Science*, 276(0):29–59, September 2011.

[144] H. D. Soule, T. M. Maloney, S. R. Wolman, Jr Peterson, W. D., R. Brenz, C. M. McGrath, J. Russo, R. J. Pauley, R. F. Jones, and S. C. Brooks. Isolation and characterization of a spontaneously immortalized human breast epithelial cell line, MCF-10. *Cancer research*, 50(18):6075–6086, September 1990.

[145] Kaare Brandt Petersen and Michael Syskind Pedersen. *The Matrix Cookbook*. November 2008.