1 Introduction

In pharmacogenomics and related areas, a lot of research is directed towards discovering, understanding and/or controlling the outcome of some particular biological pathway. Numerous examples exist where the manipulation of a key enzyme in such a pathway did not lead to the desired effect [5]. This usually happens because the intended effect was compensated for by the genetic regulation of enzyme levels. Such examples illustrate the importance of accounting for genetic regulation.

We know that the structure of complex genetic and biochemical networks lies hidden in the sequence information of our DNA but it is far from trivial to predict gene expression from the sequence code alone. The current availability of microarray measurements of thousands of gene expression levels during the course of an experiment or after the knockout of a gene provides a wealth of complementary information that may be exploited to unravel the complex interplay between genes. It now becomes possible to start answering some of the truly challenging questions in systems biology. For example, is it possible to model these genetic interactions as a large network of interacting elements and can these interactions be effectively learned from measured expression data?

Since Kauffman [21] introduced the concept of mathematical modeling of complex systems, the reverse engineering of genetic networks has triggered the imagination of many molecular biologists. Somogyi [31] also investigated some of the properties of Boolean networks in relation to biological systems. These researchers showed that Boolean networks possess properties like global complex behavior, self-organization, stability, redundancy and periodicity. Analogies between basins of attraction and different tissue types, as well as cyclic attractors and cell cycles have also been discussed by many other researchers.

The inference of genetic interactions from measured expression data is one of the most challenging tasks of modern functional genomics. When successful, the learned network of regulatory interactions yields a wealth of useful information. An inferred genetic network contains information about the pathway to which a gene belongs and which genes it interacts with. Furthermore, it explains the genes function in terms of how it influences other genes and indicates which genes are pathway initiators and therefore potential drug targets.

Obviously, such wealth comes at a price and that of genetic network modeling is that it is an extremely complex task. Although the behavior and properties of artificial networks match the observations made in real biological systems well, the field of genetic network modeling has yet to reach its full maturity. The automatic discovery

\[\text{For reasons of brevity, the authors consistently refer only to the first author of each reference.}\]
of genetic networks from expression data alone is far from trivial because of the combina-
torial nature of the problem and the poor information content of the data. First, to
model genetic regulation, one needs to take into account the fact that gene expression
levels are regulated by the combined action of multiple gene products [17]. Second,
the number of measurements (arrays) is relatively small compared to the number of
measured objects (genes) and the data are corrupted with a substantial amount of
measurement noise. Together, these two complicating factors make the construction of
genetic networks from empirical observations extremely difficult. In addition, results
are further complicated by the presence of inherent noise caused by, for example, vari-
ations between different individuals, small numbers of molecules available in a given
cell, variations between tissues in a given individual, variations caused by effects that
are not measured etc.

The dimensionality problem (many objects and few measurements) plays a funda-
mental role in genetic network modeling causing the straightforward estimation of
model parameters to become extremely unreliable (many equally good solutions). The
common approach to avoid this problem is to either reduce the models complexity or
to apply constraints on the parameters. Consequently, the relatively young field of ge-
netic network modeling has been governed by the introduction of a plethora of different
models and learning strategies.

This abstract provides an small overview of genetic network modeling approaches
that employ expression data to automatically discover genetic interactions. Reviews
on genetic network models have also appeared recently. In a recent review [42], models
are placed in an historical context and the qualitative properties of the models and
their learning strategies are compared. De Jong [10] focuses in his review more on
the mathematical properties of the models. An experimental comparison of a limited
number of genetic network models is presented in [48, 39].

2 Reverse Engineering of Genetic Network Models

The introduction of microarray technology made it possible to measure the gene-
expression levels of thousands of genes simultaneously. This introduced a new impulse
to genetic network modeling, namely the reverse engineering of large-scale genetic
networks based on measured expression data. Starting from microarray data and a
general model of genetic interactions, the parameters of this general network model are
learned from the data. Here we will describe only dynamical models, i.e., models that
are learned on time course gene expression data.

2.1 Boolean networks

In 1998, Liang [24] started off by introducing REVEAL, an algorithm that automati-
cally constructs a large-scale Boolean network from data. In a general Boolean network
model, all gene expression levels are discretized into binary expression levels; a gene is
either on or off. The binary expression levels of all genes in the system at a certain
point in time define the state of the network at that time instant. A state transition
table defines, for each possible network state, which network state will be next (see
a) Network:

b) State-transition table:

<table>
<thead>
<tr>
<th>$g_1(t)$</th>
<th>$g_2(t)$</th>
<th>$g_3(t)$</th>
<th>$g_1(t+1)$</th>
<th>$g_2(t+1)$</th>
<th>$g_3(t+1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 1: Example of: a) a Boolean network of three genes with corresponding b) state-transition table and c) Boolean rules.

Figure 1b). From this table, a Boolean rule can be determined for each gene that describes how its expression level at the next time instant depends on some combination of the gene expression levels at the current time instant.

Typical Boolean rules contain logical operators such as AND, OR and NOT (see Figure 1c). By placing connections between each of the input genes in the rule and the output gene, the structure of the network can be determined, which expresses the interactions among all genes (see Figure 1a). A typical gene expression dataset, after discretization, represents an incomplete state-transition table, since not all possible states will have been measured.

REVEAL constructs the rule for a target gene from this incomplete table by considering the mutual information between the input states of each single gene ($k = 1$) and the output state of the target gene. If the output can be perfectly determined by one of the inputs, the corresponding rules and connections are extracted. If not, all combinations of two genes ($k = 2$) are considered as input and it is examined whether this pair can perfectly predict the target. If not, the procedure repeats for $k = k + 1$ etc. In other words, the structure is learned using a forward exhaustive search procedure that stops as soon as a perfect reconstruction is possible.

A year later, Akutsu [1] proved, using a conceptually simpler approach, that $O(\log_2 N)$ random measurements are sufficient to identify a network of $N$ genes with bounded connectivity $K$ but this algorithm takes $O(NK + 1Q)$ time, with $Q$ the number of state transitions. This implies that for a typical gene expression dataset with 1000 genes and connectivity $K = 2$, in the order of 10 independent measurements are sufficient but in that case $O(10^4) \times (10^4) = O(10^8)$ time is required! The algorithm learns a Boolean model by performing an exhaustive search not only for each possible combination of inputs but also for each possible configuration of Boolean functions (using only AND or and NOT operators) that are consistent with the given state transitions. Unfortunately, this algorithm was not suited for noisy conditions but a year later Akutsu presented an algorithm that is robust to noise [2, 3].
2.2 Continuous models

Although Boolean networks provide a good starting point, they are generally criticized because only two discrete expression levels are allowed. Many examples exist where genes are regulated in a continuous manner rather than just turned on or off [29, 20, 19]. This inspired the introduction of models with a continuous representation of gene expression.

DHæseleer [12] learned a linear model on data from the rat central nervous system (CNS), during development and injury after kainate injection [47]. He coupled two partly overlapping datasets, to utilize as much information as possible, resulting in a dataset of 65 genes and 28 time points. Even this simple linear model (with a single parameter per gene) contains more parameters than the number of measurements. This so-called dimensionality problem makes it possible to find many parameter sets that perfectly reconstruct the data. As a result, the parameter estimations become unreliable. To accommodate the fact that the datasets were differently sampled, DHæseleer employed a nonlinear interpolation method (resulting in 68 time points). By employing a nonlinear interpolation scheme, he enforces smoothness and tries to avoid the dimensionality problem.

Weaver [46] also employed the linear model but augmented it with a biologically inspired, non-linear doseresponse curve. Although nonlinear, this model is essentially a recurrent neural network without a hidden layer. By de-squashing the doseresponse curve, the model can be solved by simple linear algebra. To handle the dimensionality problem, Weaver proposed the use of the Moore-Penrose pseudo-inverse. This special matrix inverse produces a solution for under-determined problems that minimizes the sum of the squared weights but still perfectly fits the data. To introduce limited connectivity, he proposed a greedy backward search that iteratively sets the smallest weight to zero and then recomputes the pseudo-inverse on the now slightly less under-determined problem. Unfortunately, the de-squashing step is quite sensitive to small changes in the data. Rather than a discrete-time model, Wahde [43] employed a continuous-time recurrent neural network. A genetic algorithm (GA) was employed to find the parameters of small networks (four genes) learned on the average profiles of clustered data. A genetic algorithm [26] is an optimization technique based on natural selection in which a set of possible solutions, called a population, is evaluated in parallel. New populations of potentially better solutions are generated and evaluated by combining (crossover) and modifying (mutation) the best solutions in the current population. After learning the parameters with a GA, a qualitative description of the parameters is given. Wahde showed results on artificial data as well as on the CNS dataset presented by Wen [47]. Using artificial data he showed that it is better to have multiple shorter time series than one long series. In later work [44, 45], he suggested a procedure that forced parameters that were not significant to zero. Repeated elimination of the most unreliable parameters can also be viewed as a form of backward search.

Chen [9] proposed an even more realistic model based on a system of differential equations that models both mRNA and protein levels, including degradation. Chen showed that, provided that both mRNA and protein levels are given, solving this model is similar to the problem of finding minimum weight solutions to linear equations.
Unfortunately, this problem is known to be NP complete. However, for a constant connectivity, K, the problem can be solved in $O(QNK+1)$ time (using a dataset of N genes and Q time points) by just checking all NK possible structures. Chen also reasoned that, as many genes showed periodic expression, the Fourier transform for stable systems (FTSS) might be employed as an alternative approach.

A year earlier, Spirov [32] had also suggested the use of a system of differential equations but for a smaller network and with more data points. For learning the parameters, he suggested first using a genetic algorithm to come up with an initial population of globally optimal solutions, which is then used as seeds for a parallel simulated annealing (SA) search. Simulated annealing is a sequential optimization technique that is based on evaluating random changes to the current solution. Better solutions are always accepted, whereas worse solutions are accepted with a probability that decreases during optimization. As a result, SA moves consistently to better solutions but is able to jump out of local optima. When these runs have almost converged, a local gradient descent (GD) approach is employed.

2.3 Modeling concerns

Apart from many papers that introduced a new reverse engineering approach based on yet another model, gradually more papers emerged that addressed the issues associated with genetic network modeling itself. With the reductionists approach, the combinatorial nature of genetic regulation had largely been ignored [35]. Therefore, it took some time before researchers realized the immense complexity that learning genetic networks from expression data involved and the early enthusiasm subdued. Szallasi [35] claimed that there are four factors inherent in biological systems that influence the reverse engineering of genetic networks from expression data. First, the nature of genetic networks is undoubtedly stochastic but microarray measurements are population averaged, which may mask the real individual regulatory interactions. Also, a faster sampling rate is not always possible because the measurement error determines a lower bound on the sampling interval, i.e., the expected difference in expression within one sampling interval should be larger than the measurement noise. Secondly, there are also many regulatory factors that are not modeled, such as (de-)stabilization of mRNA, translocation, phosphorylization etc. Thirdly, he reasons that the information content of the data is not as large as its size would suggest (12 orders of magnitude smaller), as only a few genes cycle and even fewer show frequent changes during cell cycle. On the other hand, a property that is favorable for network analysis is that networks are believed to exhibit a high level of compartmentalization.

Spirtes [33] also discussed some of the complicating issues of data acquisition in relation to construction of genetic networks. Apart from the above-mentioned issues of small sample sizes (dimensionality problem), the substantial measurement error and the masking effect of population averaged measurements, he also points to the fact that the final results can be influenced by hidden (e.g., not modeled) effects and the loss of synchronization of cells.

Erb [14] experimentally examined the influence of measurement noise. He performed Khalil’s sensitivity analysis on a complex non-linear model proposed by Mjolsness [13], employing a fully connected network of only three genes. Already with such
a small network, the parameters turned out to be very sensitive to noise in the data.

A comparative study done by Wessels [48, 39] proposed a set of mathematical properties that genetic network models should possess and by means of which they can be compared. In a small experimental study of continuous models, in which the models were learned on data generated by the other models, he reported disappointing results in terms of how well models can reveal the underlying interactions when faced with noise and limited data. The results favor simple, i.e., linear or pair-wise, models that are less sensitive to unfavorable data conditions.

2.4 Pairwise models

One way to overcome the dimensionality problem is to restrict the complexity of the model, for example, by only considering pair-wise relationships. Arkin [4] was the first to suggest the construction of biochemical pathways by means of timeshifted pair-wise correlations. First, the position and magnitude at which the maximal timeshifted cross-correlation occurs is computed in a pair-wise fashion. From this, a distance measure is constructed and single linkage hierarchical clustering is employed, resulting in a singly linked tree that connects associated genes. Augmented with directional and time-lag information this association diagram reveals temporal interactions. Arkin suggested that his approach could also be used to learn genetic networks.

Later, Chen [8] proposed a similar scheme, based on matching peaks in the signals rather than using correlation. After thresholding and clustering, the remaining profiles are represented as a set of peaks. Then peaks in the profiles are compared in a pair-wise fashion to determine the causal activation scores. Similarly, inhibition scores are determined. From these scores a putative regulation network is constructed using simulated annealing.

Woolf [50] was the first to describe a fuzzy model for learning genetic interactions. He searched for all possible triplets of an activator and a repressor (two inputs) that influence a target gene (one output). All triplets are scored and ordered on how well they fit the expression data and on whether the inputs showed enough variation. Unfortunately, these pair-wise (triple-wise) models are fundamentally limited to considering only singly (doubly) connected networks.

2.5 Qualitative models

A different way to cope with the limitations of the data is to learn qualitative models, thus avoiding the necessity to estimate model parameters precisely. Akutsu [2, 3] described a collection of algorithms that are an intermediate solution, somewhere between Boolean models and continuous differential models. These qualitative models are based on linear differential equations but instead of trying to learn the exact parameters, the researcher derives qualitative abstractions of the parameters. For instance, it is only relevant whether the differences are positive, negative or zero. In this case, a solution can be found by solving a set of inequality relations. Provided that a lot of data are available, these inequalities can be solved using linear programming (LP).

\[\text{The GA of the Wahde model converged slowly and was therefore stopped early, not allowing the model to converge completely.}\]
Alternatively, the parameters of a non-linear S-system (power-law) can be found using linear algebra by taking the logarithm on both sides of the equations. An S-system is a set of non-linear differential equations of a special form belonging to the power-law formalism (products of exponentially weighted inputs). If the logarithm is taken, the obtained parameter values only portray a relative meaning. But this was exactly the goal: to obtain a qualitative description.

Because of the multitude of detailed biological information acquired over the years, a qualitative model provides an excellent tool to describe the working hypothesis of researchers. Shrager [30] proposed an automatic scheme to revise an initial qualitative model such that it better matches the expression data. This scheme is based on comparing the expected pair-wise correlations of all pairs in the initial scheme with the correlations in the expression data. This measure of data fit is used to construct a fitness function, which is augmented with terms to reduce the number of variables and links in the model. With this fitness function a simple greedy search is performed based on considering single changes in the model. Unfortunately, the employed pairwise correlation measure does not fully capture the combinatorial nature of the qualitative model.

2.6 Modeling revisited

A better understanding of the consequences of the dimensionality problem resulted in modeling approaches that were better adapted to handle the limitations of the data. For example, strategies started to focus on first reducing the problem (e.g., taking a smaller network, using clustering or structure determination) such that the resulting parameters are estimated more reliably. As a result, the boundaries between the analytic and synthetic approaches gradually became blurred.

Van Someren suggested a number of general approaches to reduce the dimensionality problem by incorporating biologically motivated constraints and showed results from artificial data generated with linear networks. The reduction of the number of genes by clustering gene expression profiles was considered by many [12, 43, 44, 45, 32, 8, 38, 27, 11]. However, Van Someren [38] studied the relationship between clustering and its effect on the dimensionality problem when learning linear genetic network models. In [40, 42], he showed that genetic network models could be made robust to noise by minimizing the first-order derivative of the models output with respect to its input. For non-linear models, robustness is imposed by learning the model on a set of noisy profiles. To impose limited connectivity of the models, Van Someren [41] compared a number of search algorithms that search for structures with limited connectivity. In this comparison, a forward beam search approach proved to be the best. Mjolsness [27] also suggested the use of clustered data and learned a system of non-linear differential equations using simulated annealing. Apart from minimizing the prediction error, he included a weight-decay term to minimize the weight values and an exponential term that keeps the parameters bounded in the cost function. Koza [22] employed genetic programming to determine the structure and rate constants of small metabolic pathways. He showed that it was possible to automatically create a metabolic pathway involved in the phospholipid cycle using 270 time points of E-CELL simulations of a 4-enzyme network where all enzymes were perturbed. Unfortunately, a large amount of
data were required. Maki [25] proposed a two-step approach in which first the structure of a pair-wise Boolean network is learned from the steady-state expressions obtained after perturbation of each gene in the network. The resulting network structure is used to define smaller networks modeled by S-systems. The parameters of these systems are then learned using a GA applied on dynamic data. Unfortunately, this approach still needs a lot of measurements, i.e., at least perturbation experiments of all genes.

2.7 Trend towards Integrated Approaches

Ideker [18] presented a fully integrated approach on large-scale data in which four main steps were taken:

- define an initial model of a pathway
- perturb components in the pathway and measure the responses in mRNA and protein levels
- check the responses with the model
- refine the model to explain the unpredicted responses

He was the first to present mRNA expression data (microarrays) as well as protein abundance data, using isotope-coded affinity tag (ICAT) reagents and tandem mass spectrometry (MS/MS) and to integrate this with information from databases of known physical interactions of the galactose pathway.

Clearly the integration of different information sources is playing an essential part in modern approaches towards genetic network modeling. The modeling trend that is revealed by this quick review is the use of a larger variety of information for learning genetic network models, be it in terms of other types of measurements, information stored in databases or desired properties of networks. Therefore, we might expect that, in the near future, results from pathway scoring [16, 23, 28, 34, 49] and promoter analysis [7, 6, 36, 37, 15] approaches will become integrated within the learning algorithms of genetic network models. Advancements like these, will unlock and exploit the full potential that genetic network modeling has to offer.

References


