Beyond Co-Expression: Gene Network Inference[1]

Patrik D'haeseleer
Harvard University
patrik@genetics.med.harvard.edu

March 3, 2001

[1] This document contains extracts from the author's dissertation work "*Reconstructing Gene Networks from Large Scale Gene Expression Data*". The full version can be downloaded at http://www.cs.unm.edu/~patrik/networks/diss.pdf or diss.ps.

Contents

1 Introduction

Everything's connected, all along the line.
Cause and effect. That's the beauty of it.
Our job is to trace the connections and reveal them.
— Terry Gilliam ("Brazil")

1.1 Functional Genomics

All science is either physics or stamp collecting.
— Ernest Rutherford, physicist

Genes code for proteins, some of which in turn regulate other genes. This network of gene regulation, combined with protein interactions, can be very complex. The traditional approach to research in Molecular Biology has been an inherently local one, examining and collecting data on a single gene, a single protein or a single reaction at a time. This is, of course, the classical reductionist stance: To understand the whole, one must first understand the parts. Over the years, this approach has led to some remarkable achievements, allowing us to make highly accurate biochemical models of such favorites as bacteriophage Lambda [71, 7].

However, with the advent of the "Age of Genomics" an entirely new class of data is emerging. As the goal of *structural* genomics—sequencing entire genomes—comes into sight, the focus is gradually shifting to *functional* genomics.

> Specifically, functional genomics refers to the development and application of global (genome-wide or system-wide) experimental approaches to assess gene function by making use of the information and reagents provided by structural genomics. It is characterized by high throughput or large scale experimental methodologies combined with statistical and computational analysis of the results. [42]

Biology used to be a *data-poor* science, out of necessity having to rely on carefully designed hypothesis and meticulously planned experiments. Over the past couple of years, however, it has been rapidly evolving into a *data-rich* field, opening up the possibility of data-driven research—for which Hood coined the term "discovery science" [1]—rather than hypothesis-driven research. Such analysis-without-hypothesis has often been compared pejoratively to a fishing expedition. But perhaps, as Geshwind [30] states, it is "fishing, but with a stick of dynamite in a stocked pond". Obviously, there is a trade-off to be made between *unbiased* analysis—allowing for the possibility of entirely innovative conclusions—and *uninformed* analysis—ignoring all the accumulated wisdom of the field.

Unfortunately, the arrival of this flood of large scale data has so far not been accompanied by an equal abundance of computational techniques to handle the

data. Researchers who were used to looking at perhaps a few tens of measurements from very focused experiments are suddenly faced with literally tens of thousands or even millions of measurements. Initially, analysis of this data was mainly of a descriptive nature, consisting of little more than lists of how many genes were previously unknown, which genes are under or overexpressed under certain circumstances, etc. More recently, simple statistical techniques such as clustering and classification are being discovered, and—occasionally—reinvented. The goal of this dissertation is to develop computational tools to analyze this data at a higher level of complexity, by attempting to determine the underlying network of regulatory interactions that causes the behavior observed in these large scale measurements.

Of course, this large-scale data is an equally valuable resource for researchers who are focusing on individual genes. But can we really expect to construct a detailed biochemical model of, say, an entire yeast cell with some 6000 genes (only about 1000 of which were defined before sequencing began, and about 50% of which are clearly related to other known genes), by analyzing each gene and determining all the binding and reaction constants one by one?

Rather than waiting until we have worked out all the biochemical details, we would like to be able to analyze such large systems in a genome-wide fashion at some intermediate level of representation, without having to go all the way down to the exact biochemical reactions. At the very least, such an intermediate-level analysis could help guide the traditional biochemical approach towards those genes most worthy of attention among these thousands of newly discovered genes. Ideally, a sufficiently predictive and explanatory model at an intermediate level might obviate the need for an exact understanding of the system at the biochemical level. For now, we will be satisfied with "cherry-picking" the most salient features of the regulatory networks, without trying to achieve an accurate model of the entire system.

1.2 An intermediate representation

Everything is deeply intertwingled.
— Theodor Holm Nelson

I intend to focus on genetic regulatory networks at the level of single cells. This ignores the extra complexity that comes with cell to cell interactions and spatial differentiation (see Reinitz and Sharp [72] for example), but is still of major importance to cellular biology. A biological system can be considered to be a state machine, where the change in internal state of the system depends on both its current internal state and any external inputs. The goal is to observe the state of a cell and how it changes under different circumstances, and from this to derive a model of how these state changes are generated. The state of a cell consists of all those variables—both internally and externally—which determine its behavior. Included are the concentrations of all the chemical species (DNA, RNA, proteins, metabolites, etc.) involved in the inner working of the cell, concentrations in the environment of the cell, receptors presented on the

membrane, volume, position in the cell cycle, location of structural components within the cell, and so on. A sufficiently informative subset of these will have to be chosen, usually consisting of concentrations of certain key elements within the cell.

It is unlikely we will ever achieve a simultaneous measurement of the full set of important variables within a cell. In the immediate future, it seems likely we will primarily be focusing on mRNA data, plus perhaps protein data. Exogenous inputs or important intermediates which are missing in our set of measurements are impossible (or at least very difficult) to model. It should be emphasized that these models are, therefore, not intended to imply biochemical *mechanism*, but merely a higher-level view of regulation. This distinction is especially important for the small data sets used in Section 4.

The intermediate representation most familiar to molecular and cell biologists is a directed graph, with the nodes representing the key elements—often genes, proteins or metabolites—being modeled, and the arcs representing how these influence the production or destruction of others. To formalize this sort of description, we might want to add weights—positive or negative—to these arcs, and define how the inputs to a node interact. Figure 1 illustrates how a simple network model might be represented. Even though it consists of only six nodes, the dynamical behavior of the network is far from obvious. Nevertheless, the network representation provides a clear and concise summary of the regulatory interactions, and higher-level structures (such as the two pathways from a to e) can easily be extracted.

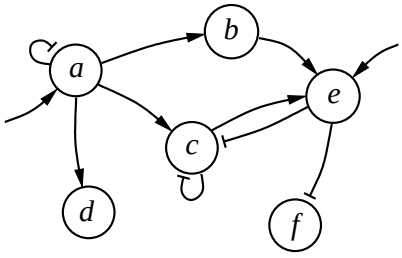

Figure 1: Example of a simple, 6-node regulatory network. For simplicity, no input-output mapping is specified, and interactions have been given a sign (regular arrowheads are positive, flat ones are negative) but not a specific weight. Nodes a and e receive external inputs (e.g. signaling molecules). Nodes a and c are auto-inhibitory, i.e. they will repress their own activation. Notice also the two pathways for upregulation of e by a.

1.3 Additive regulation models: A simple model of gene interaction

One of the simplest ways to model a system of interacting variables is to assume that the change in each variable over time is given by a weighted sum of all other

variables[1]:

$$\Delta y_i = \sum_j w_{ji} y_j + b_i \tag{1}$$

where y_i is the level of the ith variable, b_i is a bias term indicating whether i is expressed or not in the absence of regulatory inputs, and weight w_{ji} represents the influence of j on the regulation of i. We will say that A is a regulator of B if the network model predicts a causal relationship between the level of A and the change in level of B (i.e., an "arrow" in the network), regardless of the underlying mechanism of this regulation. Note that this is a more general interpretation of the terms "regulator" and "regulate" than is normally used in biology.

For a continuous-time system we get the corresponding differential equation:

$$\frac{dy_i}{dt} = \sum_j w_{ji} y_j + b_i \tag{2}$$

Because of the nature of interactions between regulatory factors, gene regulation is often context sensitive, e.g. A upregulates C, but only if B is present as well. The model presented here cannot implement such a nonlinear interaction between A and B in the regulation of C. However, the model should be able to extract the linear component of this regulation, i.e. that both A and B upregulate C, even if the regulation is not independent.

Obviously, an additive model like this will be a gross simplification for almost any natural system, but modeling a gene network with such a minimal model might allow us to extract at least the "Most Significant Bits" of information we're looking for: Which genes regulate which other genes (i.e. which interaction factors w_{ji} are nonzero)? If gene j regulates gene i, is j an inducer or repressor of i (i.e. is w_{ji} positive or negative)?

In Chapter 4, we will examine a purely linear model such as this, apply it to real gene expression data, and compare the results with the literature on the genes involved.

Note that the variables in Equation 2 can theoretically become negative, or unboundedly large. Since these variables typically correspond to concentration levels, we may want to impose realistic upper and lower bounds. Most genes exhibit a sigmoidal dose response curve: As the concentration of the inducing regulatory signals increases, the gene activation at first increases slowly, then more rapidly, and finally saturates at a maximum level. For an added level of realism, we therefore add a sigmoidal transfer function to Equation 2:

$$\frac{dy_i}{dt} = S\left(\sum_j w_{ji} y_j + b_i\right) \tag{3}$$

[1] With an additional noise component $\epsilon(t)$, such a system is generally called a first-order auto-regressive, or AR(1) time series model [39].

where $S(\cdot)$ is some sigmoidal function, e.g. $S(x) = (1+e^{-x})^{-1}$, $S(x) = \tanh(x)$, or a more biologically justified dose-response curve (although it should be noted that some studies indicate that the behavior of the entire network may not be very sensitive to the exact shape of the sigmoid [32]).

Note that the addition of a nonlinear response also allows us to model a large class of interesting nonlinear interactions between regulators. For example in the example above, where both A and B must be present to upregulate C, w_{AC} and w_{BC} individually may be too small to exceed the lower threshold of the sigmoidal $S(\cdot)$, but their combination may be large enough to cause a significant upregulation of gene C.

Because decay of gene products is often an important factor in their regulation, we can also add an extra decay term to each gene as follows:

$$\frac{dy_i}{dt} = S\left(\sum_j w_{ji} y_j + b_i\right) - D_i y_i \tag{4}$$

where D_i is the decay rate for gene i. The resulting model is very close in form to a specific type of recurrent neural networks, and can be fitted to real data in the same manner.

The idea to use a neural network representation to model regulatory networks is not new, dating back at least to Bray's work on cell signaling and parallel distributed processing networks [14]. The reasons for using a neural network model, rather than a more general differential equation model, are twofold. The neural network has a straightforward graphical representation which is close to what researchers are already used to—a very important advantage considering that refinement of these sorts of models usually benefits greatly from collaboration with scientists in the field. Secondly, a large variety of efficient learning algorithms have already been developed for neural networks, whereas determining the parameters in a more general differential equation model would require more general-purpose optimization methods.

Various researchers have used variants of this representation to model genetic regulatory networks. Most notably, Mjolsness, Reinitz and Sharp [66] used a gene regulation model as in Equation 4, interspersed with a simple model of cell division, to model small gene networks involved in pattern formation during the blastoderm stage of development in Drosophila. Weaver *et al.* [94] used a discrete-time version of Equation 3, and showed it is possible to reconstruct randomly created networks of this kind, given enough time series data generated by the network.

Unfortunately, with so many researchers arriving at similar models independently, a variety of different names have been invented for them: connectionist model (Mjolsness *et al.* [66]), linear model (D'haeseleer *et al.* [23]), linear transcription model (Chen *et al.* [17]), weight matrix model (Weaver *et al.* [94]). Considering the core of all these models is the use of a weighted sum to implement gene regulation, I propose we file them under the more general classification of *additive regulation models*[2]. This distinguishes these models from other

[2]In statistics, models consisting of a nonlinear function of a weighted sum of inputs are

representations which may make use of weight matrices, such as Savageau's power law formalism[75]: $dy_i/dt = \alpha_i \prod y_j^{v_{ji}} - \beta_i \prod y_j^{w_{ji}}$, where the two terms account for the production and destruction of the gene product i, v_{ji} and w_{ji} are the kinetic orders, and α_i and β_i the rate constants for these elemental processes.

One implicit assumption of these models is that the concentrations of the chemical species are continuous, i.e. that stochastic fluctuations due to single molecules can be ignored. We know that this does not hold at least for some proteins which are present in concentrations of only a couple of molecules per cell. Indeed, there are indications that stochastic fluctuations may actually be exploited by some organisms [7]. However, differential equations are widely used to model biochemical systems. Hopefully, a continuous approach will prove to be appropriate for the majority of interesting mechanisms.

1.4 Caution to the reader

A number of new technologies are producing a flood of genomic-scale data about the internal state of a cell. Unfortunately, even though these data sets look large to a biologist, they are large "along the wrong dimension", i.e. a large number of variables are measured, but the number of individual measurements of any one variable is still relatively small.

The network models employed here require a substantial number of data points. For example, a common rule of thumb in the neural network community is to use at least a couple times more measurements than weights in the network. This would imply hundreds of data points for the small set of 65 genes used in Chapter 4, or tens of thousands of data points for yeast ($\approx$ 6000 genes). Conventional wisdom would suggest that these sorts of models are underdetermined given the small number of data points currently available.

I intend to show that a shortage of data points does not invalidate the use of these models, as long as we can determine which parts of the model are well determined versus poorly determined. Indeed, much of this dissertation could be viewed as an exercise in distinguishing the few nuggets of well determined interactions from an otherwise poorly determined model. Unfortunately this does mean that it is not yet possible to infer a complete network model as in Figure 1. For now, we will settle for being able to infer those individual connections within the network which are best supported by the data.

Even for those relatively well determined parts of the model, we may not be able to show results with the same level of significance as with some simpler (but less powerful) methods. However, I view this approach not so much as a direct way to find "scientific truth" (however that is defined in one's favorite discipline: $P < 0.05$?), but rather as a way to derive interesting new hypotheses to guide experimentalists in further investigation.

No doubt, as the measurement technologies mature, and larger data sets become publicly available (and calibrated with each other), the usefulness and

also called *Generalized Linear* models [65], whereas models consisting of a weighted sum of nonlinear (nonparametric) functions of inputs are called *Generalized Additive* models [91].

accuracy of the network models developed in this dissertation will increase. The trend towards data sets with large numbers of measurements definitely bodes well in that respect.

2 Modeling issues

By a model is meant a mathematical construct, which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work.
— John von Neumann

Various types of gene regulation network models have been proposed, and the model of choice is often determined by the question one is trying to answer. In this Chapter we will briefly address some of the decisions that need to be made when constructing a network model, the tradeoffs associated with each, and the choices made for the modeling approaches in this dissertation.

2.1 Level of biochemical detail

Gene regulation models can vary from the very abstract—such as Kauffman's random Boolean networks [50]—to the very concrete—like the full biochemical interaction models with stochastic kinetics in Arkin *et al.* [7]. The former approach is the most mathematically tractable, and its simplicity allows examination of very large systems (thousands of genes). The latter fits the biochemical reality better and may carry more weight with the experimental biologists, but its complexity necessarily restricts it to very small systems. For example, the detailed biochemical model of the five-gene lysis-lysogeny switch in Lambda phage [7] included a total of 67 parameters—resulting from almost 50 years of research on Lambda—and required supercomputers for its stochastic simulation (in 1998).

In-depth biochemical modeling is very important for understanding the precise interactions involved in common regulatory mechanisms. However, it is doubtful we could construct such a detailed molecular model of, say, an entire yeast cell with some 6000 genes by analyzing each gene individually and determining all the binding and reaction constants for each molecular interaction one-by-one—at least not in the near future. Likewise, from the perspective of drug target identification for human disease, we cannot realistically hope to characterize all the relevant molecular interactions one-by-one as a requirement for building a predictive disease model. There is a need for methods that can handle large-scale data in a global fashion, and that can analyze these large systems at some intermediate level, without going all the way down to the exact biochemical reactions. For this reason, and because of the limited amount of data available, we will choose a more abstract model, and attempt to infer very general regulatory interactions without specifying the precise mechanism.

2.2 Boolean or continuous

The Boolean (ON/OFF) approximation implicitly assumes highly cooperative binding (very "sharp" activation response curves) and/or positive feedback loops to make the variables saturate in ON or OFF positions. However, if one examines real gene expression data, it seems clear that genes spend a lot of their time at intermediate values: gene expression levels tend to be continuous rather than binary. Furthermore, important concepts in control theory that seem indispensable for gene regulation systems either cannot be implemented with Boolean variables, or lead to a radically different dynamical behavior: amplification, subtraction and addition of signals; smoothly varying an internal parameter to compensate for a continuously varying environmental parameter; smoothly varying the period of a periodic phenomenon like the cell cycle, etc. Feedback control (see e.g. [27]) is one of the most important tools used in control theory to regulate system variables to a desired level, and reduce sensitivity to both external disturbances and variation of system parameters. Negative feedback with a moderate feedback gain has a stabilizing effect on the output of the system. However, negative feedback in Boolean circuits will always cause oscillations, rather than increased stability, because the Boolean transfer function effectively has an infinite slope (saturating at 0 and 1). Moreover, Savageau [76] identified several rules for gene circuitry (bacterial operons) that can only be captured by continuous analysis methods. Positive and negative modes of regulation were respectively linked to high and low demand for expression, and a relationship was established between the coupling of regulator and effector genes and circuit capacity and demand.

Some of these problems can be alleviated by hybrid Boolean systems. In particular, Glass [31, 33] has proposed sets of piecewise linear differential equations, where each gene has a continuous-valued internal state, and a Boolean external state. Researchers at the Free University of Brussels [90, 89] have proposed an asynchronously updated logic with intermediate threshold values. These systems allow easy analysis of certain properties of networks, and have been used for qualitative models of small gene networks, but still do not seem appropriate for quantitative modeling of real, large-scale gene expression data.

Since we are primarily interested in modeling real gene expression data, we will opt for a continuous-valued model.

2.3 Deterministic or stochastic

One implicit assumption in continuous-valued models is that fluctuations in the range of single molecules can be ignored. Differential equations are already widely used to model biochemical systems, and a continuous approach may be sufficient for a large variety of interesting mechanisms. However, molecules present at only a few copies per cell do play an important role in some biological phenomena, such as the lysis-lysogeny switch in Lambda phage [71]. In that case, it may be impossible to model the behavior of the system exactly with a purely deterministic model.

These stochastic effects—which have mainly been observed in prokaryotes—may not play as much of a role in the larger eukaryotic cells. In yeast, most mRNA species seem to occur at close to one mRNA copy per cell [92, 44], down to 0.1 mRNA/cell or less (i.e. the mRNA is only present 10% of the time or less in any one cell). Low copy numbers like these could be due to leaky transcription and not have any regulatory role. Also, if the half-life of the corresponding protein (typically measured in hours or days) is much larger than the half-life of the mRNA (averaging around 20 min in yeast [43]), the protein level may not be affected by stochastic fluctuations in mRNA. Analysis of mRNA and protein decay rates and abundances may allow us to identify those few genes for which stochastic modeling may prove necessary.

Particle-based models can keep track of individual molecule counts, and often include much biochemical detail and/or spatial structure. Of course, keeping track of all this detail is computationally expensive, so they are typically only used for small systems. A related modeling technique is Stochastic Petri Nets (SPN's), which can be considered a subset of Markov processes, and can be used to model molecular interactions [36]. Whereas fitting the parameters of a general particle model to real data can be quite difficult, optimization algorithms exist for SPN's. Hybrid Petri Nets [4, 63] include both discrete and continuous variables, allowing them to model both small-copy number and mass action interactions.

Additional sources of unpredictability can include external noise, or errors on measured data. The Bayesian approach to unpredictability is to construct models that can manipulate probability distributions rather than just single values. Stochastic differential equations could be used for example. Of course, this does add a whole new level of complexity to the models. Alternatively, a deterministic model can sometimes be extended by a simplified analysis of the variance on the expected behavior.

Since the role of stochasticity is unclear for the systems we're interested in (typically eukaryotes), we will choose for the simpler of the two approaches: a deterministic model.

2.4 Spatial or non-spatial

Spatiality can play an important role, both at the level of intercellular interactions, and at the level of cell compartments (e.g. nucleus vs. cytoplasm vs. membrane). Most processes in multicellular organisms, especially during development, involve interactions between different cells types, or even between cells of the same type. Some useful information can probably be extracted using a nonspatial model, but eventually a spatial model may be needed.

Spatiality adds yet another dimension of complexity to the models: spatial development, cell type interactions, reservoirs, diffusion constants, etc. In some cases, the abundance of data—spatial patterns—can more than make up for the extra complexity of the model. For example, Mjolsness *et al.* [66] used a time series of one-dimensional spatial patterns to fit a simple model of eve stripe formation in Drosophila. Models like the ones proposed by Marnellos

and Mjolsness [62] for the role of lateral interactions in early Drosophila neurogenesis provide experimentally testable predictions about potentially important interactions.

Current large-scale gene expression data typically does not include any spatial aspects, so we will use a non-spatial model.

2.5 Forward and inverse modeling

Some of the more detailed modeling methodologies listed above have been used to construct computer models of small, well-described regulatory networks. Of course, this requires an extensive knowledge of the system in question, often resulting from decades of research. In this dissertation, we will not focus on this forward modeling approach, but rather on the inverse modeling, or reverse engineering problem: given a specific set of measurements, what can we deduce about the unknown underlying regulatory network? Reverse engineering typically requires the use of a parametric model, the parameters of which are then fit to the real-world data. If the connection structure of the regulatory network (i.e. which genes have a regulatory effect on each other) is unknown, the parametric model will necessarily have to be very general and simplistic, providing little insight into the actual molecular mechanisms involved. Once the network structure is well known, a more detailed model might be used to estimate individual mechanism-related parameters, such as binding and decay constants.

3 Data requirements for network inference

Number is the ruler of forms and ideas,
and is the cause of gods and demons.
— Pythagoras

In this Chapter, we will start by examining the amount of data needed to be able to reconstruct various different network models—a question with great practical importance, but unfortunately no exact answers. As we increase the number of variables to model, the size of the parameter space increases exponentially. This "Curse of Dimensionality" is examined in Section 3.2. Lastly, the data requirements for network inference imply we may need to combine data sets from different sources and of different types. These issues are explored in the final two Sections.

3.1 Sample complexity

The ambitious goal of network reverse engineering comes at the price of requiring more data points. How many data points are needed to infer a gene network of N genes depends on the complexity of the model used to do the inference. As we will see, constraining the connectivity of the network (number of regulatory

inputs per gene) and the nature of the regulatory interactions can dramatically reduce the amount of data needed.

3.1.1 General network models

We can derive an absolute lower bound on the amount of information—in bits—needed to construct general network models, using Information Theory[3]. Suppose we want to derive a *sparse* network model of N genes, where each gene is only affected by K other genes on average (the "connectivity" of the network). This corresponds to constructing a sparsely connected, directed graph with N nodes and NK edges. There are N^2 possible edges between all N genes, and only NK actual edges, so there are $\binom{N^2}{NK}$ possible models of N genes with K interactions on average. To specify the correct model, we then need

$$\log \binom{N^2}{NK} = \log \frac{N^2!}{(NK)!(N^2 - NK)!} \tag{5}$$

bits of information. We can use Stirling's approximation to the factorial ($n! \approx \sqrt{2\pi n}(n/e)^n$) to derive an approximation for $\log \binom{a}{b}$ (see, e.g., [20]):

$$\log \binom{a}{b} \approx a\log(a) - b\log(b) - (a-b)\log(a-b) \tag{6}$$

for $a, b \gg 1$. Equation 5 then becomes:

$$\begin{aligned}
&\log \binom{N^2}{NK} \\
&\quad\approx\quad N^2 \log\left(N^2\right) - NK\log(NK) - \left(N^2 - NK\right)\log\left(N^2 - NK\right) && (7)\\
&\quad=\quad N\Big(N\log(N) - K\log(K) - (N-K)\log(N-K)\Big) && (8)\\
&\quad\approx\quad NK\log(N/K) && (9)
\end{aligned}$$

bits of information. The last approximation holds for $K \ll N$, such that $\log(N - K) \approx \log(N)$. Since each data point consists of N measurements, we will need at least $\Omega(K\log(N/K))$ data points to fully specify a model of this kind. Note that, by Equations 8 and 6, we would get the same growth rate for a model with *exactly* K inputs per gene: $N\log\binom{N}{K} \approx \log\binom{N^2}{NK}$. A similar derivation for undirected graphs (i.e. inferring regulatory interactions, without specifying the *causal* relationship) leads to a lower bound of $\Omega(K\log(N/2K))$ data points.

If we further want to specify whether the interaction is positive or negative, this only requires one extra bit of information per connection in the network. In general, if we want to specify p_n parameters per gene, with λ_n bits of precision each; and p_k parameters per interaction, with λ_k bits of precision, we get $NK\log(N/K) + \lambda_n p_n N + \lambda_k p_k NK$ bits, or at least $\Omega(K\log(N/K) + \lambda_n p_n +$

[3]first developed by Shannon [79], see Cover and Thomas [20] for a good introduction.

$\lambda_k p_k K$) data points. Note that we have not specified how these regulatory inputs are combined, whether the regulatory function is linear or nonlinear, etc. Each link and each node in the network could correspond to some arbitrary, parametrized nonlinear function.

3.1.2 Boolean, fully connected

In a fully connected Boolean network, the output of each gene is modeled as a general Boolean function of the outputs of all N genes. This means we need to specify the output of each single gene, for each of the 2^N possible different states of the network. In other words, we need to measure all possible 2^N input-output pairs. This is clearly inconceivable for even fairly small numbers of genes.

3.1.3 Boolean, connectivity K

If we reduce the connectivity of the Boolean network to an average of K regulatory inputs per gene, the data requirements decrease significantly. To fully specify a Boolean network with limited connectivity, we need to specify the connection pattern between the N nodes (genes) and the rule table for a function of K inputs at each. An absolute lower bound of $\Omega\left(2^K + K\log(N/K)\right)$ can be derived using information theory.[4] A tighter lower bound can be found by looking at a slightly simpler model, where we assume the pattern of connectivity is given, by calculating how the number of independently chosen data points should scale with K and N. Since this is a simpler model, its data requirements should be a lower bound to the requirements for the model with unknown connections.

Every data point (i.e. every input-output pair, specifying the state of the entire Boolean network at time t and $t+1$), specifies exactly one of 2^K entries in each rule table: Given this particular combination of the K inputs to each gene at time t, the output of the gene is given by its state at time $t+1$. We will estimate the probability P that all N rule tables are fully specified by n data points, and calculate how the number of data points n needs to scale with P, the number of genes N, and connectivity K.

The probability that one of 2^K entries in a specific rule table is *not* specified by a single data point is equal to $1 - 2^{-K}$. For n (independent) data points this becomes $(1 - 2^{-K})^n$. Since every data point has to specify exactly one entry in each rule table, the probabilities for each individual entry in a rule table to be unspecified are not entirely independent (e.g. if $2^K - 1$ entries are unspecified, the remaining entry has to be specified). However, for $P \approx 1$ (i.e. we have enough data to have a good chance at a fully specified model), these probabilities will be extremely close to zero, and we can approximate them as being independent. The probability that all 2^K entries in a single rule table are specified by n data points is then approximately:

[4] As shown in Section 3.1.1, we need $K\log(N/K)$ bits per gene to specify the connection pattern, and 2^K bits per gene to specify the Boolean function.

$$1 - 2^K \left(1 - 2^{-K}\right)^n \tag{10}$$

The probability that all N rule tables are fully specified by n data points then becomes:

$$P \approx \left(1 - 2^K \left(1 - 2^{-K}\right)^n\right)^N \tag{11}$$

Taking base-2 logarithms, we find:

$$\begin{aligned} C_1 &= -\log(P) && (12)\\ &\approx -N \log\left(1 - 2^K \left(1 - 2^{-K}\right)^n\right) && (13) \end{aligned}$$

Further simplifying using $\log_2(1 - z) \approx -z \log_2(e)$ for $z \ll 1$ (keeping in mind that the quantity in Equation 10 is very close to 1), and taking logs again:

$$C_1 \approx N 2^K \left(1 - 2^{-K}\right)^n \log(e) \tag{14}$$

$$\begin{aligned} C_2 &= -\log\left(C_1 / \log(e)\right) && (15)\\ &\approx -\log(N) - K - n \log\left(1 - 2^{-K}\right) && (16)\\ &\approx -\log(N) - K + n 2^{-K} \log(e) && (17) \end{aligned}$$

If $P \approx 1$, C_1 will be a small, and C_2 a moderate positive constant (e.g. for $P = 0.9$ and $P = 0.999$, C_2 is 3.25 and 9.97 respectively). We can now express n, the number of data points needed, in terms of N, K and C_2:

$$n \approx 2^K \left(K + \log(N) + C_2\right) / log(e) \tag{18}$$

which is $\Theta\left(2^K \left(K + log(N)\right)\right)$. This estimate agrees well with preliminary experimental results by Liang *et al.* [56] and Akutsu *et al.* [3].

3.1.4 Boolean, linearly separable, connectivity K

In addition to constraining the number of inputs per gene, we could also constrain the type of Boolean functions used in the network. A natural choice is the set of linearly separable Boolean functions, i.e., those that can be implemented using a weighted sum of the inputs, followed by a threshold function. Linearly separable functions are well-behaved, in the sense that inputs always have either an upregulating or downregulating effect. Non-linearly separable functions can have inputs that are upregulating or downregulating, depending on the state of the other inputs (the classical example of this is the Boolean XOR). Interestingly, the vast majority of genes whose regulation is described in the literature seem to have regulation functions which are linearly separable, when abstracted

down to the Boolean level [40].[5] Combining a reduced connectivity with linearly separable Boolean functions reduces the data requirements to $\Omega(K \log(N/K))$ [41].

3.1.5 Continuous, additive, fully connected

When we look at network models with continuous-valued expression levels, we need to choose a parametrized model of regulation functions. (As opposed to Boolean functions, functions over the reals are not enumerable, so we would need infinite amounts of data to fit a "general" continuous-valued function). As mentioned in Section 1.3, and in analogy with Section 3.1.4, we will focus on *additive regulation models*. For models with continuous expression levels, the data requirements are less clear than for the Boolean models. In the case of linear (D'haeseleer et al., 1999) or quasi-linear[6] additive models [94], fitting the model is equivalent to performing a multiple regression, so at least $N + 1$ data points are needed for a fully connected model of N genes[7].

3.1.6 Continuous, additive, connectivity K

Data requirements for sparse additive regulation models are as yet unknown, but based on the similarity with the equivalent Boolean model, we speculate it to be of the form $\Omega(K \log(N/K))$. A promising avenue of further research in this area may be the results on sample complexity for recurrent neural networks, which have a very similar structure to the models presented here. An analysis based on PAC-learning shows that the number of training instances needed to accurately learn the *dynamical behavior* (as opposed to the network weights) for a fully connected network is lower-bounded by $\Omega(N)$ and upper-bounded by $\mathcal{O}(N^4)$ [54]. However, this is based on a worst-case analysis, and might be reduced to $\mathcal{O}(N)$ for the general case [83]. There are a few neural network techniques (such as Winnow and Weighted Majority [57, 58, 59]) that are known to scale as $\mathcal{O}(K \log(N)$ and perform quite well in the presence of many irrelevant inputs. However, these techniques are specific for classification tasks with feedforward networks, using a multiplicative weight update (one could think of them as doing a binary search on the decision surface). It is unclear how these algorithms could be extended to a recurrent network with continuous outputs.

3.1.7 Clustering

Finally, to allow for comparison with gene clustering methods, we examined data requirements for clustering based on pairwise correlation comparisons. In

[5] Although notable exceptions to this certainly exist: in Drosophila, *hunchback*, one of the key regulatory genes in embryonic development, has a concentration-dependent regulatory effect on *Krüppel* [77].

[6] Also known as *generalized linear*.

[7] Note that this result is not directly comparable to the Boolean case: the fully connected Boolean network uses arbitrary Boolean functions, and the estimate for linearly separable Boolean functions (equivalent to the additive functions used here) assumes $K \ll N$

that case, as the number of genes being compared increases, the number of data points will have to increase proportional to $\log(N)$, in order to maintain a constant, low level of false positives. Claverie [18] arrived at a similar logarithmic scaling for binary data (absent/detected).

For this simple abstraction of clustering, we will say that two genes cluster together if their correlation is significantly greater (with a significance level α) than a certain cutoff value ρ_c. We test whether we can exclude the null hypothesis $\rho < \rho_c$ based on the measured correlation coefficient r over the available data points. Because of the large number of comparisons being made, we need to reduce the significance level for the correlation test with the number of tests each gene is involved in. We can use the *Bonferroni correction*, $\alpha = \alpha'/N$, in order to keep the expected number of false positives for each gene constant.[8] In order to be able to use the same cutoff-value for the measured correlation r_α to decide whether two genes cluster together, the number of data points will have to increase as the significance level for each test grows smaller.

If the real correlation coefficient ρ is close to 1.0, the distribution of the measured correlation coefficient r is very asymmetrical. The following z-transformation, developed by Fisher [26], is approximately normally distributed with mean $z(\rho)$ and variance $1/(n-3)$ (with n the number of data points):

$$z(r) = \frac{1}{2}\ln\left(\frac{1+r}{1-r}\right) \tag{19}$$

We can now devise a single-sided test on $z(r)$ to answer the question: If $z(r) > z(r_\alpha)$, what is the significance level with which we can reject the hypothesis $z(\rho) < z(\rho_c)$ (and thus $\rho < \rho_c$)? At the tail of the normal distribution, the area under the normal curve to the right of $z(r_\alpha)$ can be approximated by:

$$\alpha = \int_{z=z(r_\alpha)}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-z(\rho_c))^2}{2\sigma^2}}\, dz \approx \frac{\sigma}{\sqrt{2\pi}\,(z(r_\alpha)-z(\rho_c))} e^{-\frac{(z(r_\alpha)-z(\rho_c))^2}{2\sigma^2}} \tag{20}$$

Taking natural logs, replacing α with the Bonferroni correction $\alpha = \alpha'/N$, and with $\sigma = 1/\sqrt{n-3}$, we arrive at:

$$\begin{aligned} \ln(\alpha') - \ln(N) \quad \approx \quad & -\frac{1}{2}\ln(n-3) - \ln\left(\sqrt{2\pi}\,(z(r_\alpha)-z(\rho_c))\right) \\ & -(n-3)\,(z(r_\alpha)-z(\rho_c))^2\,/2 \end{aligned} \tag{21}$$

[8] In fact, it is sufficient that the false positives do not grow faster than the *true* correlations. If we assume the number of true correlations per gene increases at least as N^γ with the number of genes (with $0 < \gamma < 1$, i.e. both the number of clusters and number of genes per cluster increases), then $\alpha = \alpha'/N^{1-\gamma}$ suffices. (For example, when the number of true correlations grows linearly with N, i.e. $\gamma = 1$, we can allow the number of false positives to grow linearly as well, so no correction is needed.) When we plug $\alpha = \alpha'/N^{1-\gamma}$ into Equation 20, the resulting growth rate for n is similar to the one for $\alpha = \alpha'/N$.

$$n \approx 3 + \frac{2}{\left(z(r_\alpha) - z(\rho_c)\right)^2} \cdot \left(\ln(N) + \ln(1/\alpha') - \ln(n-3)/2 - \ln\left(\sqrt{2\pi}\left(z(r_\alpha) - z(\rho_c)\right)\right)\right) \quad (22)$$

Although this defines n recursively, as a function of $\ln(n-3)$, the dominant term will be $\ln(N)$. In other words, if we want to use the same cutoff value r_α to decide whether $\rho > \rho_c$, we need to scale the number of data points logarithmically with the number of genes. Strictly speaking, this analysis only holds for correlation tests, but we can expect similar effects to play a role in other clustering algorithms.

3.1.8 Summary

Table 1 provides an overview of some of the models considered, and estimates of the amount of data needed for each. These estimates hold for independently chosen data points, and only indicate asymptotic growth rates, ignoring any constant factors. Note also that these estimates reflect the amount of data needed to be able to reconstruct the *entire* network correctly.[9] As mentioned before, we are content with being able to extract the most significant interactions.

Model	Data needed	
General:	$K\log(N/K) + \lambda_n p_n + \lambda_k p_k K$	
Boolean:		
fully connected	2^N	
connectivity K	$2^K(K + \log(N))$	[21, 56, 3]
linearly separable, connectivity K	$K\log(N/K)$	[41]
Continuous:		
additive, fully connected	$N + 1$	
additive, connectivity K	$K\log(N/K)$ (*)	[21]
Clustering:		
pairwise correlation	$\log(N)$	[21]

Table 1: Sample complexity for various network models. Fully connected: each gene can receive regulatory inputs from all other genes. Connectivity K: at most K regulatory inputs per gene. Additive, linearly separable: regulation can be modeled using a weighted sum. Pairwise correlation: significance level for pairwise comparisons based on correlation must decrease inversely proportional to number of comparisons. (*): conjecture.

For reasonably constrained models, the number of data points needed tends to scale with $\log(N)$ rather than N, and that the data requirements for network inference are at least a factor K larger than for clustering.

[9] For example, the sample complexity for reconstructing a *single* gene in a Boolean, sparsely connected network scales with $2^K K$, rather than $2^K(K + \log(N))$ (using Equation 10 rather than 11 in Section 3.1.3)

In practice, the amount of data may need to be orders of magnitude higher because of non-independence and large measurement errors (see also [86]). Higher accuracy methods such as RT-PCR yield more bits of information per data point than cDNA microarrays or oligonucleotide chips, so fewer data points may be required to achieve the same accuracy in the model. (Conversely, if measurement accuracy is low, more data points may be required.) So far none of the sample complexity estimates on this sort of network models includes accuracy of the data. However, we may be able to use a rough guideline provided by information theory, by looking at the information capacity of a Gaussian channel. It can be shown that the maximum amount of information (in bits) encoded in a Gaussian distributed variable with variance $P = \sigma_P^2$, when measured together with an additional Gaussian noise with variance $N = \sigma_N^2$, is given by:

$$I = \frac{1}{2}\log_2(1 + \frac{P}{N}) \approx \log_2(\sigma_P) - \log_2(\sigma_N) \tag{23}$$

where the approximation is within 5% for $\sigma_N < \sigma_P/3$. In other words, every halving of the measurement error increases the amount of information per measurement by one bit. This may not sound much, but consider that with a 10% measurement noise, the information capacity is only $I \approx 3.3$ bits per measurement. The logarithmic scaling does indicate a decreasing usefulness of improving accuracy much further, especially in view of significant amounts of inherent variability in the systems being measured.

Note that modeling real data with Boolean networks discards a lot of information in the data sets, because the expression levels need to be discretized to one bit per measurement. In the example above, with $I \approx 3.3$ bits per measurement, discretizing to one bit would throw away almost 70% of the information contained in the signal. Continuous models will tend to take better advantage of the available information in the data.

Another important issue in design of gene expression experiments is whether to allocate the—so far—often limited and expensive supply of microarrays or oligonucleotide chips to collecting more replicates, or more individual data points. Again, from an information theoretic point of view, n replicates reduce the noise variance by a factor of n, increasing the information content at most with $\log(n)$. Independent measurements on the other hand increase the information content proportional to n, and are therefore—*theoretically*—preferred. However, if the noise on the measurements is significant, it will generally be much harder to extract this additional information without a very good model of the noise involved. Replicates have often been required for publication for other types of biological experiments (usually at least triplicates, so a standard error can be estimated), and it seems like the consensus may be moving in that direction for expression data as well [34, 55]. As the cost per experiment decreases, this issue will likely resolve itself in favor of doing more measurements altogether, i.e., more experiments and more replicates per experiment.

3.2 The Curse of Dimensionality

Measuring more variables allows for a *more exact* model, but makes the *correct* model exponentially harder to find.

When faced with the task of modeling an unknown process, our intuition tells us to observe as many parameters of the system as possible. This is clearly reflected in the current tendency to measure the expression levels of more and more genes simultaneously, rather than to measure these expression levels as often as possible.

However, in Machine Learning it is well known that the more variables one models, the *harder* the modeling task becomes, because the space of models to be searched increases exponentially with the number of parameters of the model, and therefore with the number of variables. This is often referred to as the Curse of Dimensionality [12].

Does this mean that our intuition about modeling is wrong? Not necessarily. Although we humans do want to be able to look at as many variables of the problem as possible, we rather quickly select those we think are really important to the system, and simply ignore the others. Our reason for wanting to know all the variables is so we wouldn't miss any of the important ones, not so we could include all the non-important ones in our model. In order to achieve an accurate model, we must at least measure those variables which are important to the process being studied. If some intermediate variables are not measured, it may be possible to infer them during the modeling process, but this can be very hard. We should be as inclusive as possible in which variables we measure, and try to eliminate redundant variables after the data is collected. Careful selection of the input variables is crucial to get around the Curse of Dimensionality. Use of a priori information can also help narrow down the range of plausible models. As we saw in Section 3.1, narrowing down the range of plausible models by putting on additional—realistic—constraints can simplify the search for the best model considerably. For example, constraining the genes to be regulated by no more than 5-7 other genes will simplify the number of regulatory interactions we need to consider. Similarly, for Boolean networks, constraining the types of Boolean functions to those that are biologically plausible can significantly reduce the number of Boolean rules that match the data.

Constraining the model by using a priori information about what is biologically known or plausible is probably the most important weapon we have to fight the Curse of Dimensionality. How precisely to include this information into the inference process is the true art of modeling.

3.3 Types of data

To infer the regulation of a single gene, we need to observe the expression of that gene under many different combinations of expression levels of its regulatory inputs. This implies sampling a wide variety of different environmental conditions and perturbations. Therefore, the gene network inference techniques we will cover all have one thing in common: they tend to be data-intensive.

Gene expression time series yield a lot of data, but all the data points tend to be about a single dynamical process in the cell, and will be related to the surrounding time points. Therefore, a 10-point time series can generally be expected to contain less information than a data set of ten independent expression measurements under different environmental conditions, or with mutations in different pathways. The advantage of a time series is that it can provide crucial insights into the dynamics of the process. On the other hand, data sets consisting of individual measurements provide an efficient way to map the attractors of the network. Both types of data, and multiple data sets of each, will be needed to unravel the regulatory interactions of the genes.

3.4 Combining different data types

The need for large amounts of data means that successful network modeling efforts will probably have to use data from different sources, and deal with different data types such as time series and steady-state data, different error levels, incomplete data, etc. Whereas clustering methods can use data from different strains, in different growth media etc., combining data sets for reverse engineering of regulatory networks requires that differences between the experimental conditions be quantified much more precisely. Likewise, data will have to be calibrated properly to allow comparison between data sets. Relative expression ratios have limited usefulness unless they can be calibrated with respect to other data sets post facto (e.g. using expression levels relative to a given standard). In this respect, there is a growing need for a reliable reference in relative expression measurements. An obvious approach could be to agree on a standard strain or tissue pool and carefully controlled growth conditions to use in all data collection efforts on the same organism. Alternatively, a reference mRNA population with fixed relative concentrations of mRNA's could be generated artificially, or perhaps even derived directly from the genomic DNA.

As individual data sets become larger, the amount of analysis that can be done within a single data set increases as well. But unless we can have confidence in comparing results from different experimenters, we potentially miss out on an enormous resource: the combined data of all researchers examining the same organism.

4 A linear model of CNS development and injury

We will start by examining the most simple form of additive regulation models: a purely linear one, where changes in expression levels are linearly correlated with expression levels of other genes. This first-order approximation model is then applied to a set of real-world gene expression time series on development and injury of rat central nervous system. We first examine some of the higher-level properties of the resulting linear model (such as limited connectivity of the network), and find that they are biologically plausible. Next, we develop a

methodology to identify those specific weights in the network which are well-defined by the data. The results of this analysis compare favorably with what can be found in the literature regarding these genes.

4.1 A first-order approximation

Have no fear of perfection – you'll never reach it.
— Salvador Dali

The whole idea of correctness is totally overrated.
— Stephanie Forrest, 10/29/99

As we will show, even the simplest form of the additive regulation model (Equation 1) can give interesting and suggestive results. Of course, a linear model such as this is unlikely to be much more than a caricature of the real system, and should be thought of as a first-order approximation. This is because its purely linear form cannot correctly model nonlinear interactions. However, we do expect it to be able to capture many important linear components of gene regulation. In that sense, it has similar strengths and weaknesses as using linear (Pearson) correlation to analyze any real-world variables. Although it is not an optimally fitting model, the majority of applied statistics is, similarly, based on linear correlations. The value of a coarse model like this is mainly exploratory. It serves to direct further detailed investigation by suggesting novel hypotheses about the system.

Let us first rewrite Equation 1 as a difference equation, explicitly introducing the time step Δt:

$$\frac{\Delta y_i(t)}{\Delta t} = \sum_j w_{ji}\, y_j(t) + b_i \tag{24}$$

where $y_i(t)$ is the expression level of gene i at time t, $\Delta y_i(t) = y_i(t+\Delta t) - y_i(t)$, w_{ji} indicates how much the level of gene j influences gene i, and b_i is a constant bias factor to model the activation level of the gene in the absence of any other regulatory inputs. Each "node" in the regulatory network model performs a simple summation of its inputs, as illustrated in Figure 2.

Note that we could equivalently rewrite this equation as an *update rule*, by multiplying both sides by Δt and adding $y_i(t)$:

$$y_i(t+\Delta t) = \sum_j w'_{ji}\, y_j(t) + b'_i \tag{25}$$

where $w'_{ji} = \Delta t\, w_{ji}$ ($+1$ if $i = j$), and $b'_i = \Delta t\, b_i$. In this more general form of an update rule, there is no implicit assumption that $\mathbf{y}(t)$ should be a smooth—or even continuous—function in time. It is included here mainly to illustrate the similarity with the Boolean network formulation, and some earlier work

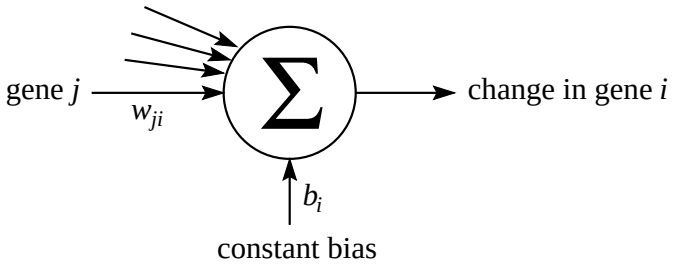

Figure 2: Schematic illustration of a node in the linear network model. The input from all regulatory genes is summed up, together with a constant bias term. The result determines the change (i.e., slope) in expression level of the corresponding gene.

on continuous models using an update rule formulation.[10] Note also that the parameters w'_{ji} and b'_i are dependent on the time step Δt in this formulation.

It is important to keep in mind that it is not the current expression *level* which is regulated in the first place, but rather the transcriptional state of the gene. Whereas the transcriptional state may show an on-off behavior at small time scales, the actual expression level is due to the accumulation of mRNA, essentially related to the integral of the transcriptional state of the gene over time[11]. Since we want to model real gene expression, with expression levels $\mathbf{y}$ that are smooth in time, i.e. $\mathbf{y}(t + \Delta t) \approx \mathbf{y}(t)$, we will instead use the difference equation formulation of Equation 24 If we choose Δt small enough, the parameters w_{ji} and b_i of Equation 24 will approach the parameters of the corresponding differential equation (and therefore be independent of the time step Δt):

$$\frac{dy_i(t)}{dt} = \sum_j w_{ji}\, y_j(t) + b_i \tag{26}$$

In addition to regulation by other genes within the data set, the genes may also be affected by changes in a number of exogenous inputs which we will have to include in the model (e.g. externally added chemicals in a toxicological experiment, depletion of nutrients in the growth medium, changing temperature, etc.):

[10] For example, Weaver *et al.* [94] generated random sparse weight matrices $\mathbf{w}'$ for an update rule similar to Equation 25, and showed that the corresponding network models can be reconstructed given enough data generated by the network. In their experiments, the generated "expression levels" $\mathbf{y}(t)$ often jumped around erratically from time point to time point. Comparing Equations 24 and 25 shows that $\mathbf{w}' = \Delta t\, \mathbf{w} + \mathbf{I}$, so in order to get a smooth time series for small Δt, $\mathbf{w}'$ should be close to the identity matrix. It is the weight matrix $\mathbf{w}$ which corresponds to our intuitive notion of a connection matrix, not $\mathbf{w}'$.

[11] This observation also was the inspiration for Glass's work on modeling gene regulation using a hybrid Boolean model with piecewise linear dynamics [31, 33], where the expression level increases or decreases linearly, depending on whether the gene is ON or OFF.

$$\frac{\Delta y_i(t)}{\Delta t} = \sum_j w_{ji}\, y_j(t) + \sum_k v_{ki}\, x_k(t) + b_i \tag{27}$$

where $x_k(t)$ is the level of exogenous input k at time t, v_{ki} accounts for the effect of this input on the expression level of gene i.

Because of the need for fairly large amounts of data, measured under different conditions, we may need to combine several data sets. In fact, the data I will be using (see Section 4.2 contains measurements on two different tissue types. Differences in gene expression between tissues are caused by regulatory inputs to the genes. Some of these regulatory inputs will be included as variables in our model, others might not. We could account for those extra regulatory variables which are purely tissue-specific (i.e. they vary depending on tissue, but do not vary within a given tissue) by adding an additional "endogenous" input for each. However, under the linear assumption, the total effect of all these tissue-specific inputs can be summarized with a single tissue-specific term T_{li} for each additional tissue l:

$$\frac{\Delta y_i(t)}{\Delta t} = \sum_j w_{ji}\, y_j(t) + \sum_k v_{ki}\, x_k(t) + \sum_l T_{li}\, \tau_l + b_i \tag{28}$$

where τ_l is an indicator variable which is 1 *iff* the particular data we are modeling comes from tissue l (otherwise 0), and T_{li} sums up the tissue-specific differences in regulation by other variables that are not included in the data set. Equivalently, we can think of genes having a different default expression state within each tissue. For a single tissue, this default expression state was modeled using the bias term b_i. Likewise, we can think of $T_{li}\,\tau_l + b_i$ (which is constant, but different for each tissue type) as modeling the default expression state in tissue l.

Given the time series $y_i(t)$, finding these parameters requires solving a least squares system of linear equations, or, equivalently, performing a multiple regression of each gene on all other genes. In Section 4.3 we will show how we can apply a model such as this on real data.

4.2 Data sets

It is a capital mistake to theorize before one has data.
Insensibly one begins to twist the facts to suit theories,
instead of theories to suit facts.
— A. Conan Doyle

Wen *et al.* [96] have published a Gene Expression Matrix of 112 mRNA species measured at nine different stages during the development of rat cervical spinal cord: embryonic days 11-21 (E11, E13, E15, E18, E21), postnatal days 0-14 (P0=E22, P7, P14), and adult (A=P90). More recently, the same team developed a similar data set [80] of 70 mRNA species measured at nine time points during development of rat hippocampus (E15, E18, P0, P3, P7, P10,

P13, P25, A=P60), and at ten more time points (0h=P25, 0.5h, 1.5h, 3h, 6h, 24h, 48h, 10d, 21d, 32d, 49d) following injury of the central nervous system by injection with kainate (kainic acid), a glutamatergic agonist which causes seizures, localized cell death, and severely disrupts the normal gene expression patterns.

The unequal spacing of time points was carefully chosen to coincide with the varying rate of development and response to injury of the rat central nervous system. The genes measured are only a tiny fraction of the total number of genes expressed in these tissues. However, they were selected to be representative of some of the major gene families assumed to play an important role in CNS development, intracellular signaling or transcriptional regulation in general: neurotransmitter synthesizing and metabolizing enzymes, neurotransmitter receptors, various signaling peptides (neurotrophins, heparin binding growth factors, insulin-like growth factors) and their receptors, cell cycle proteins, transcription factors, as well as developmental marker proteins and some expressed sequence tags (EST's).

Each data point in these time series is the result of measurements on three separate animals. This ensures high accuracy, eliminates some of the variability between individuals, and gives us an idea of the variability at each point ("triplicate standard deviation", see Section 4.4.2 for an example of how this additional information can be exploited). When I started working on these data sets, these were the largest publicly available gene expression time series in terms of number of time points, using a high fidelity gene expression assay. As of this writing, they still stand out for their relatively high quality, although they have since been surpassed in terms of number of genes and number of data points.

Considering the large amount of overlap between the mRNA species for the data sets (65 species in common) and the related tissue types (rat cervical spinal cord and hippocampus), it is possible to join them into one larger data set of 65 genes by 28 time points, consisting of 1) cervical spinal cord development, 2) hippocampus development, and 3) hippocampus injury. The regulatory "hardware" of the genes is the same, though different parts of it might be active in different contexts. Combining data from different tissues allows us to get a more complete picture of the regulatory interactions, provided we account for tissue-specific differences in regulation.

As mentioned before, The choice of these data sets should be viewed in a historic perspective (even though they are only a couple of years old!): they were the best that was available at the time. However, it should be pointed out that they are far from optimal for the sort of models we are interested in. In particular, they consist essentially of whole-tissue samples, measuring the average expression levels in the entire cervical spinal cord or entire hippocampus of an individual. These tissue can be further subdivided into different anatomical regions, each of these regions typically consists of several functionally different layers of cells, and each of these layers consist of different cell types. This obviously violates our earlier statement that we want to focus on genetic regulatory networks at the level of single cells, ignoring cell to cell interactions and spatial

differentiation. Yet, as we shall show, even from these coarse-grained, whole-tissue measurements, we are able to derive genetic regulatory interactions which compare well with the existing literature.

Initial analysis of the Gene Expression Matrix presented in Wen *et al.* [96] was based mainly on similarities between temporal gene expression patterns measured using a Euclidean distance metric [81]. Genes were clustered hierarchically, and waves of activation were identified, representing sets of genes that were turned on in a sequential manner during the course of development.

I have previously presented a preliminary statistical analysis of this data set (D'haeseleer *et al.* [22]), in which relationships between individual genes were inferred based on linear correlation, rank correlation and mutual information. Several gene pairs with high linear correlation were identified, as well as a number of genes with high rank correlation but non-significant linear correlation. Although the number of data points per gene was insufficient to derive real results, the use of mutual information (see, e.g. [79, 20]) to derive causal inferences was illustrated. Since then, a few other groups have analyzed this data as well. For example, Wahde and Hertz [93] used the clusters derived in Wen *et al.* [96] to construct a little cluster network.

4.3 Fitting the model

Truth . . . and if mine eyes
Can bear its blaze, and trace its symmetries,
Measure its distance, and its advent wait,
I am no prophet - I but calculate.
— Charles MacKay

The data sets used here cover two tissue types, and include one single exogenous output to the system (kainate). Equation 28 becomes:

$$\frac{\Delta y_i(t)}{\Delta t} = \sum_j w_{ji}\, y_j(t) + K_i\, \kappa(t) + T_i\, \tau + b_i \tag{29}$$

where $\kappa(t)$ is the kainate level at time t, K_i is the influence of kainate on gene i, τ is an indicator variable for tissue type ($\tau = 0$ for spinal cord, $\tau = 1$ for hippocampus), and T_i accounts for all the differences in regulation between tissue types. Figure 3 shows schematically what a "node" in the corresponding linear network model looks like.

Because the original data sets consist of raw ratiometric RT-PCR measurements, we first normalize the expression level of each gene with respect to its maximum level over all three data sets. This gives us a basis to compare the interaction strengths of the genes. Normalization is more commonly done with respect to the average signal, or with respect to the standard deviation of the signal. However, since this data is a coarse and non-uniform sampling of a time-series, these concepts are ambiguous (Should we average over the data set? Over the interpolated time series? Should we weight the time series based

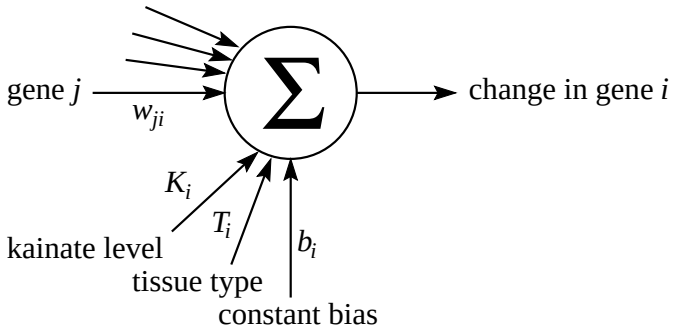

Figure 3: Schematic illustration of a node in the linear network model for the CNS development and injury data. The input from all regulatory genes is summed up, together with an input from the kainate level, a constant bias term, and an additional term to cover tissue-specific differences in regulation. The result determines the change (i.e., slope) in expression level of the corresponding gene.

on developmental speed?). In addition, the maximum expression level of a gene is a useful biochemical concept, related to its production and decay rates.

The linear model in Equation 29 can be fit to a time series finely sampled at equidistant time points Δt. Considering the extremely non-uniform spacing of the measurements (half hour interval after kainate injection, more than two months interval before the final adult cervical spinal cord measurement), we next constructed a finely interpolated time series from the data. Because the modeled variables correspond to concentration levels, we need to avoid negative values in the interpolation. This is achieved by first taking the logarithm of the expression values, applying the interpolation on these log expression levels, and then taking the exponential of the resulting interpolation. We use a piecewise cubic interpolation method, more specifically a multivariate variant of Akima interpolation [2]. This is a local, C^1 (continuous in the first derivative) method, where the interpolation only depends on the nearest data points, and which does not tend to show the spurious excursions between data points common to, for example, cubic spline interpolation (which also imposes C^2 continuity). An interpolation rate of 10 time points per hour gives us 5 interpolated points between the two closest measurements: fine enough to yield a reasonable approximation to the differential equation, while still allowing us to calculate the least squares fit over the entire 7-month data set. We get 24241 interpolated time points for the spinal cord data (101 days), 16081 for the hippocampus development data set (67 days), and 11761 for the hippocampus kainate injury data set (49 days), for a total of 52083 interpolated time points.

The kainate concentration $\kappa(t)$ is zero during the spinal cord and hippocampus time series, jumps from zero to one at 0h for the kainate time series, and then exponentially decays back to zero: $\kappa(t) = e^{-(t-0h)/D_{KA}}$. Kainate tends to disappear from the brain after several hours [95]. We chose an estimated decay

constant of $D_{KA} = 100$ min, corresponding to a half-life of 69.3 min.[12]

Note that the (nonlinear) interpolation has a crucial side-effect: it introduces an implicit additional smoothness constraint on the time series between the measured data points. This smoothness constraint is justified by the effort that went into determining at what time points measurements should be taken. If the measurement rate is fast enough to keep up with the fastest developmental or perturbational changes in the system, we can assume the trajectory of the system between data points to be smooth.

Normally, trying to fit a model with 68×65 parameters (including the additional terms in Equation 29) using only 28×65 data points would lead to a highly underdetermined system. In other words, we would be able to find infinitely many models—with different sets of parameters—that all fit the data perfectly. However, the additional smoothness constraint on the data, allows us to exclude all those models that behave very erratic in between the measured data points. In addition, it also assures that the system has a single optimum, so the fitting becomes (barely) feasible. We do expect there to be many dimensions in which the optimum is poorly determined, corresponding to parts of the model for which not enough data is available. Section 4.4.2 will illustrate how one can identify which parts of the model are well or underdetermined.

The actual fitting of the model to the data requires a small amount of linear algebra, which is summarized in Appendix 5.3. The end result is a matrix $\mathbf{W}^+$, containing the least squares fit of the parameters w_{ji}, K_i, T_i and b_i in Equation 29. The computational complexity of finding a least-squares solution for a linear model is $\mathcal{O}(TN^2)$, where T is the total number of time points in the interpolation, and N is the number of genes. Not surprisingly, the shortage of original data points relative to the number of dimensions of the problem results in a poorly conditioned system, with condition number $6.1 \cdot 10^4$. This condition number gives an upper bound for how much the relative error in $\mathbf{Y}$ (the interpolated gene expression time series) could be magnified in the least squares solution, $\mathbf{W}^+$.[13] In other words, if we are given $\mathbf{Y}$ plus some small error term $\delta\mathbf{Y}$, the resulting weight matrix will be $\mathbf{W}^+$ plus some error $\delta\mathbf{W}^+$. For a poorly conditioned system, the *relative* error $\|\delta\mathbf{W}^+\|/\|\mathbf{W}^+\|$ may be much larger than the relative error in the input, $\|\delta\mathbf{Y}\|/\|\mathbf{Y}\|$, and the magnification of this error is upper-bounded by the condition number of the system (in this case, the condition number of the augmented input matrix, $\widetilde{\mathbf{Y}}$, see Appendix 5.3):

$$\frac{\|\delta\mathbf{W}^+\|}{\|\mathbf{W}^+\|} \leq cond(\widetilde{\mathbf{Y}}) \frac{\|\delta\mathbf{Y}\|}{\|\mathbf{Y}\|} \tag{30}$$

However, this is a worst-case scenario and assumes, among other things, that the error in $\mathbf{Y}$ can vary independently for each interpolated time point. In reality, the nonlinear interpolation spreads out any errors in the original data

[12] As we will see in Section 4.4.2, knowing the exact *in vivo* decay rate D_{KA} for kainate is not crucial, as randomly varying D_{KA} within a fairly large range has little effect on the results.

[13] This is just yet another way of saying the model is poorly determined: a large range of parameter sets $\mathbf{W}$ all show a good fit with the input data $\mathbf{Y}$

sets over a range of interpolated time points, improving the conditioning of the system with respect to the original data sets. In Section 4.4.2, we show that the noise in the input data gets multiplied by a factor of "only" 29.7 in $\mathbf{W}^+$: not as bad as $6.1 \cdot 10^4$, but still poorly conditioned. In fact, the main goal of Section 4.4.2 is to determine which parts of $\mathbf{W}$ are the least affected by the poor conditioning of the system (and the noise in the input data).

Likewise, the condition number of $\mathbf{W}$ is $6.3 \cdot 10^4$, indicating that small amounts of noise in $\mathbf{Y}(t)$ could result in large changes in the slope $d\mathbf{Y}(t)/dt$. However, it turns out that the dynamical behavior of the system is surprisingly robust. If we initialize the system with the gene expression levels measured at the very first time point and apply the model iteratively, we can reconstruct the trajectory through state space almost perfectly for all three data sets. Figure 4 shows the original and reconstructed time series for three representative genes. The interpolated time series (not shown) are nearly indistinguishable from the reconstruction. The very close fit is likely due to overfitting, but it does show that errors do not accumulate, despite the poor conditioning of $\mathbf{W}$. Analysis of the eigenvectors of the linear system also reveals that the final expression levels are close to fixed points of the system (within 3% for the spinal cord and hippocampus "adult" expression levels, within 9% for the final hippocampus injury expression levels): the linear model settles into an attractor in state space corresponding to the adult expression levels of the real organism.

4.4 Results and validation

A theory has only the alternatives of being right or wrong.
A model has the third possibility: it may be right, but irrelevant.
— M. Eigen

Before we address the issue of which individual parameters are well determined versus poorly determined, Section 4.4.1 will look at some of the overall properties of $\mathbf{W}$. Just as the average of a large number of poor estimators can yield a good estimator, the hope is that these global properties may be better determined than the individual parameters. Next, Section 4.4.2 shows how we can "separate the wheat from the chaff": identify the few well determined interactions in the network model. In Section 4.4.3 we put the class of most robust parameters (those due to the effect of kainate on the genes) to the test by comparing them with what is known in the literature. Section 4.4.4 does the same for the most robust gene-to-gene parameters.

4.4.1 Biologically plausible properties?

The linear model assumes that every gene is regulated by every other gene. However, when we look at the least squares fit of the model to the real data, we find that many of the parameters of the model are close to zero. Figure 5 shows a distribution of interaction weights that is very sharply peaked around zero (with 25th and 75th percentiles at $\pm 0.258\,\sigma$, compared to $\pm 0.674\,\sigma$ for a normal

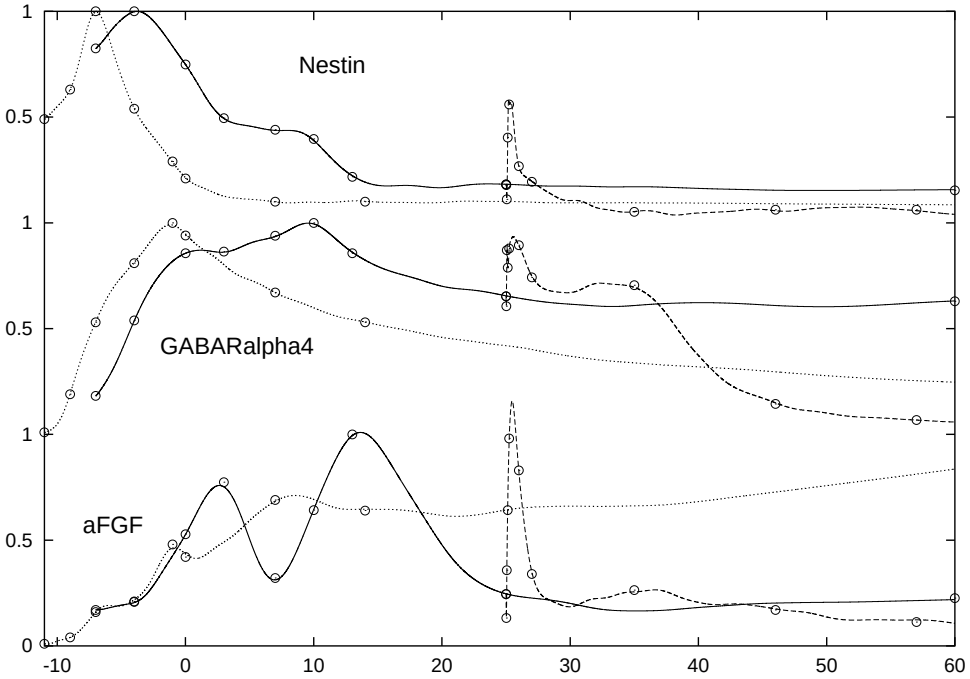

Figure 4: Original (dots) and reconstructed time series (lines) for nestin(top), GRa4 (middle) and aFGF (bottom). Time is in days from birth (day 0, corresponding to postnatal day P0 or embryonic day E22). Dotted line: spinal cord, starting day -11 (E11). Solid line: hippocampus development, starting day -7 (E15). Dashed line: hippocampus kainate injury, starting day 25 (P25)

distribution). This means the connection matrix is a good approximation to a sparse matrix, i.e., each gene is only influenced by a limited number of others, as we would expect for the real connection matrix. For a rough estimate of the number of "nonzero" parameters, we can fit the distribution with a mixture of two zero-centered Gaussians: more than 80% of the parameters get assigned to the narrowest Gaussian ($\sigma_1 = 0.068$), the rest to the much broader second Gaussian ($\sigma_2 = 0.375$).

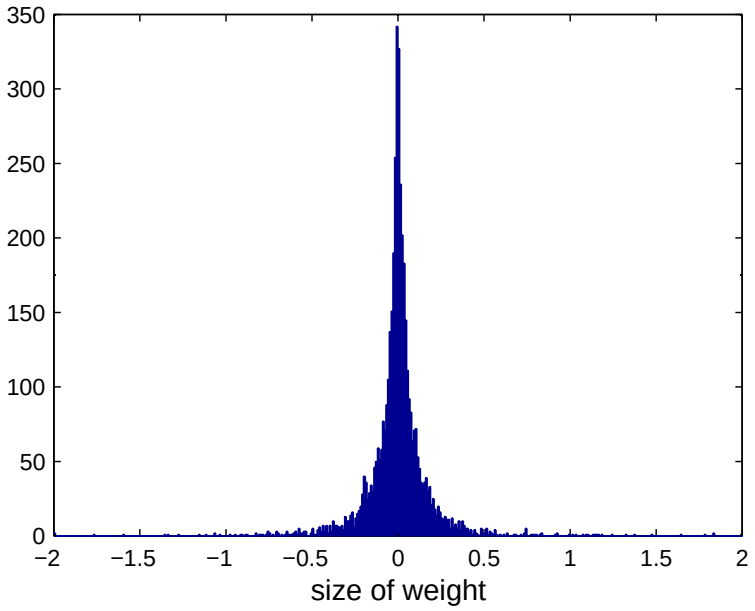

Figure 5: Histogram of average parameter values $\bar{w}$. Note the sharp peak at zero.

The sum of input weights to each gene is close to zero, i.e. there seem to be no genes that are primarily upregulated or downregulated. In fact, the distribution of the sums of input parameters is significantly much closer to zero than would be expected based on the distribution of parameters ($\sigma = 0.542$ versus expected $\sigma = 1.648$).[14] This may partially be an artifact of the model, because there are no fixed upper and lower expression thresholds for each gene. Predominantly positive (or negative) inputs to a gene would cause a increasing (decreasing) expression level, so positive and negative inputs must be balanced. More surprisingly, the distribution of input sum is close to zero even if we exclude the bias term b_i for each gene. In other words, we see few instances of genes which are "OFF" in the absence of regulatory inputs and which are

[14]The expected standard deviation of the sum (assuming the parameters are picked randomly from the distribution of parameters) is $\sqrt{68}\,\sigma_{params}$ for the input vector, $\sqrt{65}\,\sigma_{params}$ for the output vector, with $\sigma_{params} = 0.200$

upregulated by those inputs, or "ON" in the absence of regulatory inputs and downregulated.

Looking at the sum of output parameters from each gene (or other input such as kainate etc.), we see that this sum varies significantly more than expected based on the distribution of parameters ($\sigma = 2.513$ versus expected $\sigma = 1.611$).[14]In other words, there seem to be genes that have a predominantly positive (e.g. GRγ1) or negative (e.g. IGF II) regulatory effect on the other genes, which is in agreement with our biological knowledge.

Whereas the sum of input or output vectors tells us about the sign of regulation, the magnitude of the vectors informs us about the strength of regulation. We see a few significantly larger (e.g. GRγ1) and especially more small magnitude output vectors than expected, given the distribution of parameters.[15] It seems likely that the model has discovered that some genes are important regulators, while many others are not. This explanation is reinforced by a significant negative correlation ($r = -0.46$) between output magnitude and average triplicate variability for the gene, i.e. genes with less variation among the three replicates per time point had higher output magnitude. Important regulators are presumably more tightly regulated themselves, and thus would be expected to show less variability.

Surprisingly, we see a similar pattern for the input vectors: a few genes have large regulatory input parameters,[16] many others have all small regulatory inputs. Here, this variation is explained by a significant correlation ($r = 0.79$) between the magnitude of input vectors, and the standard deviation of slopes between time points. Since the linear model correlates expression levels with changes in expression levels, genes with rapid changes between time points will tend to have larger regulatory input parameters.[17] Each gene has a characteristic scale for its input parameters, corresponding to how fast the expression level of the gene changes throughout the time series. Instead of a mixture of "zero" and "nonzero" parameters, distribution of parameter values in Figure 5 should probably be considered as a mixture of distributions at these different scales.

When we divide the genes into functional categories, other interesting patterns emerge. The categories used were: *5HTR* (Serotonin Receptors), *AChR* (Acetylcholine Receptors), *GABA-R* (GABA Receptors), *GluR* (Glutamate Receptors), *ICS* (Intracellular Signaling), *NME* (Neurotransmitter Metabolizing Enzymes, including GAD), *cell cycle*, *glial*, *growth factor*, *insulin and IGF*, *neuronal*, *neurotrophin*, *progenitor*, *synaptic*, *trans-regulation*, and *other*.

NME and *GluR* are the main input classes, with weights coming from these genes on average more than twice as large as from other genes. Both categories

[15]Based on 100 random permutations of the parameter matrix.

[16]In our earlier work [23], this was assumed to be a sign of a poorly determined gene: poorly determined variables are often fitted using a number of very large inputs, which mostly cancel each other out. However, as we will see in Sections 4.4.4 and 4.4.3, one of the genes with highest input magnitude (BDNF) also has very well determined input parameters.

[17]No such correlation was found between output vector magnitude and average expression level, and no other significant correlations were found between input and output magnitudes, average and standard deviation of expression levels, slopes, or average triplicate standard deviation.

are known to play an important role in development and injury of the central nervous system. Also important are *ICS* (46% larger weights), *5HTR* (45% larger) and *trans regulation* (35% larger). A notable exception to primary regulation by *NME* and *GluR* is *growth factor*, which gets most input from *ICS*. We also observed that there is a tendency for genes in one functional class to receive more inputs from genes in the same class.

In summary, the least squares solution for the linear model results in a sparsely connected network, in which all genes have both positive and negative inputs, some genes are predominantly positive or negative regulators, there are a small number of important regulators with stable expression patterns, regulatory inputs are scaled by the speed of change in expression level of each gene, some of the main regulatory gene categories are known to play an important role, and there is more regulation within a functional category than between categories. All these high-level properties can be considered plausible from a biological point of view.

4.4.2 Robust parameters

All theorems are true.
All models are wrong.
And all data are inaccurate.
What are we to do?
We must be sure to remain uncertain.
— Leonard A. Smith

Because we expect large parts of the model to be underconstrained, we performed a Monte Carlo analysis to assess the effect of noise in the input data on the resulting parameters, and used this to determine the most robust parameters. As mentioned earlier, every value in the original data sets is really an average of triplicate experiments. This gives us high accuracy, and a rough estimate of the standard deviation at each measurement. We used this information to construct 40 new input data sets, adding a small amount of Gaussian noise (with the same standard deviation) to each. We then generated the linear model for each of these perturbed data sets, and analyzed the variability of the parameters over those 40 perturbed models.

To reflect our uncertainty about the kainate decay constant D_{KA} used to generate the kainate time series, we also lognormally perturbed D_{KA} around its estimated value of 100 min. This did not qualitatively change the results[18]. Similarly, continuity or discontinuity in the slope of the interpolated kainate time series[19] at the time of kainate injection had little effect on the results,

[18]Over the 40 perturbed models, D_{KA} varied from a minimum of 45.55 min (half-life of 31.57 min) to a maximum of 205.13 min (half-life of 142.18 min).

[19]When interpolating the kainate time series, the default slope at 0h is determined by the 0h and 0.5h data points. In contrast, the slope for unperturbed animals (at postnatal day 25, but without kainate injection) can be estimated from the hippocampus development time series. Using the slope calculated based solely on the kainate time series would therefore be equivalent to introducing a discontinuous jump in slope. Alternatively, we can force the

indicating that the time resolution in the kainate time series is sufficient to capture the initial dynamics of the response. The results listed below are for D_{KA} lognormally perturbed around 100 min, and a discontinuous slope of the interpolated kainate time series.

For each parameter in the model, we calculated the average magnitude $\bar{w}$ of the parameter, and compared it to its standard deviation σ_w over all 40 perturbed models. Note that although the original triplicate standard deviations are only a very rough estimate of the real variability for each gene and each time point, σ_w will be the result of some weighted average of a large number of these. In fact, for the specific weight w_{ji}, $\sigma_{w_{ji}}$ will be some weighted average of *all* the triplicate standard deviations for both y_i and y_j, at all time points, over all data sets. Just as the average of two poor estimators is itself a more accurate estimator (in fact, with half the variance of the original estimators), $\sigma_{w_{ji}}$ will have much greater accuracy than any of the triplicate standard deviations it is based on.

The *Z-score* of a parameter w is defined as $Z_w = |\bar{w}|/\sigma_w$, and indicates how many standard deviations the mean of the parameter is away from zero. From this Z-score, we then compute a *P value*, indicating the probability that the "real" value of the parameter for the best-fit linear model is zero, or even has opposite sign from $\bar{w}$ (i.e., the probability that this weight w is a *false positive*). We could simply count what fraction of the perturbed models have zero or even the opposite sign for the parameter in question. However, this would require many more than 40 runs to get sufficient accuracy in the P values. If we assume each parameter has a similar distribution, we can look at the distribution of all parameters, each normalized with respect to its mean $\bar{w}$ and standard deviation σ_w.[20] To estimate the P value for a specific value of Z, we count the number of instances where $(w - \bar{w})/\sigma_w > Z$, and divide by the total number of parameters $(40 \times 68 \times 65)$. The Z-scores (or their derived P values) are then used to identify robust parameters.

Note that the P values used here do not necessarily indicate the probability that the parameters found correspond to real biological regulatory interactions. They simply reflect the probability, given the noise on the input data, that the best-fit linear model for the true expression time series includes a parameter with this sign. In some instances, fitting a nonlinear interaction using a linear model may require a number of spurious linear terms. These parameters may be necessary for a good fit, and thus receive a high Z-score. Our hope is that gene regulation has sufficiently strong linear component that this first-order approximation with a linear model will mainly yield biologically relevant results.

kainate time series interpolation to start with the slope found at P25 in the developmental time series.

[20]The perhaps more standard—but less accurate—approach would be to assume all parameters have a Gaussian distribution over the 40 perturbed models. The distribution estimated above turns out to have a slightly sharper peak and longer tails than the Gaussian, resulting in larger P values for high Z, and smaller P values for low Z.

4.4.3 Results: Kainate parameters

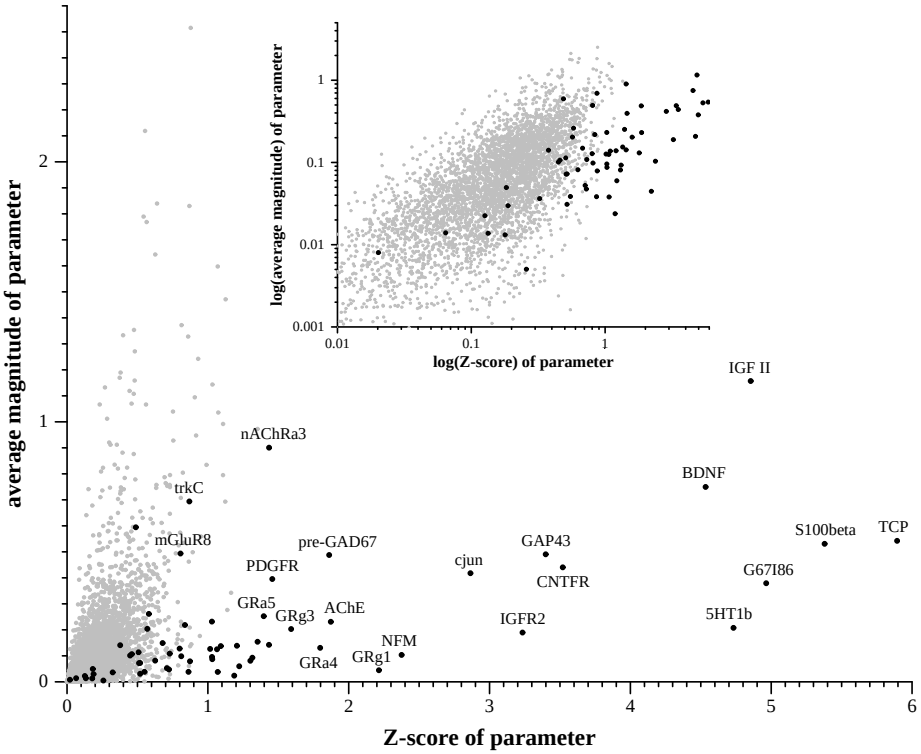

Figure 6: Left: Average magnitude of parameters vs. Z-score. Black points are kainate → gene parameters. Inset: log-log plot shows clearly that the kainate parameters on average have significantly higher Z_w and higher $|\bar{w}|$.

Figure 6 shows the Z-scores and average magnitude of all parameters. Surprisingly, the most robust parameters in the model are the parameters K_i, indicating the effect of kainate on each gene (black dots in Figure 6). This is probably because of the very fast and drastic effect of kainate-induced seizures on the system, as compared to the slow and subtle changes during development. Table 2 lists a number of the kainate → gene parameters with the highest Z-scores. Note that a few parameters (e.g. Kainate → 5-HT_{1B}) have a high Z-score but a low average magnitude. Such highly consistent but small parameter values may reflect a real but minor regulatory influence, or simply an absence of regulation (e.g. compensation for nonlinear effects).

Since it is unlikely that kainate actually regulates all these genes directly, we must assume there are some intermediate steps missing. Including an even earlier time point may shed some light on the precise sequence of regulation, especially for the BDNF/IGF II/S100β trio which also show gene-to-gene interactions (see Section 4.4.4). It should be noted, however, that most of the

existing literature on kainate response looks at much coarser time scales (typically hours), and that the 0.5 hr time interval in this data set is the shortest reported in the literature for kainate response.

Parameter	$\bar{w}$	σ_w	Z_w	P_w
Kainate $\dashv$ IGF II	-1.157	0.238	4.854	$4.355 \cdot 10^{-4}$
Kainate $\rightarrow$ BDNF	$+0.750$	0.165	4.534	$7.834 \cdot 10^{-4}$
Kainate $\rightarrow$ TCP	$+0.542$	0.092	5.894	$2.828 \cdot 10^{-5}$
Kainate $\rightarrow$ S100β	$+0.531$	0.099	5.379	$1.923 \cdot 10^{-4}$
Kainate $\rightarrow$ G67I86	$+0.379$	0.076	4.964	$3.620 \cdot 10^{-4}$
Kainate $\rightarrow$ 5-HT$_{1B}$	$+0.208$	0.044	4.732	$5.430 \cdot 10^{-4}$

Table 2: Robust kainate parameters. IGF II: insulin-like growth factor II; BDNF: brain-derived neurotrophic factor; TCP: T-complex protein; G67I86: glutamate decarboxylase 67 (GAD67) splice variant I86; 5-HT$_{1B}$: serotonin (5-hydroxytryptamine) receptor 1B

Kainate $\dashv$ IGF II: Kar *et al.* [49] found that IGF I, IGF II and insulin receptor sites show a marked decrease after kainate administration, suggesting "possible involvement of these growth factors in the cascade of neurotrophic events that is associated with the reorganization of the hippocampal formation observed following kainate-induced seizures." We found a four-fold decrease in IGF II mRNA levels one half hour after onset of seizures, followed by a two-fold increase in IGF I after 6 hours, and a large decrease of all IGF's and IGF receptors around 10-21 days. Our model suggests that it is IGF II which initially sets off the widespread changes in expression levels of insulin, the insulin-like growth factors, and their receptors following kainate administration.

Kainate $\rightarrow$ BDNF: BDNF is upregulated by kainate via two different promoters in hippocampal neurons [68], and the BDNF mRNA increase due to kainate is not blocked by protein synthesis inhibitors, indicating BDNF is regulated as an immediate early gene [16]. In the kainate injury time series, BDNF expression levels increase five-fold one half hour after onset of seizures. In the adult brain, BDNF is thought to play a major role in the development of kainate-induced hypertrophy in granular neurons of the dentate gyrus region of the hippocampus: administration of antisense deoxynucleotides for BDNF (sequestering the complementary BDNF mRNA) after kainate administration totally prevented neuronal hypertrophy [38]. Hippocampal BDNF levels are also correlated with severity of seizures and the extent of neuronal loss in the CA1 and CA3 regions of the hippocampus, and administration of exogenous BDNF exacerbates the damage to CA3 neurons [73]. Interestingly, in immature (20-day-old) rats, which normally do not show neuronal loss following kainic-acid induced seizures, BDNF apparently has a neuroprotective effect: antisense deoxynucleotide administration results in longer seizure duration and loss of CA1 and CA3 pyramidal cells and hilar interneurons inside the dentate gyrus [87].

Kainate $\rightarrow$ TCP: The case for kainate regulation of TCP (T-complex pro-

tein) is rather speculative, even though it is the single most robust parameter in our linear model. One intriguing link is the mapping of the epilepsy susceptibility locus EJM1 on chromosome 6 [74], near a human homologue of the mouse T-complex [25]. If TCP is indeed a major gene involved in kainate neurotoxicity, TCP gene defects might cause increased susceptibility to epilepsy.

Kainate → S100β: S100β is known to be upregulated five fold in human temporal lobe epilepsy [37]. S100β protects hippocampal neurons from damage induced by glucose deprivation [9], "suggesting that its elevation in neurological disorders may be a compensatory response." Perhaps its overexpression is a consequence of glucose deprivation due to neuronal hyperexcitation by kainate. Lastly, S100β induces apoptotic cell death in astrocytes [46], which protect against kainate neurotoxicity [64]. Hence, overexpression of S100β might cause aggravation of kainate toxicity by astrocyte apoptosis.

Kainate → G67I86: G67I86 is an embryonic splice variant[21] of GAD67 [13], expressed in mice from E10.5 to E15.5 (corresponding to rat E12 to E17), and not detectable in adult brain [85]. GAD synthesizes the fast-acting neurotransmitter GABA from glutamate. The short leader peptide translated from G67I86 is not enzymatically active, but is thought to exert some unknown regulatory function [85]. Mature GAD67 mRNA was known to be upregulated in hippocampal dentate granule cells four hours after kainic acid injection [78, 24]. However, the more fine-grained time series used here shows that G67I86 mRNA levels increase first (0.5h-1.5h), followed by a second embryonic splice variant G67I80[22] (1.5h), and finally the adult GAD67 mRNA (1.5h-24h). This is the same sequence in which these splice variants occur during development [85], indicating that GAD67 expression after kainate injury may be recapitulating its developmental program. Such recapitulation of developmental processes plays an important role in regeneration of the peripheral nervous system following injury, and has also been implicated in the central nervous system [19, 98, 97].

Kainate → 5-HT$_{1B}$: Kainate administration causes a release of serotonin in the hippocampus [88], which would be expected to provoke a compensatory down-regulation of the 5-HT$_{1B}$ serotonin receptor instead [48]. However, the interaction between serotonin and 5-HT$_{1B}$ is more complicated than that: 5-HT$_{1B}$ is an autoreceptor [48], i.e., activation of the receptor causes an inhibition of serotonin release. In addition, receptor activation will also cause a desensitization of the 5-HT$_{1B}$ receptor [70]. It is conceivable that this complex set of feedback loops might cause a transient upregulation of 5-HT$_{1B}$ by kainate. Indeed, the 5-HT$_{1B}$ expression time series shows a transient upregulation, peaking at 1.5-3h, followed by a decrease below the original expression level.

The direct effect of kainate is a transient phenomenon, lasting at most a couple of hours. It could be argued that we might be able to derive these kainate parameters directly from the first few time points in the kainate injury

[21] G67I86 contains a 80 bp insert not found in the adult GAD67 mRNA. This insert includes a stop codon which truncates the translation of the mRNA, resulting in a short leader peptide rather than the full-length GAD67 protein.

[22] G67I80 contains a 6bp shorter insert than G67I86, and is translated into the leader peptide, plus a truncated but enzymatically active GAD67 protein.

time series, in which all the effort involved in integrating results from three separate time series is entirely superfluous.

One can indeed find reasonable, intuitive estimators for K_i and the corresponding Z-score Z_{K_i} using the change in expression level between the first two time points, $\Delta y_i(0, 0.5)$ and the triplicate error $\sigma_i(0)$ and $\sigma_i(0.5)$ at those time points (see also Figure 7):

$$k_i = \frac{\Delta y_i(0, 0.5)}{0.5} \approx K_i \qquad \text{(slope of the time series)} \tag{31}$$

$$z_i = \frac{\Delta y_i(0, 0.5)}{\sigma_i(0) + \sigma_i(0.5)} \approx Z_{K_i} \tag{32}$$

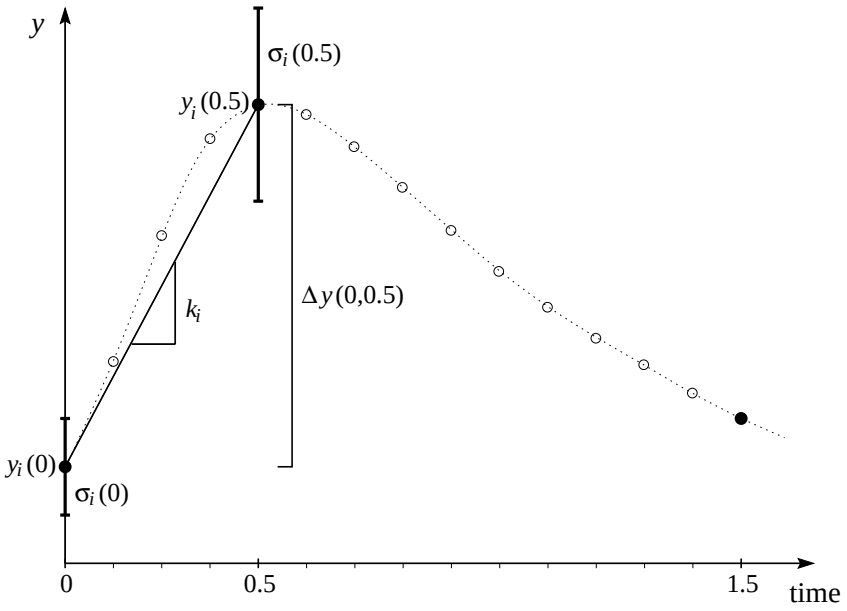

Figure 7: Estimates for kainate parameters K_i and their Z-score Z_{K_i}, based on the first two time points in the kainate time series. The original data points are solid, interpolated time points hollow. k_i is the slope between the two first time points in the data set, z_i is a measure of the significance of the change in expression level.

These simple estimators k_i and z_i show a reasonable correlation to the results obtained by fitting the linear model to all three complete time series: $r = 0.84$ and $r = 0.83$ respectively. Unfortunately, this nice correlation starts to break down in the most interesting region: for those genes with high Z-score. For the ten genes with highest Z-score, the correlation between k_i and K_i is only barely significant: $r = 0.67$, and the correlation between z_i and Z_{K_i} is no longer

significant: $r = 0.47$. The estimates are even worse for the six genes listed in Table 2:

- k_i strongly underestimates K_i for IGF II ($k_i = 0.696$; $K_i = 1.157$)
- k_i strongly overestimates K_i for BDNF ($k_i = 1.215$; $K_i = 0.750$).
- z_i strongly underestimates Z_{K_i} for S100β ($z_i = 3.333$; $Z_{K_i} = 5.379$)
- z_i strongly underestimates Z_{K_i} for 5-HT$_{1B}$ ($z_i = 3.667$; $Z_{K_i} = 4.732$),
- z_i strongly overestimates Z_{K_i} for G67I86 ($z_i = 12.333$; $Z_{K_i} = 4.964$)
- z_i strongly overestimates Z_{K_i} for BDNF ($z_i = 7.156$; $Z_{K_i} = 4.534$).

Of the six genes with largest Z-scores, TCP is the only one with accurate estimations. Both estimators are particularly unreliable for high values. Although their accuracy is probably sufficient to pick up most of the important interactions, it is clear that adding in the rest of the kainate time series, as well as the two developmental time series, significantly improves the results.

4.4.4 Results: Gene-to-gene parameters

Kainate is an exogenous input to the system, so the immediate effects of kainate administration are easy to isolate. In addition, kainate injury as a model of temporal lobe epilepsy is very well studied. The gene-to-gene interactions on the other hand are much harder to unravel, both *in vivo* as *in vitro*, and consequently less information about them is available in the literature.

In the linear model, the parameters accounting for the gene-to-gene interactions have much smaller Z-scores than for the kainate-to-gene interactions. The gene-to-gene parameters with $Z_w > 1.0$ are listed in Table 3. Interestingly, GRγ1 and IGF II—accounting for seven out of ten entries in the table—also have the highest magnitude output vectors, which we interpreted as a sign of important regulatory genes in Section 4.4.1. None of the Z-scores are significant at the $P = 0.05$ level, although in total we only expect about one false positive in this table of ten parameters. Remember that the goal of this model is primarily to generate interesting new hypothesis to guide further research. From that point of view, nine out of ten is quite acceptable.

GFAP $\dashv$ GFAP; BDNF $\dashv$ BDNF: It is interesting to note that two out of the ten gene-to-gene parameters in Table 3 are autoregulatory, i.e., a gene downregulating itself. Although these specific genes are not known to regulate themselves, in general such negative feedback loops are an important homeostatic mechanism.

BDNF, IGF II $\dashv$ BDNF, S100β: BDNF and S100β seem to be regulated by BDNF itself, and IGF II. Moreover, the regulation by IGF II is in both cases roughly twice as strong as the regulation by BDNF (2.02 times as strong for S100β, 1.64 times as strong for BDNF, well within the error bounds on

Parameter	$\bar{w}$	σ_w	Z_w	P_w
GFAP $\dashv$ GFAP	-0.277	0.243	1.138	0.097
BDNF $\dashv$ BDNF	-0.973	0.719	1.353	0.072
IGF II $\dashv$ BDNF	-1.598	1.494	1.070	0.106
BDNF $\dashv$ S100β	-0.343	0.294	1.165	0.093
IGF II $\dashv$ S100β	-0.693	0.617	1.123	0.099
GRγ1 $\rightarrow$ GRα4	$+1.144$	1.108	1.032	0.112
GRγ1 $\rightarrow$ GRβ2	$+1.036$	0.965	1.074	0.106
GRγ1 $\rightarrow$ G67I80/86	$+1.471$	1.307	1.126	0.098
GRγ1 $\rightarrow$ AChE	$+0.992$	0.895	1.108	0.101
GRγ1 $\rightarrow$ NFM	$+0.795$	0.718	1.108	0.101

Table 3: All gene-to-gene parameters with Z-score greater than 1.0. GFAP: glial fibrillary acidic protein; BDNF: brain-derived neurotrophic factor; IGF II: insulin-like growth factor II; G67I80/86: glutamate decarboxylase 67 (GAD67) splice variants I80 and I86; AChE: acetylcholinesterase; NFM: neurofilament medium; GRα4, GRβ2, and GRγ1: $GABA_A$ receptor subunits α4, β2, and γ1.

these parameters). This might lead us to infer the presence of a hidden node[23] regulating BDNF and S100β, as in Figure 8. All three of these genes are growth factors, playing a role in differentiation and development of neurons, as well as in neurite outgrowth. Both IGF II and BDNF induce differentiation of CNS stem cells-derived neuronal precursors, and IGF I and BDNF may act together or sequentially to promote differentiation [8] (the combination of IGF II and BDNF was not examined, but IGF II was found to have a similar effect as IGF I on differentiation). Furthermore, IGF II and S100β have almost opposite effects on the growth of developing serotonin and dopamine neurons in vitro [60]. The interactions between BDNF, IGF II and S100β may play an important role in differentiation of developing neurons into different cell types.

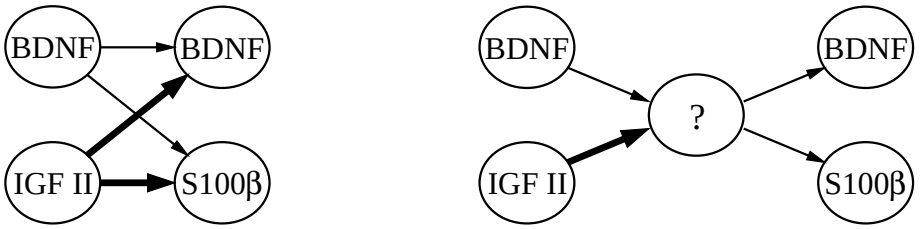

Figure 8: Alternative models for the interaction between BDNF, IGF II, and S100β. Note that BDNF was drawn twice for clarity.

[23]Interestingly, the combination (BDNF + 1.8 IGF II) shows an even higher Z-score for regulating BDNF and S100β (1.434 and 1.332), and a higher average Z-score overall (0.503 versus 0.476 for BDNF alone and 0.436 for IGF II alone)

GRγ1 $\rightarrow$ GRα4, GRβ2: GRγ1 is a $GABA_A$ receptor subunit. Each pentameric $GABA_A$ receptor consists of five subunits, and so far, 19 mammalian subunit types (plus several splice variants) have been identified, grouped into seven classes: 6 α subunit types, 4 β, 3 γ, 1 δ, 1 ϵ, 1 π, and 3 ρ [10]. In the CNS, $GABA_A$ receptors generally consist of combinations of α and β subunits, plus one or more of the γ, δ, or ϵ types, allowing for possibly hundreds of different $GABA_A$ receptors. The upregulation of α4 and β2 by γ1 seems to imply the coordinate regulation of an $\alpha 4\beta 2\gamma 1$ $GABA_A$ receptor by its γ1 subunit. This specific receptor combination has not previously been described, perhaps because α4 and γ1 are less common subunits, α4 antibodies have yielded inconsistent results, and a frequently used $\beta 2/3$ antibody does not distinguish between β2 and β3 subunits. However, $\alpha 4\beta\gamma$ [24] has been detected in the cortex, striatum and hippocampal pyramidal cells [69], $\alpha 4\beta 2\delta$ has been detected in thalamus and hippocampal dentate granule cells [69] (δ is known to substitute for γ in some receptors), α4, β2, and δ-mRNA levels are tightly correlated in individual dentate granule cells [15], and the hippocampus does contain some of the highest concentrations of both α4 [52] and γ1 [53].

GRγ1 $\rightarrow$ G67I80/86: GABA is implicated in neuronal development, and it is thought that GAD (the enzymes(s) which synthesize GABA from glutamate) regulates the expression of GABA receptors via GABA, and that GABA receptor activation in turn regulates GAD expression [82]. GAD67, α4, β1 and γ1 expression is associated with proliferation and development in the rat embryonic and early postnatal CNS [61]. Considering the timing, this GAD67 expression presumably consists mainly of the embryonic splice variants G67I80 and G67I86. Total $GABA_A$ receptor mRNA was found to be highly correlated (R=0.99) with total GAD mRNA in cervical spinal cord [82], and it seems likely that the $GABA_A$ receptor subunits which appear transiently during spinal cord development (α4, α5, β1, β2, γ1, and γ3) would be highly correlated with the transiently expressed GAD67 variants.

GRγ1 $\rightarrow$ AChE: GABA has been conjectured to control the development of cholinergic neurons, and indeed, AChE expression is downregulated by activation of $GABA_A$ receptors [51]. Exposure to GABA has also been shown to downregulate $GABA_A$ receptor subunit γ1, as well as α1, β2, β4, and γ2 [11], so perhaps the effect of GABA on AChE (and β2) is due to downregulation of γ1.

GRγ1 $\rightarrow$ NFM: NFM (neurofilament medium) is a neuronal marker, so it is not surprising that NFM would be upregulated in conjunction with a number of neurotransmitter receptors (α4, β2, γ1) and neurotransmitter metabolizing enzymes (G67I80/86, AChE). It has also been noted that GAD family mRNA expression parallels neurofilament expression [82].

Interestingly, GRγ1 and IGF II—accounting for seven out of ten entries in the table—also have the highest magnitude output vectors, which we interpreted as a sign of important regulatory genes in Section 4.4.1. Whereas IGF II is known to be an important regulator, no such role for GRγ1 has been pos-

[24] The precise subtype of β and γ was not identified

tulated before. The GRγ1 gene product is part of a receptor complex, and would not be expected to play any direct regulatory role. Nevertheless, it is not unheard off for a protein with a primarily structural role in the cell to also have a regulatory effect (for example, CASK, a cytoskeleton protein acting as a structural girder for cell junctions, is known to enter the nucleus and directly regulate gene expression [45]). Alternatively, some other factor not included in our data set may be driving the coordinate regulation of these genes, and be most highly correlated with GRγ1 (e.g., GRγ1 may have few other regulatory inputs, and may show a faster response to this regulator than the other genes), in which case the best causal explanation within the scope of the data set would be regulation by GRγ1. Either way, the model shows a significant coordinate regulation of these gene and, lacking any other explanations, further investigation of the role of GRγ1 may be warranted.

5 Conclusions

This is not the end.
It is not even the beginning of the end.
But it is, perhaps, the end of the beginning.
— Winston Churchill, 10 November 1942

5.1 The story so far...

Rather than giving up on these network models because they are officially "underdetermined", I have shown that they can indeed be applied to infer at least *part* of the regulatory interactions between genes from large-scale gene expression data. The first important result is a rather theoretical one: the estimates of data requirements in Chapter 3 show that, as long as we impose sufficient constraints on the network models, their data requirements might only scale logarithmically with the number of variables (number of genes). This compares favorably with the data requirements for clustering, although it is still perhaps an order of magnitude or more larger.

In practice, the lack of data compared with the number of parameters of the data turned out to be much more of a stumbling block than I had originally anticipated. In retrospect, the underdetermined nature of the model should not have come as a surprise, simply based on the dimensionality of the data, and the significant correlations between the measurements. Nevertheless, I showed that it is indeed possible to identify some portion of the significant weights in the model, using the knowledge we have regarding the variability of the individual measurements. This points out yet again how crucial it is to know the error behavior of the data one is working with. A common trend towards the usage of replicate experiments may allow for more widespread use of this technique.

The linear model used in Chapter 4 is an extreme simplification, and should be regarded only as a first-order approximation. The tissues studied consist of multiple functional regions, multiple layers within each region, and multiple cell

types within each layer, all of which can be expected to exhibit different expression patterns during development and injury. Also, the number of variables used is only a small fraction of the important variables that play a role in these tissues. Protein, neurotransmitter, and neuropeptide levels are missing entirely. Nevertheless, we find we can isolate several important known regulatory interactions. Other predictions generated by the model seem quite plausible when compared with current knowledge, and form useful new hypotheses that can guide further experimentation.

5.2 Directions for future research

Some further refinements could still be made to the linear model. For example, to capture the change in developmental speed around birth, we could explicitly add an additional input to the system for the "birth" event.

Rather than using the ordinary least squares solution, we could use a weighted least squares. This would allow us to (1) weigh expression levels according to the corresponding triplicate standard deviations on the measurements, (2) weigh interpolated time points based on the location within the interpolation interval (higher weight close to the real data points), (3) give equal weight to all the intervals between real data points (at the moment, their "weight" in the least squares solution is essentially proportional to the length of the interval, giving much higher weight to the final data points which are months apart). Note that the use of the Monte Carlo analysis in Chapter 4 essentially already covers the first two points: measurement with a larger triplicate standard deviation will get perturbed more, resulting in a smaller contribution to the Z-score of the associated weights. Similarly, perturbations in the real data points will probably cause larger perturbations in the interpolated time points, particularly in those interpolated points farthest away from the real data points.

Lastly, since the linear model essentially performs a multiple regression of all genes on all genes, perhaps we could exploit some of the techniques developed to determine the significant inputs in multiple regression. Some of these are based on adding additional penalty terms to the optimization to account for the number of inputs, size of input weights, etc.

The introduction of dynamic Bayesian network methodology for gene expression analysis is an especially promising development. The nonlinear neural network presented in the Introduction is essentially similar to a nonlinear dynamic Bayesian network, as pointed out by Murphy and Mian [67]. Bayesian networks do not so much provide a different model, but rather a new perspective from an area which has a very thorough theoretical foundation, just as the neural network perspective provides us with useful insights and efficient tools to tackle a set of nonlinear differential equations.

5.3 A look towards the future

As of this writing, we are still in the exponential phase of deployment of large-scale gene expression measurement technologies. Frost & Sullivan [28] estimate

approximately a doubling in the number of arrays used for each of the next two years, with a prediction of well over 1.5 million arrays used in 2003. Considering the nearly constant stream of new technologies, this may very well be an underestimate. Miniaturization, automation and mass-production will likely reduces the cost per gene expression experiment to a few dollars per chip. Once these technologies start influencing our daily lives—probably primarily as diagnostic tools in a hospital setting—there is no predicting how pervasive they will become.

As these genome-scale technologies mature, we can expect to see:

- More whole-genome measurements, rather than selected subsets of genes, increasing the need for analysis tools that can deal with large amounts of superfluous variables.

- Higher accuracy, allowing better distinction between genes with similar expression patterns. At the moment, some people still view array data essentially as qualitative data: useful as a first approach, but in need of validation by other means if one actually wants to publish a result. With increasing accuracy, automation, and understanding of the errors, large scale gene expression technology will likely become generally accepted as a quantitative measurement tool.

- Possibly higher time resolution, as we get more experience with response times of the very fastest genes. To observe the very fastest changing genes, we may very well have to resort to lab-on-a-chip approaches to do the measurement *in situ*, before the mRNA decays. For now, time resolutions on the order of a few minutes are definitely feasible, and sufficient for the vast majority of mRNA species.

- More data points, making the sorts of approaches presented here more effective. As mentioned before, in order to infer the regulation of any gene, one has to thoroughly exercise the different inputs to the gene. The recent trickle of very large data sets (such as Hughes *et al.* [47]: 300 separate measurements on yeast, all calibrated) are likely only the beginning.

- More replicates, resulting in better error models and a better appreciation of why and when genes show increased variability. Currently, replicates are mainly used for averaging (thus reducing the error variance), and for assigning significance levels to the amount of up- or down-regulation of a gene. However, as I have illustrated in Chapter 4, they also provide a crucial tool to identify well-determined regulatory interactions in the data.

Especially the advent of larger data sets and more data sets with replicates Should make the modeling methodologies developed here more widely applicable. We can also expect to see production of more large-scale non-mRNA data, bringing with it an increased need for integration between disparate data types within the same computational analysis, as well as integration with other information sources, such as literature data bases, etc. The Bayesian approach to

learning neural networks—adding additional knowledge as priors on the resulting network—provides for a very flexible tool to integrate these disparate types of data.

After the explosion of genomic-scale data, we are finally starting to see a smattering of computational tools that can deal with this data. I hope the techniques I have developed here will be a useful addition to this growing genomic biologist's tool chest. Much work yet remains to be done, and as the technologies and analysis tools develop, we will likely identify other challenges.

As the saying goes: "in the land of the blind, the one-eyed man is King" large-scale gene expression technology has given us an "eye" into the internal workings of cells. It's still only one eye, so we're only seeing half the picture. And it's still somewhat blurry, but we're furiously developing lenses. But what a difference one eye makes...

Appendix A: Fitting the linear model

I believe the day will come when the biologist will—without being a mathematician—not hesitate to use mathematical analysis when he requires it.
— Karl Pearson, in Nature, 1901

First, we rewrite Equation 29 in matrix notation:

$$\frac{\Delta\vec{\mathbf{y}}(t)}{\Delta t} = \mathbf{W}\vec{\mathbf{y}}(t) + \vec{\mathbf{K}}\,\kappa(t) + \vec{\mathbf{T}}\,\tau + \vec{\mathbf{B}} \tag{33}$$

where $\Delta\vec{\mathbf{y}}(t)$, $\vec{\mathbf{y}}(t)$, $\vec{\mathbf{K}}$, $\vec{\mathbf{T}}$ and $\vec{\mathbf{B}}$ are now column vectors containing the corresponding values of $\Delta y_i(t) = y_i(t+\Delta t) - y_i(t)$, $y_i(t)$, K_i, T_i and b_i for all 65 genes, and $\mathbf{W}$ is a 65×65 matrix containing the parameters w_{ji}. To simplify, we can include the parameters $\vec{\mathbf{K}}$, $\vec{\mathbf{T}}$ and $\vec{\mathbf{B}}$ as extra columns in the matrix $\mathbf{W}$ (which now becomes a rectangular 65×68 matrix), provided we add $\kappa(t)$, τ, and a unit constant as additional "inputs" to $\vec{\mathbf{y}}(t)$ in the right hand side.

$$\frac{\Delta\vec{\mathbf{y}}(t)}{\Delta t} = \begin{bmatrix} \mathbf{W} & \vec{\mathbf{K}} & \vec{\mathbf{T}} & \vec{\mathbf{B}} \end{bmatrix} \cdot \begin{bmatrix} \vec{\mathbf{y}}(t) \\ \kappa(t) \\ \tau \\ \mathbf{1} \end{bmatrix} \tag{34}$$

For convenience, we will call the augmented weight matrix $\widetilde{\mathbf{W}}$, and the augmented input vector $\widetilde{\mathbf{y}}(t)$. Note that we have one such equation for each time interval in each of the interpolated time series. We can combine these into a single matrix equation:

$$\frac{\Delta\mathbf{Y}}{\Delta t} = \widetilde{\mathbf{W}}\,\widetilde{\mathbf{Y}} \tag{35}$$

where $\Delta\mathbf{Y}$ is a 65×52080 matrix, containing $\Delta\vec{\mathbf{y}}(t) = \vec{\mathbf{y}}(t+\Delta t) - \vec{\mathbf{y}}(t)$ for all 52080 interpolated time intervals of each time series; and $\widetilde{\mathbf{Y}}$ is a 68×52080 matrix, containing $\vec{\mathbf{y}}(t)$, $\kappa(t)$, τ, and a unit constant for all but the *last* time points of each time series:

$$\Delta\mathbf{Y} = \begin{bmatrix} \underbrace{\begin{matrix} y_1^s(2)\text{-}y_1^s(1) & \cdots & y_1^s(n_s)\text{-}y_1^s(n_s\text{-}1) \\ \vdots & \ddots & \vdots \\ y_N^s(2)\text{-}y_N^s(1) & \cdots & y_N^s(n_s)\text{-}y_N^s(n_s\text{-}1) \end{matrix}}_{\text{spinal cord development}} & \underbrace{\begin{matrix} y_1^h(2)\text{-}y_1^h(1) & \cdots & y_1^h(n_h)\text{-}y_1^h(n_h\text{-}1) \\ \vdots & \ddots & \vdots \\ y_N^h(2)\text{-}y_N^h(1) & \cdots & y_N^h(n_h)\text{-}y_N^h(n_h\text{-}1) \end{matrix}}_{\text{hippocampus development}} & \underbrace{\begin{matrix} y_1^k(2)\text{-}y_1^k(1) & \cdots & y_1^k(n_k)\text{-}y_1^k(n_k\text{-}1) \\ \vdots & \ddots & \vdots \\ y_N^k(2)\text{-}y_N^k(1) & \cdots & y_N^k(n_k)\text{-}y_N^k(n_k\text{-}1) \end{matrix}}_{\text{hippocampus kainate injury}} \end{bmatrix}$$

$$\widetilde{\mathbf{Y}} = \begin{bmatrix} y_1^s(1) & \cdots & y_1^s(n_s-1) & y_1^h(1) & \cdots & y_1^h(n_h-1) & y_1^k(1) & \cdots & y_1^k(n_k-1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ y_N^s(1) & \cdots & y_N^s(n_s-1) & y_N^h(1) & \cdots & y_N^h(n_h-1) & y_N^k(1) & \cdots & y_N^k(n_k-1) \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \kappa(1) & \cdots & \kappa(n_k-1) \\ 0 & \cdots & 0 & 1 & \cdots & 1 & 1 & \cdots & 1 \\ 1 & \cdots & 1 & 1 & \cdots & 1 & 1 & \cdots & 1 \end{bmatrix} \tag{36}$$

$$\widetilde{\mathbf{W}} = \begin{bmatrix} w_{1,1} & \cdots & w_{N,1} & K_1 & T_1 & b_1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ w_{1,N} & \cdots & w_{N,N} & K_N & T_N & b_N \end{bmatrix} \tag{37}$$

where N is the number of genes ($N = 65$), n_s, n_h and n_k are the number of interpolated time points in the spinal cord development, hippocampus development and hippocampus kainate injury time series, respectively (for convenience, the interpolated time points are ordered from 1 to n in each time series), and $y_i^s(t)$, $y_i^h(t)$ and $y_i^k(t)$ are the interpolated expression levels in those three time series. Note that the kainate level $\kappa(t)$ (third row from the bottom in Equation 36) is zero except for the kainate injury time series, and that the tissue indicator variable τ (second row from the bottom in Equation 36) is 0 for spinal cord and 1 for the two hippocampus time series.

If $\widetilde{\mathbf{Y}}$ were an invertible square matrix, we could solve for $\widetilde{\mathbf{W}}$ exactly using $\widetilde{\mathbf{W}} = \Delta\mathbf{Y}/\Delta t \cdot \widetilde{\mathbf{Y}}^{-1}$. Since $\widetilde{\mathbf{Y}}$ is rectangular, and has more rows than columns, the system is overdetermined and no exact solution for Equation 35 is possible. However, we can find the least squares solution $\mathbf{W}^+$ using the following formula (see, e.g., [84, 35]):

$$\mathbf{W}^+ = \frac{\Delta\mathbf{Y}}{\Delta t}\widetilde{\mathbf{Y}}^{\mathrm{T}}\left(\widetilde{\mathbf{Y}}^{\mathrm{T}}\widetilde{\mathbf{Y}}\right)^{-1} \tag{38}$$

(Or, if $\widetilde{\mathbf{Y}}$ is rank-deficient, we could use the pseudoinverse [84, 35] to find a unique least squares solution). The resulting 65-by-68 matrix $\mathbf{W}^+$ gives us the least squares fit for the parameters w_{ji}, K_i, T_i and b_i in Equation 29.

References

[1] Aebersold, R., Hood, L. E., and Watts, J. D. Equipping scientists for the new biology. *Nature Biotechnology 18*, 4 (April 2000), 359.

[2] Akima, H. A new method of interpolation and smooth curve fitting based on local procedures. *J. ACM 17*, 4 (1970), 589–602.

[3] Akutsu, T., Miyano, S., and Kuhara, S. Identification of genetic networks from a small number of gene expression pattern under the Boolean network model. In Altman et al. [6], pp. 17–28.

[4] Alla, H., and David, R. Continuous and hybrid Petri nets. *Journal of Circuits, Systems, and Computers 8*, 1 (1998), 159–188.

[5] Altman, R. B., Dunker, A. K., Hunter, L., and Klein, T. E., Eds. *Pacific Symposium on Biocomputing '98* (Singapore, 1998), World Scientific Publishing Co.

[6] Altman, R. B., Dunker, A. K., Hunter, L., Klein, T. E., and Lauderdale, K., Eds. *Pacific Symposium on Biocomputing '99* (Singapore, 1999), World Scientific Publishing Co.

[7] Arkin, A., Ross, J., and McAdams, H. H. Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells. *Genetics 149*, 4 (August 1998), 1633–1648.

[8] Arsenijevic, Y., and Weiss, S. Insulin-like growth factor-I is a differentiation factor for postmitotic cns stem cell-derived neuronal precursors: Distinct actions from those of brain-derived neurotrophic factor. *J. Neurosci. 18*, 6 (1998), 2118–2128.

[9] Barger, S. W., Van Eldik, L. J., and Mattson, M. P. S100 beta protects hippocampal neurons from damage induced by glucose deprivation. *Brain Res. 677*, 1 (1995), 167–170.

[10] Barnard, E. A., Skolnick, P., Olsen, R. W., Mohler, H., Sieghart, W., Biggio, G., Braestrup, C., Bateson, A. N., and Langer, S. Z. International union of pharmacology. XV. subtypes of gamma-aminobutyric acid A receptors: Classification on the basis of subunit structure and receptor function. *Pharmacol. Rev. 50*, 2 (June 1998), 291–313.

[11] Baumgartner, B. J., Harvey, R. J., Darlison, M. G., and Barnes Jr, E. M. Developmental up-regulation and agonist-dependent down-regulation of GABAA receptor subunit mRNAs in chick cortical neurons. *Brain. Res. Mol. Brain. Res. 26*, 1–2 (October 1994), 9–17.

[12] Bellman, R. *Adaptive Control Processes: A Guided Tour.* Princeton University Press, 1961.

[13] Bond, R. W., Wyborski, R. J., and Gottlieb, D. I. Developmentally regulated expression of an exon containing a stop codon in the gene for glutamic acid decarboxylase. *Proc. Natl. Acad. Sci. USA 87*, 22 (1990), 8771–8775.

[14] Bray, D. Intracellular signalling as a parallel distributed process. *J. Theor. Biol. 143* (1990), 215–231.

[15] Brooks-Kayal, A. R., Shumate, M. D., Jin, H., Lin, D. D., Rikhter, T. Y., Holloway, K. L., and Coulter, D. A. Human neuronal gamma-aminobutyric acid-A receptors: Coordinated subunit mRNA

expression and functional correlates in individual dentate granule cells. *J. Neurosci. 19*, 19 (1999), 8312–8318.

[16] Castren, E., Berninger, B., Leingartner, A., and Lindholm, D. Regulation of brain-derived neurotrophic factor mRNA levels in hippocampus by neuronal activity. *Prog. Brain. Res. 117* (1998), 57–64.

[17] Chen, T., He, H. L., and Church, G. M. Modeling gene expression with differential equations. In Altman et al. [6], pp. 29–40.

[18] Claverie, J.-M. Computational methods for the identification of differential and coordinated gene expression. *Human Molecular Genetics 8*, 10 (1999), 1821–1832.

[19] Cohen, M. J., and Hall, G. F. Control of neuron shape during development and regeneration. *Neurochem. Pathol. 5*, 3 (December 1986), 331–343.

[20] Cover, T. M., and Thomas, J. A. *Elements of information theory.* John Wiley & Sons, Inc., New York, 1991.

[21] D'haeseleer, P., Liang, S., and Somogyi, R. Genetic network inference: From co-expression clustering to reverse engineering. *Bioinformatics 16*, 8 (2000), 707–726.

[22] D'haeseleer, P., Wen, X., Fuhrman, S., and Somogyi, R. Mining the gene expression matrix: inferring gene relationships from large scale gene expression data. In *Information processing in cells and tissues* (1997), R. C. Paton and M. Holcombe, Eds., Plenum Press, pp. 203–212.

[23] D'haeseleer, P., Wen, X., Fuhrman, S., and Somogyi, R. Linear modeling of mRNA expression levels during CNS development and injury. In Altman et al. [6], pp. 41–52.

[24] Ding, R., Asada, H., and Obata, K. Changes in extracellular glutamate and GABA levels in the hippocampal CA3 and CA1 areas and the induction of glutamic acid decarboxylase-67 in dentate granule cells of rats treated with kainic acid. *Brain Res. 800*, 1 (July 1998), 105–113.

[25] Durner, M., Greenberg, D. A., and Delgado-Escueta, A. V. Is there a genetic relationship between epilepsy and birth defects? *Neurology 42*, 4 Suppl. 2 (1992), 63–67.

[26] Fisher, R. A. In *The advanced theory of statistics*, M. G. Kendall and A. Stuart, Eds., 3rd ed., vol. 1. Hafner Press, 1969, p. 391.

[27] Franklin, G. F., Powell, J. D., and Emami-Naeini, A. *Feedback control of dynamic systems*, 3rd ed. Addison-Wesley, Reading, MA, 1994.

[28] Frost & Sullivan. Opportunities for DNA microchip and array technologies. Available from http://www.frost.com/, December 1999.

[29] Fuhrman, S., D'haeseleer, P., Liang, S., and Somogyi, R. *Tracing genetic information flow from gene expression to pathways and regulatory networks.* MIT Press, Cambridge, MA, 2000. (in press).

[30] Geschwind, D. Opening talk for Short Course on DNA Microarrays, at Soc. for Neuroscience Annual Meeting, October 1999.

[31] Glass, L. Combinatorial and topological methods in nonlinear chemical kinetics. *J. Chem. Phys. 63*, 4 (1975), 1325–1335.

[32] Glass, L., and Kauffman, S. A. Co-operative components, spatial localization and oscillatory cellular dynamics. *J. Theor. Biol. 34* (1972), 219–237.

[33] Glass, L., and Pasternack, J. S. Stable oscillations in mathematical models of biological control systems. *J. Math. Biol. 6* (1978), 207–223.

[34] Goldstein, K. M., Taboada, E., Conway, A., Hess, K., Garden, M., Nadon, R., Jackson, R., Lasky, S. R., Fairfield, E., Minning, T., Eynon, B., and others. Various comments on the GENE-ARRAYS mailing list, available at listserv@listserv.ucsf.edu. See also http://www.egroups.com/message/microarray/1974, September 2000.

[35] Golub, G. H., and Loan, C. F. V. *Matrix Computations*, 3rd ed. Johns Hopkins Series in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, 1996.

[36] Goss, P. J., and Peccoud, J. Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets. *Proc. Natl. Acad. Sci. USA 95*, 12 (1998), 6750–6755.

[37] Griffin, W. S., Yeralan, O., Sheng, J. G., Boop, F. A., Mrak, R. E., Rovnaghi, C. R., Burnett, B. A., Feoktistova, A., and Van Eldik, L. J. Overexpression of the neurotrophic cytokine S100 beta in human temporal lobe epilepsy. *J. Neurochem. 65*, 1 (1995), 228–233.

[38] Guilhem, D., Dreyfus, P. A., Makiura, Y., Suzuki, F., and Onteniente, B. Short increase of BDNF messenger RNA triggers kainic acid-induced neuronal hypertrophy in adult mice. *Neuroscience 72*, 4 (1996), 923–931.

[39] Hamilton, J. D. *Time series analysis.* Princeton U. Press, Princeton, NJ, 1994.

[40] Harris, S. Unpublished data, March 1997.

[41] Hertz, J. Statistical issues in reverse engineering of genetic networks. http://www.nordita.dk/~hertz/papers/dgshort.ps.gz, 1998. Poster for Pacific Symposium on Biocomputing.

[42] Hieter, P., and Boguski, M. Functional genomics: it's all how you read it. *Science 278* (1997), 601–602.

[43] Holstege, F. C., Jennings, E. G., Wyrick, J. J., Lee, T. I., Hengartner, C. J., Green, M. R., Golub, T. R., Lander, E. S., and Young, R. A. Genome-wide expression page. Avaliable online at http://web.wi.mit.edu/young/expression/.

[44] Holstege, F. C., Jennings, E. G., Wyrick, J. J., Lee, T. I., Hengartner, C. J., Green, M. R., Golub, T. R., Lander, E. S., and Young, R. A. Dissecting the regulatory circuitry of a eukaryotic genome. *Cell 95*, 5 (November 1998), 717–728.

[45] Hsueh, Y. P., Wang, T. F., Yang, F. C., and Sheng, M. Nuclear translocation and transcription regulation by the membrane-associated guanylate kinase CASK/LIN-2. *Nature 404*, 6775 (2000), 298–302.

[46] Hu, J., and Van Eldik, L. J. S100 beta induces apoptotic cell death in cultured astrocytes via a nitric oxide-dependent pathway. *Biochim. Biophys. Acta 1313*, 3 (1996), 239–245.

[47] Hughes, T. R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton, R., Armour, C. D., Bennett, H. A., Coffey, E., Dai, H., He, Y. D., Kidd, M. J., King, A. M., Meyer, M. R., Slade, D., Lum, P. Y., Stepaniants, S. B., Shoemaker, D. D., Gachotte, D., Chakraburtty, K., Simon, J., Bard, M., and Friend, S. H. Functional discovery via a compendium of expression profiles. *Cell 102*, 1 (2000), 109–126.

[48] Johanning, H., Plenge, P., and Mellerup, E. Serotonin receptors in the brain of rats treated chronically with imipramine or RU24969: support for the 5-HT1B receptor being a 5-HT autoreceptor. *Pharmacol. Toxicol. 70*, 2 (February 1992), 131–134.

[49] Kar, S., Seto, D., Dore, S., Chabot, J. G., and Quirion, R. Systemic administration of kainic acid induces selective time dependent decrease in [125I]insulin-like growth factor I, [125I]insulin-like growth factor II and [125I]insulin receptor binding sites in adult rat hippocampal formation. *Neuroscience 80*, 4 (1997), 1041–55.

[50] Kauffman, S. A. *The Origins of Order: Self-Organization and Selection in Evolution.* Oxford University Press, New York, 1993.

[51] Kenigsberg, R. L., Hong, Y., and Théorêt, Y. Cholinergic cell expression in the developing rat medial septal nucleus in vitro is differentially controlled by GABAA and GABAB receptors. *Brain Res. 805*, 1–2 (1998), 123–130.

[52] Khan, Z. U., Gutierrez, A., Mehta, A. K., Miralles, C. P., and De Blas, A. L. The alpha 4 subunit of the GABAA receptors from rat brain and retina. *Neuropharmacology 35*, 9–10 (1996), 1315–1322.

[53] Khan, Z. U., Gutierrez, A., Miralles, C. P., and De Blas, A. L. The gamma subunits of the native GABAA/benzodiazepine receptors. *Neurochem. Res. 21*, 2 (February 1996), 147–159.

[54] Koiran, P., and Sontag, E. D. Vapnik-Chervonenkis dimension of recurrent neural networks. *Discrete Applied Math 86* (1998), 63–79.

[55] Lee, M.-L. T., Kuo, F. C., Whitmore, G. A., and Sklar, J. Importance of replication in microarray gene expression studies: Statistical methods and evidence from repetitive cDNA hybridizations. *Proc. Natl. Acad. Sci. USA 97*, 18 (2000), 9834–9839.

[56] Liang, S., Fuhrman, S., and Somogyi, R. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In Altman et al. [5], pp. 18–29.

[57] Littlestone, N. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning 2* (1988), 285–318.

[58] Littlestone, N. Redundant noisy attributes, attribute errors, and linear-threshold learning using Winnow. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory* (University of California, Santa Cruz, 1991), ACM Press, pp. 147–156.

[59] Littlestone, N., and Warmuth, M. K. The weighted majority algorithm. *Information and Computation 108*, 2 (February 1994), 212–261.

[60] Liu, J. P., and Sternberg, P. S-100 beta and insulin-like growth factor-II differentially regulate growth of developing serotonin and dopamine neurons in vitro. *J. Neurosci. Res. 33*, 2 (October 1992), 248–256.

[61] Ma, W., and Barker, J. L. GABA, GAD, and GABA(A) receptor alpha4, beta1, and gamma1 subunits are expressed in the late embryonic and early postnatal neocortical germinal matrix and coincide with gliogenesis. *Microsc. Res. Tech. 40*, 5 (1998), 398–407.

[62] Marnellos, G., and Mjolsness, E. A gene network approach to modeling early neurogenesis in Drosophila. In Altman et al. [5], pp. 30–41.

[63] Matsuno, H., Doi, A., Nagasaki, M., and Miyano, S. Hybrid Petri net representation of genetic regulatory network. In *Pacific Symposium on Biocomputing '00* (Singapore, 2000), R. B. Altman, K. Lauderdale, A. K. Dunker, L. Hunter, and T. E. Klein, Eds., World Scientific Publishing Co., pp. 338–349.

[64] MATTSON, M. P., AND RYCHLIK, B. Glia protect hippocampal neurons against excitatory amino acid-induced degeneration: involvement of fibroblast growth factor. *Int. J. Dev. Neurosci. 8*, 4 (1990), 399–415.

[65] MCCULLAGH, P., AND NELDER, J. A. *Generalized Linear Models*, 2nd ed. Chapman & Hall, London, UK, 1989.

[66] MJOLSNESS, E., SHARP, D. H., AND REINITZ, J. A connectionist model of development. *J. Theor. Biol. 152*, 4 (1991), 429–454.

[67] MURPHY, K., AND MIAN, S. Modeling gene expression data using Dynamic Bayesian Networks. Tech. rep., University of California, Berkeley, 1999. http://www.cs.berkeley.edu/~murphyk/Papers/ismb99.ps.gz.

[68] NAKAYAMA, M., GAHARA, Y., KITAMURA, T., AND OHARA, O. Distinctive four promoters collectively direct expression of brain-derived neurotrophic factor gene. *Brain. Res. Mol. Brain. Res. 21*, 3–4 (1994), 206–218.

[69] OLSEN, R. W., AND DELOREY, T. M. GABA and glycine. In *Basic neurochemistry: molecular, cellular and medical aspects*, G. J. Siegel, B. W. Agranoff, R. W. Albers, S. K. Fisher, and M. D. Uhler, Eds., 6th ed. Lippincott-Raven Publishers, Philadelphia, 1999, pp. 335–346.

[70] PLEUS, R. C., AND BYLUND, D. B. Desensitization and down-regulation of the 5-hydroxytryptamine1B receptor in the opossum kidney cell line. *J. Pharmacol. Exp. Ther. 261*, 1 (April 1992), 271–277.

[71] PTASHNE, M. *A genetic switch*, 2nd ed. Cell Press & Blackwell scientific publications, Cambridge, MA, 1992.

[72] REINITZ, J., AND SHARP, D. H. Mechanism of *eve* stripe formation. *Mechanisms of Development 49* (1995), 133–158.

[73] RUDGE, J. S., MATHER, P. E., PASNIKOWSKI, E. M., CAI, N., CORCORAN, T., ACHESON, A., ANDERSON, K., LINDSAY, R. M., AND WIEGAND, S. J. Endogenous BDNF protein is increased in adult rat hippocampus after a kainic acid induced excitotoxic insult but exogenous BDNF is not neuroprotective. *Exp. Neurol. 149*, 2 (February 1998), 398–410.

[74] SANDER, T., BOCKENKAMP, B., HILDMANN, T., BLASCZYK, R., KRETZ, R., WIENKER, T. F., SCHMITZ, B., BECK-MANNAGETTA, G., RIESS, O., EPPLEN, J. T., JANZ, D., AND ZIEGLER, A. Refined mapping of the epilepsy susceptibility locus EJM1 on chromosome 6. *Neurology 49*, 3 (1997), 842–847.

[75] SAVAGEAU, M. A. Power-law formalism: a canonical nonlinear approach to modeling and analysis. In *Proceedings of the World Congress of Nonlinear Analysts '92* (1995), V. Lakshmikantham, Ed., vol. 4, pp. 3323–3334.

[76] SAVAGEAU, M. A. Rules for the evolution of gene circuitry. In Altman et al. [5], pp. 54–65.

[77] SCHULTZ, C., AND TAUTZ, D. Autonomous concentration-dependent activation and repression of kruppel by hunchback in the drosophila embryo. *Development 120*, 10 (October 1994), 3043–3049.

[78] SCHWARTZER, C., AND SPERK, G. Hippocampal granule cells express glutamic acid decarboxylase-67 after limbic seizures in the rat. *Neuroscience 69*, 3 (December 1995), 705–709.

[79] SHANNON, C. E. A mathematical theory of communication. *Bell Sys. Tech. J. 27* (1948), 379–423,623–656.

[80] SOMOGYI, R. Unpublished data, January 1998.

[81] SOMOGYI, R., FUHRMAN, S., ASKENAZI, M., AND WUENSCHE, A. The gene expression matrix: towards the extraction of genetic network architectures. In *Proceedings of the Second World Congress of Nonlinear Analysts (WCNA96)* (1996), vol. 30 of *Nonlinear Analysis*, Pergamon Press.

[82] SOMOGYI, R., WEN, X., MA, W., AND BARKER, J. L. Developmental kinetics of GAD family mRNAs parallel neurogenesis in the rat spinal cord. *J. Neurosci. 15*, 4 (April 1995), 2575–2591.

[83] SONTAG, E. D. Shattering all sets of k points in general position requires (k-1)/2 parameters. *Neural Computation 9* (1997), 337–348.

[84] STRANG, G. *Linear Algebra and Its Applications*, 3rd ed. Harcourt College Publishers, San Diego, CA, 1988.

[85] SZABO, G., KATAROVA, Z., AND GREENSPAN, R. Distinct protein forms are produced from alternatively spliced bicistronic glutamic acid decarboxylase mRNAs during development. *Mol. Biol. Cell 14*, 11 (November 1994), 7535–7545.

[86] SZALLASI, Z. Genetic network analysis in light of massively parallel biological data acquisition. In Altman et al. [6], pp. 5–16.

[87] TANDON, P., YANG, Y., DAS, K., HOLMES, G. L., AND STAFSTROM, C. E. Neuroprotective effects of brain-derived neurotrophic factor in seizures during development. *Neuroscience 91*, 1 (1999), 293–303.

[88] TAO, R., MA, Z., AND AUERBACH, S. B. Influence of AMPA/kainate receptors on extracellular 5-hydroxytryptamine in rat midbrain raphe and forebrain. *Br. J. Pharmacol. 121*, 8 (August 1997), 1707–1715.

[89] THIEFFRY, D., AND THOMAS, R. Qualitative analysis of gene networks. In Altman et al. [5], pp. 77–88.

[90] THOMAS, R. Regulatory networks seen as asynchronous automata: a logical description. *J. Theor. Biol. 153* (1991), 1–23.

[91] TIBSHIRANI, T. J. H. R. J. *Generalized Additive Models.* Chapman $ Hall, London, UK, 1990.

[92] VELCULESCU, V. E., ZHANG, L., ZHOU, W., VOGELSTEIN, J., BASRAI, M. A., BASSETT JR, D. E., HIETER, P., VOGELSTEIN, B., AND KINZLER, K. W. Characterization of the yeast transcriptome. *Cell 88* (1997), 243–251.

[93] WAHDE, M., AND HERTZ, J. Course-grained reverse engineering of genetic regulatory networks. *Biosystems 55*, 1–3 (2000), 129–136.

[94] WEAVER, D. C., WORKMAN, C. T., AND STORMO, G. D. Modeling regulatory networks with weight matrices. In Altman et al. [6], pp. 112–123.

[95] WEN, X. personal communication, March 1998.

[96] WEN, X., FUHRMAN, S., MICHAELS, G. S., CARR, D. B., SMITH, S., BARKER, J. L., AND SOMOGYI, R. Large-scale temporal gene expression mapping of CNS development. *Proc. Natl. Acad. Sci. USA 95*, 1 (1998), 334–339.

[97] YANG, H. Y., LIESKA, N., KRIHO, V., WU, C. M., AND PAPPAS, G. D. A subpopulation of reactive astrocytes at the immediate site of cerebral cortical injury. *Exp. Neurol. 146*, 1 (July 1997), 199–205.

[98] YIN, H. S., CHOU, H. C., AND CHIU, M. M. Changes in the microtubule proteins in the developing and transected spinal cords of the bullfrog tadpole: induction of microtubule-associated protein 2c and enhanced levels of Tau and tubulin in regenerating central axons. *Neuroscience 67*, 3 (August 1995), 763–775.