BIOINFORMATICS

Vol. 25 ISMB 2009, pages i313–i320
doi:10.1093/bioinformatics/btp191

Toward a gold standard for promoter prediction evaluation

Thomas Abeel[1,2], Yves Van de Peer[1,2,*] and Yvan Saeys[1,2]

[1]Department of Plant Systems Biology, VIB and [2]Department of Plant Biotechnology and Genetics, Ghent University, Technologiepark 927, B-9052 Gent, Belgium

ABSTRACT

Motivation: Promoter prediction is an important task in genome annotation projects, and during the past years many new promoter prediction programs (PPPs) have emerged. However, many of these programs are compared inadequately to other programs. In most cases, only a small portion of the genome is used to evaluate the program, which is not a realistic setting for whole genome annotation projects. In addition, a common evaluation design to properly compare PPPs is still lacking.

Results: We present a large-scale benchmarking study of 17 state-of-the-art PPPs. A multi-faceted evaluation strategy is proposed that can be used as a gold standard for promoter prediction evaluation, allowing authors of promoter prediction software to compare their method to existing methods in a proper way. This evaluation strategy is subsequently used to compare the chosen promoter predictors, and an in-depth analysis on predictive performance, promoter class specificity, overlap between predictors and positional bias of the predictions is conducted.

Availability: We provide the implementations of the four protocols, as well as the datasets required to perform the benchmarks to the academic community free of charge on request.

Contact: yves.vandepeer@psb.ugent.be

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Promoter prediction programs (PPPs) aim to identify promoter regions in a genome using computational models. In early work, promoter prediction focused on identifying the promoter of (protein-coding) genes (Fickett and Hatzigeorgiou, 1997), but more recently it has become clear that transcription initiation does not always result in proteins, and that transcription occurs all over the genome (Carninci *et al.*, 2006; Frith *et al.*, 2008; Sandelin *et al.*, 2007).

One important question is what the different PPPs are actually trying to predict. Some programs aim to predict the exact location of the promoter region of known protein-coding genes, while others focus on finding the transcription start site (TSS). Recent research has shown that there is often no single TSS, but rather a whole transcription start region (TSR) containing multiple TSSs that are used at different frequencies (Frith *et al.*, 2008). This article analyzes the performance of 17 programs on two tasks: (i) genome-wide identification of the start of genes and (ii) genome-wide identification of TSRs.

Most PPPs that are published make use of a tailored evaluation protocol that almost always proclaims the new PPP outperforming all others. Our aim is provide an objective benchmark that allows us to test and compare PPPs. In the past few years, a number of papers have evaluated promoter prediction software. The earliest work indicated that many of the early PPPs predicted too many false positives (FPs) (Fickett and Hatzigeorgiou, 1997). A later genome-wide review included a completely new set of promoter predictors and introduced an evaluation protocol based on gene annotation (Bajic *et al.*, 2004). This protocol has later been used to validate promoter predictions for the ENCODE pilot project (Bajic *et al.*, 2006). Sonnenburg *et al.* (2006) proposed a more rigorous machine-learning-inspired validation method that uses experimentally determined promoters from DBTSS, a database of promoters. The most recent large-scale validation of PPPs included more programs than any of the earlier studies and introduced for the first time an evaluation based on all experimentally determined TSSs in the human genome (Abeel *et al.*, 2008a, b).

While many issues have been solved, there is still a large number of challenges that remain open for debate in evaluating the performance of promoter prediction software. Generally, we can distinguish two main approaches in promoter prediction. The first approach assigns scores to all single nucleotides to identify TSSs or TSRs. Usually, the scoring is done with a classification algorithm that is typically validated using cross-validation. This cross-validation provides a first insight in to the performance of the model and can be used to optimize the model parameters on a training set. The scores obtained from these techniques can be used as input for a genome annotation pipeline, where they will be aggregated in gene models. Because of their design, this type of promoter predictors will always work on a genome-wide scale. Programs using this approach include ARTS (Sonnenburg *et al.*, 2006), ProSOM (Abeel *et al.*, 2008b) and EP3 (Abeel *et al.*, 2008a). The second approach identifies a promoter region without providing scores for all nucleotides. Typically, this type of programs will output a start coordinate and a stop coordinate of the promoter, and a score that indicates the confidence in the prediction. In rare cases, only one coordinate is given as TSS. For two programs no score is provided (Wu-method and PromoterExplorer). Within this approach, we can distinguish two subclasses of programs: the ones that work on a genomic scale and the ones that do not. The latter are used to identify the promoter of a single gene. In this work we will not consider these programs, because they are usually distributed as a website and are thus not suited for large-scale analyses.

PPPs can be applied to identify the promoter of known genes, or they can be used to identify the start of any transcription event, regardless of what the final fate of the transcribed sequence is. For each application, we propose two evaluation protocols that can be used to assess the performance of a program for that particular application. Each application has an associated reference dataset which the protocol will use to evaluate a PPP. We use the same

[*]To whom correspondence should be addressed.

type of reference datasets that have previously been used to validate promoter predictions (see section 2 for details).

Several methods have been proposed to validate promoter predictions. Cross-validation on a small set of promoter and non-promoter sequences is sometimes used to validate a PPP (Xie *et al.*, 2006), but the results are often an overestimation of the performance on a complete genome (Bajic *et al.*, 2004). Other methods make use of gene annotation to evaluate promoter predictions, based on the rationale that the start of a gene corresponds with a promoter (Bajic *et al.*, 2004, 2006). However, it is clear that not all promoters are associated with protein-coding genes and, furthermore, not all transcription events start at the beginning of a gene. TSSs have been observed at the start of internal exons or at the $3'$ end of a gene (Carninci *et al.*, 2006). More recently, two large resources for promoter research in the human genome have been used to validate promoter predictions. The first source is the DBTSS database, containing a large set of experimentally determined promoters (Wakaguri *et al.*, 2008). The second source is a genome-wide screening of the human genome using the CAGE technique (Shiraki *et al.*, 2003), providing all TSSs in the genome. The latter source is the most valuable as it is an exhaustive screening for all possible TSSs.

The remainder of this work proposes a set of protocols and datasets to use when validating promoter prediction software. To illustrate our methods, we analyzed 17 PPPs with the proposed validation schemes. While the methods are applicable to any genome, we focus in the current article on the human genome. Finally, we highlight some challenges that arise in selecting the best PPP for a particular task.

2 MATERIALS AND METHODS

2.1 Datasets

We used release hg18 of the human genome for all analyses. For the validation protocols, we use the RefSeq genes downloaded from the UCSC table browser. This set includes 23 799 unique gene models and is further referred to as the *gene set*. We also use the CAGE tag dataset from Carninci *et al.* (2006). The latter was preprocessed to aggregate all overlapping tags into clusters, resulting in 180 413 clusters containing a total of 4 874 272 CAGE tags. A cluster is considered to be a TSR if it contains at least two tags. Singleton clusters are removed as these could be transcriptional noise. This dataset will be referred to as the *CAGE dataset*.

2.2 Promoter prediction software

We used two criteria to select the PPPs to include in this analysis: (i) the program or predictions should be available without charge for academic use, and (ii) the program should be able to process the complete human genome or predictions should be available for the complete genome. At least 17 programs (Table 1) fulfilled these criteria and have been included. Details for settings and prediction extraction methods for each program are included in the Supplementary Material.

Table 1. Overview of all the programs analyzed

Name	References
ARTS	Sonnenburg *et al.* (2006)
CpGcluster	Hackenberg *et al.* (2006)
CpGProD	Ponger and Mouchiroud (2002)
DragonGSF	Bajic and Brusic (2003)
DragonPF	Bajic *et al.* (2002)
EP3	Abeel *et al.* (2008a)
Eponine	Down and Hubbard (2002)
FirstEF	Davuluri *et al.* (2001)
McPromoter	Ohler *et al.* (2000)
NNPP2.2	Reese (2001)
Nscan	Gross and Brent (2006)
Promoter 2.0	Knudsen (1999)
PromoterExplorer	Xie *et al.* (2006)
PromoterScan	Prestridge (1995)
ProSOM	Abeel *et al.* (2008b)
PSPA	Wang and Hannenhalli (2006)
Wu-method	Wu *et al.* (2007)

2.3 Evaluation protocols

In this article, we propose four protocols to evaluate the quality of predictions made by PPPs. The first two protocols are bin-based protocols, inspired by Sonnenburg *et al.* (2006). The latter two are distance based, inspired by Abeel *et al.* (2008b). Figure 1 shows a schematic overview of how each protocol determines the prediction performance.

For the explanation of each protocol we assume that we have a set of predictions. Furthermore, we have a reference set (the gene set or the CAGE set) that is considered to be the ground truth. The binning protocols (1A and 1B) are more machine-learning oriented. Each bin has two labels: one provided by the reference set and the other provided by the PPP. Performance can be assessed based on these labels. The distance protocols (2A and 2B) calculate the distance between a reference item and the closest prediction and will use this to calculate the performance. Protocols ending in A use the CAGE data as reference, while the ones ending in B use the gene set. Note that the B protocols discard all intergenic predictions from the evaluation. Intergenic prediction are removed because the gene set only contains known genes, so we have no idea which of the intergenic prediction are related to unknown genes or other types of transcription (Bajic *et al.*, 2004).

2.3.1 Bin-based validation Evaluation protocol 1A: this protocol uses the CAGE dataset as reference. We divide the genome in bins of 500 nt. Next, we check for each bin whether it overlaps with the center of a TSR. If it does, we label this bin as a positive TSR. With this labeling we can determine the number of true positives (TPs), FPs, false negatives (FNs) and true negatives (TNs). Each bin that is both labeled by a prediction and a TSR is considered a TP. A TN is a bin that is not labeled as predicted nor labeled as TSR. A FP is a bin that is labeled as predicted but not labeled as TSR. Finally, a FN is a bin that is not labeled as predicted but is labeled as TSR. From these we calculate the precision and recall with the following formulas.

$$\text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}$$

Evaluation protocol 1B: this protocol is a variant of protocol 1A, but it uses the gene set as reference instead of the CAGE dataset. This protocol resembles the one used in Sonnenburg *et al.* (2006). We label all the bins overlapping the start of a gene as a positive gene start bin. All bins that overlap with the gene, but not with the start of that gene, are labeled as negative gene start bins. Bins that do not overlap with a gene or gene start are ignored in the analysis.

A TP is a bin labeled as predicted and as a positive gene start. A TN is a bin not labeled as predicted and labeled as a negative gene start. A FP is a bin labeled as predicted and as a negative gene start. Finally, a FN is a bin not labeled as predicted and labeled as a positive gene start. The calculation of precision and recall are the same as in protocol 1A. Note that this protocol ignores intergenic predictions that are not close to a gene start.

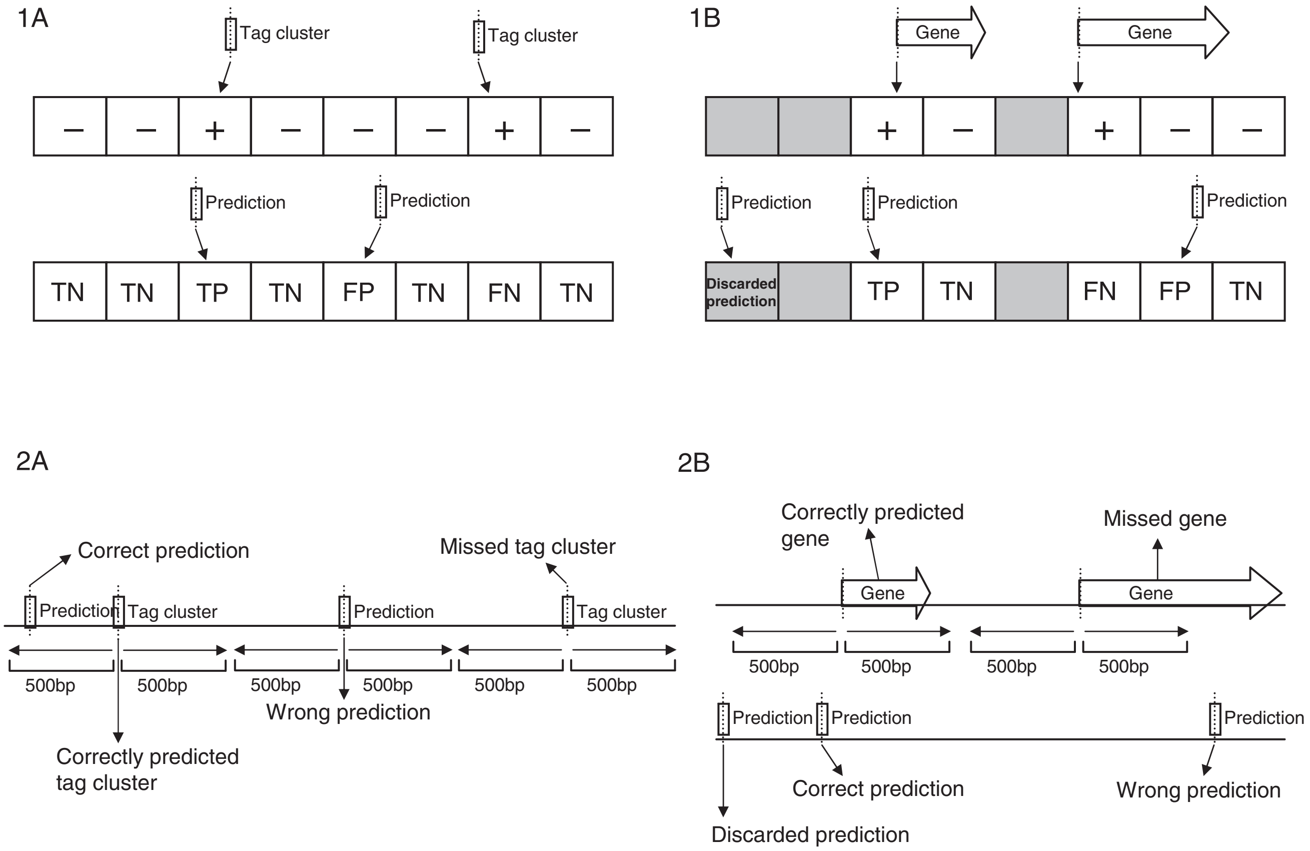

Fig. 1. Visual representation of how the different protocols work. The panel numbers refer to the protocol identifiers. Protocols starting with 1 are based on binning, the ones starting with 2 on distance. Protocols ending in A use the CAGE data as reference, and those ending in B the gene set. More details can be found in the main text.

2.3.2 Distance-based validation Evaluation protocol 2A: this protocol aims to validate predictions with the CAGE dataset as a reference. We determine three scores: (i) the number of predictions (totalPredictions); (ii) how many of these predictions are correct (correctPredictions); and (iii) how many TSRs are discovered by the predictions (discoveredTSR). A prediction is correct if the distance to the closest TSR is smaller than 500 nt. We use 500 nt as this is the same value as the binning approach and the value has been used in the past for this type of analysis (Abeel *et al.*, 2008a, b). A TSR is considered discovered if there is at least one prediction less than 500 nt away from the TSR. The CAGE dataset has 180 413 TSRs (totalTSR).

We then define recall and precision as follows:

$$\text{precision} = \frac{\text{correctPredictions}}{\text{totalPredictions}}$$

$$\text{recall} = \frac{\text{discoveredTSR}}{\text{totalTSR}}$$

Evaluation protocol 2B: this is a modification of protocol 2A to check the agreement between TSR predictions and gene annotation. This method resembles the method used in the EGASP pilot-project (Bajic *et al.*, 2006).

We determine three scores: (i) number of predictions (totalPredictions); (ii) how many of these predictions are correct (correctPredictions); and (iii) how many genes are discovered by the predictions (discoveredGenes). All predictions that are not near the start of a gene or do not overlap with a gene are discarded. A prediction is correct if the distance to the closest start of a gene is smaller than 500 nt. A start of a gene is considered discovered if there is at least one prediction less than 500 nt away from the TSR. Predictions that overlap a gene, but are not within 500 nt of the start are wrong predictions. There are 23 799 genes in the reference set (totalGenes).

$$\text{precision} = \frac{\text{correctPredictions}}{\text{totalPredictions}}$$

$$\text{recall} = \frac{\text{discoveredGenes}}{\text{totalGenes}}$$

As in protocol 1B, this method ignores intergenic predictions that are not close to a gene start.

2.4 Performance measures

Precision and recall have been defined for each protocol as their definition is dependent on the protocol. Unfortunately, it is impossible to compare two precision–recall pairs from different programs as there is a trade-off between the precision and recall. A solution that is often used in machine learning is the use of ROC curves. We will use a variant of this method called PRCs. Instead of plotting the TP rate against the FP rate, we plot the recall against the precision. The resulting graphs are comparable and provide a full overview of the potential of the PPP. So, to fairly assess the performance of each PPP, we need to calculate all possible precision–recall pairs. This can be done by a moving threshold on the score of the predictions made by a program. We use 500 thresholds equally spaced between the minimum and maximum score for each PPP. The area under the auPRC is calculated using the trapezoid method on all precision–recall pairs for each algorithm.

To quantify the performance of a PPP over all protocols with a single metric we introduce the PPP score, which is the harmonic mean of the auPRC

Table 2. Overview of the results of all protocols on all PPPs

	Name	1A	1B	2A	2B	Number of predictions	Threshold	F-score	PPP score
1	ARTS	0.19	0.36	0.47	0.64	432117	0.56362	0.47	**0.34**
2	CpGcluster	0.09	0.22	0.28	0.44	22777	42.24167	0.38	0.18
3	CpGProD	0.06	0.16	0.32	0.04	20810	0.25473	0.45	0.08
4	DragonGSF	0.06	0.16	0.25	0.42	100,046	0.26	0.45	0.14
5	DragonPF	0.05	0.08	0.18	0.26	747571	0.34	0.32	0.09
6	EP3	0.18	0.23	0.42	0.51	67807	−0.048	0.44	**0.28**
7	Eponine	0.14	0.29	0.41	0.57	1320964	0.986	0.45	**0.27**
8	FirstEF	0.08	0.23	0.28	0.52	44818	0.92938	0.28	0.18
9	McPromoter	0.04	0.10	0.12	0.23	43818	−0.01347	0.25	0.08
10	NNPP2.2	0.01	0.01	0.01	0.01	1962552	0.99	0.08	0.01
11	Nscan	0.07	0.27	0.22	0.51	23360	200.558	0.34	0.17
12	Promoter 2.0	0.01	0.01	0.02	0.01	1923610	0.5	0.10	0.01
13	PromoterExplorer	0.02	0.05	0.07	0.12	134282	NA	0.25	0.04
14	PromoterScan	0.02	0.05	0.06	0.13	248671	57.51	0.20	0.04
15	ProSOM	0.18	0.25	0.42	0.51	63228	0.65302	0.44	**0.29**
16	PSPA	0.05	0.17	0.16	0.33	25602	85.20467	0.28	0.11
17	Wu-method	0.04	0.10	0.13	0.24	23934	NA	0.31	0.08

The first two columns provide the index and the name of the PPPs. The third through sixth column show the area under the precision–recall curve (auPRC) for each of the protocols. The seventh column displays the number of predictions for the optimal threshold as determined by protocol 2A. The eighth column shows the optimal threshold determined with protocol 2A and the next column the corresponding F-score. The tenth column gives the final score for the promoter predictor as the harmonic mean of the auPRC scores for the four protocols. PPP scores over 25% are indicated in bold. These are the programs we used for in-depth analysis.

of the four protocols.

$$\text{PPP score} = \frac{4}{\frac{1}{\text{auPRC}(1A)} + \frac{1}{\text{auPRC}(1B)} + \frac{1}{\text{auPRC}(2A)} + \frac{1}{\text{auPRC}(2B)}}$$

The harmonic mean is used as it reduces the effect of high outliers, while at the same time it increases the effect of low scores. As such it will favor programs that provide a stable performance over all protocols.

For the in-depth analysis, we can only consider the predictions at one threshold. The optimal threshold is thus determined by calculating the F-score, i.e. the harmonic mean of precision and recall, for each precision–recall pair, and selecting the threshold for which the F-score is maximal.

$$F = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Determining the optimal threshold is done on the precision–recall pairs obtained by protocol 2A. We used protocol 2A, because it can be considered the most comprehensive and correct protocol: it uses the CAGE dataset (most comprehensive), and it uses the actual overlap and distance between TSRs and prediction (most correct).

2.5 Classes of promoters

We classify promoters in so-called shape classes using the method described in Carninci *et al.* (2006). Single peak (SP) promoters are TSRs that have all tags closely grouped together (the majority of TSSs are not >4 nt apart). The second category contains the promoters that have a broad distribution of TSSs (BR). To differentiate between different cases in the broad category, were two additional defined classes referred to as 'broad distribution with a dominant peak (PB)' and 'promoters with a multi-modal distribution of TSSs (MU)'.

The shape class of a tag cluster is determined by testing a condition for each shape class in a particular order. The first test that succeeds indicates the shape class. We first test for SP, next for PB and finally for MU. If none of the tests succeeds, the promoter is assigned the BR label. A TSR has the SP shape if over 50% of all individual tags starts no further than 4 nt apart. The PB shape is defined as any TSR for which the ratio of the number of tags at the two most commonly used locations exceeds 2. A TSR has a multi-modal distribution if the distance of any two subsequent 5% percentiles of the tag distribution exceeds 25% of the total length of the TSR.

We consider only clusters with at least 100 tags. When applied to our pre-processed CAGE dataset, 5570 clusters have at least 100 tags. Of these clusters, 944 have a sharp peak (SP), 498 have a broad dominant peak (PB), 3188 clusters have a multi-modal distribution (MU) and 940 do not fit in any of the other classes (BR).

Another subdivision of TSRs was made to assess the bias of PPPs toward rare and common transcription initiation event. To assess the performance on TSRs that are rarely used and TSRs that are commonly used, we create two datasets. The set with rarely used TSRs contains all TSRs that have exactly 2 tags, while the commonly used TSRs have at least 25 associated tags. This results in 14 363 common TSRs and 85 519 rare TSRs.

3 RESULTS

3.1 Benchmarking PPPs

We have applied the four protocols described in the previous section to 17 PPPs that have been published in the literature, and for which we were able to procure genome-wide predictions on the human genome or for which the software is available for free for academic use. We ran the latter programs ourselves on a grid, requiring over 30 000 CPU hours to complete the human genome. For 15 programs this resulted in predictions with scores, while for 2 programs we only have predictions without a score (Wu-method and PromoterExplorer). Results of this analysis are reported in Table 2.

In earlier work, we used the F-score to identify the PPP that performs best on a number of datasets. However, there are some drawbacks on using the F-score as single criterion. First of all, to compare programs fairly, one has to optimize the threshold of the program on the validation set. Even when this is done properly, the optimized F-score is only a single point on the PRC that can be obtained with the program. Hence, the F-score does not provide any insight in to the full potential of the PPP under investigation.

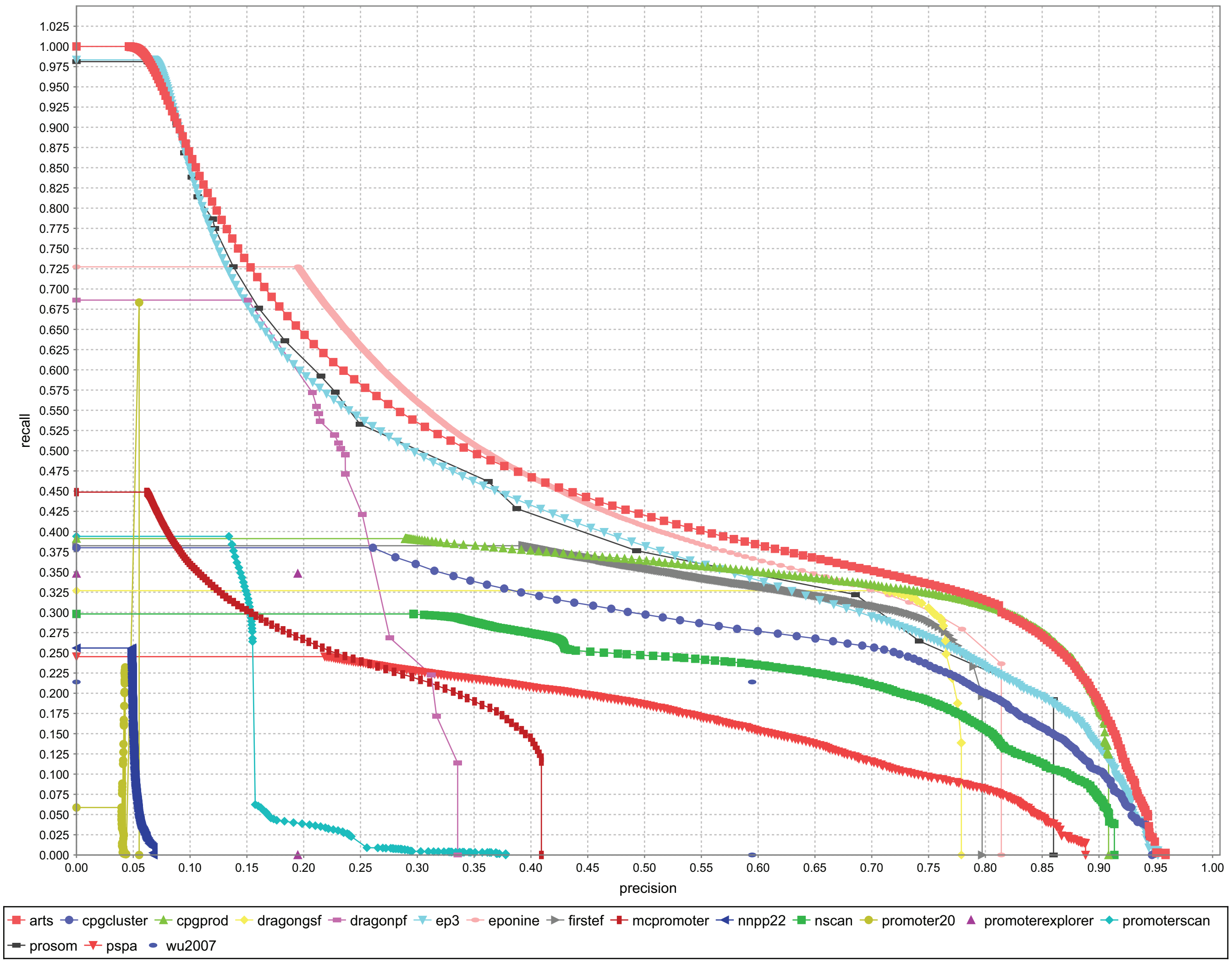

Fig. 2. PRCs for all PPPs when evaluated with protocol 2A.

For some applications one would be more interested in how the PPP behaves under very high precision conditions while other researchers could be interested in the behavior at very high recall rates. As suggested before (Sonnenburg *et al.*, 2006), the fairest way to compare PPPs is by calculating the complete PRC and then computing the area under this curve. Figure 2 shows the PRCs for all 17 PPPs for protocol 2A, and the remaining protocols result in similar plots (data not shown). In a PRC, graphs most to the top-right indicate the better performing programs. We see that there are three graphs that dominate the first part of the plot; these are the graphs corresponding to the ARTS, EP3 and ProSOM programs. At about 20% precision, the graph of Eponine starts dominating, but ARTS, EP3 and ProSOM remain closeby. PromoterExplorer and the Wu-method do not have a full graph, as they do not provide scores; they are represented by a single point in the plot.

To be able to calculate the full area under the curve, we included one extra point to close the curve. This added point has the same recall as the point with the lowest precision in the curve, but has precision value 0. Adding this point allows the auPRC to be calculated for each PPP (including those with only one precision–recall pair) and it will put programs that do not cover the complete precision spectrum on equal footing with programs that do cover it. The graphs of Eponine and DragonPF indicate that auPRC for those programs may be underestimated. However, we ran the programs at the lowest threshold that would work on our system. So it seems that Eponine and DragonPF do not allow us to explore them in an extreme setting with very low precision. On the other extreme of the plot, we see that the graph of some programs drops to 0 from a relatively high recall score. This indicates that some programs do not allow us to explore extreme high precision scores.

The area under the curve is reported in Table 2 in the columns marked with a protocol identifier. Each of the four protocols assigns the highest auPRC to ARTS. To aggregate the results of the four protocols in one measure, we calculate the harmonic mean of the auPRC of the four protocols and report it as the PPP score in the last column of Table 2. This score is an indication of the overall performance of the PPP on different tasks and using different evaluation algorithms. Four programs have a PPP score over 0.25: ARTS, Eponine, EP3 and ProSOM. ARTS clearly performs best with 34%, while the other three programs are closely together around 28%. All further analyses were performed on all 17 PPPs, but we only report results for the four best PPPs as these are the most interesting. The two methods for detection of CpG islands (CpGcluster and CpGProD) work relatively well with protocol 2A, especially since they have not been designed to predict promoters, but rather to detect CpG islands. This again indicates that CpG islands are a very strong signal for promoter detection and that the presence of a CpG island is often sufficient for promoter identification. FirstEF and NScan are two methods that try to predict more than just the core promoter. FirstEF tries to identify the structure of the first exon and NScan tries to construct a complete gene model. This additional gene-oriented modeling clearly improves the performance of the programs under the 1B and 2B protocols. In the 1A and 2A protocols, these programs have lower scores than programs that have a comparable performance on 1B and 2B. Promoter 2.0 and NNPP2.2 obtained total scores of <1% indicating that these programs are not suited to identify promoters. Striking is that Eponine, which is around since 2001, is still one of the only four promoter predictors that obtain a total score above 20%.

3.2 Positional distribution of predictions

Because the evaluation protocols allow a certain distance between the prediction and the actual TSR, one should always check how well the predictions are positioned around the target site. In this section, we analyze the positional specificity with respect to the closest TSR for the four top performing programs. For the positional specificity to the closest TSR we use the optimal threshold as determined by protocol 2A. Figure 3 shows the positional distribution of predictions relative to the closest TSRs. Note that all TSRs that overlap with a prediction have distance 0, which explains the peak at position 0 in the graph. The *x*-axis represents the distance to the TSR. The *y*-axis shows the number of tags (logarithmic scale). We can see that all programs have by far the largest fraction of the tags overlapping with predictions. ARTS and Eponine make more predictions that are not overlapping with the TSR than EP3 and ProSOM, but the predictions are mostly in the vicinity of the TSR. Further from the TSR there is little difference between the four programs. Overall, all four programs have well-localized predictions with respect to the annotated TSRs.

3.3 Classes of promoters

To analyze the bias of promoter predictors to particular shape classes, we analyzed the recall obtained by each program for each of the classes. We use the optimal threshold as determined by protocol 2A. For this threshold, we determine the number of tags of the shape class that is discovered. For these analyses only the recall is informative. The precision of a method can only be calculated on the complete reference set and for this analysis we only use a subset of the reference.

Table 3 shows the fraction of TSRs of each class that is identified at the optimal threshold. The scores in the table are the fraction of tags marked as SP, PB, MU or BR that is recovered by the program. Single peak TSRs are less recovered by PPPs than any of the broad categories (BR, PB and MU). The TATA motif is known to be overrepresented in the SP class and these promoters are commonly associated with tissue-specific genes, while the BR, PB and MU classes are strongly associated with CpG islands, commonly found in housekeeping genes (Carninci *et al.*, 2006). This indicates that the current state-of-the-art in promoter prediction is biased toward housekeeping genes that contain CpG islands.

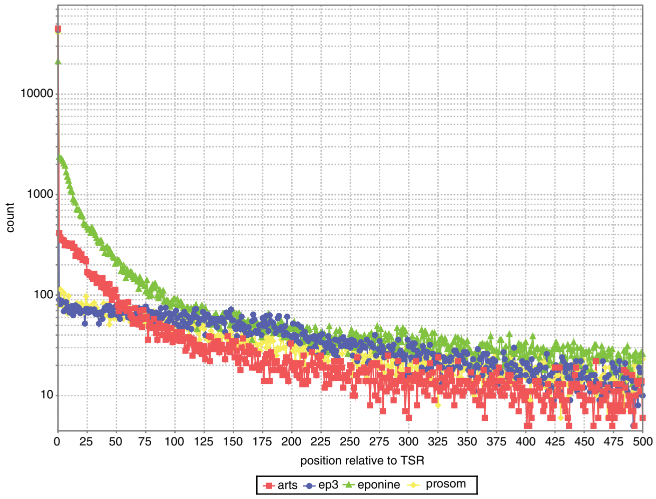

Fig. 3. Positional specificity for predictions around TSRs. The positional specificity is determined by using the optimal threshold as determined with protocol 2A.

Table 3. Recall score for each of the top four PPPs on each of the four promoter classes and on the Rare and Common TSR set

Name	SP	PB	MU	BR	Rare	Common
ARTS	0.58	0.90	0.93	0.95	0.23	0.81
EP3	0.52	0.82	0.85	0.84	0.23	0.74
Eponine	0.69	0.92	0.94	0.96	0.24	0.80
ProSOM	0.51	0.83	0.81	0.83	0.21	0.71

The recall is calculated with the optimal threshold as determined with protocol 2A.

One caveat with the last analysis is that although we make a distinction between different TSR shapes, we still look at TSRs that have at least 100 associated tags, which means that these TSRs have a high initiation rate. To compare the performance of the four PPPs on less common TSRs, we use the sets of rarely used and commonly used TSRs (see Section 2). The fraction of identified TSRs for these two sets is shown in the last two columns of Table 3. All four PPPs have a strong bias toward strong TSRs, covered by a lot of tags.

3.4 Pair-wise prediction overlap

To calculate the overlap between predictions made by different programs, we divided the genome in chunks of 500 nt. The predictions for each program are determined as predicted regions that have a score that is higher than the optimal threshold determined by protocol 2A. Table 4 shows the fraction of predictions that is shared between two PPPs. In this table, we only included the four PPPs that obtained a PPP score over 0.25 in the benchmark analysis presented in Table 2. The value in a cell with column title A and row title B should be interpreted as the fraction of predictions of program A that are contained in the predictions of program B. For example, the value in row 2, column 1 is the fraction of predictions

Table 4. Pair-wise prediction overlap for the top four programs based on PPP score

	ARTS	EP3	Eponine	ProSOM
ARTS		0.57	0.29	0.74
EP3	0.36		0.21	0.75
Eponine	0.76	0.85		0.97
ProSOM	0.37	0.59	0.18	

Details on the interpretation of the values can be found in the main text.

from ARTS that is also predicted by EP3. In this case, 36% of the predictions of ARTS are also predicted by EP3.

Some interesting observations can be made from this table. The row and column marked with Eponine indicates that the majority of the predictions made by all other programs are contained within the Eponine prediction set. All other rows indicate that the other programs generally have at least 25% unique predictions. This last observation may indicate that the predictions that are not predicted by the two programs are more likely to be wrong. Another possible explanation for this phenomenon is that since most PPPs in Table 4 are built on completely different concepts, they make use of different parts of information available in the sequence. One way to harness this insight is to aggregate multiple PPPs to use more of the information that is available in the sequence.

4 DISCUSSION AND CONCLUSION

In this article, we proposed a set of protocols to fairly evaluate PPPs. The four protocols we described can be used when different types of data are available. For the A protocols, one needs a set of experimentally determined TSRs, which is not available for all species. The B protocols can be used when only gene annotation is available for the target organism, which should be the case for most species. Because the A protocols use a more biologically inspired validation and they do not ignore intergenic predictions, one should prefer one of the two A protocols. The protocols starting with 2 are more accurate as they use the actual spatial organization of predictions and reference items, while the protocols starting with 1 reduce this organization to fixed bins. The benchmark should be done by calculating the complete PRC and computing the area under this curve. When running more protocols, one can calculate the harmonic mean of the individual auPRCs as a single score for the PPP.

We benchmarked 17 PPPs using the proposed schema and further investigated the four PPPs that performed best in the benchmark in terms of positional preference and prediction bias. While the performance of the top four is about the same (Table 2), these four programs work on different principles and were designed for different tasks. ARTS is designed to score all nucleotides in the genome, EP3 and ProSOM were designed to score putative TSRs and Eponine was designed to predict core promoter regions. One of the differences between the four programs is the number of predictions they make to obtain their scores. EP3 and ProSOM have around 65 000 predictions, while ARTS and Eponine have 432 117 and 1 320 964 predictions, respectively. Although there is such a large difference in prediction count, the final results are about the same, indicating that a lot of the predictions are redundant. In case of ARTS, this was to be expected as the program is designed to score all nucleotides in the genome. For Eponine, the large number of predictions is unexpected, since that program is meant to identify complete core promoters. Further investigation may be performed into the nature of the predictions and the extent of the redundancy. While our benchmark identified the PPPs that obtain the highest PPP scores, there are other factors that influence which PPP can or should be used. The first additional criterion is the availability. Eponine, EP3, ProSOM and ARTS are freely available for download. A second additional criterion may be the applicability domain of the software. Eponine and ProSOM have been designed to work for any mammalian genome, EP3 was designed as a generic predictor for eukaryotic genomes and ARTS has only been reported to work for the human genome.

The overlap between the sets of predictions made by the four programs is limited (Table 4). As a result, each program has a number of unique predictions indicating that each of the programs has a different information usage. It would be worthwhile to investigate how the information of multiple programs can be aggregated.

In conclusion, this article proposes a standard for the evaluation of promoter prediction software and identified four high-scoring PPPs. For these four PPPs we did an in-depth analysis of the predictive performance, promoter class specificity, overlap between predictors and positional bias of the predictions.

As future work in promoter prediction, some challenges remain. The main effort has been done in a number of model organisms, but there are plenty of other higher eukaryote genomes that will need promoter identification. In evaluating predictions, we focused on the association between predictions and TSRs or gene starts. However, a lot more data is available that may prove useful as evaluation data, e.g. promoter motifs, DNA hypersensitivity sites and chromatin structure signatures. In the near future, the importance of promoter prediction techniques will only increase, as ever more genomes are sequenced, requiring ever more accurate computational techniques to extract knowledge from these vast amounts of data.

ACKNOWLEDGEMENTS

We would like to thank Gunnar Raetsch, Soeren Sonnenburg, Alexander Zien and Uwe Ohler for early discussions on challenges in promoter prediction evaluation.

Funding: Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) (to T.A.). Research Foundation Flanders (FWO-Vlaanderen) post-doctoral grant (to Y.S.).

Conflict of Interest: none declared.

REFERENCES

Abeel,T. *et al.* (2008a) Generic eukaryotic core promoter prediction using structural features of DNA. *Genome Res.*, **18**, 310–323.

Abeel,T. *et al.* (2008b) ProSOM: Core promoter prediction based on unsupervised clustering of DNA physical profiles. *Bioinformatics*, **24**, i24–i31.

Bajic,V.B. and Brusic,V. (2003) Computational detection of vertebrate RNA polymerase II promoters. *Methods Enzymol.*, **370**, 237–250.

Bajic,V.B. *et al.* (2002) Dragon Promoter Finder: recognition of vertebrate RNA polymerase II promoters. *Bioinformatics*, **18**, 198–199.

Bajic,V.B. *et al.* (2004) Promoter prediction analysis on the whole human genome. *Nat. Biotechnol.*, **22**, 1467–1473.

Bajic,V.B. *et al.* (2006) Performance assessment of promoter predictions on ENCODE regions in the EGASP experiment. *Genome Biol.*, **7 (Suppl 1)**, S3.1–S3.13.

Carninci,P. *et al*. (2006) Genome-wide analysis of mammalian promoter architecture and evolution. *Nat. Genet.*, **38**, 626–635.

Davuluri,R.V. *et al*. (2001) Computational identification of promoters and first exons in the human genome. *Nat. Genet.*, **29**, 412–417.

Down,T.A. and Hubbard,T.J.P. (2002) Computational detection and location of transcription start sites in mammalian genomic DNA. *Genome Res.*, **12**, 458–461.

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

Frith,M.C. *et al*. (2008) A code for transcription initiation in mammalian genomes. *Genome Res.*, **18**, 1–12.

Gross,S.S. and Brent,M.R. (2006) Using multiple alignments to improve gene prediction. *J. Comput. Biol.*, **13**, 379–393.

Hackenberg,M. *et al*. (2006) CpGcluster: a distance-based algorithm for CpG-island detection. *BMC Bioinformatics*, **7**, 446.

Knudsen,S. (1999) Promoter2.0: for the recognition of PolII promoter sequences. *Bioinformatics*, **15**, 356–361.

Ohler,U. *et al*. (2000) Stochastic segment models of eukaryotic promoter regions. *Pac. Symp. Biocomput.*, **1**, 380–391.

Ponger,L. and Mouchiroud,D. (2002) CpGProD: identifying CpG islands associated with transcription start sites in large genomic mammalian sequences. *Bioinformatics*, **18**, 631–633.

Prestridge,D.S. (1995) Predicting Pol II promoter sequences using transcription factor binding sites. *J. Mol. Biol.*, **249**, 923–932.

Reese,M.G. (2001) Application of a time-delay neural network to promoter annotation in the Drosophila melanogaster genome. *Comput. Chem.*, **26**, 51–56.

Sandelin,A. *et al*. (2007) Mammalian RNA polymerase II core promoters: insights from genome-wide studies. *Nat. Rev. Genet.*, **8**, 424–436.

Shiraki,T. *et al*. (2003) Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage. *Proc. Natl Acad. Sci. USA*, **100**, 15776–15781.

Sonnenburg,S. *et al*. (2006) ARTS: accurate recognition of transcription starts in human. *Bioinformatics*, **22**, e472–e480.

Wakaguri,H. *et al*. (2008) Dbtss: database of transcription start sites, progress report 2008. *Nucleic Acids Res.*, **36**, D97–D101.

Wang,J. and Hannenhalli,S. (2006) A mammalian promoter model links cis elements to genetic networks. *Biochem. Biophys. Res. Commun.*, **347**, 166–177.

Wu,S. *et al*. (2007) Eukaryotic promoter prediction based on relative entropy and positional information. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, **75**, 041908.

Xie,X. *et al*. (2006) PromoterExplorer: an effective promoter identification method based on the AdaBoost algorithm. *Bioinformatics*, **22**, 2722–2728.