Coarse-grained reverse engineering of genetic regulatory networks

Mattias Wahde and John Hertz

{wahde, hertz}@nordita.dk

Abstract

We have modeled genetic regulatory networks in the framework of continuous-time recurrent neural networks. A method for determining the parameters of such networks, given expression level time series data, is introduced and evaluated using artificial data. The method is also applied to a set of actual expression data from the development of rat central nervous system.

1 Introduction

In recent years, the available amount of gene expression data has been increasing at a rapid rate, and there now exist large data sets with expression level measurements from different tissue types and different organisms. The existence of time series data, i.e. measurements of the time variation of the expression levels of a number of genes, raises the possibility of determining the regulatory interactions between genes. Recently, there have been many efforts to model genetic regulatory networks and to determine the interactions within them, using several different mathematical frameworks (see e.g. Kauffman 1993, Reinitz & Sharp 1995, Somogyi & Sniegoski 1996, and D'Haeseleer et al. 1999). However, usually the data available are not sufficient to determine accurately the interactions between all the genes in a given data set. Therefore it is essential to be able to construct a coarse-grained description of the system, using suitable macroscopic variables. Here we employ the approach of Wen et al. (1998), who clustered the genes into a small number of groups according to their temporal expression patterns, but our method can also be applied to other coarse-graining schemes.

In this contribution we model regulatory networks as continuous-time recurrent neural networks. Our model is introduced in Sect. 2, and the method is described

in Sect. 3. In Sects. 4 and 5, we present the results of applying our method to artificial and real data, respectively. Our conclusions, together with some directions for further work, are given in Sect. 6.

2 The Model

We have chosen to model genetic networks using a set of coupled non-linear differential equations. The use of continuous-valued functions enabled us to model the dynamics at intermediate expression levels, rather than only at the extreme on (1) and off (0) levels. Also, the use of differential equations rather than difference equations allowed us to introduce explicit rate constants. Specifically, we have modeled the networks by equations corresponding to continuous-time recurrent neural networks (Reinitz and Sharp, 1995)

$$\tau_i \dot{x_i} + x_i = g\left(b_i + \sum_j w_{ij} x_j\right), i = 1, \ldots, N, \tag{1}$$

where τ_i^{-1} are rate constants, x_i the expression levels, and $\dot{x_i}$ their derivatives with respect to time. For a set of N genes, there are $N \times N$ weights (w_{ij}), N bias terms (b_i), and N time constants τ_i, i.e. a total of $N(N+2)$ parameters to be determined. In principle, one could go beyond the linear additive summation in Eq. (1), and include higher order terms in a network of the form

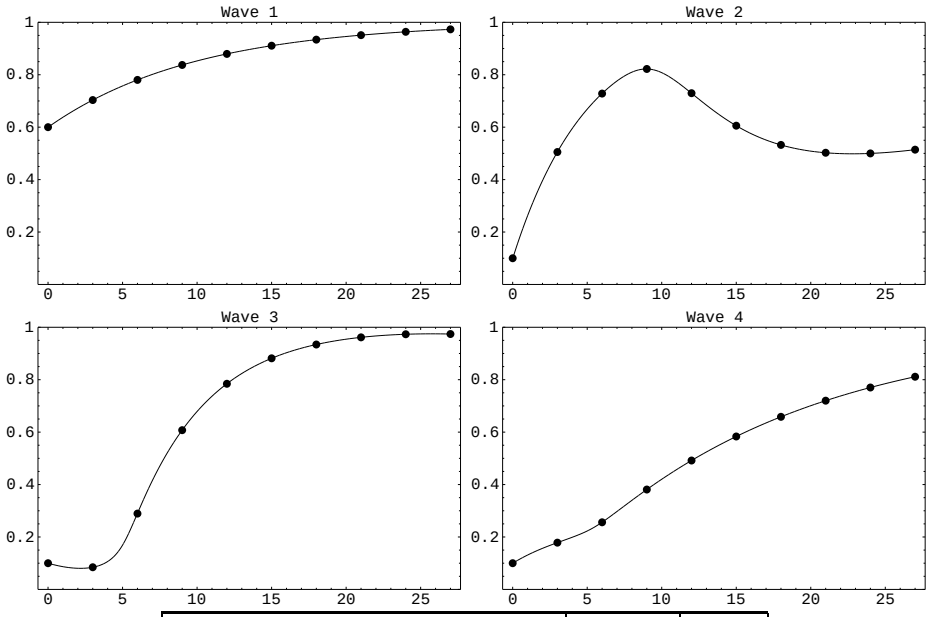
$$\tau_i \dot{x_i} + x_i = g\left(b_i + \sum_j w_{ij} x_j + \sum_{jk} \nu_{ijk} x_j x_k + \ldots\right). \tag{2}$$

Such terms can implement the complex computational logic carried out by actual regulatory elements (Yuh et al., 1998). However, the determination of the parameters of a network incorporating higher-order terms would require much more data than is generally available for these systems. Thus, we have limited ourselves to models of the form (1).

The non-linear activation function g can be chosen in several different ways. We have chosen to use

$$g(z) = \frac{1}{1 + \mathrm{e}^{-kz}}. \tag{3}$$

The choice of the numerical value of k is not very important, since the weights and bias terms always can be rescaled to give the same dynamics for different values of k. We have used $k = 1$.



w_{ij}				b_i	τ_i
20.0	5.0	0.0	0.0	0.0	10.0
25.0	-5.0	-17.0	0.0	-5.0	5.0
0.0	10.0	20.0	-20.0	-5.0	5.0
0.0	0.0	10.0	-5.0	0.0	15.0

Figure 1: The table shows an example of an artificial network, and the upper panels show the time variation of the expression levels from a given starting configuration. The dots represent data points that can be read off from the curves.

3 Method for Reverse Engineering

There are a number of methods available for determining the weights of a network described by Eq. (1), such as, for example, gradient descent methods (e.g. back-propagation), and simulated annealing. In this investigation, we have chosen to use a genetic algorithm (hereafter GA) to determine the unknown parameters of the network. While evolutionary methods, such as genetic algorithms, have been applied to many different types of problems, they have so far only had very limited use in the field of genetic regulatory networks. We decided to use a GA since, in addition to being able effectively to find the network parameters, it has the additional advantage of flexibility. For instance, it is easy to modify the GA so that it can adaptively vary the connectivity of the network during a run. While such modifications will not be used in this contribution, they may prove to be important for future research

involving larger data sets.

3.1 Genetic Algorithms

In this section, a very brief description of GAs will be given. For a more complete description, see e.g.Holland 1975, Mitchell 1996, or Davis 1991.

In a GA, the unknown parameters of the problem are encoded in strings of digits referred to as *chromosomes*. Initially, a *population* of *individuals*, each associated with one such string, is generated by assigning random values to all the locations (called *genes*) along the strings. Then, for each individual, the variables are read off from the chromosome, the relevant computation is carried out, and the *fitness* of the individual is evaluated.

The assignment of fitness values should be such that individuals close to reaching the goal set by the user obtain higher fitnesses than those who are far from the goal. When all the individuals in the first *generation* have been evaluated, the second generation is formed by first selecting *parent strings* in such a way that individuals with high fitness have a greater probability of being selected than those with low fitness, and then combining the genetic material contained in their chromosomes to form new strings, and, finally, allowing a small degree of *mutation* (i.e. random variation) of the newly formed chromosomes. The chromosomes thus formed constitute the second generation, which is evaluated by repeating the procedure used for the evaluation of the first generation. This iteration continues until a satisfactory solution has been found.

3.2 Reverse engineering

In the particular case of genetic regulatory networks of the type described by Eq. (1), the chromosomes encode the $N(N+2)$ parameters w_{ij}, b_i, and τ_i. We generally used population sizes of 10 to 100. Given network parameters and initial values (see below), Eqs. (1) were integrated for each individual. The resulting curves were read off at those discrete times for which data points were available, and the fitness was then computed as

$$f = \frac{1}{1 + \frac{1}{K}\sum_k \delta_k^2}, \tag{4}$$

where k enumerates the K data points used for computing the fitness, and $\delta_k = (x_k - x_k^d)/\sigma$, where x_k is the expression level obtained from the integration, x_k^d is the corresponding data point, and σ is a tolerance parameter, relative to which the magnitude of the error is measured. Note that, in our model, all expression levels are assumed to be scaled so that they take values between 0 and 1. K is equal to

$N(T-1)$, where N is the number of genes, and T is the length of the time series, the first point of which provides, for each gene, the initial value needed for the integration.

We used two different termination criteria for the GA. The first criterion stops a run when the fitness reaches a given level, whereas the second criterion stops a run only when all expression levels are within a distance σ from the data points. Thus, the first criterion only amounts to a requirement that the *average* deviation should be small, whereas the second requires a good fit of *all* points.

For a given fitness f there are usually many parameter sets than can fit the data, i.e. the fitness peak is not very sharp. Our procedure, therefore, was to carry out a number of runs, and then form an average of the parameters obtained. Through the standard deviations, this procedure also gave us an estimate of the error in each parameter.

For a reliable determination of the network parameters, a minimum requirement is that the number of useful data points should exceed the number of parameters: $T-1 > N+2$. Now, the gene expression data series available generally contain measurements for of order 10 different times, so this limits the number of genes to seven or less. Since the data sets generally contain measurements for more than seven genes, some kind of information compression is needed. One way of obtaining this is to group together genes that have similar expression patterns, thus forming *clusters* (or *waves*), reducing the data set from, for example, of order 100 genes to four or five waves. Below we will work with clusters and waves instead of individual genes.

4 Applications to Artificial Data

In this section, the method described above will be evaluated by applying it to artificial data, i.e. data generated by systems for which the weights, biases, and time constants are known. Clearly, it is important to know whether the method is able to establish the parameters of such networks before proceeding to the more complicated case of real data.

The artificial data was generated by providing a set of starting values $\{x_i^0\}$, and then integrating the set of equations (1), using the parameters of the artificial network. The expression levels were then read off at discrete times, so that a set of data points was obtained. For each wave, the initial value x_i^0 was included as the first point in the data set. The choice of the network parameters for the artificial network was of course arbitrary to some extent. However, we tried to select parameters such that the characteristics of the data series generated by the artificial networks

w_{ij}				b_i	τ_i
16 $\pm$11	7.5 $\pm$17	1.8 $\pm$16	12 $\pm$15	2.0 $\pm$5.6	9.5 $\pm$1.1
18 $\pm$7.7	-13 $\pm$5.8	-8.3 $\pm$11	-9.6 $\pm$14	4.8 $\pm$4.3	5.0 $\pm$0.64
-10 $\pm$12	13 $\pm$8.1	14 $\pm$11	9.7 $\pm$15	-3.3 $\pm$5.8	6.3 $\pm$0.97
-0.1 $\pm$16	6.2 $\pm$16	13 $\pm$13	5.0 $\pm$17	-0.88 $\pm$5.4	17 $\pm$1.9

Table 1: Network obtained from a single time series of artificial expression data generated using the network in Fig. 1. The results shown are the average of 50 runs.

resembled the actual data presented in Sect. 5.

Thus, for our artificial networks, if the starting values were chosen such that wave 1 had the highest expression level at the first point in time, the dynamics of the network, as shown in Fig. 1, was such that wave 1 reached high expression levels first, followed by wave 2 etc.

4.1 Single time series

The method of reverse engineering described in the previous section was applied to artificial expression data obtained from the curves in the four upper panels of Fig. 1. Many runs were carried out for each of the two termination criteria defined in Sect. 3. There was no systematic difference in the results obtained from the two criteria, and we chose to use the first criterion.

The results of the computer runs are shown in Table 1. Individual runs were stopped at $f = 0.9$, corresponding to an average error of $\sigma/3$. σ was set to 0.1. As can be seen from the table, the results were of mixed quality: All of the weights that were supposed to be zero were correctly determined in the sense that the interval spanned by the corresponding standard deviation contained the value zero. Furthermore, all of the time constants were close to their correct values. On the other hand, of the ten weights that were supposed to be non-zero, only five were significantly non-zero, whereas the other five were not, and the standard deviation was rather large for most weights.

4.2 Multiple Time Series

In the previous subsection, the data consisted of only one time series for each wave. In such a case, the data points follow a given trajectory in state space, and are not independent of each other. Thus, even if the curves in Fig. 1 had been sampled at twice as many points, the additional information obtained would not have been very large. Therefore, in order to add new information to the data set, it appears better to

take a few measurements along different state space trajectories rather than taking many measurements along one trajectory.

w_{ij}				b_i	τ_i
11 ±11	7.8 ±12	7.5 ±12	0.80 ±13	3.7 ±4.8	11 ±1.2
14 ±7.7	-16 ±7.3	0.45 ±13	-2.4 ±14	1.8 ±4.6	7.5 ±1.8
5.9 ±14	11 ±11	6.1 ±13	3.2 ±14	1.9 ±5.8	9.1 ±0.73
3.0 ±11	1.5 ±13	5.3 ±11	-16 ±7.0	1.6 ±5.7	16 ±5.3

w_{ij}				b_i	τ_i
19 ±3.5	-17 ±6.8	9.9 ±6.6	-2.9 ±12	-3.5 ±3.4	11 ±0.28
19 ±4.0	-16 ±3.8	-0.10 ±12	0.08 ±15	-4.7 ±2.6	5.8 ±0.89
-0.89 ±4.0	-15 ±7.1	14 ±5.5	7.8 ±12	2.3 ±3.2	7.2 ±0.64
3.5 ±7.3	-5.9 ±8.9	15 ±4.6	-21 ±4.4	0.14 ±2.8	6.4 ±2.7

w_{ij}				b_i	τ_i
22 ±2.2	-21 ±4.3	-1.1 ±5.5	2.3 ±7.5	1.2 ±1.8	11 ±0.47
19 ±4.2	-15 ±4.7	-1.3 ±8.1	3.2 ±10	-5.9 ±2.0	6.3 ±0.82
-2.3 ±3.0	-14 ±6.5	13 ±6.5	9.8 ±8.7	1.9 ±2.7	6.3 ±0.69
-0.22 ±7.0	-1.9 ±7.6	16 ±3.9	-22 ±3.9	-0.35 ±2.7	5.8 ±3.2

w_{ij}				b_i	τ_i
20	-20	0.0	0.0	0.0	10
15	-10	0.0	0.0	-5.0	5.0
0.0	-8.0	12	0.0	0.0	5.0
0.0	0.0	8.0	-12	0.0	5.0

Table 2: Results from runs with 30 data points per wave, distributed among one time series (upper table), three time series (second table), and five time series (third table). The correct network, from which the artificial data was generated, is shown in the lowermost table.

In order to test the effect of adding more time series to the data set, three sets of runs were carried out. In all three sets, 30 data points per wave were used. In the first set, the data points for every wave were all obtained from one single trajectory (time series). In the second set, ten points per wave were measured from each of three different trajectories, and in the third set, six points per wave were measured from each of five different trajectories.

The results together with the network from which the data were generated are shown in Table 2. For the single time series used in the first set of runs, whose results are shown in the uppermost table, the expression level curves were monotonous, containing only very little information to constrain the parameters. Adding two or

w_{ij}				b_i	τ_i
0	0	0	0	0	11
$++$	$--$	0	0	0	7.5
0	0	0	0	0	9.1
0	0	0	$--$	0	16

w_{ij}				b_i	τ_i
$++$	$--$	$+$	0	$-$	11
$++$	$--$	0	0	$-$	5.8
0	$--$	$++$	0	0	7.2
0	0	$++$	$--$	0	6.4

w_{ij}				b_i	τ_i
$++$	$--$	0	0	0	11
$++$	$--$	0	0	$-$	6.3
0	$--$	$++$	$+$	0	6.3
0	0	$++$	$--$	0	5.8

w_{ij}				b_i	τ_i
$++$	$--$	0	0	0	10
$++$	$--$	0	0	$-$	5
0	$-$	$++$	0	0	5
0	0	$+$	$--$	0	5

Table 3: A coarse-grained version of table 2. The panels show the data from the runs with 1 time series (top left), 3 time series (top right), 5 time series (bottom left), as well as the actual network (bottom right).

four time series significantly improved the results (second and third tables), despite the fact that the total number of data points was unchanged.

Thus, for the first set of runs, only three of the weights were significantly non-zero, compared to eight in the correct network, and the standard deviation was very large for most weights. None of the bias terms obtained significant non-zero values.

For the second set of runs, based on three time series of data (second table), all the weights that were supposed to be non-zero were indeed significantly different from zero, which also was the case for the set of runs based on five time series (third table). For the second set of runs, two bias terms were significantly non-zero, whereas *all* the bias terms were close to their correct values for the third set of runs, with only the second bias term being significantly different from zero. In both cases, the time constants were very close to their correct values.

When presented in the form used in Table 2, the results are far from transparent and also difficult to compare. In order to facilitate the comparison, the results are presented again, in a different form, in Table 3. Here, weights and biases are given as 0, positive ($+$), strong positive ($++$), negative ($-$), or strong negative ($--$), with the border between positive (negative) and strong positive (negative) set, somewhat arbitrarily, at 10 (-10). A weight is set to 0 if the interval formed by the standard deviation contains zero. Apart from making the comparison between different networks simpler, this form of presentation will also be more realistic when the results obtained from real data are displayed: With the measurement accuracy presently available, one does not expect (or need) to find exact weights and therefore a coarse representation, as used in Table 3, should be sufficiently accurate.

5 Application to Rat Spinal Cord and Hippocampal data

We have used the data obtained by Wen et al. (1998) for the development of rat spinal cord and hippocampus. The data consists of measurements for of order 100 genes for each tissue, with a total of 65 genes in common for the two tissues. Wen et al. carried out a cluster analysis through which four distinct waves of expression and a constant wave were identified. With the constant wave included in the bias term, we were left with four dynamic variables. Wave 1 contains genes active during initial proliferation, wave 2 is associated with neurogenesis, wave 3 contains mostly genes for neurotransmitter signaling, and wave 4 is made up of genes active during the final maturation of the tissue.

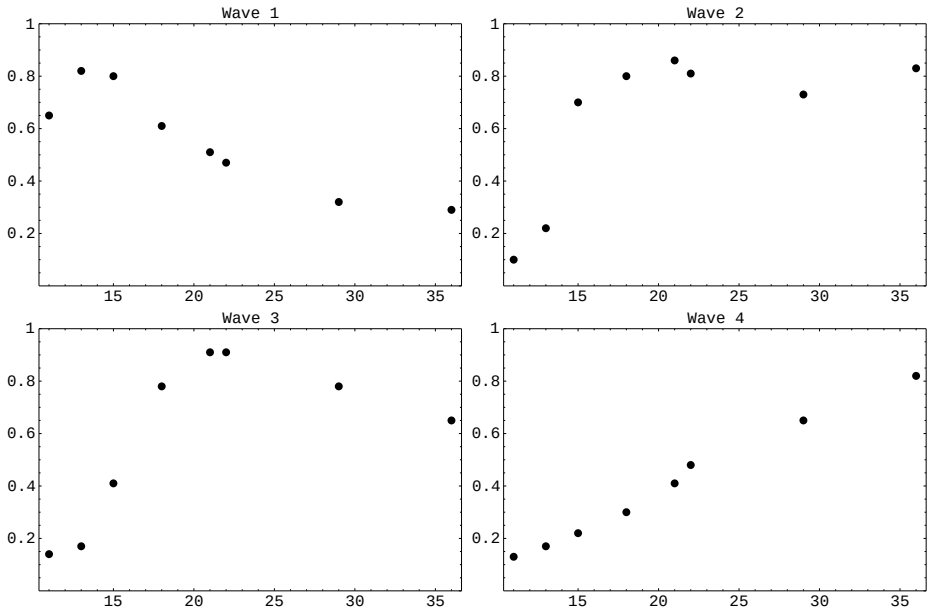


Figure 2: Expression levels for the four waves identified by Wen et al. For each panel, the horizontal axis shows time (in days), and the vertical axis the expression level, computed as an average of the expression levels of the genes in the corresponding cluster.

The four waves of expression for the spinal cord data are shown in Fig. 2. Similar curves (not shown) were also obtained for the hippocampal data. We made three sets of runs, fitting network parameters to spinal cord data (Set 1), hippocampal data (Set 2), and both data sets together (Set 3). For data obtained in different tissues, one must also include the possibility of a systematic difference between the networks describing the different tissues. Thus, in Set 3 we included, for each wave, a

w_{ij}				b_i	τ_i
0	0	0	$--$	0	14
$++$	0	0	0	0	5.1
0	$++$	0	$--$	0	5.2
$--$	$++$	0	0	0	17

w_{ij}				b_i	τ_i
0	0	0	$--$	0	19
$++$	$--$	0	0	0	9.4
$++$	0	$--$	0	0	7.0
$++$	0	0	$--$	0	16

w_{ij}				b_i	τ_i	t_i
0	0	0	$--$	$++$	18	$+$
$++$	$--$	0	0	$+$	6.2	$-$
$++$	0	$--$	0	$++$	7.1	0
0	0	$++$	$--$	0	18	0

Table 4: Results from the three sets of runs carried out with real data. Upper left table: Set 1, upper right table: Set 2, lower table: Set 3.

tissue dependent parameter t_i, and used the network equations

$$\tau_i \dot{x_i} + x_i = g\left(\alpha t_i + b_i + \sum_j w_{ij} x_j\right), i = 1, \ldots, 4, \tag{5}$$

where α was equal to -1 for the spinal cord data and $+1$ for the hippocampal data.

The runs in Set 1 and 2 were terminated when the average error was 0.7σ, and the runs in Set 3 were terminated at an average error of 0.9σ, with $\sigma = 0.1$. The results of the three runs are shown in Table 4. Set 1 and Set 2 gave almost identical results for the first two waves, i.e. for $i = 1$ and 2, but conflicting results for the last two waves. It is reassuring to note that the results of Set 3 agreed quite well with those of Set 1 and 2 in the case of the first two waves. As was the case for the artificial data, the standard deviation of the parameters decreased when additional time series were used, and so, for Set 3, several of the biases obtained significant non-zero values. Note, however, that the biases obtained from Set 3 cannot be directly compared with those obtained from Sets 1 and 2, due to the inclusion of the t_i parameters, whose coarse-grained values are also given in the table corresponding to Set 3. We note also that the time constants agree rather well for all three runs. Thus, from Table 4 we can say with some confidence that wave 4 has an inhibitory effect on wave 1, and that wave 1 has an excitatory effect on wave 2. With slightly less confidence, we conclude that wave 1 has an excitatory effect on wave 3, and that both waves 3 and 4 have an inhibitory effect on themselves. Finally, we also note that a number of the interactions are rather weak.

6 Conclusion and Further Work

We have introduced and evaluated the performance of a method for reverse engineering of genetic regulatory networks, and applied it to artificial data sets as well as to a coarse-grained representation of a data set consisting of measurements from rat central nervous system.

The evaluation shows that, for $N \sim 4$, the method is able to obtain an accurate representation of the parameters of a network, provided that there are measurements from several time series available. With only one time series of data, some features of the network can still be determined, albeit less accurately.

The next step in our analysis will be to apply the procedure described in this paper to systems with different N and larger data sets, and to improve the speed and accuracy of the method by, for example, allowing the GA to dynamically vary the connectivity of the network.

References

[1] Davis L. (Ed.), 1991, *Handbook of genetic algorithms*, Van Nostrand Reinhold, New York

[2] D'Haeseleer, P., Wen, X., Fuhrman, S., & Somogyi, R., 1999, *Pacific Symposium on Biocomputing* 99, p.41

[3] Holland J.H., 1975, *Adaptation in natural and artificial systems*, 1st ed.: University of Michigan Press, Ann Arbor; 2nd ed.: 1992, MIT Press, Camb ridge

[4] Kauffman, S.A., 1993, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford Univ. Press

[5] Mitchell M., 1996, *An introduction to genetic algorithms*, MIT Press, Cambridge

[6] Reinitz, J. & Sharp, D.H., 1995, *Mechanisms of Development*, **49**, 133

[7] Somogyi, R. & Sniegoski, C.A., 1996, *Complexity*, **1**, 45

[8] Wen, X. et al., 1998, *Proc. Natl. Acad. Sci.*, **95**, 334

[9] Yuh,. C.-H., Bolouri, H. & Davidson, E.H., 1998, *Science*, **279**, 1896