Knowledge-Based TDNN Architectures for Features Recognition in DNA Sequences

George Potamias

Institute of Computer Science, FORTH
Heraklion, Crete, GR-71110, Greece
potamias@ics.forth.gr

Elias Papanikolaou

Dept. of Computer Science, University of Crete
Heraklion, Crete, GR-71409, Greece
papanik@csd.uoc.gr

Artemis Hatzigeorgiou

Synaptic Ltd
Science and Technology Park of Crete (STEP-C), Heraklion, Crete, GR-71110, Greece
artemis@stepc.stepc

Abstract

We present work on the effective utilization of ANNs for the recognition and location of specific regulatory elements in genomic DNA, namely promoters. In particular, we rely on the utilization of knowledge-based NN (kbNN) techniques for the automatic construction of Time Delay NN (TDNN) architectures. The prior-knowledge is based and extracted from position weight-matrices of the DNA-sequence bases. The resulted kPRNN: knowledgeable Promoter Recognition Neural Network is trained, and the knowledge used to initialize the TDNN is refined. Experimental results show a significant improvement in TDNN-based promoter recognition.

1 Introduction

One of the central problems in Bioinformatics, and especially in genome-based biology, is the discovery of organizational units - the genes, and related biological functions in the different parts of DNA sequences. Roughly, about 2% - 5% of the genome provides coding for proteins [3], [6] (it should be pointed out that the present understanding of the genome structure, and the functions of specific parts of it, is not yet complete). In this environment, the huge (steadily increasing) volume of genomic sequences made available makes the use of automated computational analysis methods and tools more than a need. Such tools may provide the genome-researchers with valuable hints about the potential structure and functioning of DNA. This, will result into more targeted research with the great potential of reducing the number of the laborious and expensive laboratory experiments.

In this context, finding protein coding in DNA sequences is of major importance, with the identification of regulatory regions like promoters, being at the center of related research activities. Promoters are parts of genomic DNA that are intimately related to the initiation of the transcription process. In the process of the initiation of transcription, specific proteins – the enzymes (RNA polymerases), attempt to bind to promoter regions. Promoters generally indicate and contain the starting point of transcription, the so-called transcription start site (TSS), and regulate the rate of initiation of transcription (see figure 1, below).

![Figure 1: Possible promoter-gene relation structure for mRNA eukaryotic gene](image)

Our work regarding the promoter recognition problem focuses on promoters in eukaryotes. A promoter in eukaryotes (polymerase II or, Poll-II promoter) is defined somewhat loosely as a portion of DNA sequence around the transcription initiation site. In eukaryotic promoters, several elements or, signals (i.e., sub-regions in the DNA sequence) are scattered within the region located near the TSS. The most prominent elements - those with reasonably high consensus, are located within the core promoter region, namely: the TATA-box, the CAAT-box, and the GG-box, together with other different transcription factors (TF) binding sites (see figure 2, below).
Most eukaryotic promoter recognition methods are mainly based on a statistical framework [5], [12]. Today’s state-of-the-art algorithms are based on: (i) content search methods - based on the statistics of short words (the oligomers) [1], [10], and (ii) signal search methods - for prominent elements around the TSS [18] or, on large collections of weight matrices - for known TF binding sites [17]. Recently, methods that rely on hidden Markov models (HMM) combine various scoring functions for the signals [16].

![Figure 2: Possible structure of a Pol-II promoter.](image)

In this paper we present a novel promoter recognition approach that builds on the concept of knowledgeable NN architectures. In particular, we rely on the utilization of knowledge-based NN (kbNN) techniques [9], [25], for the automated construction of Time Delay NN (TDNN) architectures [26], [27].

Next section presents the background and the details of the approach. In section 3 we present and discuss in detail the results of the experiments performed on a set of indicative DNA sequences. In the last section we conclude and point to further research and development directions.

2 Methodology

What makes the promoter recognition problem in eukaryotes really difficult is the complex structure of Pol-II promoter sequences. The indicative elements vary a lot, they may appear in different combinations, their relative locations with respect to the TSS are different for different promoters, and not all of these specific signals need to exist in a particular promoter (for more details on the recent state-of-the-art progress on the structure and function of Pol II promoters see the reviews in [11], [19]). So, when searching for specific elements a large number of weakly conserved promoters may be missed. As a result, the so-far available computational methods for promoter recognition are plagued by a large number of false positives or imposed restrictions on the number of predictions. Indeed, the survey of Fickett and Hatzigeorgiou on the related available methods confirms this [8].

2.1 Outline and Background of the Method

The presented kPRNN: knowledgeable Promoter Recognition Neural Network system builds (actually initializes) a TDNN architecture [27], using kbNN techniques [9], [25].

TDNN: Time-Delay Neural Networks. Due to their special architecture, TDNNs are capable of learning to classify features that are invariant regarding their spatial (or temporal) translation. These networks were initially used in the problem of phoneme recognition [26], [27] and in word recognition [13]. There are some similarities in the problems of phoneme recognition and recognition of promoters, where the latter is based on detection of the promoter components. The features that need to be detected may be on different mutual spatial (or temporal) locations on the DNA sequence. TDNN provides a convenient mechanism to make neural network system insensitive to mutual distances of relevant events. The learning procedure for TDNN can be based on back-propagation [20], or other learning procedures [27]. TDNN is exposed to a sequence of learning patterns to be able to learn invariance in the space (or time) translation of the relevant patterns. The training phase of TDNNs can be long and tedious even for shorter time patterns. However, they may achieve high prediction accuracy results.

kbNN: knowledge-based Neural Networks. The kbNN family of algorithms translates a domain theory into a promising initial neural network. It exploits the theory to define the network’s topology and also, to initialize the weights of it. The approach may be summarized into the following tasks list:

- Given:
  - A list of features used to describe the observations (objects or, examples)
  - A body of prior-knowledge that approximates a domain theory
  - A set of classified training observations
- Do:
  - Translate the domain theory into a neural network
  - Train the knowledge-based network using the pre-classified observations
  - Use the trained network to classify future observations
  - (Optionally) extract a refined domain theory [23], [24].

The TDNN derived with the kbPRNN system, is initialized on the presence of prior-knowledge that indicates what elements and which DNA regions should be explored in order to recognize a sequence as eukaryotic promoter. The prior- or, background-knowledge is based and extracted from established position weight matrices (PWM) of DNA-
sequence bases, i.e., from the scoring of the various prominent promoter elements.

A weight matrix or profile (the two terms are used synonymously) is a table of position-specific amino acid weights and gap costs. These numbers (also referred to as scores) are used to calculate a similarity score for any alignment between a profile and a sequence, or parts of a profile and a sequence. An alignment with a similarity score higher than or equal to a given cut-off value constitutes a motif occurrence. As with patterns, there may be several matches to a profile in one sequence, but multiple occurrences in the same sequences must be disjoint (non-overlapping) according to a specific definition included in the profile.

Different approaches utilize PWM information [4]. In the performed experiments we rely on the widely exploited PWMs published by Bucher [2]. The TDNN is then trained and the knowledge used to initialize it is refined. Our method resembles a linear discriminative analysis approach [14], [22], and relates to the work presented in [18].

2.2 Details of the Method
The prior-knowledge exploited concerns the four major eukaryotic promoter elements, namely the TATA-box, the cap-signal, the CCAAT-box, and the GC-box.

Input. The input-file to the kPRNN consists of four basic entries that fully specify the sequences to be searched and the related weight-matrices.

- **Start position of sequence**: an integer that denotes the named start of the input sequences, relative to the TSS.
- **End position of sequence**: an integer that denotes the named end of the input sequences.
- **Number of matrices**: a positive integer that corresponds to the number of promoter elements to be searched.
- **Length of matrices**: a positive integer that denotes the length of the weight matrices. All weight matrices must have the same length, so this must be the length of the longest matrix. The shorter matrices must be padded appropriately to match this length.

Then, for each weight matrix:

- **Optimized cut-off value**: a real number that comes with the matrix data and is used as the bias of the network units; it correspond to the recognition of the respective promoter element.
- **Start & End positions of search**: two integers that denote the start and the end of the range that the specific promoter element is to be. These numbers should be within the range of the pre-specified sequence length (start, end sequence positions).

Output. The system outputs a time-delay neural network definition file. In its current implementation, kPRNN follows the SNNS format [21]. The devised network has four layers of units:

- **Layer-1.** Consists of the input units. There are a total of \[4(\text{nucleotides}) \times \text{input_sequence_length}\] units. Input_sequence_length is equal to: [end_position_of_sequence – start_position_of_sequence].
- **Layer-2.** Consists of one column of hidden units for each element (weight matrix) to be searched. All of the hidden units columns have the same length, which depends on the length of the matrices defined above.

Layer 1 to 2 connections. The first and second layers are connected according to the weight matrix data specified in the input file.
Layer-3. Consists of one unit for each element to be searched.

Layer 2 to 3 connections. Links arrive to each unit of this layer from the corresponding column of units in the previous layer. The ranges of the elements to be searched and the corresponding relaxations, all defined in the input file, determine the weights of these links.

Layer-4. Consists of the single output unit whose activation level determines the existence of a promoter in the input units or not.

In figure 3 above, the TDNN-50nt topology is shown (50nt corresponds to the specifications of the small-net used in one of our experiments, presented in the sequel).

3 Experiments

The data used to train and evaluate the constructed TDNNs comes from the EPD database [7]. The promoter sequences are all the vertebrates that were available in the EPD database at the end of 1997.

The data comprise of six sets that belong to two categories (three sets each). The first category contains sequences of 50 nucleotides each and the second, sequences of 120 nucleotides each. In each category, there is a training set, a validation set, and a test set. The validation sets were used to prevent over-training. There were no common sequences among any of the sets. For the category with the 50-nucleotide sequences, the training set contained 688 sequences, the validation set 344 sequences, and the test set 346 sequences. For the category of the 120-nucleotide sequences, the training set contained 1032 sequences, the validation set 516 sequences, and the test set 519 sequences.

3.1. Specification of Networks

The kPRNN system was tested on two indicative promoter sequences in order to build the corresponding TDNNs.

[120nt]. For the 120-nucleotide network, the input file we used contained four weight matrices corresponding to the TATA-, CAP-, CCAAT-, and GC-boxes. The data for these weight matrices, and in particular the exploited optimized cut-off value (i.e., the bias for the units that holds the corresponding promoter elements’ boxes) may be found in [2]. Moreover, the input file specifies that the input sequences are in the location –100 to 19 (total of 120 nucleotides). The relaxation values of all the sequence searches for the corresponding elements were set to 10%.

[50nt]. For the 50-nucleotide network, the input file contained three weight matrices corresponding to the TATA-, CAP-, and GC-boxes. The input file specifies that the search range of the input sequences are in the location –40 to 9 (total of 50 nucleotides). Prior-knowledge for the CCAAT-box was not utilized because the range that this element is expected, lies between –217 and –57 sequence positions. The input file also defined that the input sequences are in the location –40 to 9 (total of 50 nucleotides). Again, the relaxation value was set to 0.1.

For the same data we created two additional networks for comparison with the knowledgeable TDNNs namely, [120ntr], and [50ntr]. These networks have identical structure with their corresponding knowledgeable ones but the biases of the units, as well as the weights of the links were randomized. This process effectively wiped out any prior knowledge that was built in to the knowledgeable nets.

3.2. Results and Discussion

After training the four networks (using the corresponding training and validation datasets), we tested each one on the corresponding test set. The results are summarized in table 1 below.

Table 1: Results on running the kPRNN system on the four networks

<table>
<thead>
<tr>
<th></th>
<th>Nets with prior-knowledge</th>
<th>Random Nets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>120nt</td>
<td>50nt</td>
</tr>
<tr>
<td>Correlation Coefficent*</td>
<td>0.80</td>
<td>0.73</td>
</tr>
<tr>
<td># True Positives</td>
<td>134</td>
<td>137</td>
</tr>
<tr>
<td># True Negatives</td>
<td>339</td>
<td>161</td>
</tr>
<tr>
<td>Accuracy (%)</td>
<td>91.14</td>
<td>86.13</td>
</tr>
<tr>
<td>(Correct predictions)</td>
<td>(473)</td>
<td>(298)</td>
</tr>
<tr>
<td>False Positives (%)</td>
<td>2</td>
<td>6.9</td>
</tr>
<tr>
<td>(# FP)</td>
<td>(7)</td>
<td>(12)</td>
</tr>
<tr>
<td># False Negatives</td>
<td>39</td>
<td>26</td>
</tr>
<tr>
<td>Error (%)</td>
<td>8.86</td>
<td>13.87</td>
</tr>
<tr>
<td>(Wrong predictions)</td>
<td>(46)</td>
<td>(48)</td>
</tr>
</tbody>
</table>

* Correlation Coefficient (CC) is computed by formula [15]:

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

Discussion. The two knowledgeable networks performed much better than the corresponding random networks.

i. Accuracy. The 120nt-trained TDNN, which exploits the knowledge about all the prominent promoter elements (i.e., TATA-box, cap signal, CCAAT-box, and GC-box), predicted 91.14% of the input sequences correctly. The corresponding random network (120ntr) performance was 76.3%. The same holds for the case of the ‘small’ nets: the 50nt-trained TDNN (with the inherent knowledge) predicted 86.13% of the input sequences correctly; again, much better than its random
analogue (50nt) which exhibits a correct classification rate of 75.72%.

ii. Exploitation of prior-knowledge. In the non-random case, the ‘big’ 120nt network did better than the ‘small’ one (50nt). This may be attributed to the following facts: (a) the 120nt network ‘knows more’, i.e., the CCAAT-box, and, (b) a longer input sequence contains more information, which translates to ‘more chances’ for the identification of the various prominent elements and the final recognition of promoter sequences.

iii. False Positives. In promoter-recognition tasks it is crucial that false-positive figures are kept in low rates. The whole enterprise of applying computational methods for automated recognition of promoters gains its validity when in supplies the genomic-researchers with reliable hints (and knowledge) about the structure of promoters. Then, the researchers may focus and experiment with these sequences and confirm (or, not) the hints. In this context, the provision of models that predict with high-specificity potential promoter-sequences is a demand. The results of table 1 validate our approach towards this demand. In particular, for the 120nt/r networks the false-positive rate was drastically reduced about 69% (from 6.4 to 2.0%). The same holds for the 50nt/r networks where, the reduction is about 60% (from 17.3 to 6.9%). Furthermore, contrasting these two figures, an increase of about 14% is achieved (i.e., from 60 to 69%). This indicates that the specificity of the knowledge-based trained networks is steadily increased as the input sequences extend (of course more extensive experimentation- with large sequences, is needed in order to validate and confirm this results; see next section).

From the reported results, it is apparent that TDNN perform quite well, even when no prior-knowledge is exploited. The ~75% correct guesses of the random networks is quite a distance from the clear random guess. This indicates the good generalization properties of TDNNs on the domain. In a related work, which also utilizes TDNN architectures (using longer sequences from the EPD database, Dec.94 release; see [18]), the highest correlation-coefficient figure reported is 70%, with the respective false-positive rate being 1.2%. Our results are better (for both ‘small’ and ‘big’ networks), at least with respect to the predictive-accuracy figures. We expect that with longer sequences, a decrease in false-positive rates may be achieved with the predictive-accuracy figures kept high (see remark iii above).

4 Conclusion and Future Research

We presented a novel promoter prediction approach that combines knowledge-based NN techniques with TDNN architectures. The implemented kPRNN system builds a TDNN on the presence of prior-knowledge that indicates what elements (signals), and which DNA-sequence regions should be explored in order to recognize a sequence as eukaryotic promoter. Training refines the knowledge used to initialize the TDNN, and the network decides about what combinations can exist, their locations, and their significance. The provided prior-knowledge utilizes information from established and universally acceptable position weight-matrices of various promoter elements.

Experimental results show a significant increase in predictive-accuracy (compared with the corresponding knowledge-free random networks). This result coupled with the decrease achieved in the corresponding false-positive rates indicates the reliability of our approach.

The implemented knowledge-to-TDNN mapping heuristics performed by the kPRNN system consists just a first approach to the task. It is in our immediate future plans to experiment with other heuristics for constructing knowledge-based neural networks. Experimentation with other (more extended, and from diverse benchmark databases) will assist this research and development work.

A major issue of course, is the kind of knowledge, and the way it is mapped to the network architecture. We claim that kPRNN-like systems are easily adjustable, and may be used as a test-bed for exploiting and testing various conjectures and theories for the different promoters’ recognition problems.

References