BIOINFORMATICS

Vol. 15 no. 5 1999
Pages 356–361

Promoter2.0: for the recognition of PolII promoter sequences

Steen Knudsen

Center for Biological Sequence Analysis, The Technical University of Denmark, DK-2800 Lyngby, Denmark

Received on August 28, 1998; revised on December 16, 1998; accepted on February 4, 1999

Abstract

Motivation: *A new approach to the prediction of eukaryotic PolII promoters from DNA sequence takes advantage of a combination of elements similar to neural networks and genetic algorithms to recognize a set of discrete subpatterns with variable separation as one pattern: a promoter. The neural networks use as input a small window of DNA sequence, as well as the output of other neural networks. Through the use of genetic algorithms, the weights in the neural networks are optimized to discriminate maximally between promoters and non-promoters.*

Results: *After several thousand generations of optimization, the algorithm was able to discriminate between vertebrate promoter and non-promoter sequences in a test set with a correlation coefficient of 0.63. In addition, all five known transcription start sites on the plus strand of the complete adenovirus genome were within 161 bp of 35 predicted transcription start sites. On standardized test sets consisting of human genomic DNA, the performance of Promoter2.0 compares well with other software developed for the same purpose.*

Availability: *Promoter2.0 is available as a Web server at http://www.cbs.dtu.dk/services/promoter/*

Contact: *steen@cbs.dtu.dk*

Introduction

Eukaryotic PolII promoters, which provide start sites for the transcription of protein-coding genes, are characterized by a large number of binding sites for transcription factors, normally upstream of the initiation site (Johnson and McKnight, 1989). Each gene is characterized by a specific arrangement of transcription factor binding sites. When all transcription factors are bound to form a productive complex, transcription is initiated (Sawadogo and Sentenac, 1990). Because of the large number of degenerate binding sites, it has been difficult to identify promoters accurately based on sequence information alone. For a detailed review of the underlying biology, previous efforts, and their limitations, see Fickett and Hatzigeorgiou (1997) and Pedersen *et al.* (1998). A successful algorithm for promoter recognition will be useful in analyzing results from the Human Genome Project.

The present work uses a novel method, which has similarities to neural networks (Baldi and Brunak, 1998) and genetic algorithms (Holland, 1975; Koza, 1992), to recognize a set of discrete subpatterns, with variable separation, as one pattern. In the present example, this pattern is a promoter. The neural networks (neural network terminology will be used for convenience) use as input a small window of DNA sequence, as well as the output of other neural networks. Through the use of an optimization approach similar to genetic algorithms, the weights in the neural networks are optimized to discriminate maximally between promoters and non-promoters.

An analogy to the molecular transcription apparatus can be made by comparing the individual neural networks to transcription factors, where DNA input to the neural network models the DNA-binding domain of the transcription factor and input from other neural networks models protein–protein interactions. Genetic algorithms correspond to the evolution toward a system that optimally discriminates between a promoter and a non-promoter sequence.

Methods

The neural networks used in this work are similar to perceptrons (Minsky and Papert, 1969). Input to each neuron is summed and subjected to a threshold. DNA sequences are presented to the input neurons as orthogonal vectors of binary numbers to represent A, C, G and T (Figure 1). The present neural networks differ from the standard implementation of perceptrons in that weights are optimized using a random optimization algorithm. Hidden neurons, and output neurons, receive input from all other neurons as well as from other neural networks. Each connection between two neurons is unidirectional.

It is not known *a priori* where in each training example the transcription factor binding sites of interest reside. That is one reason why the standard back-prop algorithm (Rumelhart *et al.*, 1986) for training of neural networks is not well suited to this problem. The training target is unknown. Using

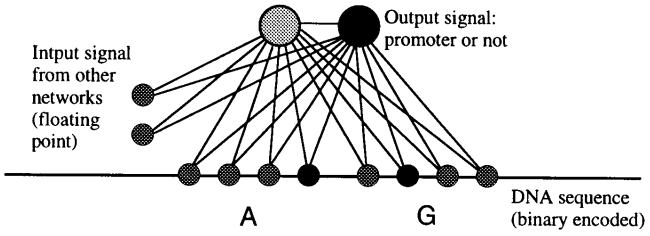

Fig. 1. Schematic illustration of a neural network. This network consists of input neurons, hidden neurons and output neurons. Here, the input neurons are two nucleotides encoded in a 4-bit scheme. Each interaction is depicted by a thin line and has a modifiable strength associated with it. Each hidden or output neuron also has a connection with a modifiable strength to an input neuron that receives output from each of the previous neural networks (two network inputs shown here) scanned over the same sequence. The input to these units is the maximum activation multiplied by a linear distance function normalized to range between zero and one.

a random optimization algorithm instead, optimization is performed on the ability to classify into promoter sequence or not, irrespective of where in the test sequence the signals reside.

A given network, with a given input window size, and a given set of connection strengths (represented by weights), scans over the sequence of a training example, typically 200 or 300 bp in size. For each position in this sequence, the activity of the output neuron is stored. When all windows have been presented, the activity and position of the window with the highest activity are stored. This represents the 'binding site', if any, of the first neural network, or 'transcription factor'. Next, another neural network with a different set of weights scans the same DNA sequence. This new network, in addition to the input window, has one input representing the highest activity for each of all previous networks, if any, on the same sequence, multiplied by a separation function. This linear separation function is normalized to be zero at maximum distance within the sequence and one at minimum distance. Hence the order of sites is not recorded, only their separation. Transcription factor binding sites that can be found on either strand and that do not possess dyad symmetry would have to be recognized by separate networks.

Neural networks are optimized (trained) one at a time by, at each generation, changing a randomly chosen individual weight by a random amount and keeping the new weight if it improves performance, and discarding it if not. More advanced features such as cross-over, commonly used in genetic algorithms, are not used in this algorithm. Performance is evaluated as the correlation coefficient (and/or the sum of squared errors) in classifying DNA sequences as promoters or not. After training of all networks that comprise the recognition algorithm, the final performance is evaluated on a separate test set.

Training and test sets were prepared from a publicly available database of promoter sequences. Bucher (Bucher and Trifonov, 1986) has prepared a set of eukaryotic promoter sequences for which biochemical evidence for the cap site assignment exists. A set of independent vertebrate PolII promoters was extracted from this set.

Positive and negative examples of promoters were generated by assigning 200 bp upstream of the cap site as the promoter, and assigning 200 bp downstream of the cap site as the negative control. A training set and a test set, each of 100 promoters, were generated. The choice of 200 bp upstream of the cap site as promoter sequence, although limiting in several respects, was based on the following facts. (i) An upstream sequence of 200 bp is available for many more known cap sites than an upstream sequence of, for example, 1000 bp. (ii) The most abundant known promoter sequence features (described in the Results) are predominantly located in this interval. The choice of 200 bp downstream of the cap site as the negative control is convenient, although not representative of non-promoter sequences. In addition, this region can contain secondary initiation sites or additional transcription factor binding elements. For these reasons, a second training and test set were generated using the region of –200 to +100 bp relative to the cap site as the positive (promoter) set and +500 to +800 as the negative (non-promoter) set. In addition, a realistic example of the adenovirus 2 full genome (ADRCG; locus J01917 of GenBank) was used as a final test. The algorithm was implemented in C language and run on a Sun Sparcstation ELC and an SGI Challenge.

Results

Four neural networks, each consisting of only one hidden and one output neuron with input from six nucleotides (i.e. $6 \times 4 = 24$ input units, of which eight are shown in Figure 1), were optimized to discriminate between 100 promoter sequences of 200 bp length and 100 non-promoter sequences of 200 bp length. All interaction strengths started out as random. When the sum of squared errors improved by <1% over 1000 generations, optimization of one network was stopped and optimization of the next network began, holding the strength matrix of the first neural network constant. When optimization of all four neural networks had finished after ~15 000 generations, 90% of the 200 sequences in the training set were correctly classified as promoter or non-promoter.

These four neural networks were then applied to a test set of the same size as the training set. The performance of one, two, three or all four of the neural networks on the test set can be summarized in the correlation coefficient, which is 0.34, 0.50, 0.62 and 0.56, respectively. A correlation coefficient of zero means no prediction ability (random classification), whereas a

correlation coefficient of one means correct classification of all examples. Optimal performance is achieved with three neural networks. The performance decreases when the fourth, last evolved, neural network is added. Figure 2 shows the effect of varying the classification cut-off on the number of false-positive predictions and number of true-positive predictions. By choosing a cut-off, one can select a desired balance between true positives and false positives.

Another experiment entailed starting with neural network–substrate interactions preset to reflect transcription factor binding to known sites: neural network 1: TATA box (weights set to give maximum response for sequence TATAAA); neural network 2: cap site (weights set to TCA), neural network 3: CCAAT box (weight set to CCAATC); neural network 4: GC box (weights set to GGGCGG). The performance after ~15 000 generations of optimization of the network ensemble was tested on the test set. The performance of one, two, three or all four of the neural networks on the test set was 0.62, 0.61, 0.63 and 0.61, respectively. Figure 2 shows the effect of varying the classification cut-off on the balance of true-positive and false-positive predictions.

Information about what features of promoters the neural networks detect can be gathered in much the same way as one would gather information on the binding of real transcription factors: a systematic presentation of sequences and registration of the resulting output. All possible 6mer sequences (4096 total) were presented to the four random start and the four preset neural networks individually (no interaction between neural networks). Table 1 shows the ten 6mers giving rise to the highest score for each of the eight evolved neural networks.

Figure 3 shows the positional distributions of sites recognized by the four randomly evolved neural networks on the sum of promoters from the training and test sets.

The apparent dissimilarity of the sequence features recognized by the random start neural networks and the preset start neural networks suggests that an improved promoter recognition could be obtained by combining the eight neural networks. Figure 2 shows the performance of all eight neural networks in concert using varying cut-offs.

Table 1. Sequences with the strongest network response

Network 1		Network 2		Network 3		Network 4	
Score	Sequence	Score	Sequence	Score	Sequence	Score	Sequence
(A) Evolution started with random interaction strengths							
0.870	AAAACG	4.725	ACACAG	3.294	TATATA	1.001	GGCTGA
0.834	AAAACA	4.349	GCACAG	3.189	TATAAA	0.990	GGCTGC
0.813	AAAAGG	4.317	ATACAG	2.402	TAAAAA	0.932	GGTTGA
0.776	AAAAGA	4.078	CCACAG	2.339	TATGTA	0.921	GGTTGC
0.692	ATAACG	4.033	ACATAG	2.269	TATTTA	0.888	GACTGA
0.685	GAAACG	3.944	ACACAA	2.196	TAGATA	0.877	GACTGC
0.656	ATAACA	3.941	GTACAG	2.091	CATAAA	0.855	CGCTGA
0.649	GAAACA	3.891	ACACAT	2.081	TAGAAA	0.844	CGCTGC
0.635	ATAAGG	3.856	GCACAA	2.063	TATAAC	0.824	GTCTGA
0.631	AAAACC	3.828	ATACAA	2.028	TATTAA	0.820	GATTGA
(B) Evolution started with preset interaction strengths							
0.870	TATAAA	43.08	TTTTCA	58.93	CCAATC	67.26	GGGCGT
0.720	TTTAAA	42.89	TCTTCA	56.35	CCAATA	66.39	GGGCGG
0.701	TATAAG	42.61	TTCTCA	53.93	CCAATG	65.79	AGGCGT
0.695	CATAAA	42.57	ATTTCA	52.99	CCATTC	65.09	GGGCAT
0.672	TATAAC	42.41	TCCTCA	52.34	CCAGTC	64.91	AGGCGG
0.664	GATAAA	42.37	ACTTCA	50.40	CCATTA	64.22	GGGCAG
0.598	TATAAT	42.24	CTTTCA	49.76	CCAGTA	63.62	AGGCAT
0.553	TAAAAA	42.10	ATCTCA	48.89	CCGATC	62.74	AGGCAG
0.551	TTTAAG	42.05	CCTTCA	47.98	CCATTG	62.49	GGGAGT
0.545	CTTAAA	41.98	TTTGCA	47.63	CCACTC	61.62	GGGAGG

Table 2. Testing the algorithm on the adenovirus 2 genome

Predicted transcription start sites			True site
Position	Score	Likelihood	
600	1.0630	Highly likely prediction	498 E1a
1800	1.1780	Highly likely prediction	1699 E1b
2200	0.760	Marginal prediction	
2900	0.5960	Marginal prediction	
3700	1.1360	Highly likely prediction	3576 IX
4400	0.6450	Marginal prediction	
6200	1.2990	Highly likely prediction	6039 major late
7600	0.5630	Marginal prediction	
8200	0.5380	Marginal prediction	
9300	0.6070	Marginal prediction	
10 800	0.6820	Marginal prediction	
11 800	0.5440	Marginal prediction	
13 400	0.6430	Marginal prediction	
14 200	0.5620	Marginal prediction	
15 200	0.6480	Marginal prediction	
16 000	0.5640	Marginal prediction	
16 700	0.6350	Marginal prediction	
17 200	0.6210	Marginal prediction	
18 300	1.1010	Highly likely prediction	
19 200	0.5890	Marginal prediction	
20 300	0.6330	Marginal prediction	
20 800	0.6010	Marginal prediction	
21 500	0.6520	Marginal prediction	
21 900	0.6040	Marginal prediction	
22 200	0.5680	Marginal prediction	
22 600	1.0560	Highly likely prediction	
23 100	0.6100	Marginal prediction	
24 400	1.1180	Highly likely prediction	
25 700	0.5930	Marginal prediction	
26 300	0.5740	Marginal prediction	
27 700	0.6510	Marginal prediction	27 610 E3
28 400	0.6470	Marginal prediction	
29 400	1.1360	Highly likely prediction	
33 200	1.2200	Highly likely prediction	
33 700	0.6930	Marginal prediction	

The window size of neural networks with preset weights was increased to 15 nucleotides from six nucleotides. This time the weights were hard-coded (no mutation possible) to previously reported weight matrices (Bucher, 1990; based on 502 promoter sequences). Thus, to compensate for the limited size of the data set, only evolution of the interaction between the four neural networks was allowed. After 1098 generations, the four neural networks were tested on a realistic set: the complete adenovirus 2 genome. To conduct this test, the algorithm had to be modified from classifying windows to classifying continuous sequence. The 35 937 bp genome (plus strand only) was divided into 50% overlapping windows of 200 bp each (i.e. a total of 359 windows), and the four neural networks assigned each window a score. Overlapping or adjacent windows having scores above the 0.5 cut-off were merged into promoter regions and assigned the highest score of the constituting windows. The 3′ end of such a region of overlapping or adjacent windows with a score above cut-off was used as the predicted transcriptional start site (TSS). Table 2 shows the 35 predicted TSSs. Of the 35 predicted TSSs, five are within 161 bp of the TSSs of known promoters in the Adenovirus genome. Among the 35 predictions, nine are labeled highly likely based on their score (>1.0; predictions > 0.8 are labeled medium likely and pre-

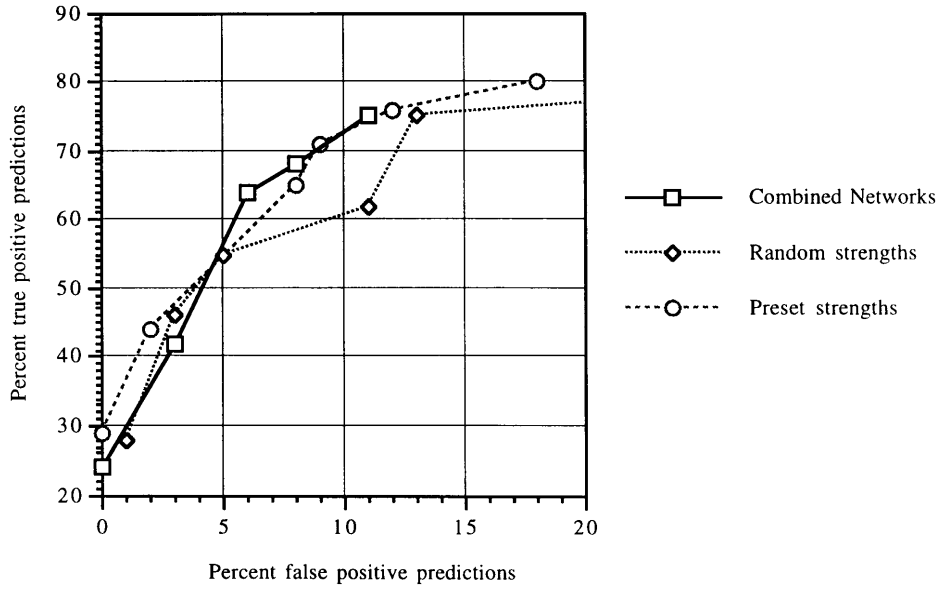

Fig. 2. Correlation between the percent of true-postive predictions and the percent of false-positive predictions on the test set at varying classification cut-offs. Diamonds, four networks with random weights; circles, four networks with preset weights; boxes, a combination of all eight networks (using their average output value).

dictions > 0.5 are labeled marginal). Of the nine highly likely predictions, four are within 161 bp of true TSSs. This performance was better than that of the random weight networks which were tested on the adenovirus 2 genome as well. The random strength networks predicted 40 TSS above cut-off. Among these, four were within 261 bp of one of the five true TSS.

The successfully trained algorithm with window size 15, modified to read complete DNA sequences in fasta format and to output predicted TSSs, was called Promoter2.0. An earlier version, Promoter1.0, predicted promoter regions instead of TSSs and contained a bug that caused it to fail to merge adjacent windows. Promoter1.0 was incorporated in the GeneID server (Guigó *et al.*, 1992; Knudsen *et al.*, 1993), and was evaluated in a large-scale test of promoter prediction software on a set of 18 human genomic DNA sequences (Fickett and Hatzigeorgiou, 1997). Promoter1.0 compared favorably to eight other software packages tested. Promoter2.0, which was completed after the published test, was tested on the same 18 genes, kindly provided by Jim Fickett. Because of the bug fix, Promoter2.0 performed slightly better than Promoter1.0, still using the evaluation criteria selected by the authors. Both predicted 10 out of 24 known TSS, but Promoter2.0 made a total of 53 predictions, whereas Promoter1.0 made 61.

The results presented so far were obtained using a version of EPD available in 1993. Training of the random and preset strength networks was repeated on a new, larger set of promoters extracted from EPD V 50 available in 1997 (Rouayda *et al.*, 1997), and using windows of –200 to +100 bp relative to the cap site as the positive (promoter) set and +500 to +800 as the negative (non-promoter) set. This did not result in any improvement of the performance, either on the independent test set or on the Fickett set (data not shown).

Discussion

The successful prediction of promoters with reasonable accuracy shows that the novel method has promise as an approach to modeling biological systems in general.

In the present example, an attempt is made to locate multiple transcription factor binding sites that may be present in a region.

In the example starting with random weights, neural networks 1, 2 and 3 all appear to recognize part of the TATA box, located at –25 to –35 relative to the transcription initiation site. It is not apparent why the sequences recognized by neural network 2 are part of the TATA box, but the approach of using a neural network ensemble allows the detection of internal correlations between different subsequences of the TATA box. So neural networks 1, 2 and 3 could recognize either different subclasses of TATA boxes or different subsequences of TATA boxes. The sequences recognized by neural network 4 do not seem to have any fixed position within the promoter region.

The experience gained from the present example shows that the evolution algorithm is robust and arrives at a good model based on the available data set. The limited size of the

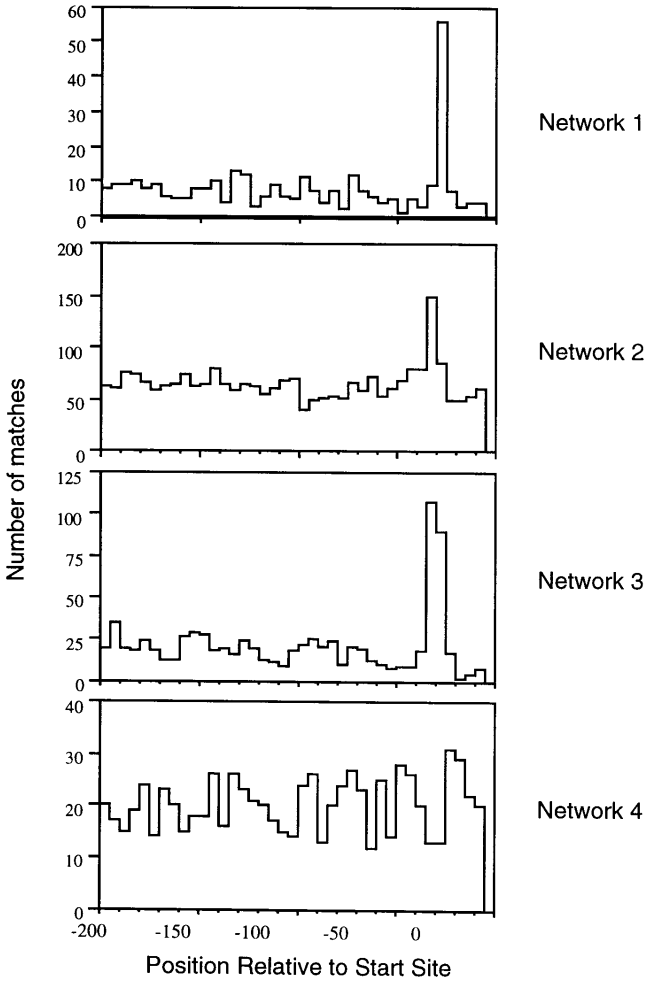

Fig. 3. Positional distributions of the sequences recognized by the four random weight networks. All windows scoring higher than the 0.5 cut-off in the 200 bp promoter sequence upstream of the initiation site (+1) were grouped in 5 bp intervals, and the number of hits in each interval plotted.

data set used in this example limits the size and number of neural networks that can be evolved. It is well known from experience with neural networks that the ability to generalize between a training set and a test set decreases when the ratio between the number of parameters in the training set and the number of parameters in the neural network decreases (Brunak *et al*., 1991). This phenomenon is likely to explain why the performance of evolved neural networks on the test set starts to fall slightly between three and four neural networks, where the ratio between data (training) set size and number of neural networks becomes too low. By adding one neural network at a time, the present method thus automatically arrives at the optimal number of neural networks in one evolution series. The performance of the neural networks on the promoter recognition problem is likely to improve if the size of the data set is increased substantially because that would support more neural networks that could recognize more sequence features. With substantially larger data sets, the input window size could also be increased. Another possible avenue to improvement could be to use a more advanced optimization approach than the random weight selection algorithm used here.

Acknowledgement

This work was supported by a grant from the Danish National Research Foundation.

References

Baldi,P. and Brunak,S. (1998) *Bioinformatics: The Machine Learning Approach*. MIT Press, Cambridge, MA.

Brunak,S., Engelbrecht,J. and Knudsen,S. (1991) Prediction of human mRNA donor and acceptor sites from the DNA sequence. *J. Mol. Biol*., **220**, 49–65.

Bucher,P. (1990) Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. *J. Mol. Biol*., **212**, 563–578.

Bucher,P. and Trifonov,E.N. (1986) Compilation and analysis of eukaryotic PolII promoter sequences. *Nucleic Acids Res*., **14**, 10009–10026.

Fickett,J.W. and Hatzigeorgiou,A.G. (1997) Eukaryotic promoter recognition. *Genome Res*., **7**, 861–878

Guigó,R., Knudsen,S., Drake,N. and Smith,T. (1992) Prediction of gene structure. *J. Mol. Biol*., **226**, 141–157.

Holland,J.H. (1975) *Adaption in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, MI.

Johnson,P.F. and McKnight,S.L. (1989) Eucaryotic transcriptional regulatory proteins. *Annu. Rev. Biochem*., **58**, 799–839.

Knudsen,S., Guigo,R. and Smith,T. (1993) GeneID—a computer server for prediction of genes in DNA sequences. In *Proceedings from the Second International Conference on Bioinformatics*. World Scientific, New York.

Koza,J.R. (1992) *Genetic Programming*. MIT Press, Cambridge, MA.

Minsky,M. and Papert,S. (1969) *Perceptrons*. MIT Press, Cambridge, MA.

Pedersen,A.G., Baldi,P., Chauvin,Y. and Brunak,S. (1998) The biology of eukaryotic promoter prediction—a review. *Comput. Chem*., in press.

Rouayda C., Junier. T. and Bucher. P (1997) *The Eukaryotic Promoter Database EPD, Release 50*. Swiss Institute for Experimental Cancer Research, 1066 Epalinges s/Lausanne, Switzerland.

Rumelhart,D.E., Hinton,G.E. and Williams,R.J. (1986) In Rumelhart,D.E., McClelland,J.L. and the PDP Research Group (eds), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, Cambridge, MA, Vol. I: Foundations, pp. 318–362.

Sawadogo,M. and Sentenac,A. (1990) RNA polymerase II and general transcription factors. *Annu. Rev. Biochem*., **59**, 711–754.