Artificial Intelligence in Medicine (2009) **45**, 109—115

ARTIFICIAL INTELLIGENCE IN MEDICINE

http://www.intl.elsevierhealth.com/journals/aiim

Virtual genetic coding and time series analysis for alternative splicing prediction in *C. elegans*

Michele Ceccarelli*, **Antonio Maratea**

Research Centre on Software Technologies, University of Sannio, via Traiano 1, 82100 Benevento, Italy

Received 21 November 2007; received in revised form 22 August 2008; accepted 22 August 2008

KEYWORDS
Alternative splicing;
Virtual genetic code;
Autoregressive model;
Support vector machine

Summary

Motivation: Prediction of alternative splicing has been traditionally based on the study of expressed sequences, helped by homology considerations and the analysis of local discriminative features. More recently, machine learning algorithms have been developed that try avoid or reduce the use of a priori information, with partial success.
Objective and method: With the aim of developing a fully automatic procedure of recognition of alternative splicing events based only on the genomic sequence, we first introduce a virtual genetic coding scheme to numerically modeling the information content of sequences in an effective way, then we use time series analysis to extract a fixed-length set of features from each sequence and finally we adopt a supervised learning method, namely the support vector machine, to predict alternative splicing events.
Results: On the base of real *C. elegans* data, we show that it is possible within this purely numeric framework to obtain results better than the state of the art, without any explicit modeling of homology or positions in the splice site, nor any use of other local features.
Conclusion: The virtual genetic coding together with time series analysis allows us to introduce an effective and powerful sequence coding scheme, that may be useful in various areas of genomics and transcriptomics.

1. Introduction

Alternative splicing is one of the key mechanisms of post-transcriptional modification [1]. Through it, a single gene can give rise to a number of different products rearranging its coding regions and splicing out its non-coding regions in many alternative configurations. The coding regions of a gene are called exons and the non-coding regions are called introns. Alternative splicing (AS) happens when exons have many different possible configurations and hence one gene can produce more than one single product (protein).

* Corresponding author. Tel.: +39 0824 305548;
fax: +39 0824 23013.
E-mail addresses: ceccarelli@unisannio.it (M. Ceccarelli), amaratea@unisannio.it (A. Maratea).

An extreme known case of AS is the gene DSCAM of *Drosophila*, that it is known to produce more than 38,000 different proteins. It is estimated that half of the human genes are alternatively spliced and this percentage does not vary much among other animals [2].

Although AS is known since the 1980s there are only a few theories on why it happens and traditional methods to predict it are based essentially on the study of expressed sequences (see [3] for an up to date overview), strongly helped by homology considerations and careful analysis of the splice site. Among the many possible forms of AS, the one which we will analyze here is exon skipping, that is the case when different gene products are obtained skipping one of gene's exons. Skipped exons are called alternative exons while exons that are not skipped are called constitutive exons (Fig. 1).

Several methods to predict AS based on the local sequence features have been proposed in literature [4—6]. Although the authors recognize many discriminative features, the most effective among them is certainly sequence homology. In spite of this result, recent studies confirm from one side that constitutive exons are more conserved than alternative exons [7] and from another side that conserved exons are subject to species-specific AS in a significant amount [8]. The use of homology information as a discriminative feature has become hence more challenging, way beyond its scarce availability on many sites. For this an other similar reasons traditional methods have been integrated and partially replaced by machine learning methods trying to infer AS on the basis of information that is always available (i.e. the crude pre-mRNA sequence) [9].

In our previous work [10] we have shown that a machine learning predictor for exon skipping based only on the numerical properties of the pre-mRNA sequence in *C. elegans* has similar or better performances than methods that account for homology, position within the splice site or length of exons. The machine learning approach proposed is here extended and generalized improving the coding step with the inclusion of sequence information content modeling. A virtual genetic code has been developed that allows to improve classification performances and to extend method's application possibilities. After proper coding, a fixed-length set of features is extracted from the coded sequences and finally a support vector machine is trained and tested on the built features. Machine learning *ab initio* recognition of alternatively splicing exons confirms to be a viable and effective procedure, that may be considered in all cases where homology information is not available or difficult to be obtained as well as when the local information on the splicing site is vague or unreliable. In this work, *C. elegans* data will be considered.

The paper is organized as follows: in the first section, an introduction to the problem of alternative splicing prediction is given together with a crash overview of main literature approaches; in the second section, the method is outlined and detailed, from the introduction of the virtual genetic code for coding, to feature extraction and splicing prediction; in the third section data and results of method's application are presented, highlighting the effect of parameters' choice; in the last section conclusions are drawn and main extension perspectives are given.

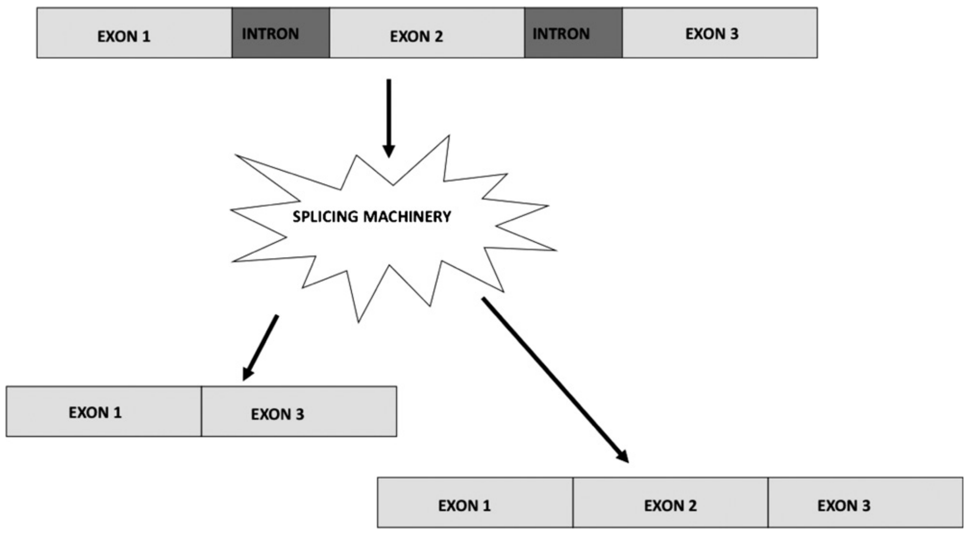


Figure 1 *Constitutive alternative splicing*: two proteins are made from the same pre-mRNA sequence, skipping one exon.

2. Proposed method

The whole process can be resumed in three steps:

- coding of each exonic or intronic sequence;
- feature extraction from coded sequences;
- training a classifier on the features' vector previously obtained.

as explained in the next sections.

2.1. Exon coding

With respect to the paper [10], coding has been improved in two ways. A first improvement is the adoption of a codon based coding instead of a single nucleotide based coding. This new coding scheme is actually equivalent to defining a "virtual genetic code", a numerical table such that each codon corresponds to a different code. A second improvement is the adoption of a sliding window of width three to replace each nucleotide with the numerical code (from the virtual genetic code table) corresponding to the codon centered on that nucleotide. Numerical codes filling the virtual genetic code matrix are computed on the base of the Shannon information content of nucleotide letters in data sequences.

If we indicate as N_i $i = \{1, \ldots, 4\}$ the four bases $\{ACGT\}$, following Shannon information theory the information content of each triplet can be written as in the following equation:

$$I(N_i, N_j, N_k) = -\log(p \times (N_i, N_j, N_k)), \quad i, j, k \in \{1, \ldots, 4\} \tag{1}$$

where log are base 2, I is the information content and $p()$ is the probability. The virtual genetic code is hence a table of size 16×4 built with the information content corresponding to each possible triplet of nucleotides. Probability is estimated by relative frequency of occurrence of each triplet in the data (computed with respect to both splicing and non-splicing groups) (Figs. 2 and 3).

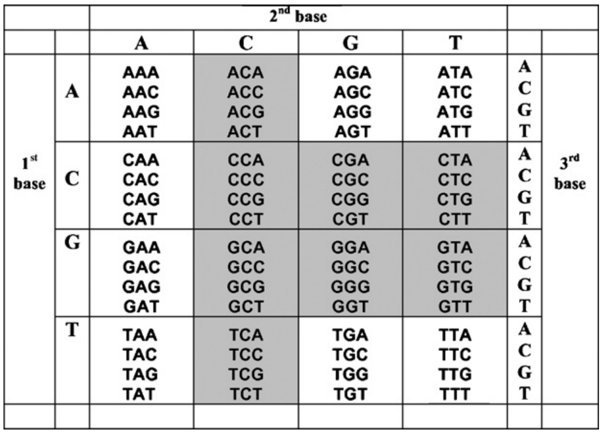
If we indicate with S a generic nucleotide sequence and with S_i its generic element of place i, the generic element of place i of the coded sequence C is:

$$C_i = I(S_{i-1}, S_i, S_{i+1}) \tag{2}$$

a profile plot of the central exon once it has been coded following this scheme for both the alternative and constitutive case can be seen in Figs. 5 and 4, respectively.

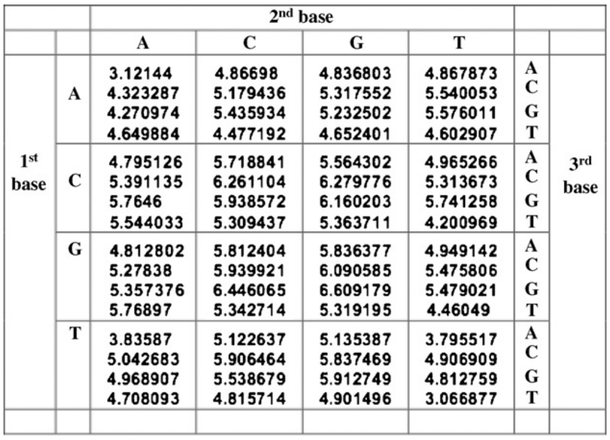
2.2. Feature extraction

Once data are coded (see Section 3), the first problem to cope with is the different length of the various sequences. As can be seen from Figs. 5 and 4, coded signals are strongly periodic and this fact candidates auto regressive (AR) models as natural descriptors of the dynamic of the phenomenon under study. Following this line, we estimate an AR model on the data and we choose its coefficients as features. In this approach, each observed coded exonic or intronic sequence C is assumed to be the output of an order p AR model driven by a white noise process $e(n)$, that is the value in position n is assumed to be equal to a weighted sum of the p previous values plus a white noise term. As an AR model has a fixed-length set of p coefficients, each sequence is replaced by a fixed-length vector of p



		2nd base					
		A	C	G	T		
1st base	A	AAA	ACA	AGA	ATA	A	3rd base
		AAC	ACC	AGC	ATC	C	
		AAG	ACG	AGG	ATG	G	
		AAT	ACT	AGT	ATT	T	
	C	CAA	CCA	CGA	CTA	A	
		CAC	CCC	CGC	CTC	C	
		CAG	CCG	CGG	CTG	G	
		CAT	CCT	CGT	CTT	T	
	G	GAA	GCA	GGA	GTA	A	
		GAC	GCC	GGC	GTC	C	
		GAG	GCG	GGG	GTG	G	
		GAT	GCT	GGT	GTT	T	
	T	TAA	TCA	TGA	TTA	A	
		TAC	TCC	TGC	TTC	C	
		TAG	TCG	TGG	TTG	G	
		TAT	TCT	TGT	TTT	T	

Figure 2 The common classification scheme of the genetic code. The four main rows indicate the first base in the codon, the four main columns indicate the second base and the rightmost column indicates the third base. The gray regions represent "family codons", where the encoded amino acid is independent of the third position. Since there are 20 different amino acids and $4^3 = 64$ possible codons the genetic code is redundant.



		2nd base					
		A	C	G	T		
1st base	A	3.12144	4.86698	4.836803	4.867873	A	3rd base
		4.323287	5.179436	5.317552	5.540053	C	
		4.270974	5.435934	5.232502	5.576011	G	
		4.649884	4.477192	4.652401	4.602907	T	
	C	4.795126	5.718841	5.564302	4.965266	A	
		5.391135	6.261104	6.279776	5.313673	C	
		5.7646	5.938572	6.160203	5.741258	G	
		5.544033	5.309437	5.363711	4.200969	T	
	G	4.812802	5.812404	5.836377	4.949142	A	
		5.27838	5.939921	6.090585	5.475806	C	
		5.357376	6.446065	6.609179	5.479021	G	
		5.76897	5.342714	5.319195	4.46049	T	
	T	3.83587	5.122637	5.135387	3.795517	A	
		5.042683	5.906464	5.837469	4.906909	C	
		4.968907	5.538679	5.912749	4.812759	G	
		4.708093	4.815714	4.901496	3.066877	T	

Figure 3 The virtual genetic code. The four main rows indicate the first base in the codon, the four main columns indicate the second base and the rightmost column indicates the third base. There are no "synonyms" coding.

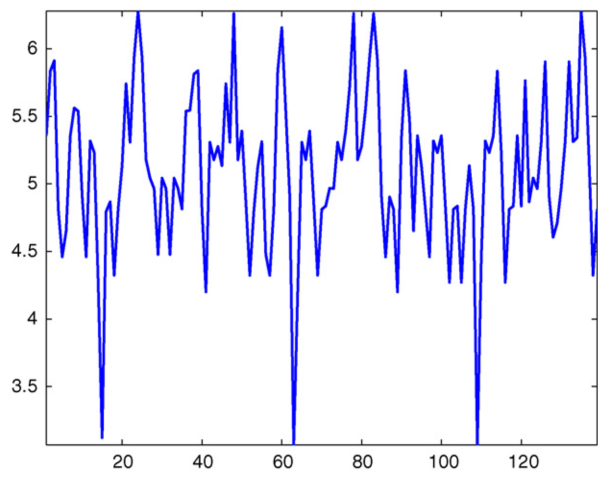


Figure 4 Profile plot of the sequence of a non-splicing exon coded with the virtual genetic code.

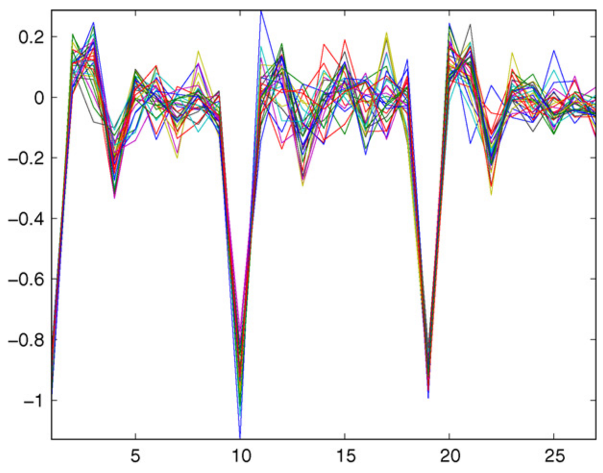


Figure 6 Profile plot of 30 randomly chosen coded sequences of splicing exon triplets. The 27 AR model coefficients are plotted versus their index.

coefficients (Figs. 6 and 7). The model is based on the following linear difference equations:

$$C(n)+\sum_{k}^{p} a_k C(n-k)=e(n) \tag{3}$$

where a_k is the kth AR parameter of an order p AR process. To estimate model parameters we used the classical Yule-Walker method, actually minimizing the forward prediction error in a least-squares sense.

At the end of process each intronic or exonic sequence S is replaced by the fixed-length vector $v=(a_1,\ldots,a_p)$ of the normalized estimate of the AR system coefficients a_k.

2.3. Classification

In the last step we consider the support vector machine (SVM) classification model. It is a classical technique for pattern recognition and data mining classification tasks [11] that has shown excellent performances in various heterogeneous fields [12]. The main advantage of SVM over i.e. neural networks, is that it has no local minima issues and that it has less free parameters.

Given a set of points in $\Re^k$ and a two-classes labels vector, SVM aims to find a linear surface that splits the data in two groups according to the indicated labels, maximizing the *margin*, that is the distance from *both* sets of points. This problem can be formulated as a constrained quadratic optimization problem:

$$\min\left(\frac{1}{2}||w||^2\right) \tag{4}$$

subject to:

$$y_i(w^T x_i+b)\geq 1 \tag{5}$$

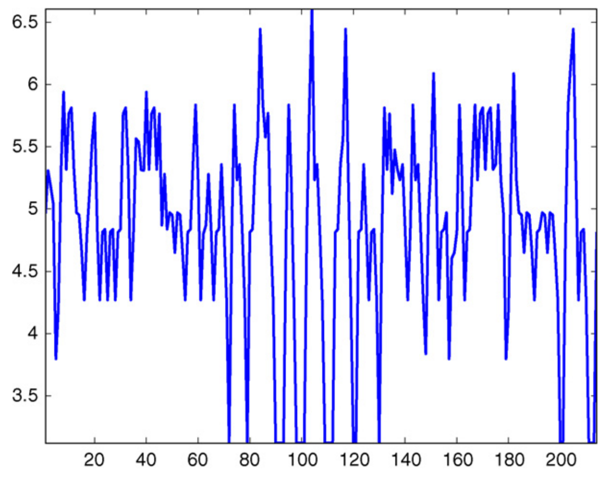


Figure 5 Profile plot of the sequence of a splicing exon coded with the virtual genetic code.

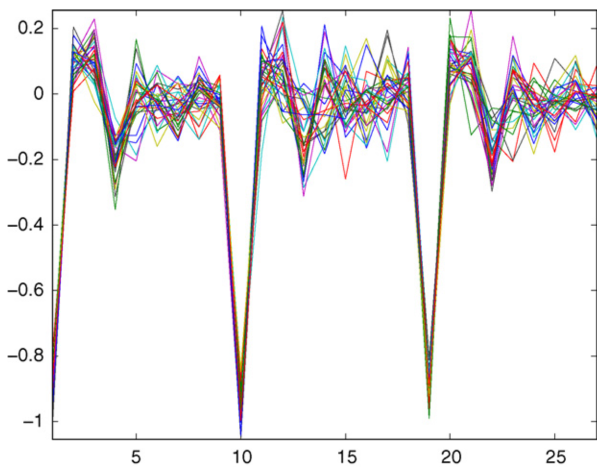


Figure 7 Profile plot of 30 randomly chosen coded sequences of non-splicing exon triplets. The 27 AR model coefficients are plotted versus their index.

where $y_i \in \{-1, 1\}$ are classes' labels, w is the normal to the hyperplane and $2/||w||$ is the margin.

If the data are not linearly separable in $\Re^k$, they can be projected nonlinearly in a Hilbert space where the classification can be performed linearly, maintaining the method almost unchanged. If we look at the optimization's problem solution, we see that data appear only in the form of dot products $x_i \cdot x_j$ and that even data transformed through a function $\Phi: \Re^k \mapsto \Gamma$ (where Γ is a space of dimension $h \geq k$) appear in the form of dot products $\Phi(x_i) \cdot \Phi(x_j)$. As a consequence, it is possible to substitute whatever dot product function $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$ in formulae and to compute the solution without even knowing the form of function Φ. Such a dot product function is called a *kernel* and there is an active field of research in the choice of the most suitable kernel for a given problem [13].

In this work we used a Gaussian kernel:

$$K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2) \gamma > 0 \quad (6)$$

Its choice was derived mainly form the following general practical considerations [14,15]:

- the radial basis function (RBF) SVM has infinite *capacity* and hence Gaussian RBF SVM of sufficiently small width can classify an arbitrarily large number of training points correctly;
- the RBF kernel includes as a special case the linear kernel;
- the RBF kernel behaves like the sigmoid kernel for certain parameters' values;
- the RBF kernel has less hyper-parameters than the polynomial kernel;
- the RBF kernel has less numerical difficulties than other kernels.

3. Experiments and results

The labeled dataset used to test our method is the same of [9]. It is a collection of 487 exons for which EST show evidence of alternative splicing and 2531 exons for which there is no evidence of alternative splicing, for a total of 3018 labeled examples. All data regard *C. elegans* and were obtained from the Wormbase, dbEST and UniGene [9].

The data matrix obtained after coding and after the AR model parameters' estimate is filled row by row by contiguous exon triplets. Rows labeled as "AS" have the central exon alternatively spliced, while rows labeled as "not AS" have the central exon constitutively spliced. The matrix has $3p$ columns, as for each exon there are p AR model coefficients. After the random division in testing and training sets (see below), we train the SVM classifier with the labeled matrix. Performances were evaluated in terms of the average area under curve (AUC) index of ROC curves. As the adopted dataset is biased towards non-splicing sequences, we generated a new dataset of alternative splicing exons resampling 5 times the original one of 487 alternative splicing sequences and adding a very small Gaussian perturbation ($\sigma = 2.5^{-4}$). Therefore the final dataset contains 4966 samples.

Once chosen the RBF kernel, there are two core parameters to tune: the width of the Gaussian function σ and the order of the AR model p.

Tuning has been performed on a single split of the full dataset in two parts, using k-fold cross-validation with $k = 5$. One training set of 4/5 of data has been randomly extracted and a series of parameters' values has been tested with k-fold cross-validation [17]. In k-fold cross-validation data are split in k subgroups and in turn $k - 1$ of them are used to predict the kth group values. Then average performance on all k groups is used to assign a predictive power to the method tested. The advantage of cross-validation is its robustness to outliers and overfitting.

For each couple of parameters' values we executed an SVM prediction and computed the average performance through cross-validation. The performances relative to each couple of parameters' values can be thought of filling a matrix P (Table 1) in which each dimension represents a parameter.

Table 1 AUC values corresponding to each tested value of parameters

$\sigma \backslash p$	2	3	4	5	6	7	8	9
0.2	0.6579	0.6324	0.5989	0.6837	0.7231	0.7405	0.6880	0.7428
0.6	0.7184	0.6051	0.6275	0.7202	0.7196	0.7834	0.8073	0.8485
1.0	0.7061	0.6677	0.6735	0.7412	0.7969	0.8395	0.8853	0.8906
1.4	0.5762	0.6823	0.6931	0.7593	0.8172	0.8556	0.9065	0.9266
1.8	0.7098	0.7404	0.7551	0.7831	0.8370	0.8855	0.9263	0.9431
2.2	0.7002	0.7207	0.8009	0.8337	0.8695	0.9039	0.9281	0.9561
2.6	0.7004	0.7476	0.8043	0.8319	0.8909	0.9263	0.9498	0.9482
3.0	0.7191	0.7462	0.8127	0.8544	0.9090	0.9200	0.9450	0.9622
3.4	0.7629	0.7356	0.8262	0.8751	0.9096	0.9388	0.9508	0.9438
3.8	0.7339	0.7589	0.8258	0.8832	0.9156	0.9412	0.9444	0.9625

In order to choose the model order, we tested all values from 2 to 9. We could not use higher values due to the presence of very short sequences in data an even to reach order 9 we have been forced to remove the 20 shortest sequences from the data set. A regular increase of average performance with the model order is clearly visible (Table 1) and we choose the maximum allowed ($p = 9$).

The other parameter was the width of the Gaussian function in the kernel. We tested 10 values, ranging from 0.2 to 3.8 in 0.4 steps and the final choice was the local maximum for $p = 9$, that is $\sigma = 3$ (see last column of Table 1).

The method has proven to be more sensible to the right choice of parameters with respect to [10]. Specifically, with low model order and narrow Gaussian kernel the performance is not so striking. On the other side, there is an evident improvement of performance increasing the value of both parameters (Table 1).

With a proper choice of parameters, the method reaches an average AUC of over 96.2% on testing sets. This result notably improves state of the art, as the best AUC previously published on these data was about 89.7% [9]. The corresponding ROC curve has a marked shift from the *y* axis due to method's failure to classify a few high scoring points from the SVM. As a consequence, comparison with state of the art in terms of true positive rate corresponding to very low false positive rate (from 1% to 5%) results in a poorer performance, while starting from 6% of false positive rate the proposed methods markedly outperforms previous ones. Ranking sequences on the base of the SVM predicted value, we noted that the top misclassified sequences at each run tend to be conserved. This fact suggests the opportunity that further biological verification is performed on these sequences, because their labeling, naturally prone to errors, could likely be effect of noise.

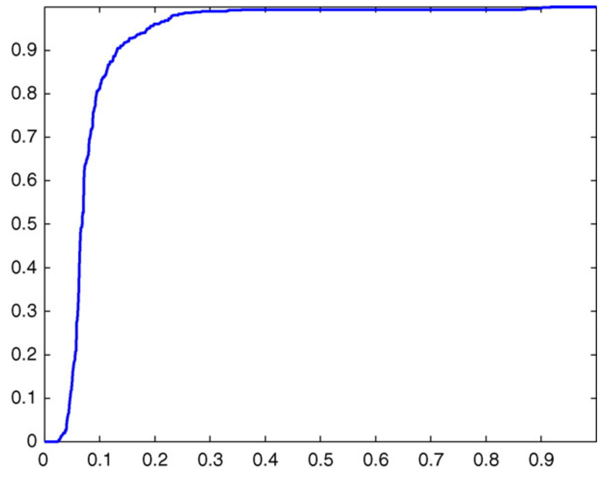


Figure 9 ROC curve for the BP classifier with 45 nodes in the hidden layer.

Just for comparison, a learning vector quantization (LVQ) and a feed forward backpropagation artificial neural network (BP) classifications have been peformed on the same features. For the former, we used a two layer network with 45 hidden nodes and a 50% typical class percentage for the two classes, while for the latter we used a two layer network, with 45 neurons in the hidden layer, *tansig* neurons on the first layer and *purelin* neurons in the output layer. How can be seen from receiver operating characteristic (ROC) curves [16] in Figs. 8—10, SVM markedly outperforms LVQ and BP in this specific application.

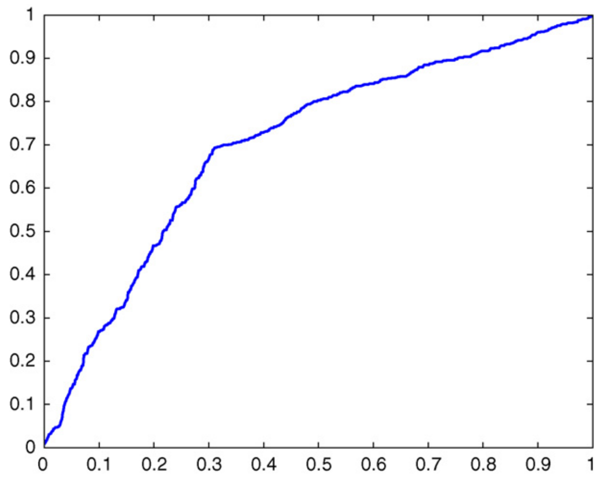


Figure 8 ROC curve for the LVQ classifier with 45 nodes in the hidden layer.

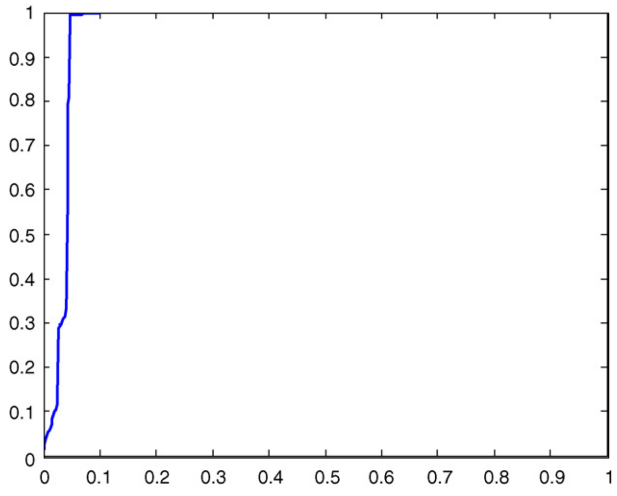


Figure 10 ROC curve built on 5-fold cross-validation data for the SVM classifier.

4. Conclusions

We have shown that using the virtual genetic code in the coding step of the *ab initio* alternative splicing prediction procedure allows to clearly improve classification performances and to extend method's application possibilities. Machine learning recognition of alternatively splicing exons confirms to be a viable and effective procedure, that may be considered in all cases where homology information is not available or difficult to obtain as well as when the local information on the splicing site are vague or unreliable. The proposed procedure reaches an AUC of over 96% on tested *C. elegans* data and does not account for homology, position within the splice site or length of exons. A few top ranked sequences from the SVM are misclassified at each run and we believe these should be biologically verified, as suggested by the overall prediction accuracy. The virtual genetic code based on Shannon information content has proven to be an effective coding scheme and should be considered as an attractive option whenever a numerical translation of a biological sequence is needed, also in other areas of genomics and transcriptomics. Finally, although more data and more work are needed to validate further this result, the need for a deeper understanding of the splicing machinery emerges, in order to use biological knowledge in a more effective way in transparent models. Future work is in studying other organisms and in trying to predict more complex form of AS.

References

[1] Watson J, Baker T, Bell S, Gann A, Levine M, Losick R. Molecular biology of the gene (international edition), 5th edition, Amsterdam: Addison-Wesley; 2004.

[2] Brett D, Pospisil H, Valrcel J, Reich J, Bork P. Alternative splicing and genome complexity. Nature Genetics 2001;30: 29—30.

[3] Bonizzoni P, Rizzi R, Pesole G. Computational methods for alternative splicing prediction. Briefings in Functional Genomics and Proteomics 2006;5(1):46—51.

[4] Dror G, Sorek R, Shamir R. Accurate identification of alternatively spliced exons using support vector machine. Bioinformatics 2004;21:897—901.

[5] Hiller M, Backofen R, Heymann S, Busch A, Glaeber TM, Freytag JC. Efficient prediction of alternative splice forms using protein domain homology. In Silico Biology 2004;4: 195—208.

[6] Sorek R, Ast G. Intronic sequences flanking alternatively spliced exons are conserved between human and mouse. Genome Research 2003;13:1631—7.

[7] Malko DBB, Makeev VJJ, Mironov AAA, Gelfand MSS. Evolution of exon—intron structure and alternative splicing in fruit flies and malarial mosquito genomes. Genome Research 2006;6:505—9.

[8] Pan Q, Bakowski MA, Morris Q, Zhang W, Frey BJ, Hughes TR, et al. Alternative splicing of conserved exons is frequently species-specific in human and mouse. Trends in Genetics 2005;21(2):73—8.

[9] Raetsch G, Sonnenburg S, Schoelkopf B. RASE: recognition of alternatively spliced exons in *C. elegans*. Bioinformatics 2005;21(Suppl. 1):i369—77.

[10] Ceccarelli M, Maratea A. An alternative splicing predictor in *C. elegans* based on time series analysis. In: Masulli F, Mitra S, Pasi G, editors. Applications of fuzzy sets theory. Proceedings of the 7th international workshop on fuzzy logic and applications (WILF2007), vol. 4578 of lecture notes in computer science. Berlin: Springer; 2007 . p. 588—95.

[11] Vapnik V. The nature of statistical learning theory. New York: Springer-Verlag; 1995.

[12] Scholkopf B, Sung K, Burges C, Girosi F, Niyogi P, Poggio T, et al. Comparing support vector machines with Gaussian kernels to Radial Basis Function Classifiers. IEEE Transactions on Signal Processing 1997;45(11):2758—65.

[13] Cristianini N, Taylor JS. Kernel methods for pattern analysis. Cambridge: Cambridge University Press; 2004.

[14] Keerthi SS, Lin CJ. Asymptotic behaviors of support vector machines with Gaussian Kernel. Neural Computation 2003;15(7): 1667—89.

[15] Verri A, Pontil M. Properties of support vector machines. Neural Computation 1998;10(4):955—74.

[16] Egan JP. Signal detection theory and ROC analysis. New York: Academic Press; 1975.

[17] Stone M. Cross-validation: a review. Mathematics Operations and Statistics 1978;9:127—40.