ELSEVIER

Journal of Molecular Graphics and Modelling 21 (2003) 323–332

Journal of Molecular Graphics and Modelling

www.elsevier.com/locate/JMGM

Computer model for recognition of functional transcription start sites in RNA polymerase II promoters of vertebrates

Vladimir B. Bajic*, Seng Hong Seah, Allen Chong, S.P.T. Krishnan, Judice L.Y. Koh, Vladimir Brusic

Computational Immunology Group, BIC-LIT, Laboratories for Information Technology, 21 Heng Mui Keng Terrace, 119613 Singapore, Singapore

Received 25 October 2001; received in revised form 9 March 2002

Abstract

This paper introduces a new computer system for recognition of functional transcription start sites (TSSs) in RNA polymerase II promoter regions of vertebrates. This system allows scanning complete vertebrate genomes for promoters with significantly reduced number of false positive predictions. It can be used in the context of gene finding through its recognition of the 5′ end of genes. The implemented recognition model uses a composite-hierarchical approach, artificial intelligence, statistics, and signal processing techniques. It also exploits the separation of promoter sequences into those that are C + G-rich or C + G-poor. The system was evaluated on a large and diverse human sequence-set and exhibited several times higher accuracy than several publicly available TSS-finding programs. Results obtained using human chromosome 22 data showed even greater specificity than the evaluation set results. The system has been implemented in the Dragon Promoter Finder package, which can be accessed at http://sdmc.krdl.org.sg:8080/promoter/.

Keywords: Promoter modelling; Promoter recognition; Transcription start site; Eukaryotic promoters

1. Introduction

One of the general, yet unsolved, problems is how to model different functional regions of a genetic DNA strand for efficient prediction of similar regions in anonymous DNA sequences. This problem was present from the very beginning of computer-based DNA analysis. Recognition of the promoter is a very difficult task and numerous attempts have been made to model promoters for computer recognition [1–4]. The general framework of promoter activity is dependent on protein–DNA interaction, as different transcription factors (TFs) have to bind to the promoter region to enable its activity. In principle, one can model the interaction of an individual TF protein and its binding site on the DNA strand [5]. These interaction models are complex because of the large number and the diversity of significant TFs and transcription factor binding sites (TFBSs) which contribute to promoter activity. This complexity prevents the application of protein–DNA interaction models for large-scale DNA screening. Because of this difficulty many simplified approaches have been used in promoter recognition problems [3]. The simplified models of promoters inevitably produce a considerable level of false positive (FP) recognition at any significant level of true positive (TP) recognition [1,3,6]. An FP prediction indicates the presence of a promoter at a location where promoters do not exist, while a TP prediction correctly identifies the location of a promoter. The promoter recognition systems for large-scale screening require acceptable ratios of TP and FP predictions (i.e. those that maximize the TP recognition while minimize the FP recognition).

The biological activation of any gene is a complex process controlled to a great extent by the promoter region. In eukaryotes, the promoter region is usually located upstream and near the start of the gene whose transcription it controls [1,2,7]. A gene has at least one promoter [2,7]. Multiple occurrences of promoters in a single gene are also known and well documented [1,8]. Thus, it is possible to search for genes by recognizing the underlying structure of their respective promoters [9]. Moreover, this approach allows for identification of groups of genes which have similar organization of transcriptional control elements. To develop promoter models for such a specific search, one needs a system that can accurately locate TSSs in anonymous DNA. The existing TSS-finding systems, such as NNPP2.1

Abbreviations: bp, base-pair; DNA, deoxyribonucleic acid; FP, false positive; Mbp, one million bp; nt, nucleotide; ppv, positive predictive value; RNA, ribonucleic acid; SPB, signal processing block; TF, transcription factor; TFBS, transcription factor binding site; TP, true positive; TSS, transcription start site; UTR, untranslated region

* Corresponding author.

1093-3263/02/$ – see front matter
PII: S1093-3263(02)00179-1

[10,11], Promoter2.0 [12], McPromoter [13], etc., produce a large number of FP predictions [2,3,6,13], making them unsuitable for promoter finding in large genomic sequences.

Computational gene recognition approaches [4] are based upon the homology analysis of potential gene products or use the analysis of different signals that indicate gene presence. However, one can also search for the 5′ end of a gene by recognizing associated promoters. The promoter recognition approach to gene prediction increases chances for discovery of non-typical genes in targeted groups, since only similarity of the organization of promoter elements is required. Although the idea of searching for genes by recognizing their promoter regions is conceptually interesting and attractive, the lack of sufficiently accurate promoter finding programs has prohibited efficient gene hunting by this method for a long time.

A recently developed system, PromoterInspector [14], was reported to produce a considerably reduced level of FP recognition compared to other publicly available promoter recognition programs. After its introduction, two other systems of similar performance were reported [15,16]. These three systems predict regions that either overlap promoter regions or are in close proximity. Yet, for efficient hunting of genes within a specific genetic class, one needs a model that recognizes relevant features of promoter regions for the targeted class of genes. Therefore, it is necessary to pinpoint the TSS in order to localize the promoter region with high precision and develop a promoter model to be used in such a search. The localization of TSSs is not tackled by the previously mentioned programs [14–16].

It is well known that TSS-finding systems produce a high rate of FPs [1–3,6,13]. In this article, we present a TSS-finding system that considerably reduces this problem. To our best knowledge, this is the first TSS-finding system that has a relatively low rate of FPs with sufficient sensitivity to make the system suitable for large-scale genomic analysis. Contrary to solutions [15,16] which are aimed at recognition of specialized classes of promoters, such as CpG-island [17–20] related promoters, our system is aimed at analysis of general vertebrate polymerase II promoters. We also present some details on its performance obtained on a diverse evaluation set and part of the results obtained on human chromosome 22. Our system has been implemented in the Dragon Promoter Finder program that can be freely accessed at http://sdmc.krdl.org.sg:8080/promoter/. This system uses a hierarchical multi-model structure with models specialized for (a) different promoter groups and (b) different sensitivity/specificity ranges. To the best of our knowledge this is the first reported composite-model structure used in promoter recognition systems based on (a) and (b). The promoter data (the short DNA segments around TSS) was separated into G + C-rich and G + C-poor groups. This separation of data and subsequent development of models for both of these promoter groups resulted in a considerably enhanced system performance. The system combines multiple hierarchically organized models optimally tuned for different sensitivity/specificity requirements, specialization of models to C + G-rich and C + G-poor promoter groups, sensor-integration, nonlinear signal processing and artificial neural networks (ANNs). This makes it conceptually different from the approaches used in other promoter-finding systems [1–3] including those that use several region sensors [13,14,21]. The system is shown to be capable of successfully recognizing promoters that are CpG-island related and those that are not. This makes it quite universal as opposed, for example, to solutions [15,16] which are specialized in recognizing CpG-islands related promoters.

A critical issue in developing promoter recognition systems is the lack of suitable standardized training and testing sets. We compiled a large set of sequences for evaluation purpose, creating one of the most diverse sets ever used in evaluation of TSS-finding programs. This sequence-set contains more than 1.15 Mbp and 159 promoters from 146 human and human-virus DNA sequences. We compared the performance of our system to several other promoter recognition programs on the evaluation set. Our results are several fold more accurate than those achieved by the other programs [10–12,14] used in the comparison. The practical utility of Dragon Promoter Finder is in its use in identification and annotation of promoters in anonymous DNA, as well as in enhancement of gene hunting by more accurate determination of the 5′ end of the gene and parts of its regulatory regions.

2. Model

The conceptual structure of our system is depicted as shown in Figs. 1–3. The overall system (Fig. 1) comprises a collection of five independent models, each tuned for optimal performance for a particular sensitivity/specificity range. Each of the independent models (Figs. 2 and 3) has the same structure made up of two sub-models, A and B, as presented in Fig. 2. Models were trained for different ranges of sensitivity/specificity. Each sub-model was trained separately for the best performance. The activation of relevant model is based on user's request, where user can select one of five pre-defined specificity/sensitivity levels. Data processing for each of the models (Figs. 2 and 3) is similar. The system analyses the content of data-windows, obtained by shifting a window of length L along the DNA sequence. The shifting of the data-window is one bp ahead. First, data is classified (Fig. 2) as being G + C-rich or G + C-poor. The criterion for separating promoters is based on

$$(\mathrm{C}+\mathrm{G}) \geq 0.5, \quad \text{for C + G rich sequences}$$

$$(\mathrm{C}+\mathrm{G}) < 0.5, \quad \text{for C + G poor sequences}$$

where

$$(\mathrm{C}+\mathrm{G}) = \frac{\#\mathrm{C}+\#\mathrm{G}}{L}$$

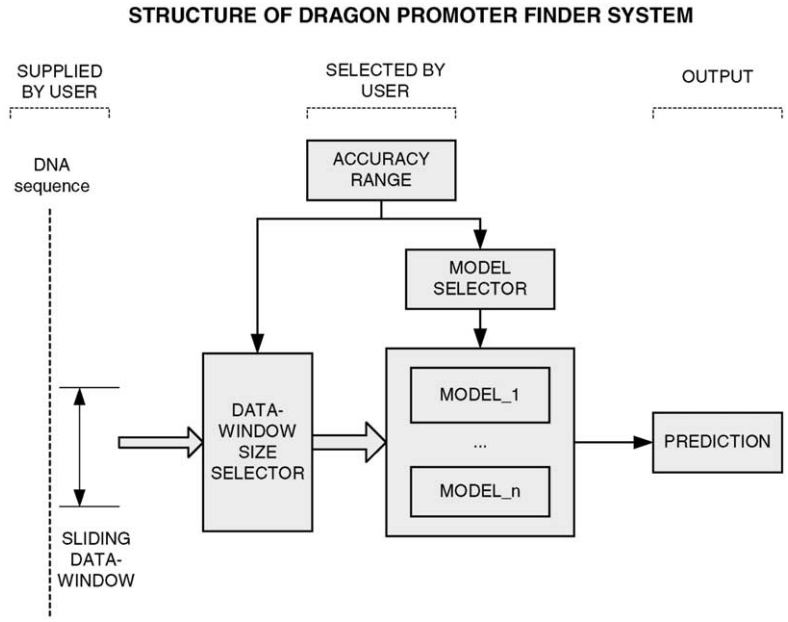

Fig. 1. Overall structure of the Dragon Promoter Finder system.

where #C and #G are the respective numbers of cytosine nucleotides and the numbers of guanine nucleotides found in the data window. After data is classified by its C + G content, it passes through either the sub-model A (G + C-rich window) or B (G+C-poor window). Within the sub-models, data passes three parallel sensors (Fig. 3). Each of the sensors represents a model of a functional region of a gene: promoter, coding exon or intron. Models of these regions were derived as positional distributions of overlapping pentamers (all sequences of five consecutive nucleotides) contained in the region.

The positional distributions of pentamers are represented by their positional weight matrices (PWMs). The PWMs were generated from the training set for each of the three functional groups considered by counting frequencies of all pentamers at each position. The position weight matrix of overlapping pentamers has dimensions $1024(L - 4)$ for a data-window of length L. The data-window was compared to the weight matrices to calculate scores that represented data-window content. Let a data-window contain the sequence $W = n_1 n_2 \ldots n_{L-1} n_L$, where $n_j \in \{A, C, G, T\}$ are nucleotides from the DNA sequence. Let the corresponding sequence P of successive overlapping pentamers p_j obtained from this data-window W is given by $P = p_1 p_2 \ldots p_{L-5} p_{L-4}$. The score for the data-window is obtained by the following

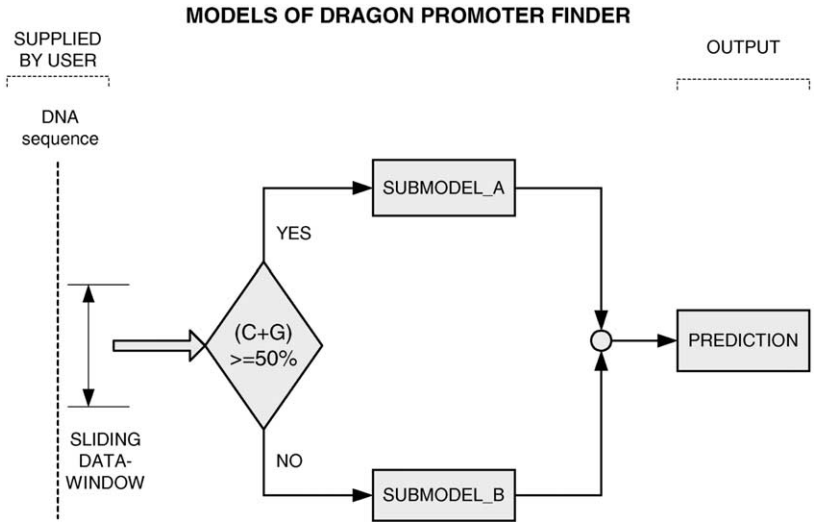

Fig. 2. Model structure in the Dragon Promoter Finder system. Different sub-models are activated based on the C + G content of the examined window.

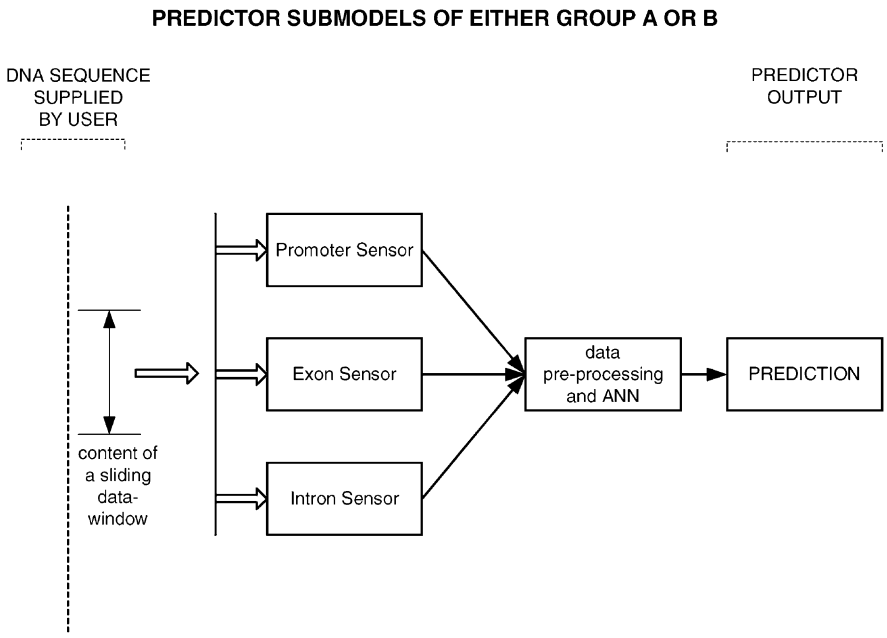

Fig. 3. Schematic representation of data processing within a sub-model.

formula

$$S = \frac{\sum_{i=1}^{L-4} p_j^i \otimes f_{j,i}}{\sum_{i=1}^{L-4} \max_j f_{j,i}}, \quad p_j^i \otimes f_{j,i} = \begin{cases} f_{j,i}, \text{ if } p_i = p_j^i, \\ 0, \text{ if } p_i \neq p_j^i \end{cases}$$

where p_j^i is the jth pentamer at position i, $f_{j,i}$ is the frequency of the jth pentamer at position i. These scores take values between 0 and 1. The working assumption was that the higher the score, the more likely that the data-window represented the respective functional region. Let the scores (signal values) of the promoter, coding exon and intron sensors be denoted by σ_p, σ_e and σ_i, respectively. These scores were used as inputs to the signal processing block (SPB). The outputs of SPB are three signals s_E, s_I, s_{EI}, which are defined as

$$s_E = \text{sat}(\sigma_p - \sigma_e, a_e, b_e)$$

$$s_I = \text{sat}(\sigma_p - \sigma_i, a_i, b_i),$$

$$s_{EI} = \text{sat}(\sigma_e - \sigma_i, a_{ei}, b_{ei})$$

where the function sat is defined by

$$\text{sat}(x, a, b) = \begin{cases} a, \text{ if } x > a \\ x, \text{ if } b \leq x \leq a \\ b, \text{ if } b > x \end{cases}$$

Parameters a_k, b_k, $k = \text{e, i, ei}$, are part of the tuning parameters of the system. The three SPB output signals were transformed by principal component analysis and fed as inputs to the ANN system. The ANN performs a multi-sensor integration. The ANN in each sub-models was a simple feed-forward network combined with the nonlinear SPB. The ANNs have three 'tansig' neurons in the hidden layer, and one 'tansig' neuron in the output layer. 'Tansig' function is described by $\tanh(x) = (e^x - e^{-x}/e^x + e^{-x})$ These ANNs are very simple and of low complexity, so that they efficiently perform the necessary signal smoothing since the outputs of the promoter and coding exon sensors are quite noisy. The ANN was trained by the Bayesian regularization method for best separation between the classes of input signals. Consequently, each of the models contained two ANN systems, one for each sub-model. The ANN system for each sub-model was trained to separate promoter regions from the non-promoter regions. Its output ranges from –1 to 1. The output that best separated the promoters from non-promoter sequences was selected as a threshold. All ANN outputs greater than the threshold are considered to indicate the presence of the promoter region in the data-window.

3. Data, system training and tuning

We used the following criteria for compilation of a data set for training and testing the Dragon Promoter Finder:

(a) the dataset should contain both, positive (promoter-containing) and negative (non-promoter) sequences;
(b) both of these sets need to have sufficient diversity, resembling the varieties of different transcription initiation mechanisms of promoters and, in the case of non-promoter data, the diversity of different functional non-promoter regions of DNA;
(c) data used has to be sanitized as much as possible retaining only those sequences that has TSS unambiguously identified.

There are objective problems in compiling a statistically representative, diverse, and clean dataset. The Eukaryotic Promoter Database (EPD) [8], Rel. 67, was the source of positive (promoter-containing) training data for most promoter recognition systems. Currently, the EPD is the most reliable source of promoter sequences, but unfortunately it is limited in content (less than 1400 promoter data across diverse species). We used EPD data to generate our positive training dataset.

3.1. Training sets

Dragon Promoter Finder system was trained using data from a collection of promoter and non-promoter sequences. We used 793 different vertebrate promoter sequences from EPD; each sequence was 250 bp in length, extending from 200 bp upstream of the TSS to 49 bp downstream of the TSS, i.e. promoter sequences covered the segment (−200, +50) relative to the TSS. By convention, there is no nucleotide position '0'. The nucleotide at the TSS is assigned the position '+1' and the nucleotide immediately preceding the TSS is assigned the position '−1'. These 250 bp long sequences represented the positive training data. We also collected, by random selection, a set of non-overlapping human coding exon and intron sequences, each 250 bp in length, from Genbank Rel. 121 [22]. The non-promoter sequences in these two groups were checked for similarity using the Blast 2 sequences program [23] to ensure that any two sequences within the group have less than 50% identity relative to each other. In total, we used 800 coding exon and 4000 intron sequences and divided the data into G + C-rich and G+C-poor sets. Consequently, we generated the training set (P_{cg+}, N_{cg+}) where P_{cg+} denotes the positive (promoter) C+G-rich sequences, while N_{cg+} denotes the non-promoter C + G-rich sequences. In an analogous way we generated the training set, P_{cg-}, N_{cg-}, that contained C + G-poor sequences. Our system was trained separately on (P_{cg+}, N_{cg+}) and on (P_{cg-}, N_{cg-}) sets. As a result of the training, the position weight matrices of pentamers and parameters of the ANNs were determined.

3.2. Tuning set and system tuning

To tune other adjustable system parameters, such as the bounds of the sensor signals, or the threshold levels for sensor signals and ANNs, we created a tuning set by expanding the training sets with 1600 non-overlapping human sequences from the 3′UTR regions taken from the UTRdb database [24], additional 500 non-overlapping human coding exon sequences, and 500 non-overlapping human intron sequences, where each sequence was 250 bp in length. These coding exon and intron sequences were extracted from the Genebank Rel. 121. Additionally, 20 full-length gene sequences with known TSSs were included in the tuning data. The promoters of these 20 genes and the EPD data were mutually exclusive. The goal of tuning was to maximize the level of TPs versus FPs over the entire range of accuracy settings. Different models were trained and each was tuned for best performance at a predefined sensitivity range. Here we define the specificity and the positive predictive value (ppv) which is sometimes called specificity in bioinformatics [25] as it is used as a proxy for true specificity particularly when the prevalence of true positives is very low. The sensitivity and ppv used are given by

$$S_e = \frac{TP}{TP + FN}, \quad ppv = \frac{TP}{TP + FP}.$$

In the above expression for sensitivity, FN stands for false negatives, i.e. FN equals the number of true promoters not predicted by the promoter prediction programs. For higher specificity range (from 0.8 to 1) we used a data-window of 250 bp, since the models using this sequence length allowed for very high specificity. For the lower specificity ranges (from 0.2 to 0.7) we used a data-window of 200 bp, as our previous experiments showed that with this window length we are able to achieve sufficiently high sensitivity (over 0.9) that was not possible with the window length of 250. We tuned five different models, where each of the models contained sub-models of classes A and B.

3.3. Evaluation set

The evaluation set (test set) was compiled by sequences taken from a larger set of human and human-virus sequences that were used by other researchers for training of gene-finding and -analysis prediction systems, as well as in promoter-finding systems. The following criteria were used in deciding which sequences to include in the set:

1. The sequence should be from a large collection of human sequences used either in the evaluation of promoter recognition systems or in the training and evaluation of gene recognition systems. By this approach we reduced the bias contained in the collected data. This allowed for the preservation of diversity of the promoter regions and ensured a representative dataset. General gene recognition programs typically recognize broad classes of genes and, therefore, they are trained on diverse sets of gene sequences. Consequently, such sets of sequences are a convenient source of potential candidate sequences for evaluation of promoter recognition programs. Based on this selection criterion we included, sequences used in training gene-finding and -analysis programs including Genie [26], GENSCAN [27], NetGene [28], GeneId [29], and some others [30,31]. We also included sequences used in testing a ANN-based promoter recognition program [32] and those used in the analysis of TATA box motifs and TSSs [33]. The sequences used in training the Genie program made up the greater part of our evaluation set.
2. Sequences included in the evaluation set were not used, in part or wholly, for training or tuning of our system.

3. The sequences should have a sufficiently detailed and complete annotation of the 5' flanking region of the gene, so that it was possible to deduce the location of the TSS.
4. The evaluation sequence-set was cleaned to remove possible redundancy by the CLEANUP program [34], allowing maximum mutual similarity of 50% in the promoter regions [−200, +50] relative to TSS. The comparison of the evaluation gene sequences with the training and tuning data by the Blast 2 sequences program resulted in less than 50% similarity of any sequence in the evaluation set versus sequences in the training or tuning sets.
5. Finally, the sequences that satisfied the above four criteria were additionally checked against the pertinent literature for accuracy of annotation.

The final evaluation set of 146 human and human-virus sequences contained 159 TSSs and had a cumulative length of more than 1.15 Mbp.

4. Comparison systems

We selected three promoter recognition systems PromoterInspector [14], NNPP2.1 [10,11], and Promoter2.0 [12], to compare their performance against the performance of our Dragon Promoter Finder program. These three programs were selected for the following practical reasons:

- they are accessible through www,
- they allow for the analysis of long sequences.

The NNPP2.1 and Promoter2.0 use ANNs, while PromoterInspector uses a statistics based approach. PromoterInspector is not a TSS-finding but a 'region-finding' promoter recognition system. For this reason, a complete comparison of its performance with our system is not possible. However, a comparative study [14] of PromoterInspector and the other five promoter recognition systems has shown that the former significantly outperforms others. We, thus, decided to include it in our comparative study. We were not able to include McPromoter program [13] in the comparison since it allows analysis of only short sequences on the www.

5. Results

Our system was compared to several other promoter finding systems on the evaluation set. In these comparisons Dragon Promoter Finder system performed extremely well. The analysis on human chromosome 22 confirmed the excellent performance of our system.

5.1. Results on the evaluation set

The criterion for deciding which prediction is correct and which is not, is as stated in [1]. The predicted TSS is thereby considered correct if it is within 200 nucleotides upstream, or 100 nucleotides downstream of the real TSS. We used this criterion for predictors that make the prediction of the TSSs as individual locations such as in the case of Dragon Promoter Finder, Promoter2.0, and NNPP2.1. The NNPP2.1 gives the region around the predicted TSS location in the ranges of [−40, +10], and their predicted TSS location is 40 nt downstream of the predicted 5′ boundary of the region.

In the case of PromoterInspector, which gives the prediction of the promoter region, the prediction is calculated as correct if the TSS falls within that region. Otherwise it is calculated as false positive. This criterion is based on ([9], p. 169) which states: "a promoter must contain a transcription start site (TSS)". Consequently, a reasonable definition of promoter region should include the TSS. One should also note that [14] used a different calculation method that assesses only the location in the genomic sequence, disregarding the strand orientation, which generates one FP for each correctly predicted promoter and two FPs for each false positive prediction. We used a more stringent screening of both strands.

Results achieved on the evaluation set are as presented in Fig. 4 for all four compared programs.

When the system is set to the same sensitivities as the other three tested systems, then our system achieves many times less FPs (as shown in Table 1).

5.2. Results on human chromosome 22

In addition to testing on the evaluation set, we performed further testing of Dragon Promoter Finder system on the annotated known genes on human chromosome 22. The results were generally consistent with those obtained on the evaluation set (Table 2).

For PromoterInspector, the TP predictions are determined according to the criteria explained in detail in [35], while the FP predictions were considered those that fall on the annotated part of the human chromosome 22 covered by known genes, but not sufficiently close to the 5' end of these gene, thus not representing the TP predictions. For Dragon Promoter Finder, we considered the predicted TSS position correct if they fall within the interval of length equal to the average length of the promoter region predicted by PromoterInspector (555 bp), while the other conditions being the same as in [35]. This makes the criteria for comparison between the two programs equivalent.

The annotation data (Rel. 2.3) for human chromosome 22 were produced by the Chromosome 22 Gene Annotation Group at the Sanger Centre and were obtained from the world wide web [36]. Here we present the Se and the ppv obtained on the sections of chromosome containing annotated known genes. For example, Table 2 implies that for each TP at Se = 80%, an average of 3.34 FP predictions were made; at Se = 0.58%, 1.4 FP predic-

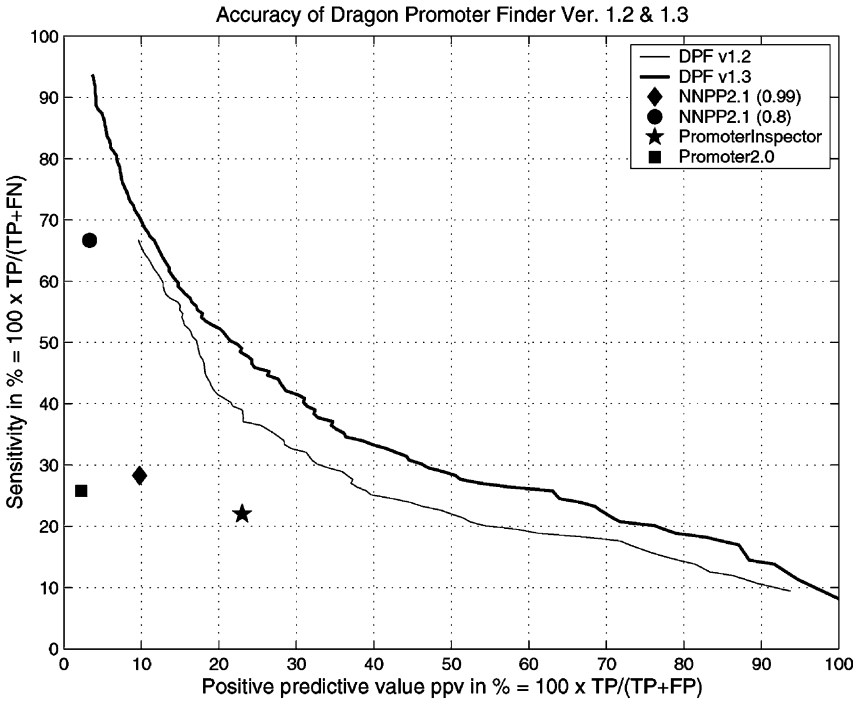

Fig. 4. Accuracy of Dragon Promoter Finder system and comparison with the results of other programs on the evaluation set. DPF v.1.3 shows performance of Dragon Promoter Finder. The DPF v.1.2 shows performance of Dragon Promoter Finder trained without separation of training data into C + G-rich and C + G-poor sets.

Table 1
FP is the number of false positive predictions made by the promoter recognition programs other than our system; FP_{DPF} is the number of false positive predictions made by Dragon Promoter Finder at the same sensitivity level as the program used in comparison

	Promoter-Inspector	NNPP2.1 (threshold 0.8)	NNPP2.1 (threshold 0.99)	Promoter2.0
FP/FP_{DPF}	6.88	3.8	8.82	56.9

FP/FP_{DPF} shows the decrease in FP predictions of Dragon Promoter Finder on the evaluation set at the same sensitivity of predictions as the comparison programs (Fig. 4). The results for NNPP2.1 are given for two levels of threshold (0.8 and 0.99).

tions were made; and at Se = 0.49% on average 1.1 FP predictions were made. These results provide additional evidence of excellent performance of Dragon Promoter Finder.

Table 2
The results of the promoter prediction by Dragon Promoter Finder on human chromosome 22

Se (%)[a]	Ppv (%)[a]
49	48
58	42
64	33
74	30
80	23

[a] Human chromosome 22 (known genes).

6. Discussion and comments

6.1. C + G content and CpG-islands

Promoters can be divided by the C+G content of the region around the TSS. There are several motivating factors for dividing promoters in such a manner. Firstly, CpG-islands, known to be associated with promoters of approximately 50% of all known genes and virtually all housekeeping genes [19,20], are only contained in G + C-rich regions. Secondly, the EPD database has about 2/3 of vertebrate promoters in this group. Consequently, one should expect that the separation of the data by C + G content will result in more specialized models and, when combined, they may enhance overall promoter prediction. This specialization of the models indeed resulted in the improvement of predictions as shown in Fig. 4. Another possibility is to associate promoters with the CpG-islands, as implemented in certain specialized promoter recognition programs [15,16]. However, to exactly determine the CpG-islands in large genomic sequences, one needs enormous amount of computation time. Therefore, various simplified approaches are in use for obtaining approximate CpG-island information within reasonable time. Unfortunately these approaches produce different results [15]. Thus, we resorted to the use of the C+G content since its determination is unambiguous and direct, keeping in mind that the C + G-rich sequences are those intimately related to the CpG-islands.

The CpG-island information has been used in some programs [15,16] for predictions of specific promoter groups. The CpGpromoter system [15] uses quadratic discriminant analysis and three parameters of the CpG-islands, such as their length, CpG score, and the C+G content, to distinguish between promoter-associated CpG-islands and the ones that are not promoter-associated. CpGpromoter has a reported sensitivity of 0.47 and ppv of 0.34. In addition to the parameters that relate to the CpG-islands, several other parameters related to density of specific motifs were introduced and linear discriminant analysis was applied in [16] to predict promoters associated with the CpG-islands. That system [16] was shown to perform slightly better than the PromoterInspector program (sensitivity of 0.4 and ppv of 0.4) on genomic size sequences. These systems [14–16] showed a strong bias toward CpG-island related promoters. Moreover, all these systems predict a region that may overlap or is in the close proximity to the promoter region, and they do not localize the TSS. The Dragon Promoter Finder has a different goal: it makes predictions of TSSs; it achieves sensitivity of 0.35 and ppv of 0.35 on our evaluation set, while it achieves sensitivity of 0.49 and ppv of 0.48 on human chromosome 22. This is very comparable with the performances of the region-predicting promoter-finding systems [14–16].

By using information from relatively short DNA segments, Dragon Promoter Finder makes only a local assessment of the DNA content in producing TSS predictions and lacks the more global context analysis that can potentially localize promoter regions. Consequently, there is a possibility to enhance the TSS predictions of our system by combining its predictions with predictions made by the region-predicting promoter finders [14–16], since the latter are made based on information from a more global context analysis.

It is important to note that the two submodels (A and B) of the system developed for C+G-rich and C+G-poor cases (at any sensitivity/specificity range) always make mutually exclusive predictions. For the two very similar promoter regions, but one with $C + G > 50\%$ and the other with $C + G < 50\%$, the two models will not always produce the same predictions, meaning that the models were trained on distinct features.

6.2. Selection of sensors of functional regions

The main functional features of genes are the promoters, 5′UTRs, coding exons, introns, and 3′UTRs. We used models of promoters, coding exons and introns to build our TSS recognition system. Our assumption was that the characteristics of these three functional regions are sufficient to help Dragon Promoter Finder distinguish true predictions from false predictions on the gene and in the intergenic regions. The use of sensors for promoters, coding exons and intron regions in the development of promoter models was first reported recently [21]. A similar use of three sensors for promoters, intronic and cds regions in promoter recognition is applied in McPromoter program. Also, PromoterInspector uses four sensors for promoters, introns, exons and 3′UTRs. In our model, we did not find significant contribution of 3'UTR sequences to the recognition performance and, thus, we did not use them in the model development. The 5′UTR regions were also not used since we were not able to find sufficient number of long enough 5′UTRs that were sufficiently diverse (less than 50% mutual similarity). Moreover, the inclusion of the 5'UTR sensor would change the design of the whole system and, thus, we omitted its use.

6.3. Evaluation and tuning sets selection

The reasons for the selection of sequences used within the tuning-set are as follows:

(a) we did not have sufficient number of diverse promoter sequences to make two sufficiently large separate sets, one for training and one for tuning, for each of the two cases considered: G+C-rich and G+C-poor sets;
(b) by adding the 3'UTR sequences, and new coding exon and intron sequences, we increased the size of the negative data, and
(c) by adding 20 full-length gene sequences, we provided a more realistic environment for the operation of the final, tuned system.

The evaluation set that we used is one of the most diverse sets applied in assessing the accuracy of TSS recognition systems. Each of the sequences included in the evaluation set has already been used by other researchers in the field, so no new sequences were introduced to this set by the authors. This helped to partly reduce the bias, as we put emphasis on the diversity of the promoter regions and the previous use of the sequences mainly for training or testing gene finding programs. The bias, however, is not eliminated fully and remains in the data. For example, one of its manifestations is that intergenic regions are not well represented.

6.4. Comments on program comparisons

The accuracy of Dragon Promoter Finder achieved on the evaluation set was compared to that of several other programs. At the same levels of sensitivity our system achieves approximately 4–57-fold less FP predictions than any of the other programs compared within this study.

The PromoterInspector program was shown to have a considerably improved level of TP/FP compared to several other promoter recognition programs [14]. However, on the evaluation set our program achieved 6.88 times less FP predictions than PromoterInspector at the same sensitivity level of $Se = 0.22$. This shows that the model we used is suitable and that it picks up many (although not all) relevant features of promoters. We made a comparative study with PromoterInspector because this is currently deemed the best region-predicting promoter-finding system.

The NNPP2.1 program is based on the recognition of two specific signals within the promoter region: the TATA-box and the initiator (Inr), as well as their mutual distance. This system uses three time-delay ANNs. One ANN recognizes TATA-box, the other recognizes Inr, while the third one combines the outputs of the two and takes into account the spatial distance between the TATA-box and Inr signals. A significant portion of sequences (61%) in our evaluation set were from the training set of Genie program. Intron and cds sequences from the Genie training set were used in the training of NNPP2.1 program, thus providing significant advantage to this program in the comparative analysis on our evaluation set. However, even with this advantage of the NNPP2.1 program, Dragon Promoter Finder outperforms it by roughly 4 to 9 times with regards to specificity.

The other system, Promoter2.0, is also based on the ANN and trained to recognize four specific signals very commonly present in eukaryotic promoters: TATA-box, Inr, GC-box, CCAAT-box, as well as their mutual distances. Dragon Promoter Finder achieves specificity of more than 50 times higher than Promoter2.0 at the same sensitivity level.

7. Conclusions

We present here the Dragon Promoter Finder program aimed at recognizing TSSs of RNA polymerase II promoters in anonymous DNA sequences of vertebrates. It is based on a conceptually new promoter model and conveniently exploits multi-sensor integration via ANNs and multi-model system structure. The prediction accuracy of Dragon Promoter Finder, achieved on a large and diverse evaluation-set, appears to be superior to other reported web-accessible TSS recognition systems. The algorithm can be used for promoter search in large contigs of anonymous DNA to make gene hunting easier, since it allows considerably reduced number of FP recognition in comparison to other TSS-finding systems. Due to its modular conceptual design this model opens up new avenues for developing systems to localize groups of signals that may characterize some aspects of gene structure, such as translation initiation site, polyA site or splice sites. Some possible improvements of the program's accuracy might be found in combining the TSS search of the Dragon Promoter Finder with some of the region-predicting promoter-finding programs.

References

[1] J.W. Fickett, A.G. Hatzigeorgiou, Eukaryotic promoter recognition, Genome Res. 7 (1997) 861–878.
[2] A.G. Pedersen, P. Baldi, Y. Chauvin, S. Brunak, The biology of eukaryotic promoter prediction—a review, Comput. Chem. 23 (1999) 191–207.
[3] D.S. Prestridge, Computer Software for Eukaryotic Promoter Analysis, published on internet, http://biosci.umn.edu/class/bioc/8140/Promoter.html, 1999.
[4] G.D. Stormo, Gene-finding approaches for eukaryotes, Genome Res. 10 (2000) 394–397.
[5] B. Dreier, D.J. Segal, C.F. Barbas III, Insights into the molecular recognition of the 5′-GNN-3′ family of DNA sequences by zinc finger domains, J. Mol. Biol. 303 (2000) 489–502.
[6] M.G. Reese, G. Hartzell, N.L. Harris, U. Ohler, J.F. Abril, S.E. Lewis, Genome annotation assessment in *Drosophila melanogaster*, Genome Res. 10 (2000) 483–501.
[7] R.O.J. Weinzierl, Mechanism of Gene Expression, Imperial College Press, London, 1999.
[8] R.C. Périer, V. Praz, T. Junier, C. Bonnard, P. Bucher, The Eukaryotic Promoter Database (EPD), Nucl. Acid Res. 28 (2000) 302–303.
[9] T. Werner, Models for prediction and recognition of eukaryotic promoters, Mammal. Genome 10 (1999) 168–175.
[10] M.G. Reese, F.H. Eeckman, Time-delay neural network for eukaryotic promoter prediction, 1999, unpublished.
[11] M.G. Reese, N.L. Harris, F.H. Eeckman, Large scale sequencing specific neural networks for promoter and splice site recognition, in: Proceedings of the 1996 Pacific Symposium on Biocomputing, L. Hunter, T.E. Klein (Eds.), World Scientific Publishing Co., Singapore, 2–7 January, 1996, http://www.fruitfly.org/seq_tools/promoter.html.
[12] S. Knudsen, Promoter2.0: for the recognition of Pol II promoter sequences, Bioinformatics 15 (1999) 356–361.
[13] U. Ohler, H. Niemman, G.-C. Liao, G.M. Rubin, Joint modeling of DNA sequence and physical properties to improve eukaryotic promoter recognition, Bioinformatics 17 (Suppl. 1) (2001) S199–S206.
[14] M. Scherf, A. Klingenhoff, T. Werner, Highly specific localisation of promoter regions in large genomic sequences by PromoterInspector: a novel context analysis approach, J. Mol. Biol. 297 (2000) 599–606.
[15] I.P. Ioshikhes, M.Q. Zhang, Large-scale human promoter mapping using CpG islands, Nat. Genet. 26 (2000) 61–63.
[16] S. Hannenhalli, S. Levy, Promoter prediction in the human genome, Bioinformatics 17 (Suppl 1) (2001) S90–S96.
[17] A.P. Bird, M.H. Taggart, R.D. Nichollas, D.R. Higgs, Non-methylated CpG-rich islands at the human alpha-globin locus: implications for evolution of the alpha-globin pseudogene, Embo J. 6 (1986) 999–1004.
[18] S.H. Cross, A.P. Bird, CpG islands and genes, Curr. Opin. Genet. Dev. 5 (1995) 309–314.
[19] M. Gardiner-Garden, M. Frommer, CpG islands in vertebrate genomes, J. Mol. Biol. 196 (1987) 261–282.
[20] F. Larsen, G. Gundersen, R. Lopez, H. Prydz H, CpG islands as gene markers in the human genome, Genomics 13 (1992) 1095–1107.
[21] S. Levy, L. Compagnoni, E.W. Myers, G.D. Stormo, Xlandscape: the graphical display of word frequencies in sequences, Bioinformatics 14 (1998) 74–80.
[22] D.A. Benson, I. Karsch-Mizrachi, D.J. Lipman, J. Ostell, B.A. Rapp, D.L. Wheeler, GenBank, Nucl. Acid Res. 28 (2000) 15–18.
[23] T.A. Tatusova, T.L. Madden, Blast 2 sequences—a new tool for comparing protein and nucleotide sequences, FEMS Microbiol. Lett. 174 (1999) 247–250.
[24] G. Pesole, S. Liuni, G. Grillo, F. Licculli, A. Larizza, W. Makalowski, C. Saccone, UTRdb and UTRsite: specialized databases of sequences and functional elements of 5′ and 3′ untranslated regions of eukaryotic mRNAs, Nucl. Acid Res. 28 (2000) 193–196.
[25] V.B. Bajic, Comparing the success of different prediction programs in sequence analysis: a review, Brief. Bioinform. 1 (2000) 214–228.
[26] M. Reese, D. Kulp, A. Gentles, U. Ohler, http://www.fruitfly.org/seq_tools/datasets/Human, 1999.
[27] C. Burge, S. Karlin, Prediction of complete gene structure in human genomic DNA, J. Mol. Biol. 268 (1997) 78–94.
[28] S. Brunak, J. Engelbrecht, S. Knudsen, Prediction of human mRNA donor and acceptor sites from the DNA sequence, J. Mol. Biol. 220 (1991) 49–65.

[29] R. Guigo, S. Knudsen, N. Drake, T. Smith, Prediction of gene structure, J. Mol. Biol. 226 (1992) 141–157.
[30] M. Gelfand, A.A. Mironov, P.A. Pevzner, Gene recognition via spliced sequence alignment, PNAS U.S.A. 93 (1996) 9061–9066.
[31] R. Farber, A. Lapedes, Determination of eukaryotic protein coding regions using neural networks and information theory, J. Mol. Biol. 226 (1992) 471–479.
[32] N. Mache, M. Reczko, A. Hatzigeorgiou, Multistate time-delay neural networks for the recognition of POL II promoter sequences, ISMB96, St. Louis, http://www.informatik.uni-stuttgart.de/ipvr/bv/personen/mache, 1996.
[33] F.E. Penoti, Human DNA TATA-boxes and transcriptional initiation sites, a statistical study, J. Mol. Biol. 213 (1990) 37–52.
[34] G. Grillo, M. Attimonelli, S. Liuni, G. Pesole, CLEANUP: a fast computer program for removing redundancies from nucleotide sequence databases, Comput. Appl. Biosci. 12 (1996) 1–8.
[35] M. Scherf, A. Klingenhoff, K. Frech, K. Quandt, R. Schneider, K. Grote, M. Frisch, V. Gailus-Durner, A. Seidel, R. Brack-Werner, T. Werner, First pass annotation of promoters on human chromosome 22, Genome Res. 11 (2001) 333–340.
[36] Dunham et al., Unpublished data, http://www.sanger.ac.uk/HGP/Chr22/.