BIOINFORMATICS ORIGINAL PAPER Vol. 23 no. 4 2007, pages 414–420
doi:10.1093/bioinformatics/btl639

Sequence analysis

In search of the small ones: improved prediction of short exons in vertebrates, plants, fungi and protists

Yvan Saeys, Pierre Rouzé[1] and Yves Van de Peer*

Department of Plant Systems Biology, Flanders Interuniversity Institute for Biotechnology (VIB) and [1]Laboratoire Associé de l'INRA (France) Ghent University, Technologiepark 927, B-9052 Ghent, Belgium

Received on August 30, 2006; revised on November 24, 2006; accepted on December 14, 2006
Advance Access publication January 4, 2007
Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: Prediction of the coding potential for stretches of DNA is crucial in gene calling and genome annotation, where it is used to identify potential exons and to position their boundaries in conjunction with functional sites, such as splice sites and translation initiation sites. The ability to discriminate between coding and non-coding sequences relates to the structure of coding sequences, which are organized in codons, and by their biased usage. For statistical reasons, the longer the sequences, the easier it is to detect this codon bias. However, in many eukaryotic genomes, where genes harbour many introns, both introns and exons might be small and hard to distinguish based on coding potential.

Results: Here, we present novel approaches that specifically aim at a better detection of coding potential in short sequences. The methods use complementary sequence features, combined with identification of which features are relevant in discriminating between coding and non-coding sequences. These newly developed methods are evaluated on different species, representative of four major eukaryotic kingdoms, and extensively compared to state-of-the-art Markov models, which are often used for predicting coding potential. The main conclusions drawn from our analyses are that (1) combining complementary sequence features clearly outperforms current Markov models for coding potential prediction in short sequence fragments, (2) coding potential prediction benefits from length-specific models, and these models are not necessarily the same for different sequence lengths and (3) comparing the results across several species indicates that, although our combined method consistently performs extremely well, there are important differences across genomes.

Supplementary data: http://bioinformatics.psb.ugent.be/

Contact: yvan.saeys@psb.ugent.be

1 INTRODUCTION

With the ever-increasing pace of genomes currently being sequenced, there is a great need for fast and accurate computational tools that automatically identify the genes in these genomes. This is a very challenging problem, because in most eukaryotic genomes only a small fraction of the genome sequence actually does code for genes. In human, for example, it is estimated that only 1.2% of the genome encodes proteins (International Human Genome Sequencing Consortium, 2004), making it very hard for automatic methods to reliably identify the informational parts in the sequence. In addition to the sparseness of protein coding regions in the DNA, another challenge is to uncover the exact gene structure, in particular when coding regions (exons) are interrupted by non-coding regions (introns) as is usually the case in eukaryotic genomes. Exon size can vary from a few base pairs to thousands of base pairs. Depending on the genome, a large fraction of these exons might be considered small (<200 nt) and in particular these pose a major problem for gene calling since they are easily missed by current gene prediction tools, resulting in incomplete or wrong annotations (Mathé *et al*., 2002; Wang *et al*., 2002; Brent and Guigo, 2004).

Therefore, methods that assess the coding potential in short sequences are of major importance in gene prediction and genome annotation. The coding potential is a measure of how likely it is that a given sequence encodes a protein or at least part of it, and is based on the particular structure of coding sequences. Nucleotides in protein coding sequences are translated into amino acids by triplets (codons), and these are most often not equally used by the organism, a phenomenon referred to as codon bias. The prediction of coding potential lies at the heart of every gene predictor (Borodovsky and McIninch, 1993; Salzberg *et al*., 1998; Schiex *et al*., 2001; Majoros *et al*., 2004; Stanke *et al*., 2006). On the one hand, coding potential is used to assign a score to potential exons, while on the other hand it is used as an additional measure to predict functional sites, such as translation initiation sites and splice sites. These functional sites are all characterized by the fact that they represent a transition from a coding/non-coding sequence to a non-coding/coding sequence. As a result, information regarding the coding potential of the upstream and downstream flanking sequences of a functional site provides vital information to correctly identify the site.

To capture peculiarities from coding sequences, such as codon bias, a large number of protein coding measures have been developed (Fickett and Tung, 1992; Tiwari *et al*., 1997; Kotlar and Lavner, 2003; Gao and Zhang, 2004). Particular classes of methods that are frequently being used to detect coding potential are Markov chain models. These were pioneered in the gene finder GenMark (Borodovsky and McIninch, 1993), and since then a variety of these methods have been used for coding potential prediction (CPP; Salzberg *et al*., 1998, 1999; Majoros *et al*., 2003, 2004; Stanke *et al*., 2006). While Markov models are known to perform well for CPP for longer sequences (Borodovsky and McIninch, 1993), little is known about their performance on shorter sequences, although genes containing small exons are often miss-predicted,

*To whom correspondence should be addressed.

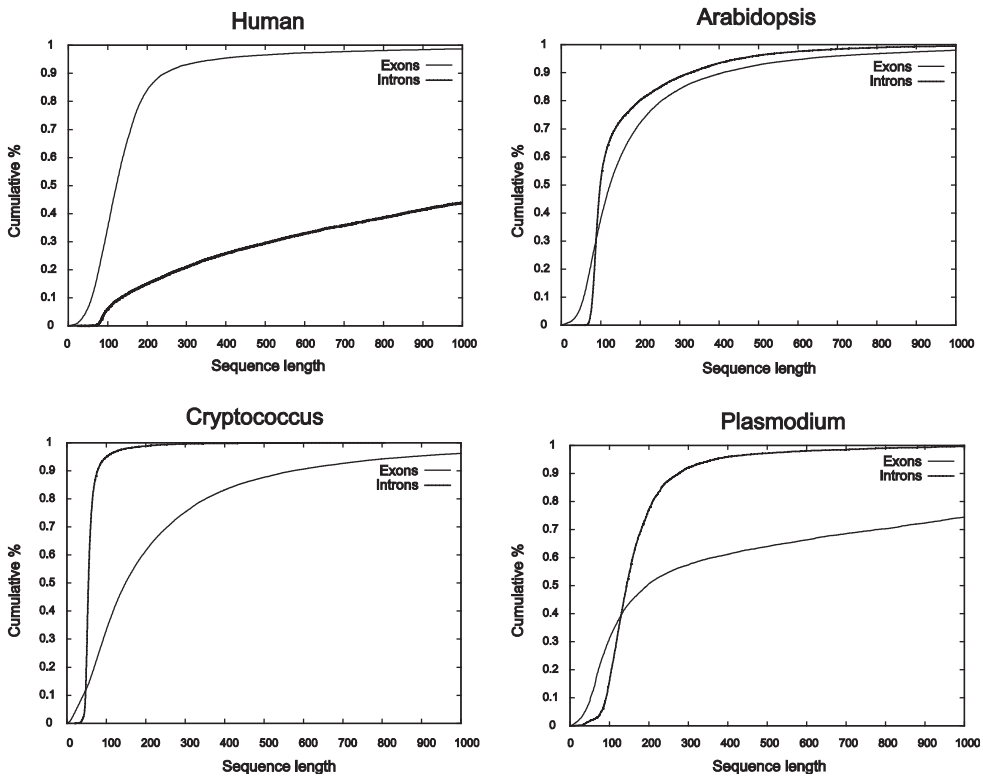

Fig. 1. Cumulative frequencies for the distribution of exon lengths in the genomic sequences of Human, *Arabidopsis*, *Cryptococcus* and *Plasmodium*. Figures are shown for exon and intron lengths up to 1000 nt.

as was recently demonstrated for several organisms by identification of transcribed sequences using genome tiling arrays (Stolc *et al*., 2005; Bertone *et al*., 2006). However, assessing the coding potential of short sequences is not self-evident because intrinsic signals, such as codon biases are harder to detect with shorter sequence lengths. As a result, more advanced methods are needed to specifically deal with CPP in short sequences.

Although identifying short protein coding sequences is considered an important issue, up to now a large-scale analysis of the problem has not been performed. In the literature, the issue was only briefly touched upon by Gao and Zhang (2004), who performed an analysis on short human sequences using the Zcurve method. Their method starts from a transform (Z-transform) of the DNA sequence from which they calculated several features, such as frequencies of frame-dependent *k*-mers, which are subsequently fed into a linear discriminant classifier. However, their method is not compared to state-of-the-art Markov models, which makes it difficult to assess the real value of it.

Here, we propose a novel method for the prediction of CPP using complementary sequence features and compare it to a large number of state-of-the-art models. In addition, we have applied our approach to representatives of different eukaryotic kingdoms, such as animals, plants, Fungi and Apicomplexa, and provide a cross-species comparison of the results. Figure 1 shows the cumulative distribution of exon and intron lengths in the four species used in the current study. As can be seen, depending on the genome, approximately one-third of all exons is smaller than 100 nt, while 50–85% is smaller than 200 nt. Small exons thus represent a considerable fraction of all exons in a genome and consequently, any improvement in the detection of small exons will have a significant impact in gene prediction and annotation. In the current study, we start from the conjecture made by Fickett and Tung (1992) in the early nineties, namely those future algorithms for CPP should combine different characteristics. By doing so, and by adding additional information, an extensive set of discriminative features for distinguishing between coding and non-coding sequences is obtained. Subsequently, we have designed two new classification models that combine these features, and perform an extensive comparative analysis of these models with current models for CPP. We focus on the analysis of short sequence fragments (<200 nt), as the challenge for coding prediction lies exactly in the analysis of such short sequences.

2 METHODS

2.1 Dataset construction

Genomic data from three different species was analyzed, each representing a peculiar genome style as well as a major class of eukaryotes, namely Human as a representative for vertebrates, *Arabidopsis thaliana* for plants, and *Cryptococcus neoformans* for fungi. In addition, we also included *Plasmodium falciparum* as a representative for obligatory parasites belonging to the kingdom Apicomplexa. Data for these four species was downloaded from publicly available databases (Table 1). For each of these datasets, the following procedure was used to extract the datasets for CPP. In a first step, the datasets were cleaned by removing genes with wrong start or stop codons, in-frame stop codons or genes whose length was not a multiple of three. Next, coding exons were extracted as positive

Table 1. Publicly available websites for the sequence and annotation data for the four species analyzed in this study.

Species	Provider	Website
Human, *A. thaliana*	Tigr	http://www.tigr.org/software/traindata.shtml
C. neoformans	The Broad Institute	http://www.broad.mit.edu/annotation/fungi/cryptococcus_neoformans/
P. falciparum	PlasmoDB	http://plasmodb.org/restricted/Gridddpf.shtml

Table 2. Statistics of the datasets for the four organisms analyzed here.

	Human	*Arabidopsis*	*Cryptococcus*	*Plasmodium*
#Genes	1500	8501	6500	5363
#Exons	11 497	44 328	41 142	12 759
#Introns	9997	35 960	34 757	7469
Exon distribution				
<42	438	1780	4053	750
[42,63]	838	3944	2893	1065
[63,87]	1648	7116	4281	1476
[87,108]	1655	5662	3010	959
[108,129]	1592	4567	3121	737
[129,162]	2026	4943	3892	804
[162,192]	1227	3210	2751	518
>=192	2073	13 106	16 541	6450
Amount of training data	10 481 kB	16 385 kB	13 085 kB	14 745 kB

For each organism, the number of genes, introns and exons in the dataset is shown. The distribution of exons in terms of their lengths is shown in the middle part of the table. The lower part represents the amount of training data that can be used in each cross-validation fold.

learning examples, while introns and UTR sequences were extracted as negative learning examples. As we were interested in analyzing the accuracy of prediction algorithms for short exons, all sequences were divided into length classes (similar to Gao and Zhang, 2004): <42 nt, [42–63nt], [63–87nt], [87–108nt], [108–129nt], [129–162nt], [162–192nt] and 192 or more nt.

For testing purposes we adopted the following strategy. For each length class in the range [42–192nt], an equal number of negative examples were added to the positive examples to obtain a balanced dataset. In the case of insufficient non-coding sequences for a particular length class, additional sequences were extracted from the length class of 192 or more. This yields, for each length class, a balanced dataset, which was independently split five times in half, obtaining 10-folds to test on. For each of these 10-folds, all the other available data (including the data from all other length classes) were used as training data. The statistics of the different datasets are shown in Table 2.

2.2 Markov models for coding potential prediction

Markov models are by far the technique mostly used for CPP. When used for this purpose, Markov models are used in a generative way, which means they learn a model of the joint probability $p(x,y)$ of the sequences x and the label y (where y can be either coding or non-coding), and make their predictions by using Bayes rule to calculate $p(y \mid x)$, and then picking the most likely label y. In the current study, we evaluated different types of Markov models as baseline classifiers to compare our new algorithms with. Four types of models with increasing complexity were used, namely fixed order Markov models (FOMM), variable order Markov models (VOMM), interpolated Markov models (IMM) and interpolated context models (ICM). For our purposes, the three-periodic version of the models was used to learn p(x,coding) and the homogeneous version was used to learn p(x,non-coding). Adapted versions of the following, publicly available programs were used to train and test these models: Exonomy (Majoros *et al*., 2003) for FOMM, Tigrscan (Majoros *et al*., 2004) for VOMM and IMM, and Glimmer2.13/GlimmerHMM (Salzberg *et al*., 1998) for ICM training/testing.

2.3 Combining complementary sequence features

Most methods for CPP focus on one type of sequence information to discriminate between coding and non-coding sequences. In Markov models, the sequence information that is used essentially consists of compositional information in the form of *k*-mer occurrences. In the case of global counts of these *k*-mers, this results in a homogeneous Markov model, while in the case of frame-dependent counts the model is termed three-periodic. Another class of methods focuses on a transform of the DNA sequence, after which features that relate to the specific transformation are extracted, and used for classification. The most commonly used methods either use features based on the Fourier transform (Silverman and Linsker, 1986; Voss, 1992; Tiwari *et al*., 1997; Kotlar and Lavner, 2003), or the Z-transform (Gao and Zhang, 2004). In our work, we combined different types of features, hoping that the combined strength of these complementary sequence characteristics results in a better separation between coding and non-coding sequences, thereby focusing especially on short ($\sim \leq$200 nt) sequences. The following types of features were used:

- Frame-dependent *k*-mers. For each of the three possible reading frames, *k*-mer frequencies ($1 \leq k \leq 3$) were calculated, resulting in 252 [$= 3\times(4+16+64)$] features.
- In-frame *k*-mers. Assuming the sequence is in reading frame 1 (the start of the sequence coincides with the start of a codon), in-frame *k*-mer frequencies ($4 \leq k \leq 6$) were calculated, resulting in a set of 5376 features.
- Frameless *k*-mers. For each possible k-mer ($1 \leq k \leq 3$), the global frequencies of occurrence are calculated (i.e. without taking into account the reading frame). This results in 84 features.
- Fourier transforms features. The most common way to apply Fourier analysis to DNA sequences is to decompose them first into four binary indicator sequences (Silverman and Linsker, 1986; Voss, 1992), apply the Fourier transform to each of these sequences, and then sum the Fourier coefficients. The following features, derived from the Fourier transform, were used: (1) For each of the four indicator sequences, the magnitude of the peak at frequency 1/3 in the Fourier spectrum (four features). We refer to Voss (1992) for more details. (2) The global magnitude at frequency 1/3, which is the sum of all four magnitudes of the indicator sequences (one feature). (3) The signal-to-noise ratio of the peak at frequency 1/3 (one feature; See Tiwari *et al*., 1997 for details).
- *Z*-curve parameters. The *Z*-curve parameters are calculated for the frequencies of frame-dependent *k*-mers ($1 \leq k \leq 3$), using the *Z*-transform of DNA sequences, as exemplified in Gao and Zhang (2004). This results in a set of 189 features [frame-dependent parameters of mononucleotides (9), di-nucleotides (36) and tri-nucleotides (144)].
- Run features. For each of the non-trivial (i.e. excluding the empty subset and the full set) subsets of {A,T,C,G}, a new sequence is constructed by replacing each base present in the subset with 1 and replacing each base not in the subset with 0. Using this transform of the sequence, the number of runs of 1's of length 1, 2, 3, 4, 5 and greater than 5 are then counted. This results in a set of 84 features (Fickett and Tung, 1992).
- ORF feature. Given a sequence and an assumed reading frame, this feature denotes whether there is an in-frame stop codon presents (Wang *et al*., 2002).

2.4 A discriminative model for coding potential prediction

In order to combine all available evidence to discriminate between coding and non-coding sequences, all previously mentioned features are combined with a classification model. We decided to adopt a discriminative classification model that deals well with high-dimensional feature spaces, and that is able to handle large datasets: the linear Support Vector Machine (SVM) (Boser *et al*., 1992). In discriminative learning, the posterior $p(y \mid x)$ is modelled directly, instead of solving a more general problem as in generative learning, where the intermediate step $p(x \mid y)$ is modelled (Ng and Jordan, 2002). As input features for the SVM, all features mentioned above are combined, resulting in a set of 5992 features describing each sequence fragment. Features that depend on the length were length-normalized, and before training the SVM, all features were scaled between 0 and 1. The C-parameter of the SVM was tuned using a 5-fold cross-validation of the training set. For the ORF-feature, we applied a post-processing step to the algorithm, ensuring that sequences with in-frame stop codons are always predicted as negatives. This was done by setting the output of the SVM to a very large negative value. For the implementation, we made use of the SVMlight package (Joachims, 1998).

2.5 Feature subset selection

To investigate to what extent the features in the constructed set are all equally important or necessary to discriminate between coding and non-coding sequences, feature selection (Kohavi and John, 1997) was performed. In feature selection, one seeks a minimal subset of relevant features that achieves maximal classification performance. Benefits of applying feature selection include better classification performance, faster classification models (because less features have to be taken into account), smaller databases (less features are needed to describe the training instances), and the ability to gain more insight into the process that is being modelled (Saeys *et al*., 2004). In our work, we adopted a Markov-blanket based filter approach, introduced by Koller and Sahami (1996). This algorithm has a solid mathematical basis, and has the advantage of being reasonably fast and taking into account feature dependencies. In our experiments, feature selection was performed for each organism separately using 50% of the sequences with length between [42,192]. The Markov-blanket feature selection method returns a ranking of the features, which can be used afterwards to eliminate features. As we have no prior knowledge on the size of a good feature set for the problem of CPP, various feature subset sizes were evaluated, ranging from 10 features to the full set of 5992 features. The following set of feature subset sizes was evaluated: {10, 20, 50, 100, 150, 200, 250, 500, 750, 1000, 1500, 2000, 2500, 5992}.

2.6 A hybrid Markov-SVM model

In order to investigate if both strengths of a generative approach (Markov model) and a discriminative approach (SVM) could be combined, we designed a very simple extension of our SVM model. In this extension, the score obtained by a Markov model is subsequently used as an additional feature for the SVM model. The score for a Markov model was calculated using the log odds score: $\log[p(x,\text{coding})] - \log[p(x,\text{non-coding})]$. This way, the SVM takes into account the classification obtained by the Markov model. One could see this as an ensemble learning approach, thereby stacking an SVM on top of the results of a Markov model, and using the Markov score combined with an additional set of features to build a more complex model. Instead of trying all possible combinations of the different types of Markov models with the SVM, we focused on one model that consistently obtained good results in all experiments, namely a fifth-order ICM.

2.7 Setup and evaluation

Many classifiers use an internal threshold value to determine the class when confronted with a test example. For a given threshold value, the classification performance of a classifier can be summarized by four numbers: TP (true positives = actual positive examples predicted as positives), TN (true negatives = actual negative examples predicted as negatives), FP (false positives = actual negative examples predicted as positives) and FN (false negatives = actual positive examples predicted as negatives). In many cases, these four numbers are combined into one number that summarizes the classification performance. Commonly used measures are:

Accuracy : (TP TN)/(TP TN FP FN)
TruePositive(TP)rate/recall/sensitivity : TP/(TP FN)
FalsePositive(FP)rate : FP/(FP TN)
Precision/specificity : TP/(TP FP)

However, comparing classifiers based on only one value (e.g. accuracy) often does not provide a fair comparison. This could be the case when for instance one classifier would have high-sensitivity and low-specificity, and another one would have low-sensitivity and high-specificity. Although having different classification performance, the accuracy measure for both classifiers could be comparable. A solution to this would be to compare both classifiers at the same levels of sensitivity or specificity by varying the decision threshold. This is the idea underlying the receiver-operator-curve (ROC), introduced by Provost and Fawcett (1997). On such a curve, the FP-rate is plotted on the *x*-axis versus the TP-rate (sensitivity or recall) on the *y*-axis. By varying the decision threshold, several values can be obtained for these rates. Classifiers can then be evaluated by comparing their ROC graphs, where better classifiers have a larger area under their ROC curve. This criterion is often used in machine learning to compare classifiers, and is termed the area under the ROC curve (AUC).

We also used, in addition to the AUC, another measure to compare the classification models and provide a comparison of all models at a specific level of sensitivity (TP-rate), allowing a fair comparison between all techniques. For all models, we fixed the sensitivity level at 0.95 and compared the FP-rate (FPR). This would be equivalent to drawing a horizontal line at the ROC graph at a TP-rate of 0.95 and comparing the intersections of the graphs with the line. All results were obtained using a 10-fold cross-validation, as explained earlier.

3 RESULTS AND DISCUSSION

3.1 Classification performance

For each genome, we compared the classification performance of different types of models for the prediction of coding potential: Markov models, methods based on a transform, and two new methods using a combination of features and a linear SVM. The different types of Markov models were tested using various orders (order 2 to 7 for FOMM, order 2 to 8 for VOMM and IMM, and order 2 to 11 for ICM). Two methods that were based on a transform were included in the test, namely the signal-to-noise ratio (SNR) of the peak at frequency 1/3 in the Fourier spectrum (Tiwari *et al*., 1997) and the *Z*-curve method (Gao and Zhang, 2004). Finally, two newly introduced algorithms were included: linear SVM employing different types of features in combination with a feature selection algorithm (FSS-SVM) and a hybrid model of a 5th order ICM and the FSS-SVM model (hybrid Markov-SVM).

Table 3 shows a summary of the main results obtained in this study and displays the FP-rate at a TP-rate (sensitivity) of 0.95 for each experiment. The complete results for all models (showing both the AUC and FP-rate) are available as Supplementary material. For each organism, experiments are grouped by model and length class. For each length class, the best method is shaded in grey. The column termed 'Markov' groups all types of Markov models, but only the best result (i.e. the lowest FP-rate) is shown. The best combination

Table 3. Results of the FP-rate at a TP-rate (sensitivity) of 0.95 for all tested organisms.

Length	Markov	SNR	*Z*-curve	FSS-SVM	Hybrid Markov-SVM
Human					
42	34.79 (ICM-8)	87.82	25.08	11.91 (1000)	*11.53 (500)*
63	22.53 (ICM-8)	83.69	13.46	*5.09 (750)*	*5.21 (750)*
87	23.39 (ICM-4)	75.55	14.19	*2.40 (1500)*	*2.47 (500)*
108	9.27 (ICM-8)	72.72	5.98	*1.23 (1500)*	*1.23 (1500)*
129	4.42 (ICM-11)	56.84	3.69	0.48 (1500)	*0.43 (1500)*
162	3.08 (ICM-11)	49.41	2.29	0.22 (all)	*0.17 (500)*
Arabidopsis					
42	14.42 (ICM-8)	86.19	22.02	10.29 (500)	8.33 (200)
63	0.43 (IMM-5)	80.98	25.30	1.22 (150)	0.40 (150)
87	0.08 (IMM-5)	77.93	14.93	0.52 (150)	0.26 (50)
108	0.66 (all)	74.81	12.02	1.04 (150)	0.88 (150)
129	0.00 (all)	69.57	9.05	0.32 (200)	0.39 (20)
162	0.00 (all)	59.12	8.37	0.39 (200)	0.28 (20)
Cryptococcus					
42	8.22 (FOMM-5)	87.96	8.65	5.02 (250)	1.90 (200)
63	3.94 (IMM-7)	82.86	9.20	3.90 (250)	1.28 (150)
87	5.26 (FOMM-5)	82.47	20.61	5.51 (250)	4.04 (250)
108	2.91 (IMM-6)	77.89	18.40	3.40 (250)	2.07 (150)
129	3.45 (ICM-8)	71.44	21.28	3.94 (250)	2.95 (200)
162	2.25 (ICM-8)	67.46	17.85	2.81 (150)	1.72 (200)
Plasmodium					
42	7.00 (ICM-8)	80.70	8.45	4.46 (all)	4.76 (50)
63	3.08 (ICM-8)	79.07	3.38	0.86 (750)	0.83 (750)
87	0.32 (ICM-11)	68.29	3.19	0.08 (750)	0.04 (20)
108	0 (FOMM-2)	68.72	0.02	0 (100)	0 (100)
129	0.13 (FOMM-2)	63.09	0.20	0.05 (150)	0.05 (20)
162	0 (ICM-4)	58.49	0.50	0.03 (50)	0.03 (50)

The column Markov shows the best result (lowest FP-rate) over all types of Markov models. In brackets, the best performing model is shown. SNR represents the results of the signal-to-noise ratio of the Fourier spectrum at frequency 1/3. FSS-SVM represents the best result of a linear SVM combined with feature selection, and Hybrid Markov-SVM represents the best result of a model that combines SVM and a 5th order ICM. In brackets, the size of the best feature subset is shown. The best results are grey-shaded.

(model-order) is shown in brackets. If several methods perform equally well, the simplest method was chosen. If all methods performed equally well, this is noted as (all). In a similar way, the results of the newly developed methods FSS-SVM and Hybrid Markov-SVM are shown. Here, numbers in brackets represent the size of the best feature subset. The notation (all) denotes the full feature set (5992 features). Again, if different subsets led to the best result, the smallest feature subset is chosen. Overall, several general trends can be observed, which are described in detail below.

3.2 Combining complementary sequence features clearly outperforms current Markov models

Analyzing the results of the different classification methods, it is clear that the new models FSS-SVM and Hybrid Markov-SVM outperform state-of-the-art Markov models. Overall, our new models obtain the best result in 75% of the cases, while the Markov models only obtain the best result in 20% of the cases. Comparing the simplest of the new models (FSS-SVM) to the best Markov model reveals that FSS-SVM outperforms the Markov model in >50% of all cases. In addition to this, the method needs far less features than the Markov models (a more detailed feature analysis follows further in the text). The second new model (Hybrid Markov-SVM) outperforms the best Markov model in 75% of all cases. This clearly illustrates that the inclusion of a single additional feature (the ICM-5 score) again improves the FSS-SVM model. This result justifies a hybrid approach for CPP, and illustrates the benefit of combining complementary sequence features.

When we focus on the very short length classes ([42–63nt] and [63–87nt]) it can be observed that our two new models outperform the best Markov model in >95% of the cases, regardless of the species the genome is analyzed from. This illustrates that state-of-the-art Markov models can indeed be improved when dealing with very short coding sequences. Comparison of the two new methods to the transform-based methods SNR and *Z*-curve shows that both new methods always outperform those as well. This is not surprising, as both SNR and *Z*-curve features are used as a component in the new methods. This again provides evidence that a combination of complementary sequence features greatly improves classification performance. Looking at the results of the *Z*-curve method over different species, it can be seen that only in the case of Human the *Z*-curve method improves on the best Markov model, confirming the results by Gao and Zhang (2004). However, this observation cannot be generalized to other species and—in general—the *Z*-curve method does not outperform state-of-the-art Markov models. Strikingly, the method based on the Fourier transform (SNR) performs drastically worse than all other methods, resulting in a very high false positive rate. However, it should be stressed that this method uses only one feature, and—in contrast to all other methods tested—requires no training. Given this information, the result of the SNR method is still far better than a random guessing classifier, which would obtain an FP-rate of 95% at the same sensitivity threshold.

3.3 The best model depends on the sequence length

Analyzing the results for the different length classes within a specific organism, it can be observed that there is not a single 'best' classification model. Rather, the best model is length-dependent, and even within the different types of methods (Markov versus SVM-based) it can be observed that the best order or feature subset size tends to vary along with the sequence length. However, it has to be noted that for some species, results of the best model of a specific type are more consistent than for other species. This is for instance the case for Markov models applied to the *Arabidopsis* data, where a 5th order IMM consistently performs very good, or in the case of FSS-SVM applied to *Cryptococcus*, where a subset of 250 features gives the best results for the majority of all length classes (Table 3). A more relaxed example is the case of Markov models applied to Human, where an ICM always obtains the best results, yet with different orders.

3.4 There is a great amount of variation across genomes

To our knowledge, this is the first large-scale analysis comparing several eukaryotic organisms with a focus on CPP for short

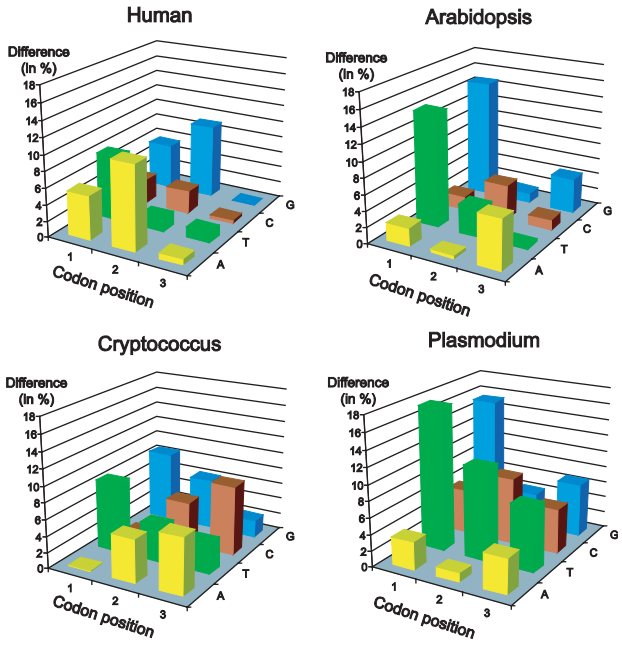

Fig. 2. Difference in percentage between the frequency of occurrence of each nucleotide at one of the three positions in a codon (1, 2 and 3). The differences are calculated as the absolute value of the difference in frequency between coding and non-coding sequences.

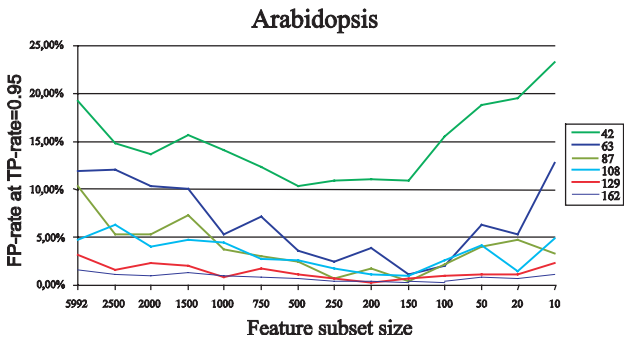

Fig. 3. Effect of feature selection on the classification performance for the new method FSS-SVM. The *x*-axis denotes the feature subset size, starting at the origin with the full feature set. Going further away from the origin, gradually more features are eliminated. The *y*-axis shows the classification performance.

sequences. From our results, it can be observed that there is great variation in CPP performance for different genomes. CPP seems to be the most difficult for Human, where the resulting FP-rates are much higher than in other organisms, regardless of the sequence length. On the other hand, CPP proves to be easier in *Plasmodium* and in particular in *Arabidopsis*, where the FP-rate quickly approaches zero with increasing sequence length. To see whether this difference in result can be explained by codon bias, we plotted the difference in frequency of occurrence of each nucleotide at one of the three possible codon positions (Fig. 2). These differences are calculated as the absolute values of the difference in frequency between coding and non-coding sequences of the length class [42–63nt]. It can be observed that the difference between coding and non-coding sequences is indeed more outspoken in *Arabidopsis* and *Plasmodium*, especially at the first codon position. On the other hand, *Cryptococcus* and especially Human show a lower difference between coding and non-coding sequences, correlating well with the differences in classification performance.

3.5 Feature analysis

3.5.1 Effect on classification performance The performance of a classifier depends on the interaction between a number of parameters, such as training set size, number of features and classifier complexity. When, for a fixed training set size, the number of features is increased, the number of unknown parameters for the classifier also increases. Correspondingly, the reliability of the parameter estimates decreases, and the performance of the classifier may degrade with an increase in the number of features (Raudys and Jain, 1991), motivating the search for a minimal subset of relevant features. It is known that Markov models need many features to build a model. A simple Markov model of order N requires 4(N + 1) features to be calculated for each model. For CPP, four models have to be built (three coding models for every possible reading frame and one non-coding model), thus requiring 4(N + 2) features to be calculated. More advanced Markov models require fewer features, yet still many thousands of parameters need to be determined. As a result, Markov models need a lot of training data to accurately determine parameter estimates. On the other hand, methods based on a transform (like SNR and *Z*-curve) require far less features to be calculated: the SNR methods uses only one feature, and the *Z*-curve method uses 189 features. In terms of simplicity, these methods provide more compact models, and are to be preferred over Markov models when only a limited amount of training data are available.

To analyze the effect of feature selection on the classification performance of the SVM-based models, the FP-rate for the different feature subsets was analyzed. Figure 3 shows these results in the case of *Arabidopsis* (results for the other species are available as Supplementary material). The *x*-axis denotes the size of the feature subset, starting at the origin with the full feature set, and ending in a very small subset of 10 features. The *y*-axis shows the classification performance (FP-rate at a TP-rate of 0.95) for each of the different length classes. From these results, it is clear that feature selection is beneficial to all genomes, regardless of the length class. In all four genomes investigated, about two-thirds of all features can be eliminated without decreasing classification performance. Two general trends can be observed. For *Arabidopsis* and *Cryptococcus*, feature selection seems to be indispensable in achieving good results, and the classification performance increases drastically when smaller feature subsets are used. On the other hand, feature selection does not significantly increase classification performance for Human and *Plasmodium*. However, many features can still be eliminated without affecting the classification performance, and thus smaller feature subsets can be used to achieve good performance.

In general, SVM-based methods can thus achieve better classification performance than state-of-the-art Markov models, while at the same time heavily reducing the number of required parameters to build the model. Resulting models will thus be more compact than existing Markov models, which will allow them to perform a better generalization when confronted with unseen examples. This

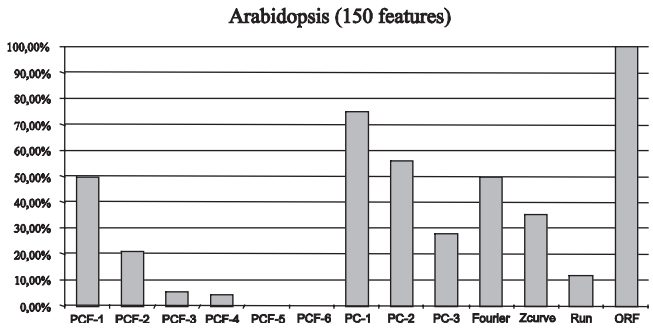

Fig. 4. Evaluation of the feature relevancies. For a good subset of features for each organism (see text for details), the representation of each type of feature (Y-axis) is shown. PCF-k (k=1..3) denote frame-dependent k-mers, PCF-k (k=4..6) denote in-frame k-mers, and PC-k features denote global (frameless) k-mers.

may especially be of importance when only a limited amount of training data is available.

3.5.2 Gaining insight into feature relevance An important advantage of feature selection methods is the ability to gain more insight into the problem under investigation by analyzing a 'minimal' subset of relevant features. For CPP, we combined complementary sequence features in one large feature set, hoping that this would enhance classification. However, doing this introduced a great deal of redundant or irrelevant features. Using the feature selection results, we can now focus on appropriate feature subsets for each organism, and identify the features that were selected as most relevant. Figure 4 summarizes the representation of each type of feature in the chosen subset in the case of *Arabidopsis* (figures for the other organisms can be found in the Supplementary material). The sizes of the considered subset of features were chosen based on the results in Table 3, and only represent an averaged view over all sequence lengths of a specific organism. For each type of feature (x-axis), the y-axis shows the representation (in terms of percentage) of this feature type in the chosen subset. For Human, the best 1500 features were chosen, and for *Arabidopsis*, *Cryptococcus* and *Plasmodium*, 150, 250 and 150 features were chosen, respectively. A representation of 100% means that all features of this type were included in the subset of most important features, thus indicating an indispensable set of features. Some general trends on feature relevance can be observed. First, it can be observed that the ORF feature is most important. This is not surprising, as it is included as a post-processing step in our method to filter out false positives, and thus will always be applied. Second, it is shown that the features based on a transform (Fourier, Z-curve and Run) are very important, as a large part of them remains relevant. Furthermore, it is clear that only a very small percentage of the higher-order in-frame k-mers are important. A surprising result may be the fact that frameless k-mers (PC-1, PC-2 and PC-3) appear to be very important, suggesting that overall compositional features of coding (exon) sequences greatly contribute to CPP. A more detailed feature list, sorted according to relevance, can be found as Supplementary material. Features that are ranked in the top 10 for all organisms studied here include ORF and in-frame stop codon frequencies, features related to AT-composition, nucleotide composition at the first codon position (especially nucleotides G and T) and the signal-to-noise ratio of the peak at frequency 1/3 in the Fourier spectrum.

Conflict of Interest: none declared.

REFERENCES

Bertone,P. *et al*. (2006) Global identification of human transcribed sequences with genome tiling arrays. *Science*, **306**, 2242–2246.

Borodovsky,M. and McIninch,J. (1993) Genmark: parallel gene recognition for both DNA strands. *Comput. Chem.*, **17**, 123–133.

Boser,B.E. *et al*. (1992) A training algorithm for optimal margin classifiers. In *Proceedings of Fifth ACM Workshop on COLT*, pp. 144–152.

Brent,M.R. and Guigó,R. (2004) Recent advances in gene structure prediction. *Curr. Opin. Struct. Biol.*, **14**, 264–272.

Fickett,J. and Tung,C. (1992) Assessment of protein coding measures. *Nucleic Acids Res.*, **20**, 6441–6450.

Gao,F. and Zhang,C.T. (2004) Comparison of various algorithms for recognizing short coding sequences of human genes. *Bioinformatics*, **20**, 673–681.

International Human Genome Sequencing Consortium (2004), Finishing the euchromatic sequence of the human genome. *Nature*, **431**, 931–945.

Joachims,T. (1998) Making large-scale support vector machine learning practical. In Schoelkopf,B. *et al*. (eds), *Advances in Kernel Methods: Support Vector Machines*, pp. 169–184.

Kohavi,R. and John,G. (1997) Wrappers for feature subset selection. *Artif. Intell.*, **97**, 273–324.

Koller,D. and Sahami,M. (1996) Toward optimal feature selection. In *Proceedings of the 13th International Conference on Machine Learning*, pp. 284–292.

Kotlar,D. and Lavner,Y. (2003) Gene prediction by spectral rotation measure: a new method for identifying protein-coding regions. *Genome Res.*, **13**, 1930–1937.

Majoros,W.H. *et al*. (2003) GlimmerM, Exonomy and Unveil: three ab initio eukaryotic genefinders. *Nucleic Acids Res.*, **31**, 3601–3604.

Majoros,W.H. *et al*. (2004) TigrScan and GlimmerHMM: two open source ab initio eukaryotic gene-finders. *Bioinformatics*, **20**, 2878–2879.

Mathé,C. *et al*. (2002) Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Res.*, **30**, 4103–4117.

Ng,A.N. and Jordan,M. (2002) On discriminative vs. generative classifiers: a comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems 14*. MIT Press, Cambridge, MA.

Provost,F.J. and Fawcett,T. (1997) Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pp. 43–48.

Raudys,S. and Jain,A. (1991) Small sample size effects in statistical pattern recognition: recommendataions for practitioners. *IEEE Trans. PAMI*, **13**, 252–264.

Saeys,Y. (2004) Feature selection for splice site prediction: a new method using EDA-based feature ranking. *BMC Bioinformatics*, **21**, 64.

Salzberg,S.L. *et al*. (1998) Microbial gene identification using interpolated Markov models. *Nucleic Acids Res.*, **26**, 544–548.

Salzberg,S. *et al*. (1999) Interpolated Markov models for eukaryotic gene finding. *Genomics*, **59**, 24–31.

Schiex,T. *et al*. (2001) EuGène: an Eukaryotic Gene Finder that combines several sources of evidence. In Gascuel, O. and Sagot, M.-F. (eds), *Proceedings of the Lect. Notes Comput. Sc. 2006*, pp. 111–125.

Silverman,B. and Linsker,R. (1986) A measure of DNA periodicity. *J. Theor. Biol.*, **118**, 295–300.

Stanke,M. *et al*. (2006) Gene prediction in eukaryotes with a generalized hidden Markov model that uses hints from external sources. *BMC Bioinformatics*, **7**, 62.

Stolc,V. *et al*. (2005) Identification of transcribed sequences in *Arabidopsis thaliana* by using high-resolution genome tiling arrays. *Proc. Natl Acad. Sci. USA*, **102**, 4453–4458.

Tiwari,S. *et al*. (1997) Prediction of probable genes by Fourier analysis of genomic sequences. *Comput. Appl. Biosci.*, **13**, 263–270.

Voss,R. (1992) Evolution of long-range fractal correlations and 1/f noise in DNA base sequences. *Phys. Rev. Lett.*, **68**, 3805–3808.

Wang,Y. *et al*. (2002) Recognizing shorter coding regions of human genes based on the statistics of stop codons. *Biopolymers*, **63**, 207–216.