StackTIS: A Stacked Generalization Approach for Effective Prediction of Translation Initiation Sites

George Tzanis[1], Christos Berberidis[2], Ioannis Vlahavas[1]
[1]Department of Informatics, Aristotle University of Thessaloniki, Greece
[2]School of Science and Technology, International Hellenic University, Greece

Abstract

The prediction of the translation initiation site in an mRNA or cDNA sequence is an essential step in gene prediction and an open research problem in bioinformatics. Although recent approaches perform well, more effective and reliable methodologies are solicited. We developed an adaptable data mining method, called StackTIS, which is modular and consists of three prediction components that are combined into a meta-classification system, using stacked generalization, in a highly effective framework. We performed extensive experiments on sequences of two diverse eukaryotic organisms (Homo sapiens and Oryza sativa), indicating that StackTIS achieves statistically significant improvement in performance.

Keywords: Translation Initiation, Bioinformatics, Data Mining, Machine Learning, Classification, Stacking

1. Introduction

As many genome sequencing projects have been completing, the need to annotate those genomes is being emerged. The observed paradigm shift from static structural genomics to dynamic functional genomics and the assignment of functional information to known sequences is deemed particularly important. Gene prediction, being the next step after sequencing, is particularly important for the annotation and understanding of a genome. One of the most important tasks of gene prediction is the recognition of Translation Initiation Sites (TISs). The problem of TIS prediction has been extensively studied since the 80s, however it is still an open research problem. The scientific community is very interested in prediction of TISs with the highest possible accuracy, not only in the framework of gene prediction. For example, TIS prediction can be used as a filtering step in finding microRNAs, since the prediction of the presence of a TIS in an mRNA sequence denotes the absence of a microRNA in this sequence. The fast and accurate recognition of TISs is not a trivial task and requires the utilization of advanced computational methods like the data mining approach we propose in this paper that includes various pre-processing steps and the construction of classification models. Although

several methods have been proposed to deal with this problem, there is the need and there are the possibilities for the improvement of the performance of these methods.

In this paper, we propose the use of a data mining approach that consists of three main prediction components, the coding component, which recognizes the coding potential of a cDNA sequence, the consensus component, which is used for scoring a candidate TIS according to its near context, and the upstream length component, which focuses on the distance of a candidate TIS from the 5′ end of the sequence, exploiting the advantages of the Ribosome Scanning Model. The output of the proposed method is not just a class label (0/1) for each candidate TIS, but it is a score that indicates the probability of the candidate being a TIS. This provides with a confidence for every decision of the method and makes possible the ranking of the candidate TISs. This is important if one wants to consider alternatives. Our approach has been trained and evaluated on two human and a rice dataset containing sequences with reduced similarity. The experimental results indicate that our method achieves improved prediction accuracy against other popular methods presented in the literature.

The proposed approach, called StackTIS, is an extended and improved version of MANTIS [1], which is a robust methodology that has been tested over a number of datasets of various organisms and has achieved increased performance over a number of other popular approaches. The main differences between the two approaches are presented below. Firstly, the feature set used for building the coding component's classifiers of MANTIS consists of a large number of features (>150) and thus feature reduction and transformation methods like Principal Components Analysis (PCA) have been utilized. In contrast, StackTIS uses a more robust feature set consisting of 64 features that represent codon frequencies avoiding the use of PCA. There are also differences in the way that the training of coding component is applied. The most important is that in StackTIS the negative examples of the non-coding frames are generated from regions that do not include any stop codons. This is done in order to force classifiers to learn recognizing the non-coding regions even in the absence of stop codons. Moreover, StackTIS selects and uses only the best classifier according to the used metric (e.g. accuracy or adjusted accuracy) for constructing each of the three components. On the contrary, this selection step was not included in MANTIS resulting sometimes a decreased performance due to the incorporation of the weakest classifiers. Furthermore, there are many differences in the procedure of the recognition of the TIS in a sequence. In MANTIS, all ATGs were considered as candidate TISs. In StackTIS, however, many ATGs are filtered out and only the ones that are included in the longest ORFs of each reading frame are considered as candidates. Also, the coding windows constructed in MANTIS were centered at each ATG, whereas in StackTIS one such window has been replaced by two separate windows one for the upstream and one for the

downstream region. In addition, there are differences in the experimental setup and validation of the two approaches. In the present study, StackTIS has been evaluated using more reliable datasets that have been undergone similarity reduction for minimizing biasness due to similarity of sequences. Furthermore, important parameters have been tested extensively (e.g. 6 window lengths have been tested in StackTIS, instead of only one in MANTIS). Finally, the statistical comparisons of the two approaches indicate a statistically significant superiority of StackTIS against MANTIS in terms of accuracy and adjusted accuracy.

This paper is organized as follows: In the next section, a brief introduction on the biological problem that is approached in our study is presented. Section 3 provides a concise review of the literature on TIS prediction. In Section 4 the proposed data mining approach, called StackTIS, is described in detail. Section 5 presents the results of the extensive experiments that we have conducted. In the last section the paper is summarized with conclusions and directions for future research.

2. Background

Translation is the second process of protein synthesis. In particular, after a DNA molecule has been transcribed into a messenger RNA (mRNA) molecule, an organelle called *ribosome* scans the mRNA sequence. The ribosome reads triplets, or codons, of nucleotides and "translates" them into amino acids. An mRNA sequence can be read in three different ways in a given direction. Each of these ways of reading is referred to as *reading frame*.

Translation, usually, initiates at the AUG codon nearest to the $5'$ end of the mRNA sequence. However, this is not always the case, since there are some escape mechanisms that allow the initiation of translation at following, but still near the $5'$ end AUG codons. Due to these mechanisms the recognition of the TIS on a given sequence becomes more difficult.

After the initiation of translation, the ribosome moves along the mRNA molecule, towards the $3'$ end (the direction of translation is $5' \rightarrow 3'$) and reads the next codon. This process is repeated until the ribosome reaches a stop codon. For each codon read the proper amino acid is brought to the protein synthesis site by a transfer RNA (tRNA) molecule. The amino acid is joined to the protein chain, which by this way is elongated.

A codon that is contained in the same reading frame with respect to another codon is referred to as *in-frame codon*. We call *upstream* the region of a nucleotide sequence from a reference point towards the $5'$ end. Respectively, the region of a nucleotide sequence from a reference point towards the $3'$ end is referred to as *downstream*. In TIS prediction problem the reference point is an AUG codon. The above are illustrated in Figure 1. The ribosome scans the mRNA sequence from the 5′ end towards the 3′ end until it reads an AUG codon. If the AUG

codon has appropriate context, the translation initiates at that site and terminates when a stop codon (i.e. UGA) is read. An in-frame codon is represented in the figure by three consecutive nucleotides that are grouped together.

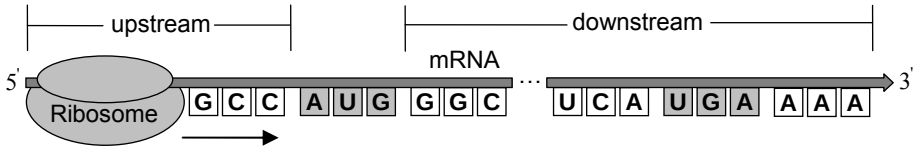

Figure 1. Translation initiation.

3. Related Work

Since 1982 the prediction of TISs has been extensively studied using biological approaches, data mining techniques and statistical models. Stormo et al. [2] used the perceptron algorithm to distinguish the TISs. Marylin Kozak developed the first weight matrix for the identification of TISs in cDNA sequences [3]. The consensus pattern GCC[**AG**]CCatg**G** was derived from this matrix. The bold residues indicate the highly conserved positions. Kozak and Shatkin [4] proposed the scanning model of translation initiation, which was later updated by Kozak [5]. According to this model, translation initiates at the first start codon that is in a particular context.

Pedersen and Nielsen [6] used artificial neural networks to predict which AUG codons are TISs achieving an overall accuracy of 88% in an Arabidopsis thaliana dataset and 85% in a vertebrate dataset. Zien *et al*. [7] studied the same vertebrate dataset, but instead of neural networks employed support vector machines using various kernel functions and got better results. Hatzigeorgiou [8] proposed an ANN system named "DIANA-TIS" consisting of two modules: the consensus ANN, sensitive to the conserved motif and the coding ANN, sensitive to the coding or non-coding context around the start codon. The proposed algorithm utilizes the simplified method of the ribosome scanning model starting a linear search at the beginning of the coding ORF and stopping when the combination of the two modules predicts a true TIS. The method applied in human cDNA data and 94% of the TISs were correctly predicted. Salamov *et al*. [9] developed the program ATGpr, using a linear discriminant approach for the recognition of TISs by estimating the probability of each ATG codon being the TIS. Nishikawa *et al*. [10] presented an improved program, ATGpr_sim, which employs a new prediction algorithm based on both statistical and similarity information. This new algorithm exploits the similarity to

known protein sequences and achieves better performance in terms of sensitivity and specificity. Zhang *et al*. [11] used CAEP, a classification algorithm based on emerging pattern aggregation for predicting TISs.

In 2002 Zeng *et al*. [12] used feature generation and correlation based feature selection along with machine learning algorithms. The most important of the raw and generated features found are positions -3 and -1 in the sequence (the A of the ATG codon is position 1 and the previous one is -1), upstream k-grams for k=3, 4 and 5, the frequency of stop codon, downstream in-frame 3-gram and the distance of ATG codon from the beginning of the sequence. Using a scanning model along with the same features an overall accuracy of 94% was attained on the vertebrate dataset of Pedersen and Nielsen. In [13] the above three-step approach (feature generation, feature selection, feature integration with machine learning method to build model for TIS prediction) is also presented. Basic k-grams, in-frame k-grams and position-specific features are the candidate features. Various methods for feature selection, such as signal-to-noise, t-statistics, entropy measure, information gain measure, information gain ratio and correlation based feature selection, are discussed. For feature integration C4.5, support vector machines, naïve Bayes and PCL are described. Later, in [14] the same three-step method used, but k-gram amino acid patterns were used, instead of k-gram nucleotide patterns. In the second step a number of the top ranked features were selected by an entropy based algorithm and finally, in the third step, a classification model is built for recognition of TIS applying support vector machines or ensembles of decision trees. In [15] Gaussian Mixture Models were used for the prediction of TISs improving classification accuracy. Finally, Nadershahi *et al*. [16] compare five methods -firstATG, ESTScan [17], Diogenes [18], Netstart [6] and ATGPr [9] - for the prediction of TIS. For the comparison a dataset of 100 Expressed Sequence Tag (EST) sequences, 50 with and 50 without TIS, was created. ATGPr appeared to outperform the other methods over this dataset.

4. Our Approach

StackTIS consists of three major prediction components as shown in Figure 2. These components contribute to the final prediction by considering different aspects of a candidate TIS. The first one is a classifier that captures differences in the coding potential around an ATG, the second is a Markov-chain based consensus pattern discovery model and the third is a model based on the location of the ATG inside the sequence.

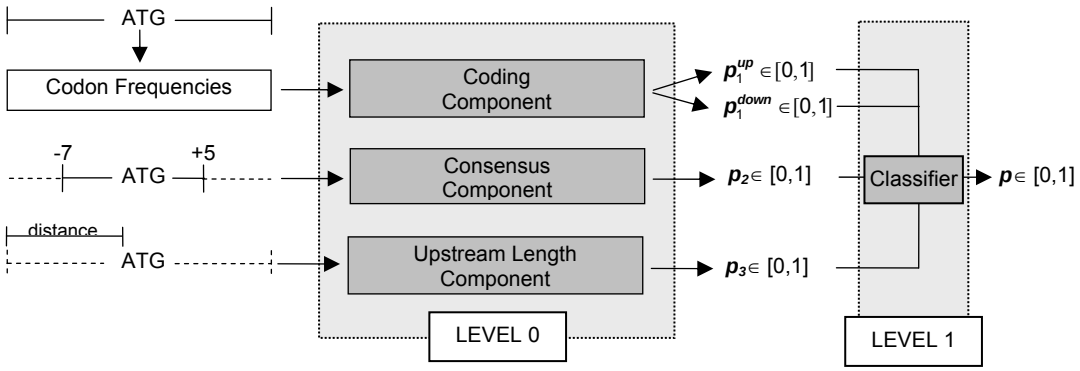

Figure 2. System architecture of StackTIS.

4.1. The Coding Component

The coding component is a classifier that identifies the coding potential of a cDNA sequence. In particular, the basic mission of this component is to recognize if an adequately long part of the region starting at an ATG codon is coding in contrast to the upstream region of this codon. In this case, the ATG codon is very possible to be the TIS.

The classifier is trained using a number of positive and negative examples that are generated from each training cDNA sequence, using a fixed-length sliding window. Every training example is represented by a vector of 64 components. Each component corresponds to the frequency of a specific codon, which is calculated inside the window. When the window slides a new vector is calculated, so that a new example is generated as shown in Figure 3. The positive examples are generated by windows of length *N* starting at the first nucleotide of the coding sequence (CDS). In our setup we experimented with various values of window length (see section 5.3.1). The negative examples are generated by windows of same length from 5′ and 3′ UTRs as well as from the two non-coding reading frames of the coding region. It is important to mention that the negative examples of the non-coding frames are generated from regions that do not include any stop codons. This constraint was set in order to make the coding classifier more sensitive to changes in codon frequencies rather than to the presence or absence of stop codons.

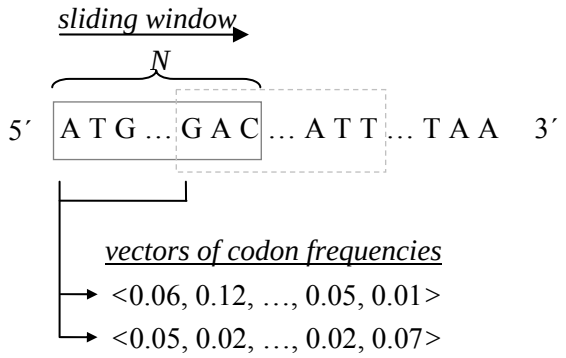

Figure 3. Extraction of coding components' s training examples.

From each cDNA sequence an equal number of positive and negative examples are extracted, so that the training set is balanced. The number of examples per sequence is determined by two parameters. The first one is the minimum window sliding length (minOffset, set to 30 in our setup). This parameter is used in order to avoid creating very similar training windows. The second parameter is the maximum number of examples per sequence per class (maxInstancesPerClass, set to 3) and is used in order to avoid generating too many examples from a single sequence. The number of instances that is generated from each class is calculated based on the above two parameters, the lengths of the CDS and UTRs, as well as on the constraint that the proportion of negatives to positives should be 1:1. It is worth mentioning that the proportion of negatives from 5′ UTR to negatives from 3′ UTR is as close as possible to 1:1.

4.2. The Consensus Component

Previous studies [8, 19] have shown that for the recognition of the TIS, it is important to examine a narrow area around it. The consensus component is a classifier constructed in order to score an ATG according to its near context. The higher the score, the more possible that the ATG is a TIS. For training the consensus classifier, one positive example (TIS) and one random negative example (an ATG that is not a TIS) were extracted from each cDNA sequence. This component uses Markov chains to capture the consensus pattern starting from position -7 and ending at position +5, as shown in Figure 4. The use of a Markov-chain based technique allows capturing not only the probability of the occurrence of a nucleotide at a certain position (as consensus pattern mining algorithms usually do), but also how the occurrence of one affects the occurrence of another. A Markov chain is a series of states of a system that has the Markov property:

$$P(s_i \mid s_{i-1}s_{i-2}...s_0) = P(s_i \mid s_{i-1}) \qquad (1)$$

Equation 1 describes 1st order Markov chains, where every state depends on the previous state only. There are, however, Markov chains of higher order. The order of a Markov chain corresponds to the number of states from which probabilities are defined to a successor. A Markov chain of order *k* is described by the following equation:

$$P(s_i \mid s_{i-1}s_{i-2}...s_0) = P(s_i \mid s_{i-1}s_{i-2}...s_{i-k}) \qquad (2)$$

position: -7 -6 -5 -4 -3 -2 -1 +1 +2 +3 +4 +5

5′ T G A A T A G **A T G** G C 3′

Figure 4. The positions of nucleotides relative to an ATG codon.

A homogeneous Markov chain (equation 3) is one that the transition probabilities do not change over time (the probability of going from state s_i to s_{i+1} is always the same), while in non-homogeneous chains (equation 4) these transition probabilities can be different. Using a non-homogeneous Markov chain we can detect the distributions of transitions separately for each position in a sequence.

$$P(s_{i+1} = x \mid s_i = y) = P(s_i = x \mid s_{i-1} = y) \qquad (3)$$

$$P(s_{i+1} = x \mid s_i = y) \neq P(s_i = x \mid s_{i-1} = y) \qquad (4)$$

The sequence segment around the ATG is modelled by a Markov chain model as follows: each state corresponds to a nucleotide at a certain position, from -7 to +5. The a posteriori probability of a sequence with respect to a given Markov chain of order *k* is given by equation 5. Equation 6 calculates the probability of a model $\hat{P}$ as the ratio of the frequencies in the training set of all partial sequences.

$$P(s_0s_1...s_L \mid M) = \prod_{i=k+1}^{L} P(s_i \mid s_{i-1}s_{i-2}...s_{i-k}) \quad (5)$$

$$\hat{P}(s_i \mid s_{i-1}s_{i-2}...s_{i-k}) = \frac{\#s_is_{i-1}s_{i-2}...s_{i-k}}{\#s_{i-1}s_{i-2}...s_{i-k}} \qquad (6)$$

One Markov chain is trained using examples of the positive class only and another Markov chain using examples of the negative class only. When a new instance arrives for classification, each of the two Markov chains produces a score for this instance. These scores are scaled so that their sum equals to 1, as follows:

$$S_+ = \frac{S_+}{S_+ + S_-} \text{ and } S_- = \frac{S_-}{S_+ + S_-} \qquad (7)$$

4.3. The Upstream Length Component

This component is based on the location of the ATG inside the sequence and the Ribosome Scanning Model (RSM), as proposed by Kozak [20]. According to this model, the ribosome scans the sequence until it finds the first ATG that is in an optimal nucleotide context. In previous studies, an ATG was chosen by the RSM as a TIS, when it was the one among those predicted by a classifier as a positive example and was closest to the 5′ end. In that case, all other ATGs were assigned to the negative class, even those that had been given a higher probability by the classifier. However, this rule has several exceptions that are explained by Kozak [20] and Pain [21].

In some cases, there are two ATG codons close to each other both having high coding and consensus predictions (Figure 5). The first ATG is selected by the RSM, even if the second one has higher predictions. Instead, without the utilization of the RSM, the second ATG codon would be selected as being the TIS, due to higher coding and consensus predictions. However, there is a dilemma on how to make the final decision. It is important to find out when one should take under consideration the decision of RSM and when not. The main parameters of this problem are the probability of an ATG being the TIS according to its distance from the 5′ end and the predictions of the coding and consensus components. All these parameters have to be incorporated into a model that learns how to combine them in order to provide a final prediction. In StackTIS this is achieved by fusing the predictions of all components using stacked generalization (see next section).

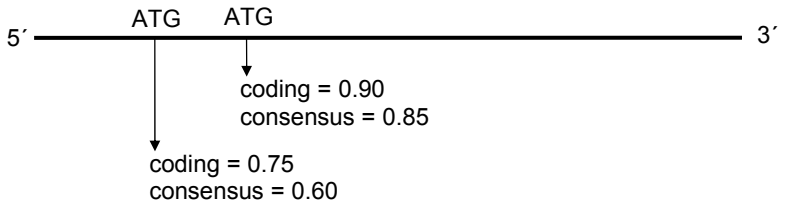

Figure 5. Two close to each other ATG codons with high coding and consensus predictions.

The upstream length component calculates the probabilities of an ATG to be the TIS according to its distance from the 5′ end. Two separate models are built, one trained on the positive examples and one on the negative examples (all ATG codons that are not TISs). The final output of the upstream length component is the scaled score that is calculated by the two models as in equation 7.

4.4. Fusion of Classifiers Using Stacked Generalization

The final stage in StackTIS is the fusion of the three components described above. The idea of meta-classifier systems is an attempt to construct more accurate classification models by combining a number of classifiers. Classifier combination includes two main paradigms: classifier selection and classifier fusion. In the first case a new instance is classified by selecting the appropriate classifier, while in the second case a new instance is classified according to the predictions of all the individual classifiers. In StackTIS a popular classifier fusion technique called Stacked Generalization or Stacking is utilized. Stacking [22] is a scheme for minimizing the generalization error rate of one or more models. According to that technique, a number of models (called level-0 models) that are trained on the original data (called level-0 data), produce the input (level-1 data) to a higher level classifier (level-1 classifier). These two levels of data and models are shown in Figure 2.

An important issue that has to be considered when using stacking is the kind of features that will be used as input to the level-1 classifier. Ting and Witten [23] proved that using probability distributions over classes instead of class labels (e.g. 0/1) improves stacking performance. Furthermore, in a more recent study [24] has been shown that using only the probabilities of class under consideration improves stacking significantly. For this reason in StackTIS the outputs of the components that are used as input to the level-1 classifier are probability estimates of the class under consideration instead of class labels.

4.5. Prediction of the TIS in a Sequence

The procedure followed in order to recognize the TIS in a given cDNA sequence consists of the steps presented below:

1) The longest regions for every reading frame that do not contain any stop codons are found.
2) All the ATG codons inside the three regions found in 1 are only considered as candidate TISs.
3) For each ATG found in 2 the following are applied:
   - Firstly, the coding component is used in order to calculate the coding prediction p_1^{up} for the sequence upstream the ATG and the coding prediction p_1^{down} for the sequence downstream the ATG. The two predictions are calculated on the specified window length (e.g. 120). If the considered ATG is the TIS, then p_1^{up} should be close to 0 and p_1^{down} close to 1.
   - Secondly, a consensus prediction p_2 is calculated for the ATG using the consensus component.

- Thirdly, the upstream length component is used in order to calculate the prediction of the ATG to be the TIS according to its distance from the 5′ end (p_3).
- Finally, the predictions calculated in the previous steps of 3 are used as input to the level-1 classifier that will provide the final prediction for the ATG.

4) Return a ranking of the ATGs according to their final prediction and consider the first in ranking (highest prediction) to be the TIS.

5. Experiments

This section describes our experimental setup and results. First, the datasets and the evaluation method that were used are described and then the experimental results are presented and discussed.

5.1. Datasets

In our study we have used three datasets. One of them (Human 1 dataset) was used in previous studies [8, 14, 15] whereas the other two (Human 2 and Rice) are new ones.

5.1.1. The First Human Dataset (Human 1)

This dataset [8] consists of 480 human sequences. It was extracted from Swissprot protein database. All the human proteins whose N-terminal sites are sequenced at the amino acid level were collected and manually checked. Then, the full-length mRNAs for the proteins with TISs that had been indirectly experimentally verified were retrieved and the corresponding human cDNAs, completely sequenced and annotated, were found.

5.1.2. The Second Human Dataset (Human 2)

For the construction of this dataset all known human proteins were first extracted from Swiss-Prot (18962 entries). Next, all the RefSeq human complementary DNA (cDNA) sequences were extracted from GenBank (52491 entries). The protein sequences underwent pair-wise global alignment against every other sequence in order to reduce similarity among all sequences. Particularly, there are not any two sequences in the resulting protein dataset that have more than 30% identity. This was done for minimizing biasness due to similarity of sequences. Then, the extracted cDNAs were translated and matched with the filtered set of proteins and a total number of 8113 human cDNAs remained. The sequences at the end were filtered in order to satisfy the UTRs and CDS length constraints we had set for proper training and evaluation. Finally, a number of 2351 human cDNAs remained.

5.1.3. The Rice Dataset

For the construction of the rice dataset all known proteins of organism Oryza stiva were first extracted from Swiss-Prot (2262 entries). Next, all the RefSeq rice complementary DNA (cDNA) sequences were extracted from GenBank (23309 entries). These protein sequences were also underwent pair-wise global alignment against every other sequence in order to reduce similarity among all sequences. There are not any two sequences in the resulting protein dataset that have more than 30% identity. Then, the extracted cDNAs were translated and matched with the filtered set of proteins and a total number of 824 rice cDNAs remained. The sequences at the end were filtered in order to satisfy our UTRs and CDS length constraints for proper training and evaluation and a number of 652 rice cDNAs remained.

5.2. Evaluation Method

For the evaluation of the different classifiers we have used stratified 10-fold cross-validation (CV). In particular, the performance of a classifier on a given dataset using 10-fold CV is evaluated as follows. The dataset is divided into 10 non-overlapping almost equal size parts (folds). In stratified CV each class is represented in each fold at the same percentage as in the entire dataset. After the dataset has been divided, a model is built using 9 of the folds as a training set and the remaining fold as a test set. This procedure is repeated 10 times with a different test set each time. The use of the 10-fold CV was based on the widely accepted study of R. Kohavi [25]. The results of this work indicate that for many real-word datasets, the best method to use for model selection is stratified 10-fold CV, even if computation power allows using more folds.

5.3. Results

For the conduction of our experiments we have utilized the Weka library of machine learning algorithms [26] and WLSVM [27], which supports several SVM methods that run much faster than Weka's SVM algorithm (SMO). WLSVM can be viewed as an implementation of the LibSVM running under Weka environment.

5.3.1. Coding Component Evaluation

For the coding component's classifier construction, we have experimented with the following four algorithms, representative of four different categories of classification algorithms:

- Naïve Bayes. A simple probabilistic classifier, based on Bayes's theorem.
- C4.5. A decision tree construction algorithm.

- k-Nearest Neighbors. An instance-based or "lazy" classifier. We used this algorithm with attribute normalization and automatic selection of the optimal number of neighbours (1 to 15) via leave-one-out cross-validation.
- SVM. An SVM with RBF kernel (parameters cost and gamma were selected via Grid search, using an internal cross-validation procedure). The output of the SVM are probability estimates.

Figure 6 presents the results of the experiments conducted for evaluating various coding component's classifiers using 10-fold cross-validation. The classifiers were built using the four algorithms presented above. Moreover, for each algorithm six different window lengths for the extraction of training and test examples were used.

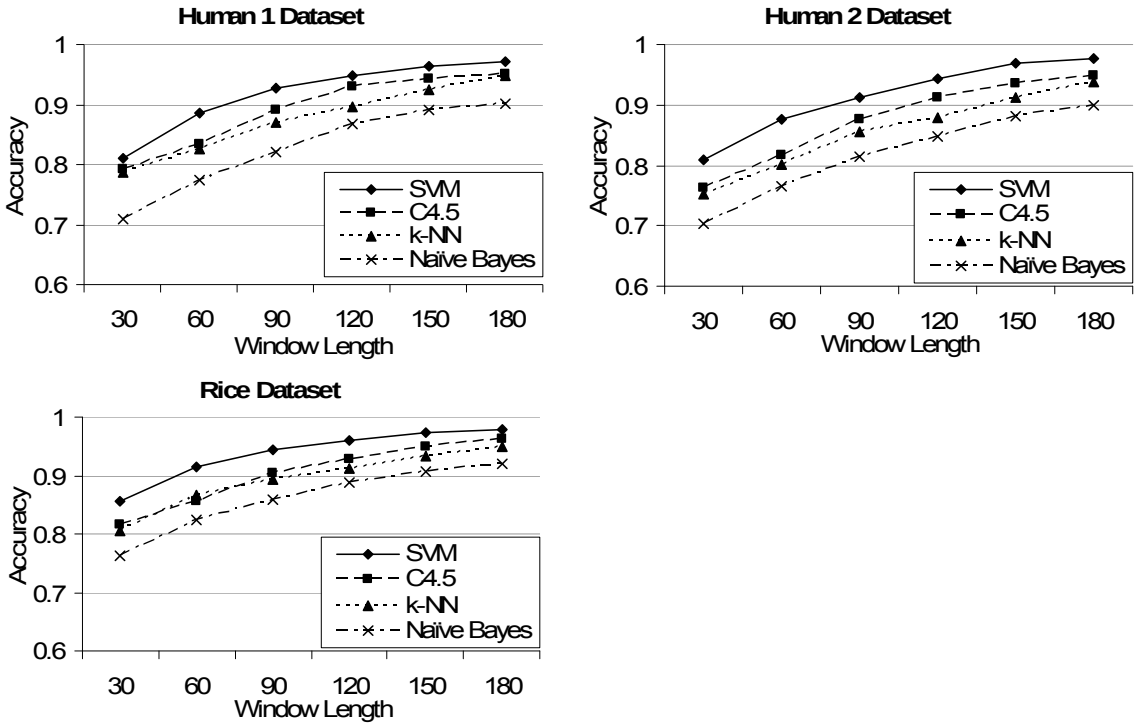

Figure 6. Evaluation of coding component's classifiers.

As shown in Figure 6, the SVM algorithm provides the best classifier in terms of classification accuracy, whereas the accuracy is also proportional to the length of the window. SVM is a very popular algorithm that usually provides very effective classifiers as presented in other related classification approaches [7, 28]. The improvement of the accuracy when larger windows are used is more or less an expected behaviour, because when a larger number of nucleotides is used for extracting the codon frequencies of training examples and test instances

there is more information available, thus the classifier can learn better and consequently can achieve better predictions. Another interesting observation is that the rate of improvement of performance when the length of the window increases, is decreasing. So, there is an upper value of window length over which the improvement of performance becomes insignificant. Moreover, the size of the sliding window should have an upper bound due to the infinite length of cDNAs. Thus a window size of 120 nucleotides is considered adequate in our setup.

5.3.2. Consensus Component Evaluation

For the consensus component's classifier construction, we have experimented with the following four Markov chains:

- A 1st order homogeneous Markov chain
- A 2nd order homogeneous Markov chain
- A 1st order non-homogeneous Markov chain.
- A 2nd order non-homogeneous Markov chain

Figure 7 presents the results of the experiments conducted for evaluating the consensus component's classifiers using 10-fold cross-validation. As shown in the figure the 1st order homogeneous Markov chain achieves the best performance among all Markov chains. The important conclusion is that a lower order homogeneous Markov chain can encapsulate better the information hidden in the near context of an ATG codon.

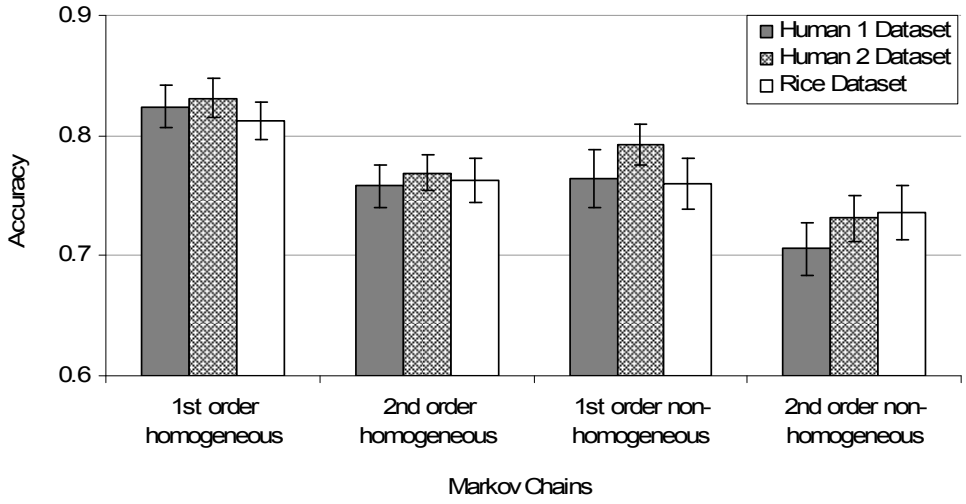

Figure 7. Evaluation of consensus component's classifiers (error bars indicate standard deviation).

5.3.3. Study of Level-0 Classifier Calibration

We have studied the effect of calibrating the probability outputs of the classifiers before passing these outputs in the level-1 classifier. A probabilistic classifier C is said to be well-calibrated if the empirical class membership probability $P(c|s(x) = s)$ converges to the score value $s(x) = s$, as the number of examples goes to infinity [29]. In order to evaluate StackTIS we experimented with both calibrated and non-calibrated probabilities on the prediction components. For that purpose we applied the Pool Adjacent Violators (PAV) algorithm [30], which performs isotonic regression.

Studying the effect of calibrating the probabilities of the components on the performance of the level-1 classifier we concluded that calibration does not improve the effectiveness and sometimes performance is degraded. These findings are compatible with the results of [31]. The authors of the above work investigated the effect of calibration on ensemble selection and found that calibrated models present improvement in squared error and cross-entropy only. Moreover, as the authors state, the improvement was due to using better models rather than taking advantage of the common scale of the calibrated models.

5.3.4. StackTIS Evaluation

In this study we use a standard approach, which is common in TIS prediction literature [8, 13, 14] as a baseline method for comparing StackTIS. This approach consists of the combination of a coding and a consensus component, followed by the RSM. This approach from now on will be referred to as "reference approach". The coding and consensus components were combined using stacking as in StackTIS, whereas the RSM was applied to the final prediction of the stacking system in order to make it comparable to StackTIS. Also, we have compared StackTIS with our previous approach MANTIS [1].

Table 1 contains the results of previous approaches with the Human 1 dataset. The performance is evaluated in terms of accuracy and adjusted accuracy, which is a skew-insensitive version of accuracy and is defined below:

$$adjusted\ accuracy = \frac{sensitivity + specificity}{2}$$

Table 1. Results of Previous Studies on Dataset Human 1

Study	Accuracy	Adjusted Accuracy
Liu et al. [14]	98.46%	85.34%
Hatzigeorgiou [8]	-	94.00%
Li et al. [15]	-	95.24%
Tzanis et al. [1]	99.14%	93.42%

In this study two classifiers have been considered as level-1 classifiers, namely Multi-response Linear Regression (MLR) and M5′. MLR is suitable for this task because in stacking it is necessary to use output class probabilities rather than class predictions (0/1) as shown in [23]. M5′ is a continuous class model tree classifier, whereas MLR involves estimating several response variables using a common set of input variables. Stacking with M5′ has been proposed as an extension of stacking with MLR in [32] and presented improved performance.

We performed extensive experiments over the three datasets, using stratified 10-fold cross-validation (CV). SVM was used as coding component's classifier and 1st order homogeneous Markov chain was used as consensus component's classifier. All stages of the knowledge discovery process (feature extraction, level-0 and level-1 model training and testing) were incorporated into the evaluation procedure. It is also important to clarify that all the experimental runs were made over the same folds for all approaches, which makes the comparison as fair as possible.

Tables 2 and 3 present the results of the comparison of StackTIS with MANTIS and the reference approach. In every case StackTIS outperforms the two approaches and the M5′ algorithm provides a slightly better stacking classifier.

Table 2. StackTIS vs. MANTIS and Reference Approach with MLR Level-1 Classifier

	Accuracy			Adjusted Accuracy		
Dataset	StackTIS	MANTIS	Reference Approach	StackTIS	MANTIS	Reference Approach
Human 1	99.32%	99.08%	98.70%	94.28%	92.99%	86.05%
Human 2	97.06%	96.14%	93.02%	96.42%	95.68%	89.14%
Rice	97.26%	97.12%	92.32%	96.87%	96.09%	88.04%

Table 3. StackTIS vs. MANTIS and Reference Approach with M5′ Level-1 Classifier

	Accuracy			Adjusted Accuracy		
Dataset	StackTIS	MANTIS	Reference Approach	StackTIS	MANTIS	Reference Approach
Human 1	99.38%	99.14%	98.90%	94.52%	93.42%	87.62%
Human 2	97.36%	96.18%	93.16%	96.97%	95.79%	90.45%
Rice	97.82%	97.24%	92.89%	97.11%	96.82%	89.93%

For the statistical comparison of StackTIS against MANTIS and the reference approach we applied the non-parametric Wilcoxon signed-rank test, in order to perform a fold to fold comparison for each dataset. The superiority of StackTIS against the reference approach on the three datasets was proved to be statistically significant at a 99% confidence level for both metrics (accuracy and adjusted accuracy) and both level-1 classifiers (MLR and M5′). Table 4 contains the statistical comparison of StackTIS against MANTIS. A +(a) denotes a statistically significant superiority of StackTIS with confidence 1-a. Note that in all cases the superiority of StackTIS against MANTIS is statistically significant with a confidence level at least 90%.

Table 4. Statistical Comparison of StackTIS vs. MANTIS

	Accuracy		Adjusted Accuracy	
	MLR	M5′	MLR	M5′
Human 1	+ (0.05)	+ (0.05)	+ (0.01)	+ (0.01)
Human 2	+ (0.01)	+ (0.01)	+ (0.01)	+ (0.01)
Rice	+ (0.10)	+ (0.05)	+ (0.05)	+ (0.05)

6. Conclusions and Future Work

In this paper we proposed a robust TIS prediction method, called StackTIS. StackTIS is an intuitive component-based approach, consisting of three main components that map the biological sub-problems identified. It is worth mentioning that the utilization of the proposed upstream length component provides the advantages of the typical RSM model and overcomes its limitations. The three components are combined using a state-of-the-art method, namely stacking. The output of StackTIS is a ranking of all candidate TISs and a probability estimate of each of them to be a TIS. The probability estimates that are returned instead of simple decisions (0/1) provide with a confidence for every decision of the system, which is very important when

more than one ATG have been predicted to be the TIS and if one wants to consider alternatives. Extensive experiments over three datasets showed that StackTIS is a method that outperforms existing ones. The improvement of StackTIS in terms of accuracy and adjusted accuracy over other popular approaches is statistically significant.

Our future plans, are focused on the extension of our approach in order to apply it on other kinds of sequences like Expressed Sequence Tags (ESTs). The incompleteness of ESTs as well as the presence of frame-shift errors in ESTs demands a number of modifications that should be made in our approach so that it can be effectively applied on the domain of EST coding region prediction.

7. References

[1] G. Tzanis, C.Berberidis, and I. Vlahavas, MANTIS: A Data Mining Methodology for Effective Translation Initiation Site Prediction. in *Proc. of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, Lyon, France, August 23-26, 2007, pp. 6443-6447.

[2] G.D. Stormo, T.D. Schneider, L.Gold, and A. Ehrenfeucht, Use of the 'Perceptron' Algorithm to Distinguish Translational Initiation Sites in E. coli, *Nucleic Acids Research*, vol. 10, no. 9, pp. 2997-3011, 1982.

[3] M. Kozak, An analysis of 5'-noncoding sequences from 699 vertebrate messenger RNAs, *Nucleic Acids Research*, vol. 15, no. 20, pp. 8125-8148, 1987.

[4] M. Kozak, and A.J.Shatkin, Migration of 40 S ribosomal subunits on messenger RNA in the presence of edeine, *Journal of Biological Chemistry*, vol. 253, no. 18, pp. 6568-6577, 1978.

[5] M. Kozak, The Scanning Model for Translation: An Update, *The Journal of Cell Biology*, vol.108, no. 2, pp. 229-241, 1989.

[6] A.G. Pedersen, and H. Nielsen, Neural Network Prediction of Translation Initiation Sites in Eukaryotes: Perspectives for EST and Genome analysis, in *Proc. of the 5th International Conference on Intelligent Systems for Molecular Biology*, Halkidiki, Greece, 1997, pp. 226-233.

[7] A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K.R. Müller, Engineering support vector machine kernels that recognize translation initiation sites, *Bioinformatics*, vol. 16, no. 9, pp. 799-807, 2000.

[8] A. Hatzigeorgiou, Translation Initiation Start Prediction in Human cDNAs with High Accuracy, *Bioinformatics*, vol. 18, no. 2, pp. 343-350, 2002.

[9] A.A. Salamov, T. Nishikawa, and M.B. Swindells. Assessing protein coding region integrity in cDNA sequencing projects, *Bioinformatics*, vol. 14, no. 5, pp. 384-390, 1998.

[10] T. Nishikawa, T. Ota, and T., Isogai, Prediction whether a human cDNA sequence contains initiation codon by combining statistical information and similarity with protein sequences *Bioinformatics*, vol. 16, no. 11, pp. 960-967, 2000.

[11] X. Zhang, G. Dong, and L. Wong, Using CAEP to Predict Translation Initiation Sites from Genomic DNA Sequences, TR2001/22, CSSE, Univ. of Melbourne, 2001.

[12] F. Zeng, H. Yap, and L. Wong, Using feature generation and feature selection for accurate prediction of translation initiation sites, in *Proc. of 13th International Conference on Genome Informatics*, Tokyo, Japan, 2002, pp. 192-200.

[13] H. Liu, and L. Wong, Data Mining Tools for Biological Sequences, *Journal of Bioinformatics and Computational Biology*, vol. 1, no. 1, pp. 139-168, 2003.

[14] H. Liu, H. Han, J. Li, and L. Wong, Using Amino Acid Patterns to Accurately Predict Translation Initiation Sites, *In Silico Biology*, vol. 4, no. 3, pp. 255-269, 2004.

[15] G. Li, T.-Y. Leong, and L. Zhang, Translation Initiation Sites Prediction with Mixture Gaussian Models in Human cDNA Sequences, *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 17, pp. 1152-1160, 2005.

[16] A. Nadershahi, S.C. Fahrenkrug, L.B.M. Ellis, Comparison of computational methods for identifying translation initiation sites in EST data, *BMC Bioinformatics*, vol. 5, no. 14, 2004.

[17] C. Iseli, C.V. Jongeneel, and P. Bucher, ESTScan: A program for detecting, evaluating, and reconstructing potential coding regions in EST sequences, in *Proc. of the 7th International Conference on Intelligent Systems for Molecular Biology*, Heidelberg, Germany, 1999, pp. 138–148.

[18] Diogenes: ORF-finding in short genomic sequences. Available: http://web.ahc.umn.edu/diogenes

[19] J. C. Rajapakse and L.S. Ho. Markov Encoding for Detecting Signals in Genomic Sequences, *IEEE/ACM Trans. Comput. Biol. Bioinform*., vol. 2, no. 2, pp. 131-142, 2005.

[20] M. Kozak, Initiation of translation in prokaryotes and eukaryotes, *Gene*, vol. 234, no. 2, pp. 187-208, 1999.

[21] V.M. Pain, Initiaton of proteins synthesis in eukaryotic cells, *Eur. J. Bio-chem.*, vol. pp. 236, 747–771, 1996.

[22] D. Wolpert, Stacked Generalization, *Neural Networks*, vol. 5, no. 2, pp. 241-259, 1992.

[23] M.T. Ting and I.H. Witten, Issues in Stacked Generalization, *J. Art. Intell. Res*., vol. 10, pp. 271-289, 1999.

[24] A.K. Seewald, How to make stacking better and faster while also taking care of an unknown weakness, in *Proc. of the 19th International Conference on Machine Learning*, 2002, pp. 554–561.

[25] R. Kohavi, The power of decision tables, in *Proc. of the 8th European Conference on Machine Learning*, Greece, 1995, pp. 174-189.

[26] I.H. Witten and E. Frank, Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[27] Y. El-Manzalawy, and V. Honavar, WLSVM: Integrating LibSVM into Weka Environment, 2005. Available: http://www.cs.iastate.edu/~yasser/wlsvm

[28] Y.-F., Sun, X.-D., Fan, and Y.-D. Li, Identifying splicing sites in eukaryotic RNA: support vector machine approach, *Computers in Biology and Medicine*, vol. 33, pp. 17–29, 2003.

[29] B. Zadrozny, and C. Elkan, Transforming classifier scores into accurate multiclass probability estimates, in *Proc. of the The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, 2002, pp. 694-699.

[30] M. Ayer, H. Brunk, G. Ewing, and W. Reid, An empirical distribution function for sampling with incomplete information, *The annals of mathematical statistics*, vol. 26, no. 4, pp. 641-647, 1955.

[31] R. Caruana, A. Munson, and A. Niculescu-Mizil, Getting the Most Out of Ensemble Selection, in *Proc. of the 6th IEEE International Conference on Data Mining*, Hong Kong, 2006, pp.828-833.

[32] S. Dzeroski, and B. Zenko, Is Combining Classifiers with Stacking Better than Selecting the Best One?, *Machine Learning*, vol. 54, pp. 255-273, 2004.

Summary

During the last decades two main scientific areas, namely biology and computer science have been characterized by major advances that have attracted the interest of all humanity. The growth of World Wide Web and the completion of Human Genome Project are two representative examples that reflect the extent of the development of these two scientific areas. However, biology and computer science have not grown separately. The need of the collaboration between biologists and computer scientists has been grown year by year as the two areas have been progressing and new scientific questions have been arising. Bioinformatics is a novel research area that has emerged as a solution to the aforementioned need. It is a very promising field that aims to provide the means to analyze and explain the vast amounts of biological data, contributing thereby to the development of other related areas like medicine.

Two relative subfields of computer science, data mining and machine learning, have provided biologists, as well as experts from other areas, a powerful set of tools to analyze new data types in order to extract various types of knowledge efficiently and effectively. These tools combine powerful techniques from different areas such as statistics, mathematics, artificial intelligence, and database technology. This fusion of technologies aims to overcome the obstacles and constraints posed by the traditional statistical methods.

In this paper we deal with the problem of the translation initiation site prediction. The accurate prediction of the translation initiation site in an mRNA or cDNA sequence is an important problem in bioinformatics. It has attracted the focus of researchers for many years and is still a very interesting topic. Translation initiation site prediction is an essential step in discovering protein coding sequences and consequently in gene prediction. Moreover, it is very useful in the framework of the complementary problem of discovering non-protein coding sequences like microRNAs, which is currently a very hot and promising area. Recent approaches perform well, however there is the need for more effective and reliable methodologies. We developed an adaptable data mining method, called StackTIS, which is modular and consists of three major prediction components: a coding component, a consensus component, and an upstream length component that allows for the utilization of the advantages of the popular Ribosome Scanning Model while overcoming its limitations. All three of them are combined into a meta-classification system, using stacked generalization, in a highly effective prediction framework. We performed extensive comparative experiments on sequences of two diverse eukaryotic organisms (Homo sapiens and Oryza sativa), indicating that StackTIS achieves statistically significant improvement in performance.