

# Modeling Genetic Regulatory Dynamics in Neural Development

Mattias Wahde<sup>1,\*</sup>

John Hertz<sup>2</sup>

1. Div. of Mechatronics, Chalmers Univ. of Technology, S-412 96 Göteborg, Sweden.

tel. +46 31 772 3727, fax. +46 31 772 3730

2. Nordita, Blegdamsvej 17, DK-2100 Copenhagen

## Abstract

We model genetic regulatory networks in the framework of continuous-time recurrent networks. The network parameters are determined from gene expression level time series data using genetic algorithms. We have applied the method to expression data from the development of rat central nervous system, where the active genes cluster into four groups, within which the temporal expression patterns are similar. The data permit us to identify approximately the interactions between these groups of genes. We find that generally a single time series is of limited value in determining the interactions in the network, but multiple time series collected in related tissues or under treatment with different drugs can fix their values much more precisely.

*Keywords:* Reverse engineering

## 1 Introduction

Large-scale gene expression data provide us with genome-wide information about the genetic regulatory networks that control basic biological processes such as development, disease, and the cell cycle. Time series of such measurements allow us to visualize this dynamics directly as changing intensity patterns on gene chips. The next step we would like to take is toward understanding how these particular changing expression states come about, that is, toward understanding and characterizing mathematically the underlying dynamics of the genetic network.

In this paper we take a first step toward such a characterization for a data set of intermediate scale (on the order of 100 genes). While this is still an order of magnitude or two smaller than a typical complete eukaryotic genome, there are enough genes that we will be able to infer something about the networks governing the particular process under study: the development of neural structures, specifically spinal cord and hippocampus, in rats. The problem also serves to illustrate important points relevant to applying such an analysis to other large-scale expression data.

The task we have is of the sort called an inverse problem, or “reverse engineering”. The problem is not, given a model of the system, to find out what its dynamics are, but rather, given dynamical data for it, to find a suitable model. In the present problem, “a model” means

specifying how the protein product of each gene influences the transcription of all the other genes. If there are  $N$  genes, this needs at least  $N^2$  parameters, even in the simplest possible description (a linear model), and these require at least  $N^2$  measurements to fix them.

This simple requirement makes our reverse engineering problem very difficult, at least if we want to solve it at the level of specifying the influence of every gene on every other one. If we have time series of  $T$  measurements for  $N$  genes, we need in principle  $T > N$ , and this is never satisfied in current data. For example, in the data we analyze (obtained by Wen *et al.* [5]),  $N = 66$  and  $T \approx 10$ . (And note further that measuring the expression levels of more genes only makes the problem worse.)

Thus we have to limit our ambitions and try to describe things as well as we can using a smaller number of variables (and thus fewer parameters in our model). We do this here by exploiting the findings of Wen *et al.* [5], who succeeded in clustering the genes into five groups, within each of which the temporal development of expression levels was highly similar. Neglecting the differences within these clusters, then, approximately speaking, there are only five independent degrees of freedom in the system, not 65. Now, with time series of length 10, we have more measurements than parameters, so we can hope to determine a model from the data.

That is the task we carry out in this paper. We construct a model of four interacting units, each associated with one of the clusters of Wen *et al.* (Genes in their fifth cluster have approximately constant expression levels, so they are dynamically trivial and are not considered in the model.) For each unit, there are six parameters: a time constant that specifies how rapidly its expression level responds to changes in the expression levels of other genes, four numbers (each of which can be either positive or negative) describing how its transcription rate is affected by the expression levels in the four clusters (including itself), and a number associated with the intrinsic transcription rate (i.e., in the absence of regulation). The model is almost the simplest set of first-order differential equations one can write for these variables; it is like a linear model except that the sum of these influences is subjected to a sigmoid nonlinearity to keep expression levels from going negative or exceeding a maximum value.

However, even for this reduced model, it will turn out that we do not have sufficient data to make an unambiguous fit. The different measurements in a time series are not independent, so we do not really have as many independent data points as we might naively have thought. To deal with this difficulty, we adopt a statistical strategy. We employ a technique (a genetic algorithm) that permits us to find many different parameter sets that fit the data for a given quality-of-fit criterion. By running it many times with different random number seeds, we obtain empirical sample distributions of the parameters in the models. Some parameters will vary wildly across their distributions; these are not well-determined. However, others will turn out not to fluctuate strongly, and we can place some confidence in their values. In this way, we will be able to make some reliable inferences about the networks governing nervous system development in this animal. Furthermore, the analysis will also indicate how to design future experiments to permit us to specify the network more accurately.

The paper is organized as follows. Section 2 describes the data we work with: how it was obtained from the measurements of Wen *et al.* and how we preprocess it before applying the model to it. Section 3 describes the model sketched above in more detail. Section 4 describes the genetic algorithm used to fit the model parameters. In section 5 we try to carry out the statistical analysis on each of the two time series (spinal cord, hippocampal) separately. We find that almost no parameters are well-determined by either single time series by itself – the data simply do not probe enough of the state space of the network. Thus, in section 6, we

apply the model to the two series taken together (with an extra parameter per unit to account for possible differences in the intrinsic transcription rate in the two tissues). Now we find that the network parameters are much better determined, and we can identify the chief influences on the transcription of genes in the respective clusters. In the Discussion, we consider especially the implications for future analysis of this kind for other systems.

## 2 The data

The analysis in this paper is based on the data obtained by Wen *et al.* [5] consisting of measurements of gene expression levels for of order one hundred genes during the development of the central nervous system of rats. Each gene was measured at 9 different points in time (of which the last, measured for the adult animal, was not used in this study). The first measurement was made 10 days before birth, and the intervals between measurements were 2 or 3 days in the period before birth and 7 days after birth.

The data contain measurements from two different developing tissues: spinal cord and hippocampus. Neglecting the genes for which data were available from only one of the two tissues, we were left with data from 66 genes. Following the procedure used by Wen *et al.*, these were clustered into four groups, each with its own characteristic temporal expression profile. (They also identified a fifth group of genes with essentially constant expression levels, but we ignore them, since they do not play any dynamical role.) As emphasized in the Introduction, this clustering was necessary to reduce the number of degrees of freedom in the analysis to something smaller than the number of data.

In Figs. 1 and 2 the four temporal expression patterns (called “waves” by Wen *et al.*) are shown for hippocampal and spinal cord tissues, respectively. Each point represents the average expression level, normalized to the range  $[0, 1]$ , of the genes comprising the corresponding wave. The numbering of the waves should be evident from the figures: genes in wave 1 are active in the early stages of development, followed by genes in wave 2, etc.

## 3 The model

It is important to appreciate that, even before clustering, gene expression time series represent a coarse-grained description of the dynamical system. Part of the coarse-graining is spatial: the measurements are averages over the different kinds of cells in the tissue in question. And part of it is temporal: a full dynamical description would include concentrations of proteins and many other molecules, since reactions involving these molecules transduce signals that influence transcription. However, the characteristic times for these reactions are orders of magnitude faster than those governing transcription. These fast, unobserved variables essentially determine the effective interactions that can be measured on longer timescales between the slow variables. Only these effective interactions can be measured in long-timescale experiments, and these are the parameters of the models we construct.

It is thus evident that it would be meaningless to try to model the system microscopically. The nature of the phenomenon we are studying, the data and the coarse-graining we have been forced to apply to it all dictate a phenomenological level of description, such as the differential equations

$$\tau_i \frac{dS_i}{dt} = -S_i(t) + g \left( b_i + \sum_j w_{ij} S_j(t) \right). \quad (1)$$

The quantities  $S_i(t)$  represent the average mRNA concentrations for the clusters, labeled by the index  $i$ . For the function  $g(x)$  we take a sigmoid form to enforce the conditions that expression levels can not get too large or fall below zero. We choose  $g(x) = (1 + e^{-x})^{-1}$ ; this means that  $S_i$  is the expression level normalized by its maximum value. A model of almost this form, with diffusion added, was used successfully at the single-gene level by Reinitz and Sharp [3], to model *eve* stripe formation in drosophila.

We make no particular assumptions about the connections  $w_{ij}$ , biases  $b_i$ , or time constants  $\tau_i$ , so a total of  $N(N + 2)$  parameters need to be determined for a network with  $N$  expression clusters.

All cells in any given individual share the same genetic material, so in a model for the complete network, the parameters would be the same in all cells and tissues. Here, we are trying to construct a reduced model in which we restrict our attention to genes whose expression levels vary in a small number (2) of tissues. The constant expression levels of the remaining genes can then be absorbed into the  $b_i$ , but as these levels might differ from tissue to tissue, the  $b_i$  must be allowed to be tissue-dependent.

The model (1) is not the most general one. In particular, the argument of the sigmoid need not be linear in the  $S_i$ . One can employ a general power series expansion:

$$\tau_i \frac{dS_i}{dt} = -S_i(t) + g \left( b_i + \sum_j w_{ij} S_j(t) + \sum_{jk} w_{ijk} S_j(t) S_k(t) + \dots \right). \quad (2)$$

Terms in the second sum where  $j$  or  $k$  is one of the genes with time-independent (but tissue-dependent) expression levels lead, in the reduced model that does not explicitly describe these genes, to effective tissue-dependent  $w_{ij}$ 's, in the same way that we obtained tissue-dependent  $b_i$ 's in the simple model (1).

A commonly-proposed modeling strategy is to take the limit where genes are treated as either fully "on" ( $S_i = 1$ ) or fully "off" ( $S_i = 0$ ). Then the most general possible description involves all possible Boolean functions of  $N$  inputs. A model like (2), with all the nonlinear terms up to order  $N$ , describes this case if we replace the sigmoid  $g(x)$  by a sharp threshold function and let the  $\tau_i \rightarrow 0$ . However, as pointed out by Kauffman [1], high-order combinatorial control (such as would be implemented by high-order terms in the expansion of the argument of  $g(\ )$  in Eq. (1) is difficult to implement biochemically. Thus one may hope that relatively few significant high-order terms will be necessary to describe real genetic networks. We are indeed forced into adopting this hope as a working hypothesis by the paucity of the data available to us in the present work. Adding all the terms with  $w_{ijk}$  would add  $N^3$  parameters, and even with the dimensionality reduction given by the clustering, the data would be totally inadequate to determine a model of this size. Thus we confine our analysis to the model of Eq. (1) with tissue-dependent biases  $b_i$ .

For computational convenience, we do not actually work with the differential equations, but rather with the discrete-time maps

$$S_i(t + \Delta t) = S_i(t) + \frac{\Delta t}{\tau_i} \left[ -S_i(t) + g \left( \sum_j w_{ij} S_j(t) + b_i + \alpha t_i \right) \right], \quad (3)$$

where we have parametrized the tissue-dependent biases as  $b_i + \alpha t_i$ , with  $\alpha = \pm 1$  for spinal cord and hippocampus, respectively. In the limit  $\Delta t / \tau_i \rightarrow 0$ , Eq. (3) converges to Eq. (1). In fact, we will find values of the  $\tau_i \gg \Delta t$ , so the discrete-time approximation should be quite good.

## 4 The algorithm

We have used a genetic algorithm (hereafter GA) to determine the parameters of the network. Evolutionary algorithms, of which GAs are a special case, have received widespread attention in recent years. Such algorithms are especially useful when the search space has a complex error landscape. Here only a short description of the GA will be given. For a more complete description, see, e.g. Mitchell [2].

In a GA, a population of  $M$  (the population size) candidate solutions, called individuals, is used. Each individual has a string of digits (the chromosome) associated with it. Initially, the values of the digits in the chromosome (the genes) are taken to be random, and the corresponding individuals comprise the first generation. Each string yields, when decoded, the parameters  $w_{ij}$ ,  $b_i$ ,  $\tau_i$ , and  $t_i$  of a network. (The weights  $w_{ij}$ , biases  $b_i$  and tissue-dependent bias differences  $t_i$  are taken initially to be uniformly distributed between  $-W_{\max}$  and  $+W_{\max}$ .)

Each network thus obtained is evaluated, by iterating Eq. (3) from a suitable starting point (see below), and the deviation between the time series obtained from the integration and the data is recorded. Thus, the total deviation for a network is given by

$$\delta = \frac{1}{N} \sum \frac{(x^i - x^d)^2}{\sigma^2}, \quad (4)$$

where  $N$  is the total number of points used in the comparison,  $x^d$  are the data points and  $x_i$  the values obtained from the integration. The parameter  $\sigma$  is simply a scale factor that plays no role in the algorithm. For convenience we assign it a value 0.1, corresponding to typical experimental error bars in the data, to make it easier to judge the overall quality of fit at a given  $\delta$ . The fitness of the network in question is (conventionally) defined by

$$f = \frac{1}{1 + \delta}. \quad (5)$$

Thus, the smaller the deviation, the larger the fitness.

When all individuals have been evaluated and fitnesses have been assigned, the next generation is formed. In order to be able better to distinguish individuals with similar fitness values, new quantities called "rank fitnesses" are assigned through a ranking process. (The original fitness values  $f$  are therefore sometimes referred to as "raw" fitnesses.) In this process, the individual with the highest raw fitness is assigned rank fitness  $M$ , the second best individual is assigned rank fitness  $M - 1$ , etc., down to the worst individual, whose rank fitness is 1. Two parent individuals are selected with probabilities directly proportional to their rank fitnesses. Then, with probability  $p_c$  (the crossover probability) the two chromosomes are cut at a randomly chosen point along the string, and the four parts are joined crosswise. With probability  $1 - p_c$  the two parents are left unchanged. Either way, the two strings are then subjected to mutation, in which the individual genes are changed to new random values with probability  $p_m$ .

This process – selection, crossover, and mutation – is repeated until  $M$  new individuals have been formed. The  $M$  new individuals make up the second generation, which is evaluated in the same way as the first. This process is repeated until a satisfactory network with high (raw) fitness is obtained. The inverse problem of determining the network parameters given the time series data is far from trivial and can be approached in several different ways, and some experimenting was required in order to find the best way. In the method chosen, first, the equations (3) were iterated from the first point to the last (and the deviations measured at

those times for which data points were available), then the equations were iterated from the second point to the last etc. until the final iteration which started with the penultimate point and continued to the final point. Thus the fitness measured the predictive ability of the model for both short and long intervals between initial and test points. This method turned out to be both efficient (in terms of integration times) and robust.

## 5 Analysis of single time series

As a first step in the analysis, the reverse engineering algorithm was applied separately to each of the two time series. For each time series, the reverse engineering algorithm was applied 250 times with different random starting weights, with  $W_{\max} = 5$ . The runs with hippocampal data were stopped at  $f = 0.6098$ , corresponding to an average deviation between measured data points and network output of  $0.8\sigma$ , and the runs with spinal cord data, for which it happened be easier to obtain a good fit, were stopped at  $f = 0.6711$ , corresponding to an average deviation of  $0.7\sigma$ . The value of  $\sigma$  was 0.1. The results are shown in the two panels of Table 1. Figures 3 and 4 show the results in a different way, as histograms of the values of the  $w_{ij}$  found in the different runs.

We also show (Table 2) a simplified representation of the results in which significant parameters (i.e. those with averages exceeding the corresponding standard deviation) with values in the ranges  $[0, \frac{1}{2}W_{\max}]$  and  $[-\frac{1}{2}W_{\max}, 0]$  are denoted by plus and minus signs, respectively, and values in the ranges  $[\frac{1}{2}W_{\max}, W_{\max}]$  and  $[-W_{\max}, -\frac{1}{2}W_{\max}]$  are denoted by double plus and minus signs, respectively. For those parameters for which (the modulus of) the average does *not* exceed the standard deviation, the corresponding entry is zero.

It is evident from all three representations of the results that a single time series is not sufficient to constrain more than a few of the network parameters. The fact that one time series is not sufficient to constrain the network parameters agrees with the results obtained for artificial data by Wahde and Hertz [4].

## 6 Analysis of multiple time series

Repeating the runs in Table 2 but using both time series, we obtained the results shown in Fig. 5 and Table 3a. For these runs, the termination criterion was  $f = 0.6098$ . We note that, with two time series, the algorithm is able to pin down more parameters. Furthermore, an important consistency check can be made: comparing Table 3a with Table 2 one sees that the parameters found to be well-determined by either single time series are also found to be well-determined by the joint data. An exceptional case is  $w_{44}$ , for which the single series disagree strongly. Here the joint series results agree with those of the hippocampal series. We can see by examining the single-series histograms for this parameter that the hippocampal data indicate a more certainly positive value than the spinal cord data do a negative one, so the outcome for the joint series is not surprising: the less equivocal data decide the question.

All the above results were obtained with the maximum weight absolute value  $W_{\max} = 5$ . The fact that many of the histograms are concentrated around either  $+W_{\max}$  or  $-W_{\max}$  suggests that we should explore fits in which larger weights are allowed. Fig. 6 shows the weight histograms obtained for the combined time series with  $W_{\max} = 10$ . It is evident that good fits can also be obtained with these large allowed weight ranges. It even looks as if even larger values would give good fits, since many of these histograms are still concentrated near

the upper or lower ends of the allowed range (though they are not as concentrated as those for  $W_{\max} = 5$ ). Indeed, we find (results not shown) that such solutions do give good fits.

Apparently, then, we conclude that the data do not exclude solutions with very large weight values. It is also apparent, however, comparing Figs. 5 and 6, that as  $W_{\max}$  increases, all the weights scale up roughly together. That is, the weights are determined (to the extent that they are) only up to an overall scale factor.

It is easy to see why this happens. If all the weights in some good solution are multiplied by a large factor, the function  $g(\cdot)$  approaches a step function. Then, at any time any unit is approaching either 1 or 0 exponentially, and it switches between these two possibilities when its net input  $b_i + \sum_j w_{ij} S_j(t)$  passes through zero. At these points,  $S_i(t)$  has abrupt changes in slope. Such behaviour is unrealistic in the present context, but its presence or absence is unlikely to affect the quality of fit on the data points we have available. Thus, we must exclude such very large weights *a priori*.

Using the average network parameters obtained with  $W_{\max} = 10$  from the combined time series and iterating from the first data points, the expression level curves shown in Figs. 7 and 8 were obtained. For the average network, the fitness was 0.6186, i.e. slightly higher than the value used for terminating the individual runs. These curves do not exhibit rapid changes in slope of the kind described above, so we conclude that  $W_{\max} = 10$  is not artificially large.

## Pruned networks

As is evident from Table 3, not all weights obtained from the reverse engineering process are significant. When the non-significant weights are put to zero, the performance of the resulting network is, in general, not very impressive. For example, for the case  $W_{\max} = 10$ , a fitness value of less than 0.3 is obtained, compared with more than 0.6 for the average network. This is perhaps not so surprising, since the parameters that are not set to zero still have significant standard deviations. However, if the reverse engineering procedure is applied again, with non-significant parameters (from the original run) clamped at zero and the other parameters initialized with random values, a much better result is obtained, and the remaining parameters are determined more precisely. We performed this procedure for both  $W_{\max} = 5$  and  $W_{\max} = 10$ .

The significance criterion we used for determining whether to remove weights is somewhat arbitrary, so we also experimented with another one: the ratio of the total numbers of positive and negative values of a given weight found across all the runs that achieved the fitness criterion. For  $W_{\max} = 5$  we found that if we required this ratio to be greater than 4 or less than 1/4 (i.e., at least 80% of the weight in the histogram on one side of zero or the other), we obtained the same results as in Table 3a except that two of the biases,  $b_2$  and  $b_3$ , now passed the test. Relaxing the criterion somewhat, now only asking that 2/3 of the weight in the histogram be on one side of zero or the other, we found two more weights,  $w_{23}$  and  $w_{33}$ , that also reached significance.

These constrained fits provide a useful consistency check on conclusions drawn from the unconstrained ones. It is therefore reassuring that they lead to nearly the same assignments of significant weights from those in Table 3.

In the light of these results, we propose the following reverse engineering procedure:

1. Run the reverse engineering algorithm, without introducing any particular constraints (except the maximum allowed values) on the network parameters. Make a sufficient number of

runs (100–250) to form histograms of the network parameters.

2. Determine the averages and standard deviations of the network parameters using the results from step 1.
3. Set non–significant parameters (if any) to zero. If there are no non–significant parameters, end the procedure.
4. Rerun the reverse engineering algorithm, with non–significant weights clamped at zero. If the results (measured by the fitness) are as good, or almost as good, as for the previous set of runs, form the network averages and return to step 3. If instead the results are significantly worse than in the previous run, discontinue the procedure.
5. Repeat the entire procedure for different values of  $W_{\max}$ .

A conservative inference strategy then dictates that only weights which appear significant in both pruned and unpruned models, for all values of  $W_{\max}$ , should be judged significant.

Based on the results above, we then summarize what we have learned about the network governing CNS development diagrammatically in Fig. 9. Only significant connections are shown. From this provisional picture we can see, for example, that

- Genes in cluster 4 (and possibly cluster 2) are responsible for lowering the expression level of those in cluster 1 as development proceeds.
- Clusters 2 and 3 get most of their their positive input from cluster 1, so their rises and subsequent falls in expression level are apparently driven by the initially high expression level of cluster 1 and its subsequent falloff. The fact that cluster 3 falls off more severely from its maximum as development proceeds is attributable to repression from cluster 4.
- The rise in expression level for cluster 4 is apparently driven mostly by clusters 1 and 3, and it is limited by repression within cluster 4 itself.

Thus, our analysis makes it possible to understand of what causes the various rises and falls in expression for different groups of genes as the genetic control network evolves in CNS development. These explanations are not evident from mere examination of the data (Figs. 1 and 2). They are, of course, provisional, because future data might suffice to determine some of the parameters that are not well–determined at present, and these could change our explanations. However, the present story is the most we can say on the basis of the present data.

## 7 Discussion

It is instructive to formulate what we have done as an exercise in statistical inference. Initially (i.e., before we have any data), all we assume about the weights in the network is that they lie in a hypercube of size  $2W_{\max}$  on each side; the *a priori* probability density of the weights is uniform within this volume. As a result of the data, this initially uniform probability cloud shrinks, so the *a posteriori* density is mostly concentrated in a much smaller region of parameter space. If we had sufficient data (and if the network were really exactly describable by a model of the form we use), the posterior density would get concentrated to a single point, representing the true network. We are far from able to achieve this unique solution (the problem remains under-determined), but we have also come a good way from *a priori* total ignorance. The histograms of Figs. 3–6 are our estimates of the marginal probability densities of the individual weights



(i.e. average cross-sections of the probability density along the different weight axes), and they are evidently far from uniform. The probability cloud is evidently quite compressed along certain axes; these correspond to our well-determined weights.

We found that a few of our decisions about significance of parameters were dependent on the significance criterion we chose, and one should exercise caution in drawing conclusions about these parameters. On the other hand, the fact that the significance of most weights is robust against such variations in the criterion gives us confidence in the main features we have identified in the structure of this regulatory network (at our coarse-grained level).

Of course, examination of the marginal distributions estimated by our histograms or of the standard deviations in, e.g., Table 1 gives only a partial picture of the anisotropic compression of the parameter probability cloud implied by the data. More information is provided by the covariances of the parameters. We have examined these. A few of the weights ( $w_{32}$  and  $w_{42}$ ) were found to be strongly anticorrelated with biases ( $b_3$  and  $b_4$ , respectively). That is, the probability density (marginalised with respect to all other parameters) exhibits a tendency toward an approximately diagonal ridge in the  $w_{32} - b_3$  resp.  $w_{42} - b_4$ -plane. This means that the respective clusters (3 and 4) have significant expression without input from other clusters, or receive significant input from cluster 2, or some combination of these, but we cannot be certain about the relative strengths of these two contributions – knowledge of this correlation does not permit us to estimate any of the individual parameters more precisely.

Our finding of significant improvement in the compression of the probability cloud when we used two time series instead of one carries an important lesson for future experimental design and analysis. It does not help much to make more closely-spaced measurements in single time series, since successive points in a single series lie along the same trajectory in the state space of the network and are therefore not independent. Multiple time series, whether (as here) in different tissues, with different drug treatments, or with different genes knocked out or artificially overexpressed, provide dynamical information in different parts of the state space and therefore constrain the model more tightly. We found this effect earlier with artificial data [4], and the present results confirm it. This provides some basis for cautious optimism about how well we will generally be able to understand genetic networks from data that are likely to be available, as well as some guidance for planning the measurements.

The degree of success we have achieved in determining the network depends crucially on the coarse-graining achieved by clustering the genes into a small number of groups. Such a dimensionality reduction is necessary quite generally for any large-scale expression data likely to be available in the near future: There are simply too many genes, and therefore too many parameters, to hope to be able to pin them all down uniquely unless thousands of time series of data are employed. Perhaps we were lucky that a simple clustering was possible in the present case, and it would be desirable to try other methods of dimensionality reduction to test the robustness of our results. We are presently exploring using principal component analysis in an alternative coarse-graining strategy.

## Acknowledgement

We would like to thank Roland Somogyi and his co-workers for providing us with their data and for many useful discussions.

## References

- [1] Kauffman, S.A. 1993. *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford Univ. Press, Oxford, UK.
- [2] Mitchell, M. 1996. *An Introduction to Genetic Algorithms*, Mit Press, Cambridge, MA.
- [3] Reinitz, J. and Sharp, D.H. 1995. Mechanisms of eve strip formation, *Mech. Dev.* 49, 133–158.
- [4] Wahde, M., and Hertz J.A. 2000. Coarse-grained reverse engineering of genetic regulatory networks, *Biosystems* 55, 129–136.
- [5] Wen, X., Fuhrman, S., Michaels, G.S., Carr, D.B., Smith, S., Barker, J.L., and Somogyi, R. 1998. Large-scale temporal gene expression mapping of central nervous system development, *PNAS*, 95, 334–339.

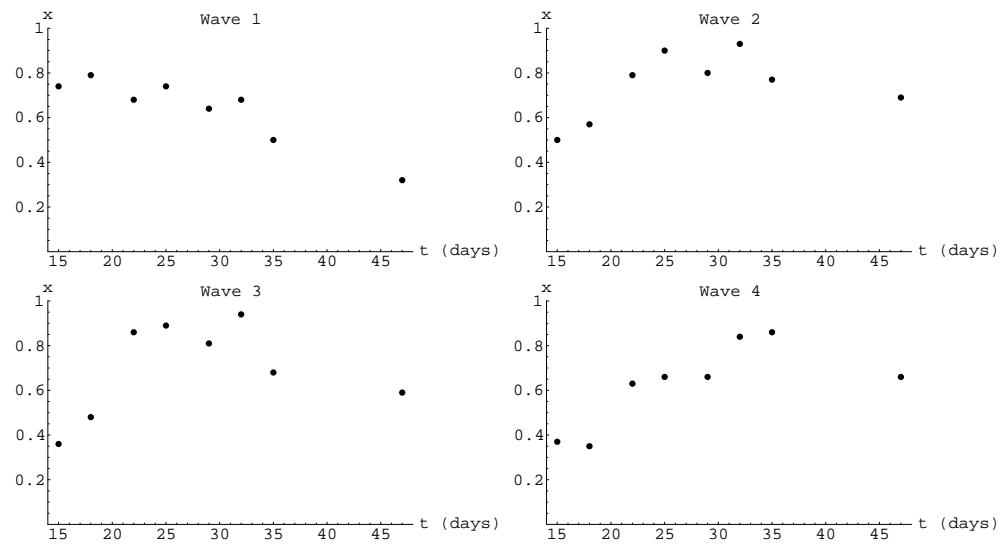


Figure 1: Waves of expression for the hippocampal data.

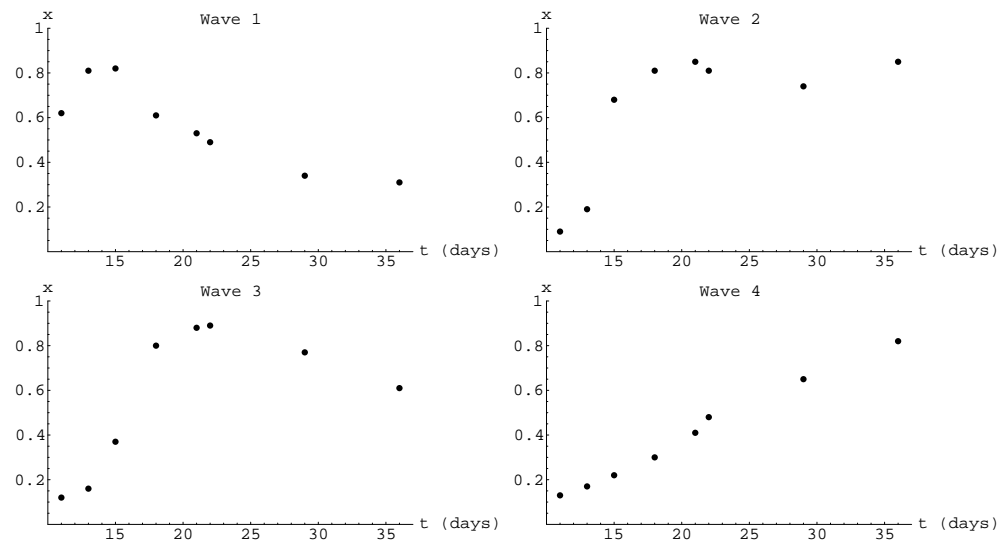


Figure 2: Waves of expression for the spinal cord data.

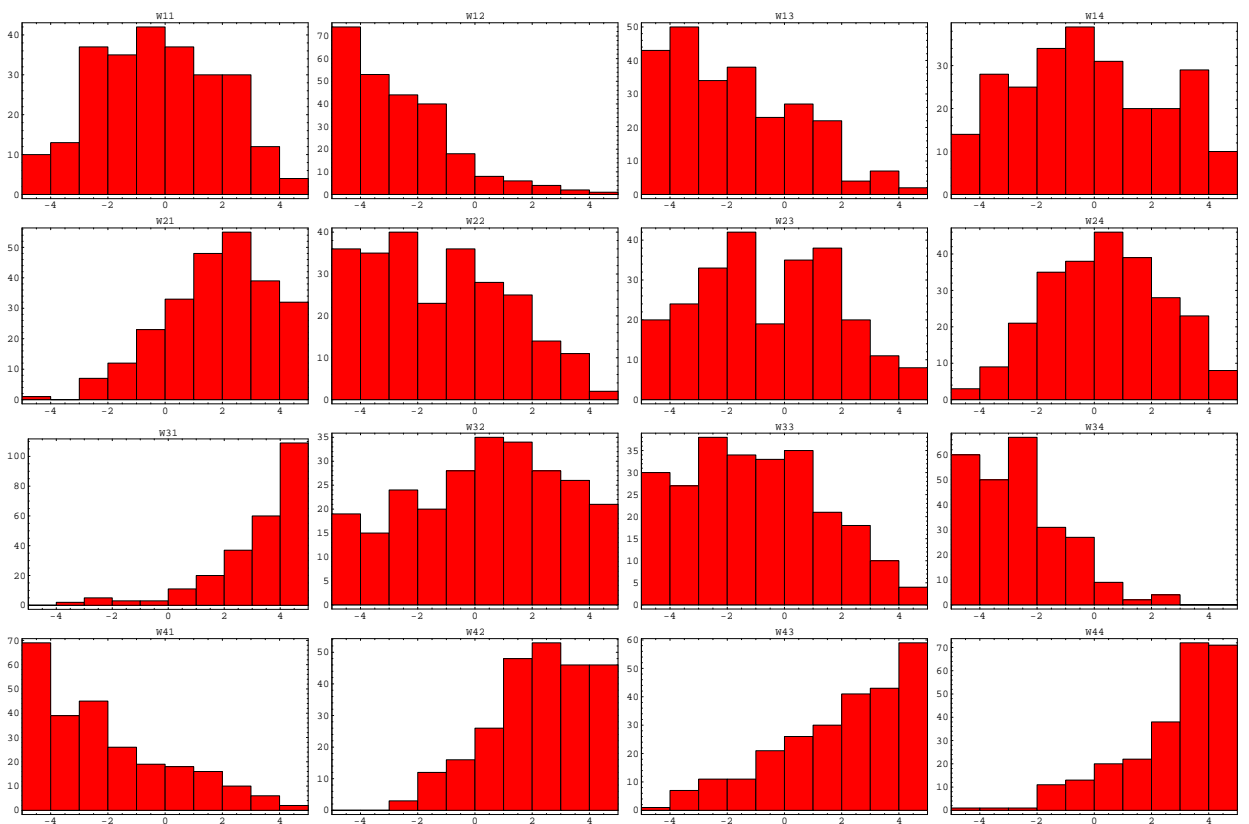


Figure 3: Histograms,  $W_{\max} = 5$ , spinal cord data.

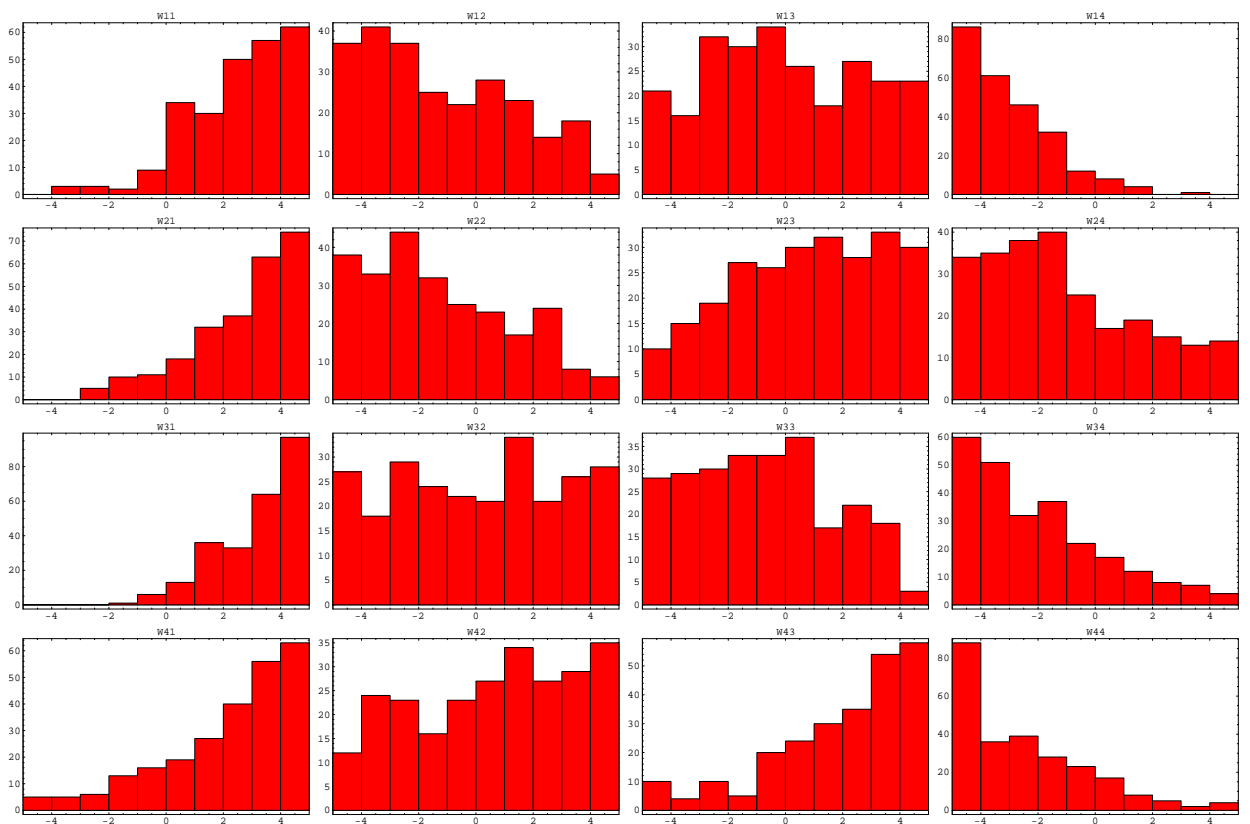


Figure 4: Histograms,  $W_{\max} = 5$ , hippocampal data.

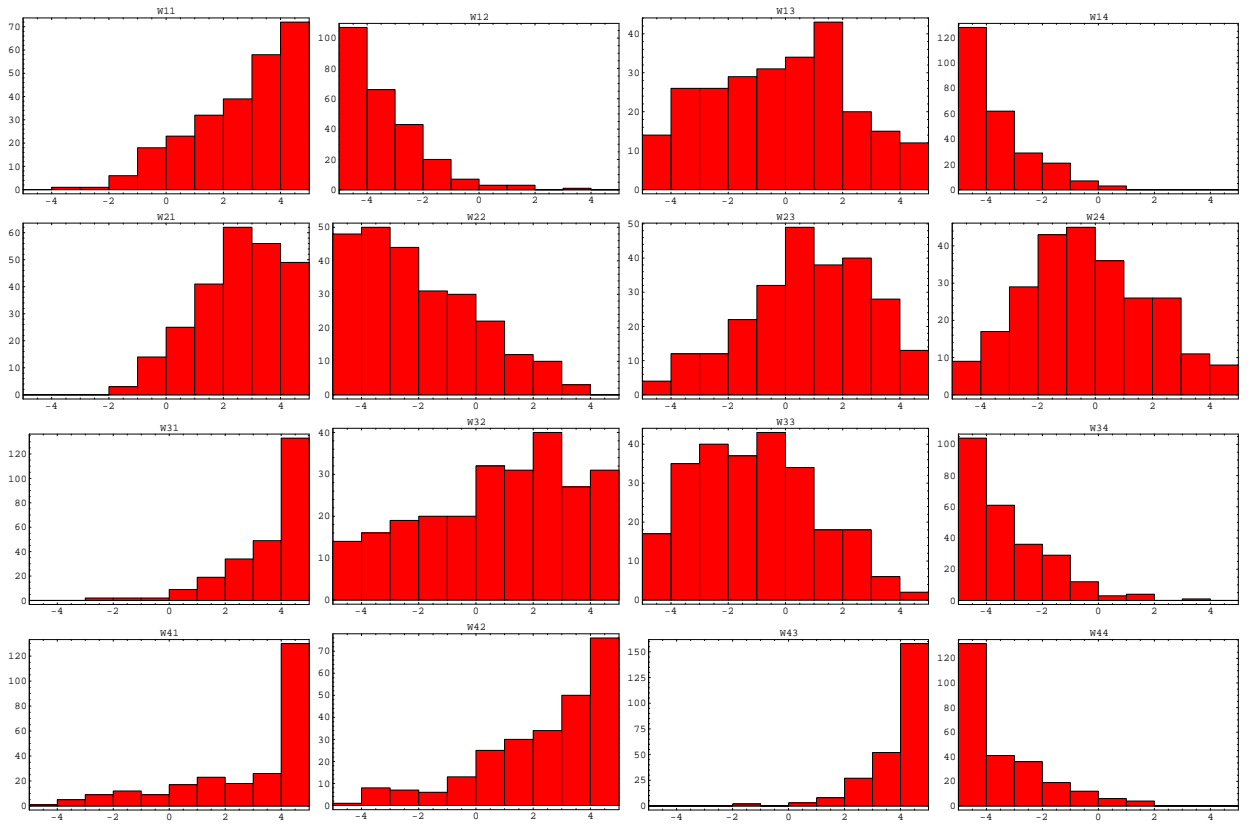


Figure 5: Histograms,  $W_{\max} = 5$ , both data sets.

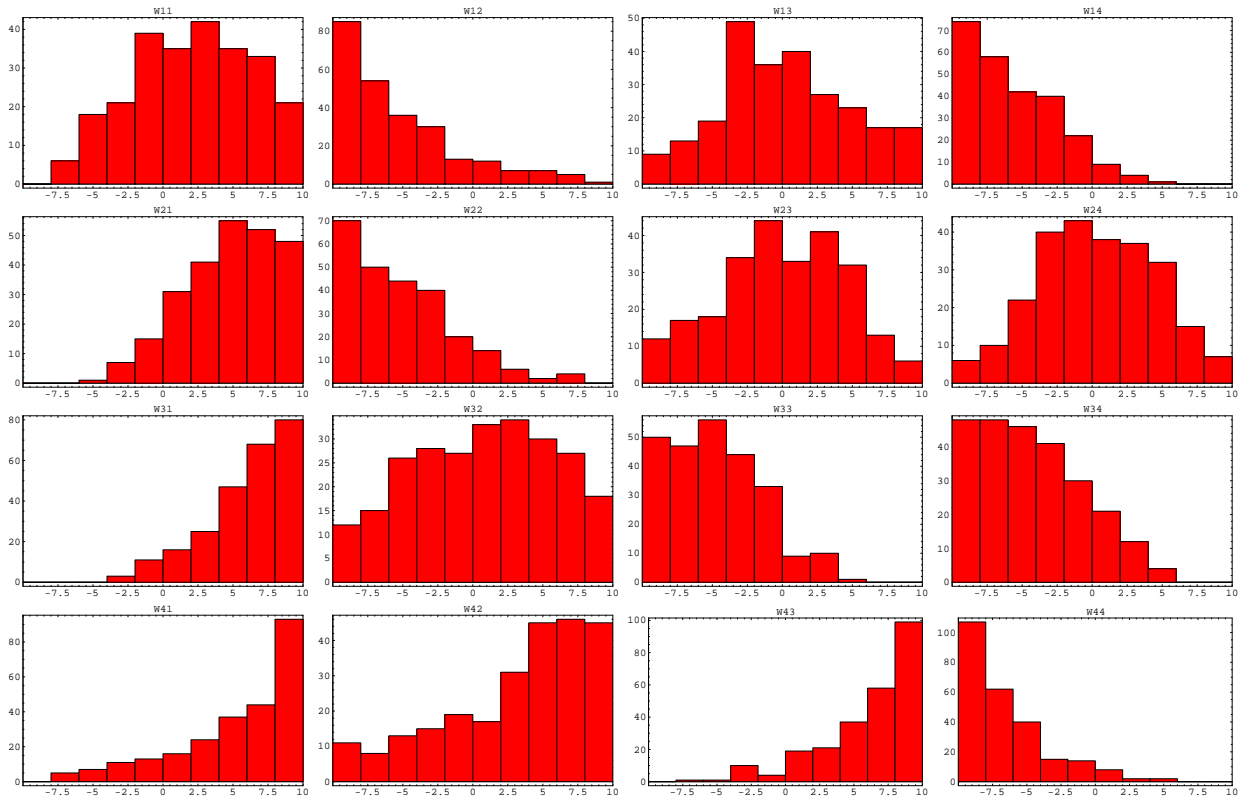


Figure 6: Histograms,  $W_{\max} = 10$ , both data sets.



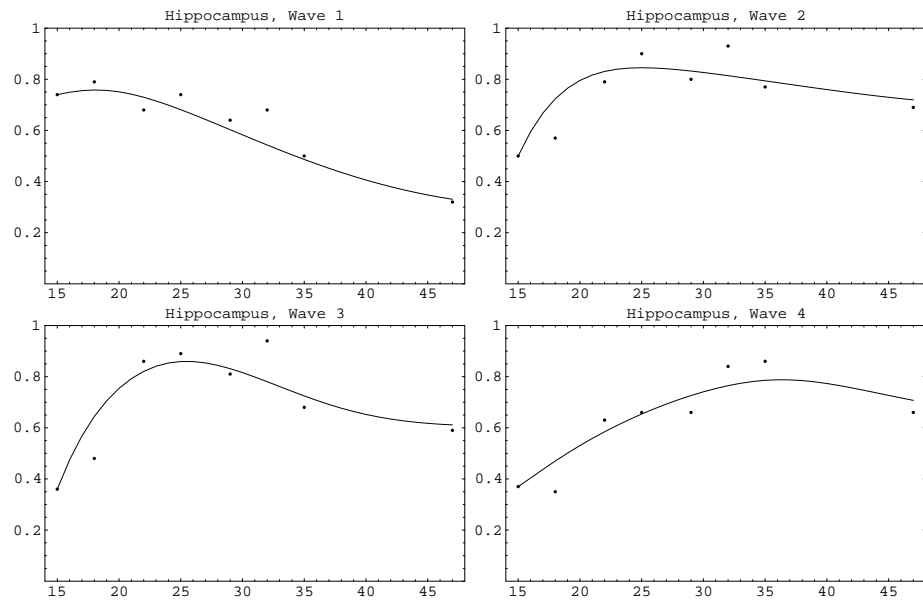


Figure 7: Curves of expression levels for the hippocampal data. Dots: data points; Solid lines: levels obtained from the average network.

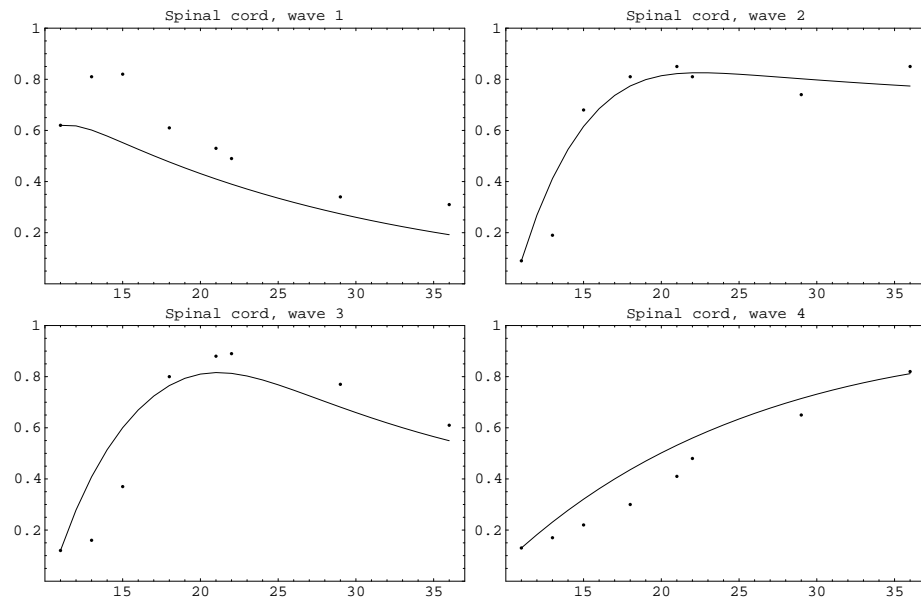


Figure 8: Curves of expression levels for the spinal cord data. Dots: data points; Solid lines: levels obtained from the average network.

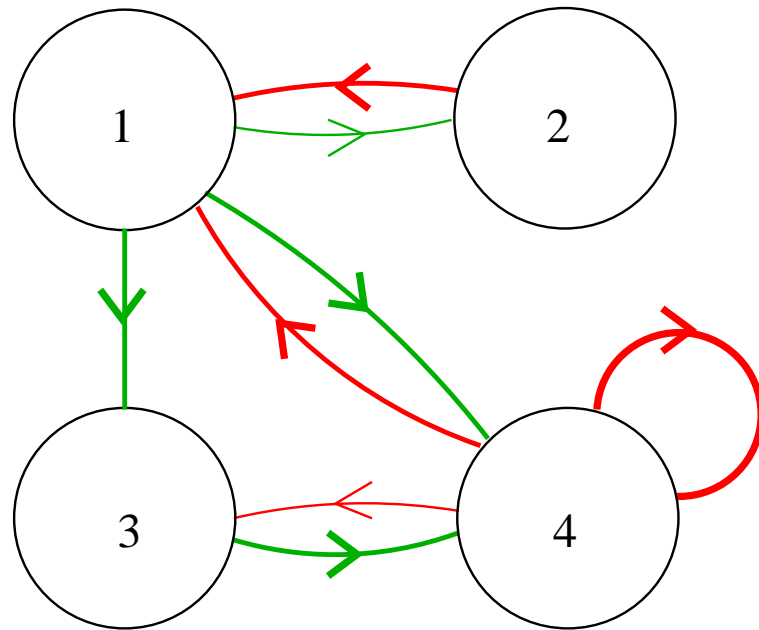


Figure 9: Diagrammatic representation of the final network obtained for  $W_{\max} = 5$ . Thick lines denote weights whose magnitude exceeds  $W_{\max}/2$ . All biases were non-significant.

$i$	$w_{i1}$	$w_{i2}$	$w_{i3}$	$w_{i4}$	$b_i$	$\tau_i$
1	$2.92_{\pm 1.73}$	$-1.15_{\pm 2.62}$	$0.17_{\pm 2.72}$	$-3.30_{\pm 1.58}$	$0.88_{\pm 2.28}$	$13.81_{\pm 4.97}$
2	$2.86_{\pm 1.80}$	$-1.34_{\pm 2.51}$	$0.82_{\pm 2.63}$	$-1.24_{\pm 2.67}$	$1.16_{\pm 2.08}$	$5.27_{\pm 1.91}$
3	$3.48_{\pm 1.40}$	$0.07_{\pm 2.99}$	$-0.73_{\pm 2.43}$	$-2.33_{\pm 2.34}$	$1.34_{\pm 2.13}$	$4.28_{\pm 1.08}$
4	$2.41_{\pm 2.28}$	$0.63_{\pm 2.85}$	$2.09_{\pm 2.47}$	$-2.67_{\pm 2.21}$	$0.05_{\pm 2.11}$	$10.26_{\pm 2.47}$

$i$	$w_{i1}$	$w_{i2}$	$w_{i3}$	$w_{i4}$	$b_i$	$\tau_i$
1	$-0.13_{\pm 2.05}$	$-2.72_{\pm 1.82}$	$-1.65_{\pm 2.16}$	$-0.11_{\pm 2.52}$	$2.17_{\pm 1.60}$	$10.81_{\pm 3.80}$
2	$1.87_{\pm 1.79}$	$-1.29_{\pm 2.34}$	$-0.74_{\pm 2.37}$	$0.42_{\pm 1.88}$	$2.14_{\pm 2.01}$	$4.11_{\pm 1.01}$
3	$3.50_{\pm 1.74}$	$0.33_{\pm 2.70}$	$-0.98_{\pm 2.28}$	$-2.95_{\pm 1.57}$	$1.91_{\pm 2.26}$	$4.97_{\pm 1.05}$
4	$-2.32_{\pm 2.33}$	$2.21_{\pm 1.72}$	$1.97_{\pm 2.24}$	$2.85_{\pm 1.88}$	$-0.01_{\pm 2.19}$	$15.04_{\pm 2.03}$

Table 1: Upper panel: hippocampal data, lower panel: spinal cord data.  $W_{\max} = 5$ .

$i$	$w_{i1}$	$w_{i2}$	$w_{i3}$	$w_{i4}$	$b_i$	$\tau_i$
1	++	0	0	--	0	13.8
2	++	0	0	0	0	5.3
3	++	0	0	0	0	4.3
4	+	0	0	--	0	10.3

$i$	$w_{i1}$	$w_{i2}$	$w_{i3}$	$w_{i4}$	$b_i$	$\tau_i$
1	0	--	0	0	+	10.8
2	+	0	0	0	+	4.1
3	++	0	0	--	0	5.0
4	0	+	0	++	0	15.0

Table 2: Simplified version of Table 1.

a)

$i$	$w_{i1}$	$w_{i2}$	$w_{i3}$	$w_{i4}$	$b_i$	$\tau_i$	$t_i$
1	++	--	0	--	0	17.5	--
2	++	-	0	0	0	4.6	+
3	++	0	0	--	0	5.0	0
4	++	+	++	--	0	15.5	++

b)

$i$	$w_{i1}$	$w_{i2}$	$w_{i3}$	$w_{i4}$	$b_i$	$\tau_i$	$t_i$
1	0	--	0	--	0	20.3	--
2	+	--	0	0	+	5.1	+
3	++	0	-	-	+	5.6	0
4	+	0	++	--	0	16.6	++

Table 3: Simplified tables from runs with both data series using  $W_{\max} = 5$  (a) and  $W_{\max} = 10$  (b).