ELSEVIER

Computers and Chemistry 26 (2001) 51–56

Computers & Chemistry

www.elsevier.com/locate/compchem

Application of a time-delay neural network to promoter annotation in the *Drosophila melanogaster* genome

Martin G. Reese *

Berkeley Drosophila Genome Project, Department of Molecular and Cell Biology, University of California, Berkeley, CA 94720-3200, USA

Received 4 December 2000; accepted 8 May 2001

Abstract

Computational methods for automated genome annotation are critical to understanding and interpreting the bewildering mass of genomic sequence data presently being generated and released. A neural network model of the structural and compositional properties of a eukaryotic core promoter region has been developed and its application for analysis of the *Drosophila melanogaster* genome is presented. The model uses a time-delay architecture, a special case of a feed-forward neural network. The structure of this model allows for variable spacing between functional binding sites, which is known to play a key role in the transcription initiation process. Application of this model to a test set of core promoters not only gave better discrimination of potential promoter sites than previous statistical or neural network models, but also revealed indirectly subtle properties of the transcription initiation signal. When tested in the *Adh* region of 2.9 Mbases of the *Drosophila* genome, the neural network for promoter prediction (NNPP) program that incorporates the time-delay neural network model gives a recognition rate of 75% (69/92) with a false positive rate of 1/547 bases. The present work can be regarded as one of the first intensive studies that applies novel gene regulation technologies to the identification of the complex gene regulation sites in the genome of *Drosophila melanogaster*.

Keywords: Neural networks; Genome annotation; Promoter recognition; DNA sequence analysis; *Drosophila melanogaster*

1. Introduction

Recent advances in sequencing technology are making the generation of whole genome sequences common place. Capillary sequencers speed the production of raw data. Changing tactics from traditional mapping and sequencing clones in series to an integrated simultaneous mapping and sequencing approach (whole genome shotgun) has significantly reduced the amount of time it takes to completely sequence a genome. These improvements in genomic sequencing are possible because of software advances that fully exploit mapped clone constraint data and directly attack the problems that repetitive sequences cause during sequence assembly (Myers et al., 2000).

At present, several very large-scale genomic sequencing projects are complete or are expected to be complete within a few months. These initial genome sequences are from key model organisms in genetics and include five eukaryotes, *Saccharomyces cerevisiae*, *Schizosaccharomyces pombe*, *Caenorhabditis elegans*, *Drosophila melanogaster* and *Arabidopsis thaliana*, as well as draft human sequence. In a few years, sequencing new genomes and individuals will become routine practice. These raw data are not immediately useful and

* Corresponding author. Present address: ValiGen, Tour Neptune, 92086 Paris-La-Défense. Tel.: +33-1-4767-6600; fax: +33-1-4906-0715.

E-mail address: mgreese@lbl.gov (M.G. Reese).

0097-8485/01/$ - see front matter
PII: S0097-8485(01)00099-7

interpreting them places major demands on the field of computational biology.

The development and application of a novel neural network system to recognize eukaryotic polymerase II promoters in the annotation of the *D. melanogaster* genome are presented. A time-delay neural network (TDNN) is developed, an architecture that was originally introduced in speech recognition (Waibel et al., 1989; Lang and Waibel, 1990), to model the complex sequence structure of a transcription start site. The transcription start site (TSS) is the location upstream of a gene where the polymerase II protein binds to the genomic DNA and initiates the transcription process. The entire region around the transcription start site is called a promoter.

A typical polymerase II promoter consists of multiple functional binding sites that are involved in the transcription initiation process. Separate neural networks for these individual binding sites (TATA box and initiator (*Inr*)) are trained and integrated into a time-delay neural network. Such an architecture is well suited to model this complex sequence structure because it allows for variable spacing between functional sites (equivalent to different time points in speech recognition), a feature common to polymerase II promoters.

These promoters have a very complex structure (for reviews see: Pugh and Tjian, 1992; Pugh, 1996; Yokomori et al., 1998; Kornberg, 1999) consisting of these multiple DNA binding sites for transcription factors. Some of these sites enhance transcription and some other repress transcription. The nucleotide pattern of the sites is often related to the strength of binding. In addition to these core promoter elements in the vicinity of the transcription start site, there exist long-range interactions through so called enhancer sites. Therefore, current methods to model these promoters are pruned for a high rate of false positives and the task of promoter recognition can be seen as one of the most difficult in the field of DNA sequence analysis.

2. Methods

2.1. Time-delay neural networks

For promoter modeling, a special neural network is chosen, the TDNN architecture developed by Waibel et al. (1989). This architecture was originally designed for processing speech sequence pattern in time series with local time shifts. The usual way of transforming sequence patterns into input activity pattern is the extraction of a subsequence using a fixed window. This window is shifted over all positions of the sequence and the subsequences are translated into input activities. The network produces an output activity or score for each input subsequence.

The following two promoter specific features have to be learned:

- The network has to recognize subsequences that may occur at non-fixed positions in the input window. Therefore the network has to learn that the subsequence is a feature independent of shifts in its position.
- The network has to recognize features even when those features appear at different relative positions. This situation arises in cases where different subsequences occur in the input window with different relative distances. This happens very frequently in genomic sequences when one or more elements (nucleotides) are inserted or deleted in a given promoter.

The TDNN architecture addresses both problems by imposing certain restrictions on the network topology and by the way in which weights are updated. Hidden units are connected to a limited number of input units that represent a consecutive pattern in the input window. These hidden units have a *receptive field*, that is, they are only sensitive to a part of the input window. The important restriction is that the same *receptive field* has to be present at each position in the input exactly once. If the input window contains, for example, ten

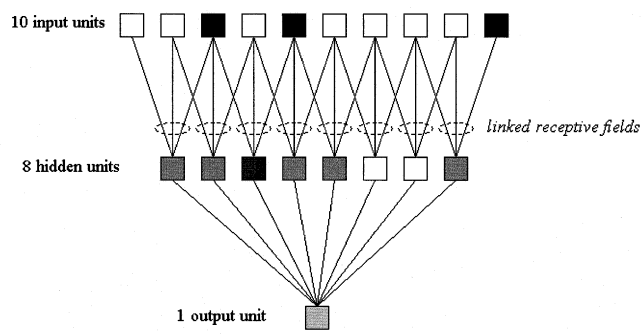

Fig. 1. An example of one-layer time-delay neural network. The small squared boxes symbolize the neurons. The input layer contains ten input units. Each window of three units is connected with one hidden unit through a linked receptive field (marked by little circles). Finally, all hidden units are connected with the output unit.

Table 1
NNPP prediction performance on the four-fold cross-validated test dataset

% Promoters recognized	TATA box FP-rate (CC)	Initiator FP-rate (CC)	Combined two-layer TDNN (CC)	Threshold (0–1) for combined TDNN	Multi-layer perceptron FP-rate (CC)
10	0.2% (0.36)	0.8% (0.28)	0.0% (0.38)	0.99	0.2% (0.35)
20	0.3% (0.45)	2.7% (0.27)	0.1% (0.38)	0.97	0.3% (0.45)
30	0.5% (0.52)	7.0% (0.28)	0.3% (0.50)	0.92	0.8% (0.48)
40	0.9% (0.56)	10.6% (0.26)	0.4% (0.60)	0.85	1.9% (0.50)
50	1.3% (0.62)	18.7% (0.25)	1.0% (0.65)	0.70	3.7% (0.51)
60	3.8% (0.60)	33.0% (0.21)	3.1% (0.61)	0.38	9.9% (0.44)
70	7.2% (0.57)	45.5% (0.18)	5.3% (0.58)	0.20	16.1% (0.40)
80	22.3% (0.39)	60.5% (0.17)	12.5% (0.52)	0.12	45.5% (0.23)

False positive (FP) rates and correlation coefficients (CC) are averaged over the four-cross validated sets.

positions and a *receptive field* covers a subsequence of three positions, there must be eight hidden units with the same *receptive field* (see Fig. 1). Since the corresponding weights in all copies of a *receptive field* are forced to have the same values, these hidden units are said to have *linked receptive fields*. In neural network terminology this is also known as *weight sharing*. Each hidden unit is called a *feature unit* because it will recognize a certain feature in the input window irrespective of its relative position. During learning, the partial derivatives of corresponding weights in *linked receptive fields* are calculated separately since these hidden units with their *receptive fields* at different positions in the input window get different activation. To adapt a *receptive field*, the weight update is averaged over all copies of a weight. This average update is then applied to all copies of that weight. In this way, it is ensured that the copies of a *receptive field* remain identical for a given feature. In the basic TDNN architecture the hidden layers (feature units) are connected to the output layer in a standard feed-forward way (Fig. 1). Training is performed using a modified backpropagation algorithm.

There are several successful applications of TDNNs in speech recognition (Waibel et al., 1989) and the recognition of handwritten characters (Lang and Waibel, 1990). These references include a detailed description of the time-delay architecture.

2.2. Implementation of the core-promoter time-delay neural network model (NNPP)

Using the time-delay architecture described above, two distinct neural networks, one for the TATA box and one for the *Inr*, were trained. An input window of 30 bp (-40 to -10) for the TATA box neural network and a window of 25 bp (-14 to $+11$) for the *Inr* network are selected. The window sizes were chosen so that the consensus sequences for both binding sites are included. The two signals occur at varying distances relative to the TSS.

The two time-delay neural networks were trained independently. It was experimentally determined that a receptive field size of 15 bp performed the best. For the TATA network, this leads to a total of 120 input units (30 bp) and 60 weights (4×15) for each unit in the hidden layer. The *Inr* network has 100 input units (25 bp) and also 60 weights (4×15) for each unit in the hidden layer.

The weights of the receptive fields for both of the two networks were initialized using the weight matrices from the literature to 'push' them to recognize particular signals. The TATA box weight matrix was taken from Bucher (1990), and the *Inr* weight matrix from Penotti (1990). These initializations were ideal to train the TDNNs to recognize the appropriate signals in the sequence (i.e. the TATA box time-delay network was forced to train only on the TATA box pattern at approximately -20 bp). The results of both networks can be seen in Table 1 and are discussed below.

2.3. Incorporation of feature detector networks into the final TDNN

To combine the above described individual feature detector neural networks for TATA and *Inr*, we use a two-layer time-delay neural network. The input to this final TDNN consists of 51 bp, spanning the transcription start site from position -40 to $+11$ and including the TATA box and the *Inr*. The hidden layers from the two previously trained single-feature time-delay neural networks are copied into the combined TDNN and training is carried out. The resulting neural network maps high order correlation between the different features and their relative distance into a complex weight matrix. A snapshot of the trained two-layer (TATA and *Inr*) TDNN is shown in Fig. 2. The weights from the hidden layers can be interpreted as the preferred position for an individual element in the input window.

All neural networks were implemented, integrated and tested using the Stuttgart Neural Network Simulator Software toolkit (Zell et al., 1999). The networks were then implemented in the neural network for promoter prediction (NNPP) program. This program is publicly accessible through a World Wide Web server (http://www.fruitfly.org/seq_tools/promoter.html).

3. Results

3.1. Application of NNPP to a cross-validated set of promoters

Table 1 shows the prediction results for the two single feature time-delay neural networks, the TATA box feature detector (column 2), the *Inr* feature detector (column 3) and the two-layer TDNN, which incorporates both (column 4 and 5). The results are averaged over four cross-validated test sets produced from the complete dataset of 429 promoters (promoter dataset including the cross-validation at http://www.fruitfly.org/sequence/human-datasets.html). The correlation coefficient is calculated as defined originally by Matthews (1975) and later adapted to the problem of gene finding evaluation by Burset and Guigó (1996) as:

$$CC = \frac{(TP \times TN) - (FN \times FP)}{\sqrt{(TP + FN) \times (TN + FP) \times (TP + FP) \times (TN + FN)}}$$

As can be seen from Table 1, the performance of the feature detecting networks used in isolation is rather poor. The TATA box network has the better performance of the two, since over 60% of the vertebrate promoters contain a TATA box. The predictive power of the initiator network is weaker because there is no real consensus sequence for vertebrate *Inr*s. The TATA box network recognizes on average 64 (60%) of the 107 promoter sequences in each test set (four-fold cross-validated) with an average of 38 (3.8%) false positive predictions. If we adjust the threshold so that on average 75 (70%) of the promoters are predicted correctly, there are 72 (7.2%) false positive predictions. The *Inr* neural network can only detect 11 (10%) of the promoters, with a false positive rate of 0.8%. The combination of both neural networks increases the prediction rate. If on average in the four cross-validated sets 54 (50%) promoters are correctly predicted, the false positive rate drops down to 1.0% (ten coding DNA regions wrongly predicted as promoters; correlation coefficient of 0.65), but that is similar to the TATA-only results. Even if 75 (70%) promoters are correctly predicted, the

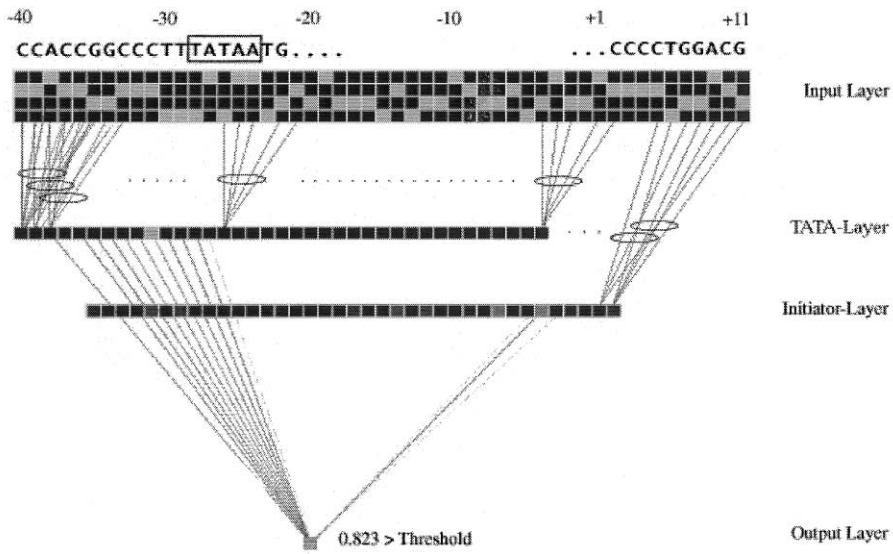

Fig. 2. The trained two-layer time-delay neural network. The small squared boxes symbolize the neurons. The input layer is on top with the window 'reading in' the DNA sequence. The receptive fields, indicated with a circle grouping connections from the input layer to the two hidden layers (TATA and *Inr*), show the structure of the time-delay connections. Both hidden layers connect to the single output neuron on the bottom. For clarity, only strong weights are shown. For example, the only significant weights shown from the TATA-layer to the output unit are the ones that localize the position of the TATA box at the beginning of the input window (below CCACCGG). The TATA box is boxed. This test sequence of CCACC···GGACG received a score of 0.823 from NNPP.

average number of false predictions is only 53 (vs. 72 for TATA alone). At a threshold of 0.12, 80% of the promoters predicted, the number of false positive predictions goes up to 125 (12.5%). Twenty-one (19.6%) promoter sites on average in the test sets cannot be predicted at all using this two-layer neural network.

For comparison, the results for a 'standard' feed-forward backpropagation neural network with one hidden layer trained on the same datasets are shown in the last column of Table 1. The number of hidden units and the number of training cycles were optimized the same way as for the time-delay neural network. The results show the superiority of the two-layer TDNN. At a threshold that gives 64 (60%) correct predictions, the number of false positive predictions is more than three times higher for the standard network (99 (9.9%) false predictions) than for the two-layer TDNN (31 (3.1%) false predictions). This shows that reducing the parameter space from 3091 adjustable weights in the standard network to 169 in the TDNN, improves the prediction accuracy on a limited training dataset (419 promoter sequences).

3.2. Application of NNPP in Drosophila melanogaster: the Adh region

To apply the two-layer time-delay neural network to contiguous genomic sequence, a window of 51 base pairs is shifted over the sequence base by base. In this way, a score is computed for every position in the sequence. These individual scores are subsequently smoothed by a simple but efficient function, which selects the position of the highest score in a window of ten neighboring positions as the final prediction. The smoothing function is implemented as a post-processing procedure and is part of the final NNPP program.

To test the accuracy of NNPP in *Drosophila melanogaster*, NNPP was applied to the 2.9 Mbase genomic sequence of the *Adh* region (Ashburner et al., 1999) (dataset at http://www.fruitfly.org/GASP1/). A careful promoter analysis in this region (Reese et al., 2000a) resulted in high quality full-length cDNA alignments for 92 genes out of the original 222 gene annotations.

In Table 2 the NNPP results are reported on this test set of genes in the *Adh* region (Ashburner et al., 1999) in comparison to CoreInspector (Scherf et al., 2000) and MCPromoter (Ohler et al., 1999), both evaluated in a recent annotation experiment (Reese et al., 2000a). Although NNPP is far from accurate, this test shows good results similar to those in a review by Fickett and Hatzigeorgiou, 1997. In this paper they reported a recognition rate for NNPP of 54% of the known promoters at a threshold of 0.8. In *Adh*, the same threshold identifies 69 or 75% of the total of 92 annotated promoters with a false positive rate of 1/547, similar to the rate of 1/460 reported in Fickett and Hatzigeorgiou (1997). It has to be noted that Fickett and Hatzigeorgiou used both strands to calculate the false positive rate while for *Adh* only the gene strand was used. If one applies a more stringent threshold of 0.97, 35 of the 92 promoters are still recognized with a much lower false positive rate of 1/2416. The higher classification rate in the *Adh* region might be due to the small number of promoters or the difference in composition in the Fickett and Hatzigeorgiou (1997) dataset.

4. Discussion

The presented tool is an artificial neural network model using a time-delay network architecture. This network has two feature layers: one for the TATA box and one for the *Inr* (initiator). The output of both feature layers is combined in a time-delay neural network. It is shown that such a neural network detects the TATA box and the *Inr* and is insensitive to their relative spacing. It is therefore an excellent model for

Table 2
Evaluation of promoter prediction systems in the *Adh* region

	Program name	Identified TSS	Rate of false predictions in annotated *Adh* region (total of 853 180 bases)
From (Reese et al., 2000a)	CoreInspector	1 (1.0%)	1/853 180 (0.00012%)
NNPP	MCPromoter v2.0	31 (33.6%)	1/2437 (0.041%)
	NNPP ($t = 0.99$)	20 (21.7%)	1/6227 (0.016%)
	NNPP ($t = 0.97$)	35 (38.0%)	1/2416 (0.041%)
	NNPP ($t = 0.80$)	69 (75.0%)	1/547 (0.183%)
	NNPP ($t = 0.70$)	80 (86.9%)	1/400 (0.250%)

The table shows the results of the 'search by signal' program (CoreInspector) and 'search by content' program (MCPromoter) from the experiment of Reese et al. (2000a) and the prediction sets from NNPP with different thresholds. The rate of false positives is shown for the sequence where cDNA annotations define the region as non-promoter.

the compositional sequence properties of a eukaryotic core promoter region. The discriminative ability of such a model for the short core promoter region of -40 to $+11$ bases spanning the transcription start site is so strong that this model can be used to predict an entire promoter in genomic DNA. These results show that the highest information content in a promoter region exists in the core promoter region.

The program is able to predict over 70% of transcription start sites in genomic DNA when used with the default parameters. The false positive rate calculated on the *Adh* region in *Drosophila melanogaster* is 1/547 bases. Matthew's correlation coefficient (Matthews, 1975) is 0.58. Thirty percent of all promoter sequences remain undetected and this is probably due to the non-local structure of the promoter region, where initiation control elements can occur at positions many kilo bases distant from the transcription start site.

The NNPP program can easily be extended to incorporate novel information as it becomes available. Other known promoter elements such as the CAAT box, GC box, DPE (downstream promoter element; so far only known to exist in *Drosophila*), and conserved transcription factor binding sites can also be used within the existing framework. The extended parameter space of such an extended model would require more data for training.

The positive results obtained using the time-delay architecture will hopefully lead to more widespread application of neural networks to similarly complex problems in molecular biology, such as the detection of splice sites and protein–protein interaction motifs.

For the application to complete genome annotations the NNPP code needs to be integrated into a more global annotation system such as Genie (Kulp et al., 1996; Reese et al., 1997, 2000b).

Since the NNPP program is made available on the World Wide Web it has been widely used in the scientific community to hypothesize about potential transcription start sites. Recently the program was used to correct an important *C. elegans* gene, *unc-86*, that encodes a POU IV class transcription factor. In this study, the transcription start site prediction by NNPP was experimentally verified (Roehrig, 2000, personal communication).

This example demonstrates how useful a program like NNPP can be in the right context. It is clear that a program cannot substitute for the final experimental proof but the example shows that it can give direction and guidance for such experiments to verify computational predictions.

Acknowledgements

M.G.R. would like to thank Gerry M. Rubin and Anette Preiss for their continuing scientific support and advice for developing this system as part of his doctoral thesis. M.G.R. was supported by NIH grant HG00750.

References

Ashburner, M., Misra, S., Roote, J., Lewis, S.E., Blazej, R., Davis, T., Doyle, C., Galle, R., George, R., Harris, N., Hartzell, G., Harvey, D., Hong, L., Houston, K., Hoskins, R., Johnson, G., Martin, C., Moshrefi, A., Palazzolo, M., Reese, M.G., Spradling, A., Tsang, G., Wan, K., Whitelaw, K., Kimmel, B., et al., 1999. Genetics 153 (1), 179.

Bucher, P., 1990. J. Mol. Biol. 212 (4), 563.

Burset, M., Guigó, R., 1996. Genomics 34 (3), 353.

Fickett, J.W., Hatzigeorgiou, A.G., 1997. Genome Res. 7 (9), 861.

Kornberg, R.D., 1999. Trends Cell Biol. 9 (12), M46.

Kulp, D., Haussler, D., Reese, M.G., Eeckman, F.H., 1996. Ismb 4, 134.

Lang, K.J., Waibel, A.H., 1990. Neural Netw. 3, 23.

Matthews, B.W., 1975. Biochim. Biophys. Acta 405 (2), 442.

Myers, E.W., Sutton, G.G., Delcher, A.L., Dew, I.M., Fasulo, D.P., Flanigan, M.J., et al., 2000. Science 287 (5461), 2196.

Ohler, U., Harbeck, S., Niemann, H., Noth, E., Reese, M.G., 1999. Bioinformatics 15 (5), 362.

Penotti, F.E., 1990. J. Mol. Biol. 213 (1), 37.

Pugh, B.F., 1996. Curr. Opin. Cell Biol. 8 (3), 303.

Pugh, B.F., Tjian, R., 1992. J. Biol. Chem. 267 (2), 679.

Reese, M.G., Eeckman, F.H., Kulp, D., Haussler, D., 1997. J. Comput. Biol. 4 (3), 311.

Reese, M.G., Hartzell, G., Harris, N.L., Ohler, U., Lewis, S.E., 2000a. Genome Res. 10 (4), 483.

Reese, M.G., Kulp, D., Tammana, H., Haussler, D., 2000b. Genome Res. 10 (4), 529.

Scherf, M., Klingenhoff, A., Werner, T., 2000. J. Mol. Biol. 297 (3), 599.

Waibel, A.H., Hanazawa, T., Hinton, G.E., Shikano, K., Lang, K.J., 1989. IEEE Trans. Acoust. Speech Signal Process. 37 (3), 328.

Yokomori, K., Verrijzer, C.P., Tjian, R., 1998. Proc. Natl. Acad. Sci. USA 95 (12), 6722.

Zell, A. et al., 1999. Stuttgart Neural Network Simulator (SNNS) 4.2. http://www-ra.informatik.uni-tuebingen.de/SNNS/.