CHAPTER 4

TECHNIQUES FOR RECOGNITION OF TRANSLATION INITIATION SITES

Jinyan Li

Institute for Infocomm Research
jinyan@i2r.a-star.edu.sg

Huiqing Liu

Institute for Infocomm Research
huiqing@i2r.a-star.edu.sg

Limsoon Wong

Institute for Infocomm Research
limsoon@i2r.a-star.edu.sg

Roland H.C. Yap

National University of Singapore
ryap@comp.nus.edu.sg

Correct prediction of the translation initiation site is an important issue in genomic research. In this chapter, an in-depth survey of half a dozen methods for computational recognization of translation initiation sites from mRNA, cDNA, and genomic DNA sequences are given. These methods span two decades of research on this topic, from the perceptron of Stormo *et al.* in 1982[805] to the systematic method of explicit feature generation and selection of Wong *et al.* in 2002.[928]

ORGANIZATION.

Section 1. We begin with an introduction to the biological background of protein translation initiation. We also explain some of the difficulties involved in recognizing translation initiation sites from mRNA, cDNA, and genomic DNA sequences.

Section 2. We describe the dataset of Pedersen and Nielsen.[658] This is the most popular dataset used for investigating translation initiation sites.

Sections 3–9. After that, we give an in-depth survey of a number of translation initiation site recognition methods. Specifically, we present the method of Stormo *et al.*[805] that is based on perceptrons, the NetStart system of Pedersen and Nielsen[658] that is based on artificial neural networks, the method of Zien *et al.*[940, 941] that is based on kernel engineering of support vector machines, the approach of Wong *et al.*[494, 515, 928] that is based on explicit feature generation and selection, the method of Salamov *et al.*[737] that is based on linear discriminant function, and the method of Hatzigeorgiou[335] that is based on a complicated architecting of two artificial neural networks and the ribosome scanning rule.

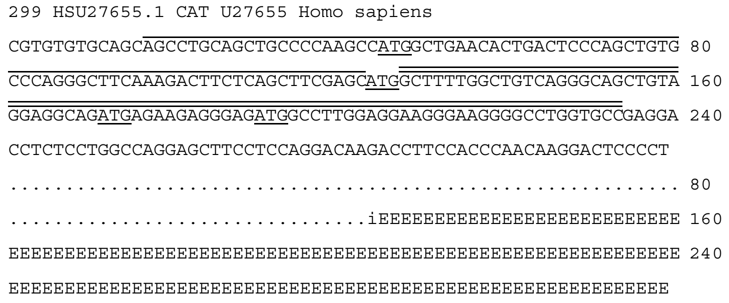
Section 10. Finally, the performance of these methods are summarized—most of them achieve close to or above 90% accuracy. A qualitative discussion of these methods are also given.

1. Translation Initiation Sites

Proteins are synthesized from mRNAs by a process called translation. The process can be divided into three distinct stages: initiation, elongation of the polypeptide chain, and termination.[107] The region at which the process initiates is called the Translation Initiation Site (TIS). The coding sequence is flanked by non-coding regions which are the 5' and 3' untranslated regions respectively. The translation initiation site prediction problem is to correctly identify TIS in a mRNA, cDNA, or genomic sequence. This forms an important step in genomic analysis to determine protein coding from nucleotide sequences.

In eukaryotes, the scanning model postulates that the ribosome attaches first to the 5' end of the mRNA and scans along the 5'-to-3' direction until it encounters the first AUG.[451] While this simple rule of first AUG holds in many cases, there are also exceptions. Some mechanisms proposed to explain the exceptions are: leaky scanning where the first AUG is bypassed for reasons such as poor context; reinitiation where a short upstream open reading frame causes a second initiation to occur; and also other alternative proposed mechanisms.[451, 620] Translation can also occur with non-AUG codons, but this is rare in eukaryotes[451] and is not considered here.

The problem of recognizing TIS is compounded in real-life sequence analysis by the difficulty of obtaining full-length and error-free mRNA sequences. Pedersen and Nielsen found that almost 40% of the mRNAs extracted from GenBank contain upstream AUGs.[658] The problem becomes more complex when using unannotated genome data or analyzing expressed sequence tags (ESTs), which usually contain more errors, and are not guaranteed to give the correct 5' end.[87, 88] Thus, the prediction of the correct TIS is a non-trivial task since the biological mechanisms are not fully understood and sequences may have errors and may not be complete.



```
299 HSU27655.1 CAT U27655 Homo sapiens
CGTGTGTGCAGCAGCCTGCAGCTGCCCCAAGCCATGGCTGAACACTGACTCCCAGCTGTG 80
CCCAGGGCTTCAAAGACTTCTCAGCTTCGAGCATGGCTTTTGGCTGTCAGGGCAGCTGTA 160
GGAGGCAGATGAGAAGAGGGAGATGGCCTTGGAGGAAGGGAAGGGGCCTGGTGCCGAGGA 240
CCTCTCCTGGCCAGGAGCTTCCTCCAGGACAAGACCTTCCACCCAACAAGGACTCCCCT
............................................................ 80
................................iEEEEEEEEEEEEEEEEEEEEEEEEEEE 160
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE 240
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
```

Fig. 1. An example annotated sequence from the dataset of Pedersen and Nielsen. The 4 occurrences of ATG are underlined. The second ATG is the TIS. The other 3 ATGs are non-TIS. The 100 nucleotides upstream of the TIS are marked by an overline. The 100 nucleotides downstream of the TIS are marked by a double overline. The ".", "i", and "E" are annotations indicating whether the corresponding nucleotide is upstream (.), TIS (i), or downstream (E).

2. Data Set and Evaluation

The vertebrate dataset of Pedersen and Nielsen[658] is used in most of the approaches that we plan to present. So, let us first describe this data set. The sequences contained in this data set are processed by removing possible introns and joining the exons[658] to obtain the corresponding mRNA sequences. From these sequences, only those with an annotated TIS, and with at least 10 upstream nucleotides as well as 150 downstream nucleotides are selected. The sequences are then filtered to remove those belonging to same gene families, homologous genes from different organisms, and sequences which have been added multiple times to the database. Since the dataset is processed DNA, the TIS site is ATG (rather than AUG).

An example entry from this dataset is given in Figure 1. There are 4 ATGs in the example sequence shown. The second ATG is the TIS. The other 3 ATGS are non-TIS. ATGs to the left of the TIS are termed upstream ATGs. So the first ATG in the figure is an upstream ATG. ATGs to the right of the TIS are termed downstream ATGs. So the third and fourth ATGs in the figure are downstream ATGs.

There are a total of 13375 ATG sites in the Pedersen and Nielsen dataset. Of these ATG sites, 3312 (24.76%) are the true TIS, while the other 10063 (75.23%) are non-TIS. Of the non-TIS, 2077 (15.5%) are upstream of a true TIS. At each of these ATG sites, a sequence segment consisting of 100 nucleotides upstream and 100 nucleotides downstream is extracted. These 13375 sequence segments are the inputs upon which TIS recognition is performed. If an ATG does not have enough upstream or downstream context, the missing context is padded with the appropriate number of dont-care symbols.

With the exception of the perceptron method presented in Section 3 and the ribosome-scanning method presented in Section 9, the selection of features and the training and testing of the machine learning methods are all performed with a 3-fold cross validation process on this dataset of Pedersen and Nielsen. That is, the dataset is divided into three equal parts, and each part is in turn reserved for testing the classification trained on the features selected from the other two parts of the data.[328]

The results of cross validation are evaluated using standard performance measures defined as follows. Sensitivity measures the proportion of TIS that are correctly recognized as TIS. Specificity measures the proportion of non-TIS that are correctly recognized as non-TIS. Precision measures the proportion of the claimed TIS that are indeed TIS. Accuracy measures the proportion of predictions, both for TIS and non-TIS, that are correct.

3. Recognition by Perceptrons

One of the earliest attempts at predicting TIS using supervised learning methods is the work of Stormo *et al.*[805] Unlike most of the work surveyed here which deals with TIS recognition for eukaryotes, the problem in Stormo *et al.* was to distinguish TIS in *E. coli*. They trained a perceptron[569] model to recognize TIS for sequences of different lengths. A perceptron is a simple artificial neural net with no hidden units and a single output unit. The output unit is activated—that is, recognizes the TIS—when $W \cdot X \geq T$ (and vice versa, $W \cdot X < T$ is not a TIS), where X is a vector encoding the candidate sequence and T is a chosen threshold value. The perceptron is trained to find a suitable weight vector W by iterating the following steps in Figure 2.

The encoding of a nucleotide is as four bits rather than the less redundant two bit encoding—*i.e.*, “A” as 1000, “C” as “0100”, “G” as 0010, and “T” as 0001. This encoding is used because a major limitation of perceptrons is that they can only be used to distinguish linearly separable functions. In particular, it cannot be used to encode the “exclusive or” (XOR) function. Suppose instead that the 2-bit binary representation was used with the following encoding,

“A” as 00, “C” as 01, “G” as 10, and “T” as 11.

It would not be possible then to train the perceptron to recognize XOR which in this encoding is the rule

A or T versus C or G.

```
    TEST:  choose a sequence $X$ from the training set
           if $X$ is a TIS and $W \cdot X \geq T$, goto TEST
           if $X$ is a TIS and $W \cdot X < T$, goto ADD
           if $X$ is not a TIS and $W \cdot X < T$, goto TEST
           if $X$ is not a TIS and $W \cdot X \geq T$, goto SUBTRACT
     ADD:  replace $W$ by $W + S$, goto TEST
SUBTRACT:  replace $W$ by $W - S$, goto TEST
```

Fig. 2. The training of a perceptron is by iterating the steps above. The perceptron convergence theorem guarantees that the solution is found in a finite number of steps if its solution exists. Alternatively, if the changes to the weight vector W plateaus, the training can also be stopped.

Stormo *et al.*[805] try using the 4-bit encoding and sequences of windows of sizes 51, 71, and 101 bases roughly centered around the position of the TIS to find the weights for the perceptron model. Of the three sizes, Stormo *et al.* find that the window of 101 bases is the best. Not surpringly, the initiation codon and the Shine-Dalgarno region is found to have the greatest significance for the weights W. The authors have also worked on a different rule-based consensus model [804] and the perceptron approach is found to be more accurate and also more precise.

4. Recognition by Artificial Neural Networks

In Pedersen and Nielsen,[658] an artificial neural network (ANN) is trained on a 203 nucleotide window centered on the AUG. It is a feed-forward ANN with three layers of neurons which can be thought of as generalizing the earlier work of Stormo *et al.* to ANNs with hidden units. As with Stormo *et al.*, the nucleotides are encoded using a redundant 4-bit representation. While ANNs with hidden units can overcome the limitations of perceptrons, the redundant representation has the advantage of not introducing encoding bias. The output layer has two neurons. The first neuron predicts if the input is a TIS. The second neuron predicts if the the input is a non-TIS. Whichever of these neurons gives the bigger score wins.

According to Pedersen and Nielsen,[658] the number of neurons in the hidden layer of the ANN does not significantly affect the performance of the ANN. They obtain results of 78% sensitivity on start AUGs and 87% specificity on non-start AUGs on their vertebrate dataset described in Section 2, giving an overall accuracy of 85%. Their system is available on the Internet as the NetStart 1.0 prediction server accessible at `http://www.cbs.dtu.dk/services/NetStart`.

Pedersen and Nielsen[658] also carry out additional analysis to try to uncover features in their sequences that are important for distinguishing TIS from non-TIS. In one of the analysis, they supply their neural network with input windows which cover the aforementioned 203 nucleotides, except for one position—a "hole"—from which input is disregarded. The hole is shifted along the input window in a series of runs of the neural network and the impact of the hole in each position is noted. This experiment reveals that position –3 is crucial to TIS recognition, as the accuracy of the neural network drops precipitously when a hole is present in this position.

Pedersen and Nielsen[658] also analyse the positions of non-translation initiating ATGs that are misclassified by their neural network as TIS. In this analysis, they discover that ATGs that are in-frame to the TIS are more likely to be misclassified as TIS regardless of whether they are upstream or downstream of the TIS.

5. Recognition by Engineering of Support Vector Machine Kernels

Zien *et al.*[940, 941] work on the same vertebrate dataset from Pedersen and Nielsen by using support vector machines (SVM) instead. The same 203 nucleotide window is used as the underlying features to be learnt. Each nucleotide is encoded using the same sparse binary encoding as Pedersen and Nielsen.

Homogeneous polynomial kernels[757] of degree d,

$$k(X, Y) = (X \cdot Y)^d = \sum_{i_1} \ldots \sum_{i_d} X[i_1] \times \ldots \times X[i_d] \times Y[i_1] \times \ldots \times Y[i_d]$$

are commonly used in SVM. Due to the encoding used for nucleotides, the position of each bit that is set indicates it is A, C, G, or T. Consequently, the dot product $(X \cdot Y)$ is equivalent to a count of the number of nucleotides that coincide in the two sequences represented by vectors X and Y. Similarly, $(X \cdot Y)^d$ is equivalent to a correlation of the nucleotide frequencies at any d sequence positions. Zien *et al.*[940] report that SVM achieves TIS recognition performance comparable to Pedersen and Nielsen's ANN using this standard type of kernels on the dataset of Pedersen and Nielsen described in Section 2.

In the polynomial kernel $(X \cdot Y)^d$ above, the correlation of nucleotide frequencies at any d sequence positions is used. However, there are a number of biological reasons that suggest we should only consider sequence positions that are not too far apart. In particular, each amino acid is coded by a triplet of adjacent nucleotides and the region upstream of a TIS is non-coding but the region downstream of a TIS is coding. Thus a kernel that zooms into such localized correlations may be appropriate for TIS recognition.

Inspired by this reasoning, Zien *et al.*[940] show how to obtain improvements by appropriate engineering of the kernel function—using a locality-improved kernel with a small window on each position. The locality-improved kernel emphasizes correlations between sequence positions that are close together, and a span of 3 nucleotides up- and down-stream is empirically determined as optimal. The locality-improved kernel is thus defined as

$$k(X,Y) = \sum_{p=1}^{l} win_p(X,Y)$$

where

$$\begin{aligned} win_p(X,Y) &= \left(\sum_{j=-3}^{3} w_j \times (X =_{p+j} Y) \right)^4 \\ &= \sum_{j_1=-3}^{3} \dots \sum_{j_4=-3}^{3} w_{j_1} \times (X =_{p+j_1} Y) \times \dots \times w_{j_4} \times (X =_{p+j_4} Y) \end{aligned}$$

Here, w_j are appropriate weights that are increasing from the boundaries to the center of the window, and

$$(X =_{p+j} Y) = \begin{cases} 1, \text{ if the nucleotides at position } p+j \text{ of} \\ \quad X \text{ and } Y \text{ are the same} \\ 0, \text{ otherwise} \end{cases}$$

With the locality-improved kernel, Zien *et al.*[940] obtain an accuracy of 69.9% and 94.1% on start and non-start AUGs respectively, giving an overall accuracy of 88.1% on the dataset described in Section 2.

Zien *et al.*[941] further improve their previous results by engineering a more sophisticated kernel—a so-called Salzberg kernel. The Salzberg kernel is essentially a conditional probabilitistic model of positional di-nucleotides. The Salzberg kernel gives an overall accuracy of 88.6% on the dataset described in Section 2.

6. Recognition by Feature Generation, Feature Selection, and Feature Integration

Wong *et al.*[515, 928] show that good performance comparable to the best results can be obtained by a methodology based on these three steps:

(1) generate candidate features from the sequences,
(2) select relevant features from the candidates, and

(3) integrate the selected features using a machine learning method to build a system to recognize specific properties—in this case, TIS—in sequence data.

We present these three steps in the next three subsections

6.1. *Feature Generation*

The sequence segments prepared in the previous section are not suitable for direct application of most machine learning techniques, as these techniques rely on explicit signals of high quality. It is necessary to devise various signals and sensors for these signals, so that given a sequence segment, a score is produced to indicate the possible presence of such a signal in that sequence segment. The obvious strategy to devising these signals and sensors is to generate a large number of candidate features, and to evaluate them against an annotated dataset to decide which ones are signals and which ones are noise.

Wong *et al.*[928] make use of the general technique of k-grams and a few refinements for producing candidate features. A k-gram is simply a pattern of k consecutive letters, which can be amino acid symbols or nucleic acid symbols. K-grams can also be restricted those in-frame ones. Each k-gram and its frequency in the said sequence fragment becomes a candidate feature. Another general technique for producing these candidate features is the idea of position-specific k-gram. The sensor for such a feature simply reports what k-gram is seen in a particular position in the sequence fragment.

For ease of discussion, given a sequence segment, we refer to each position in the sequence segment relative to the target ATG of that segment. The "A" in the target ATG is numbered as +1 and consecutive downstream positions—that is, to the right—from the target ATG are numbered from +4 onwards. The first upstream position—that is, to the left—adjacent to the target ATG is –1 and decreases for consecutive positions towards the 5' end—that is, the left end of the sequence fragment.

Let us use the sequence segment centered at the second ATG in Figure 1 and comprising 100 nucleotides upstream and 100 nucleotides downstream of this ATG for illustration of the various k-gram features described by Wong *et al.*.[928] These upstream and downstream nucleotides are marked using overline and double overline in the figure.

For the basic k-grams, k is the length of a nucleotide pattern to be generated. Some typical values for k are 1, 2, 3, 4, and 5. Since there are 4 possible letters for each position, there are 4^k possible basic k-grams for each value of k. For example, for k = 3, one of the k-grams is ATG and the frequency of this k-gram is 4 in our example sequence segment. The candidate feature (and value assignment) corresponding to this is "ATG=4".

The upstream region of a TIS is non-coding and the downstream region of a TIS is coding. It can therefore be expected that they have some different underlying features. So it is wise to introduce additional classes of k-grams to attempt to capture these differences. These are the upstream and downstream k-grams.

For the upstream k-grams, Wong *et al.*[928] count only occurrences of the corresponding patterns upstream of the target ATG. Again, for each value of k, there are 4^k upstream k-grams. For example, for k = 3, some k-grams are: ATG, which has frequency 1 in this context; GCT, which has frequency 5 in this context; and TTT, which has frequency 0 in this context. The candidate features and value assignments corresponding to these k-grams are "upstream ATG=1", "upstream GCT=5", and "upstream TTT=0".

For the downstream k-grams, Wong *et al.*[928] count only occurences of the corresponding patterns downstream of the target ATG. Again, for each value of k, there are 4^k downstream k-grams. For example, for k = 3, some k-grams are: ATG, which has frequency 2 in this context; GCT, which has frequency 3 in this context; and TTT, which has frequency 2 in this context. The candidate features and value assignments corresponding to these k-grams are "downstream ATG=2", "downstream GCT=3", and "downstream TTT=2".

The biological process of translating from nucleotides to amino acids is to have 3 nucleotides—the so-called codons—codes for one amino acid, starting from the TIS. Therefore, 3-grams in positions ..., -9, -6, -3, +4, +7, +10, ... are aligned to the TIS. Wong *et al.*[928] call those 3-grams in positions ..., -9, -6, and -3, the in-frame upstream 3-grams, and those 3-grams in positions +4, +7, +10, ..., the in-frame downstream 3-grams. As these 3-grams are in positions that have biological meaning, they are also good candidate features. There are 2×4^3 such 3-grams. In our example sequence fragment, some in-frame downstream 3-grams are: GCT, which has frequency 1 in this context; TTT, which has frequency 1 in this context; ATG, which has frequency 1 in this context. The corresponding candidate features and value assignments are "in-frame downstream GCT=1", "in-frame downstream TTT=1", and "in-frame downstream ATG=1". Some in-frame upstream 3-grams are: GCT, which has frequency 2 in this context; TTT, which has frequency 0 in this context; ATG, which has frequency 0 in this context. The corresponding candidate features and value assignments are "in-frame upstream GCT=2", "in-frame upstream TTT=0", and "in-frame upstream ATG=0".

Another type of features are what Wong *et al.*[928] call the position-specific k-grams. For this type of k-grams, they simply record which k-gram appears in a particular position in the sequence segment. It is sufficient to consider only 1-grams, that is, k-grams for k = 1. Since our sequence segment has 100 nucleotides flanking each side of the target ATG, there are 200 position-specific 1-grams. In our example sequence segment, some position-specific 1-grams are: at position +4 is G and at position –3 is G. The corresponding candidate features and value assignments are "position+4=G" and "position–3=G".

Combining all the features discussed above, for $k = 1, ..., 5$, each sequence segment is coded into a record having $(\sum_{k=1}^{5} 4^k + 4^k + 4^k) + 2 \times 4^3 + 200 = 4436$ features. For illustration, our example sequence segment is coded into this record: $\{$..., "ATG=4", ..., "upstream ATG=1", "upstream GCT=5", "upstream TTT=0", ..., "downstream ATG=2", "downstream GCT=3", downstream TTT=2", ..., "in-frame downstream GCT=1", "in-frame downstream TTT=1", "in-frame downstream ATG=1", ..., "in-frame upstream GCT=2", "in-frame upstream TTT=0", "in-frame upstream ATG=0", ..., ..., "position–3=G", ..., "position+4=G", ...$\}$. Such a record is often called a feature vector.

These 4436 features as described above are generated for each of the 13375 sequence segments corresponding to the 13375 ATG sites in the Pedersen and Nielsen dataset.[658] We note that other techniques for generating candidate features are possible. For example, we can compute a specific statistic on the sequence segment, such as its GC ratio. Specific biological knowledge can also be used to devise specialized sensors and features, such as CpG island,[271] Kozak consensus pattern,[452] *etc*. An exhaustive exposition is outside the scope of this chapter. In the next two subsections, we show how to reliably recognize TIS on the basis of a small subset of our 4436 candidate features.

6.2. *Feature Selection*

This number of candidate features is clearly too many. Most of them can be expected to be noise that can confuse typical machine learning algorithms. So the next step in the methodology proposed by Wong *et al.*[928] is to apply a feature selection technique to pick those features that are most likely to help in distinguishing TIS from non-TIS.

Any feature selection techniques can be used, inclusing signal-to-noise measure,[297] t-test statistical measure,[133] χ^2 statistical measure,[514] entropy measure,[242] information gain measure,[692] information gain ratio,[693] Fisher criterion score,[251] Wilcoxon rank sum test,[742] correlation-based feature selection method (CFS),[316] and so on. All of these methods are described in Chapter 3.

Wong *et al.*[928] apply CFS to the feature vectors derived from the Pedersen and Nielsen dataset as described in Section 2 and Subsection 6.1 in a 3-fold cross validation setting. It turns out that in each fold, exactly the same 9 features are selected by CFS, *viz.*

(1) “position–3”,
(2) “in-frame upstream ATG”,
(3) “in-frame downstream TAA”,
(4) “in-frame downstream TAG”,
(5) “in-frame downstream TGA”,
(6) “in-frame downstream CTG”,
(7) “in-frame downstream GAC”,
(8) “in-frame downstream GAG”, and
(9) “in-frame downstream GCC”.

These 9 features are thus very robust differentiators of TIS from non-TIS. Furthermore, there are good biological reasons for most of them.

“Position–3” can be explained by the known correspondence to the well-known Kozak consensus sequence, GCC[AG]CC$\underline{\text{AUG}}$G, for vertebrate translation initiation sites.[403, 450] Hence having a “A” or “G” in this position indicates that the target ATG is more likely to be a TIS. This is the same feature deduced as important by Pedersen and Nielsen[658] in their “hole-shifting” experiment discussed in Section 4.

“In-frame upstream ATG” can be explained by the ribosome scanning model.[9, 451] The ribosome scans the mRNA in a 5’-to-3’ (that is, left-to-right) manner until it encounters the first ATG with the right context for translation initiation. Thus a ATG that is closer to the 5’ end have a higher probability to be a TIS. Consequently, the presence of an in-frame ATG upstream of the target ATG indicates that the target ATG is less likely to be a TIS. This is also consistent with the observation by Rogozin *et al.*[718] that a negative correlation exists between the strength of the start context and the number of upstream ATGs. This is also a feature deduced by Pedersen and Nielsen[658] in a detailed analysis on the erroneous predictions made by their neural network.

“In-frame downstream TAA”, “in-frame downstream TAG”, and “in-frame downstream TGA” can be explained as they correspond to in-frame stop codons downstream from the target ATG. These 3 nucleotide triplets—TAA, TAG, TGA—do not code for amino acids. They are called the stop codons. The biological process of translating in-frame codons into amino acids stops upon encountering an in-frame stop codon. Thus the presence of any of these three features means there is an in-frame stop codon within 100 nucleotides downstream of the target ATG. Consequently, the protein product corresponding to the sequence is no more than 33 amino acids. This is smaller than most proteins. Hence the target ATG is not likely to be a TIS. This group of stop codon features are not reported by Pedersen and Nielsen,[658] Zien *et al.*,[940, 941] nor Hatzigeorgiou,[335] presumably the complex and/or low-level nature of their systems prevented them from noticing this important group.

Wong *et al.*[928] do not have a clear biological explanation for the remaining 4 selected features, other than that of codon biases.

6.3. *Feature Integration for Decision Making*

In order to show that the 9 features identified in the previous step are indeed relevant and reliable differentiators of TIS from non-TIS, Wong *et al.* [928] demonstrate in a 3-fold cross validation setting that practically any machine learning methods can be trained on these 9 features to produce TIS recognizers of extremely competitive accuracy. In particular, among those classification methods described in Chapter 3, Wong *et al.*[928] give the results obtained on Naive Bayes (NB),[471] Support Vector Machine (SVM),[855] and C4.5.[693]

According to Wong *et al.*,[928] NB trained on the 9 features selected above yields an effective TIS recognizer with sensitivity = 84.3%, specificity = 86.1%, precision = 66.3%, and accuracy = 85.7%. SVM trained on these 9 features yields an accurate TIS recognizer with sensitivity = 73.9%, specificity = 93.2%, precision = 77.9%, and accuracy = 88.5%. C4.5 trained on these 9 features also yields an accurate TIS recognizer with sensitivity = 74.0%, specificity = 94.4%, precision = 81.1%, and accuracy = 89.4%.

7. Improved Recognition by Feature Generation, Feature Selection, and Feature Integration

As every 3 nucleotides code for an amino acid, it is legitimate to investigate if an alternative approach to generating features based on amino acids can produce effective TIS recognizers. Also, in the previous sections, Wong *et al.* [928] use features selected by CFS, hence it is legitimate to investigate if features selected by other methods can produce effective TIS recognizers. Li *et al.* [494] and Liu and Wong[515] pursue these two alternatives. In this section, we discuss their results.

For generating features, Li *et al.*[494] and Liu and Wong[515] take the sequence segments of 100 nucleotides upstream and 100 nucleotides downstream of the target ATG as before. Then they consider 3-grams that are in-frame—that is, those 3-grams that are aligned to the ATG at positions ..., -6, -3, +4, +7, ... Those in-frame 3-grams that code for amino acids are converted into the corresponding amino acid letters. Those in-frame 3-grams that are stop codons are converted into a special letter symbolizing a stop codon. From the conversion above, Li *et al.* [494] generate the following types of k-grams:

(1) up-X, which counts the number of times the amino acid letter X appears in

the upstream part, for X ranging over the standard 20 amino acid letters and the special stop symbol.

(2) down-X, which counts the number of times the amino acid letter X appears in the downstream part, for X ranging over the standard 20 amino acid letters and the special stop symbol.

(3) up-XY, which counts the number of times the two amino acid letters XY appear as a substring in the upstream part, for X and Y ranging over the standard 20 amino acid letters and the special stop symbol.

(4) down-XY, which counts the number of times the two amino acid letters XY appear as a substring in the upstream part, for X and Y ranging over the standard 20 amino acid letters and the special stop symbol.

Li *et al.*[494] also generate the following Boolean features from the original sequence fragments: up-ATG, which indicates that an in-frame ATG occurs in the upstream part; up3-AorG, which indicates that an "A" or a "G" appears in position –3 ; down4-G, which indicates that a "G" appears in position +4. These last two features are inspired by the Kozak consensus sequence, GCC[AG]CCAUGG, for vertebrate translation initiation sites.[403, 450] A total of $2 \times 21 + 2 \times 21^2 + 3 = 927$ features are thus generated as described above.

For selecting features, they use the entropy measure to rank the relevance of each of these 927 candidate features in a 3-fold cross validation setting. In each fold, the top 100 features are selected. The following features are consistently among the top 10 features in each of the 3 folds: up-ATG, down-STOP, down-L, down-D, down-E, down-A, up3-AorG, up-A, down-V. Up-M also appears among the top features in each fold, but we exclude it as it is redundant given that up-ATG is true if and only if up-M > 0. The detailed ranking of these features in each fold is given in Figure 3.

Interestingly, most of these features, except up-A and down-V, correspond to those selected by CFS on the original nucleotide sequence fragments in Section 6.2. Specifically, up-ATG corresponds to "in-frame upstream ATG"; down-stop corresponds to "in-frame downstream TAA", "in-frame downstream TAG", and "in-frame downstream TGA"; up3-AorG corresponds to "position–3"; down-L corresponds to "in-frame downstream CTG"; down-D corresponds to "in-frame downstream GAC"; down-E corresponds to "in-frame downstream GAG"; and down-A corresponds to "in-frame downstream GCC".

For validating whether accurate systems for recognizing TIS can be developed using features based on amino acids, Liu and Wong[515] test the C4.5, SVM, and NB machine learning methods in 3-fold cross validations. The top 100 features selected by the entropy measure are used in each fold.

Fold	up ATG	down STOP	up3 AorG	down A	down V	up A	down L	down D	down E
1	1	2	4	3	6	5	8	9	7
2	1	2	3	4	5	6	7	8	9
3	1	2	3	4	5	6	8	9	7

Fig. 3. Ranking of the top 9 features selected by the entropy measure method as relevant in each of the 3 folds.

For C4.5, they obtain sensivity = 74.88%, specificity = 93.65%, precision = 79.51%, and accuracy = 89.00%. This is comparable to the performance of C4.5 using the 9 features selected by CFS in Section 6.2.

For SVM, they obtain sensitivity = 80.19%, specificity = 96.48%, precision = 88.24%, and accuracy = 92.45%. This is significantly better than the performance of SVM using the 9 features selected by CFS in Section 6.2. In fact, it is the best reported results on the Pedersen and Nielsen dataset[658] that we know of.

For NB, they obtain sensitivity = 70.53%, specificity = 87.76%, precision = 65.47%, and accuracy = 83.49%. This is considerably worse than the other results. The increase from 9 features in Subsection 6.3 to 100 features here has apparently confused NB.

Li *et al.*[494] use just the top 10 features selected by the entropy measure in their study. They use the PCL (Prediction by Collective Likelihood of emerging patterns) classifier,[497] described in Chapter 3, to integrate these top 10 entropy features. They report that PCL achieves 84.7% sensitivity, 88.7% specificity and 87.7% overall accuracy.

8. Recognition by Linear Discriminant Function

Linear discriminant analysis provides a linear function that separates two classes while minimizing misclassification. Assume a p-feature variable X is given as a vector, then the linear discriminant function

$$y = \sum_{i=1}^{p} \alpha[i] \times X[i]$$

classifies X into the first class if $y \geq c$ and into the second class if $y < c$. The optimal selection of the vector of feature weights $\alpha[\_]$ and the decision threshold

c is typically determined by maximizing the ratio of between-class-variation to within-class-variation.

The ATGpr program of Salamov *et al.*[737] uses a linear discriminant function that combines several statistical features derived from training sequences. Specifically, ATGpr uses the following characteristics extracted from their training sequences:

(1) Positional triplet weight matrix around an ATG. For each triplet nucleotides $i = 1, 2, \ldots, 64$, and position $j = -14, -13, \ldots, +4, +5$, the frequencies for true TIS $f_{TIS}(i, j)$ and all candidate ATGs $f_{totalATG}(i, j)$ are calculated. Then the propensity for a particular triplet i to be in a specific position j relative to the initiation codon is given as:

$$P_{triplet}(i, j) = \frac{f_{TIS}(i, j)}{f_{totalATG}(i, j)}$$

To use these propensities,the total score around each candidate ATG region is added together for the window –14 to +5.

(2) In-frame hexanucleotide weight matrix. For each hexanucleotide $k = 1, 2, \ldots, 4^6$, the frequencies of it appearing in an in-frame position upto 300 nucleotide downstream of the candidate ATG for true TIS $f_{coding}(k)$ and downstream of false TIS in a noncoding region $f_{noncoding}(k)$ are calculated. Then the propensity for a particular hexanucleotide is given as:

$$P_{hexamer}(k) = \frac{f_{coding}(k)}{f_{noncoding}(k)}$$

(3) 5’ UTR-ORF hexanucleotide difference. A true TIS has a higher value of the average hexamer coding propensities in the potential 5’UTR region [–1, –50] and potential coding region [+1, +50], where the positions are relative to the candidate ATG. So Salamov *et al.* also uses the difference between the average hexanucleotide coding propensities between these two regions as a feature.

(4) Signal peptide characteristic. Within a 30 amino acid window down-stream of each ATG, the most hydrophobic 8-residue peptide was identified. Hydrophobicity is calculated using the hydropathy scale of Kyte and Doolittle.[466]

(5) Presence of another upstream in-frame ATG. This is a simple Boolean-valued feature. If an extra ATG is found upstream of the candidate ATG without encountering an in-frame stop codon, the likelihood of the ATG being an TIS is down-weighted.

(6) Upstream cytosine nucleotide characteristic. The frequency of cytosine in the region [–7, –36] upstream of a candidate ATG is counted, as it has been observed that 5’ UTRs of human genes are cytosine rich.[524]

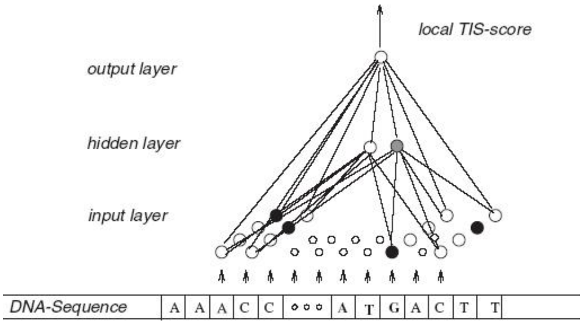


Fig. 4. The consensus ANN of DIANA-TIS. A window of 12 nucleotides is presented to the trained ANN. A high score at the output indicates a possible TIS. (*Image credit: Artemis Hatzigeorgiou.*)

In a more recent work,[618] an improved version of ATGpr called ATGpr_sim is developed, which uses both statistical information and similarities with other known proteins to obtain higher accuracy. ATGpr can be accessed at `http://www.hri.co.jp/atgpr/`. When searching TIS in a given cDNA or mRNA sequence, the system outputs several ATGs—5 by default—in the order of decreasing confidence. If we always take the ATG with highest confidence as TIS, then for the 3312 sequences in the Pedersen and Nielsen dataset, ATGpr can predict correctly true TIS in 2941 (88.80%) of them. Note that by taking only the ATG with the highest confidence as the TIS, only 1 prediction is made per sequence. Hence this figure is not directly comparable to the 3-fold cross validation figures reported on other methods earlier.

9. Recognition by Ribosome Scanning

Hatzigeorgiou[335] reports a highly accurate TIS prediction program, DIANA-TIS, using artificial neural networks trained on human sequences. Their dataset contains full-length cDNA sequences which has been filtered for errors. An overall accuracy of 94% is obtained using an integrated method which combines a consensus ANN with a coding ANN together with the ribosome scanning model.

The consensus ANN assesses the candidate TIS and its immediate surrounding comprising a window from positions –7 to +5 relative to the candidate TIS. The consensus ANN is a feed-forward ANN with short cut connections and two hidden units, as shown in Figure 4. The coding ANN is used to assess the coding potential of the regions upstream and downstream of a candidate TIS. The coding ANN works on a window of 54 nucleotides. As every three nucleotides form a codon

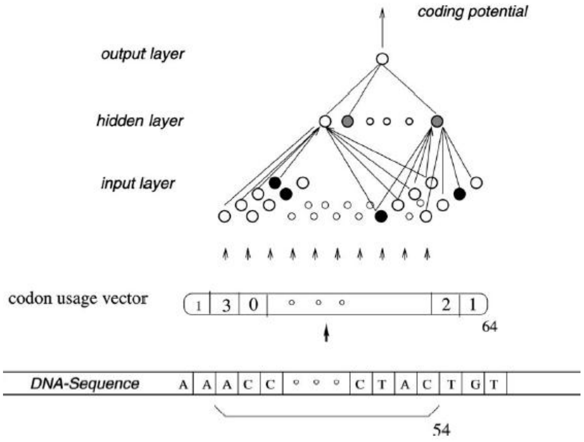


Fig. 5. The coding ANN of DIANA-TIS. A window of 54 nucleotides is presented to the trained ANN. A high score at the output indicates a coding nucleotide. (*Image credit: Artemis Hatzigeorgiou.*)

that translates into an amino acid, there are 64 possible codons. To assess the coding potential of the window of 54 nucleotides, this window is transformed into a vector of 64 units before feeding into the coding ANN. Each unit represents one of the 64 codons and gives the normalized frequency of the corresponding codon appearing in the window. The coding ANN is a feed-forward ANN and has two hidden units, but no short cut connection, as shown in Figure 5. Both the consensus ANN and the coding ANN produce a score between 0 and 1.

The two ANNs are basically integrated as follows. Given a candidate ATG, the consensus ANN is applied to to a window of 12 nucleotides at positions –7 to +5 to calculate a consensus score s_1. Then the coding ANN is applied to the in-frame 60 positions before the ATG by sliding along a window of 54 nucleotides and summing the output of the coding ANN at each position to obtain an upstream coding score s_2. The coding ANN is also applied to the in-frame 60 positions after the ATG to obtain a downstream coding score s_3 in a similar fashion. The final score for the ATG is then obtained as $s_1 \times (s_3 - s_2)$. The score calculations above are applied to all the ATGs in a mRNA sequence one after another, and the first ATG to score at above 0.2 is taken as the TIS of the mRNA sequence. This preference to accept the first ATG in a favour context as the TIS is the so-call ribosome scanning model.

Note that in the ribosome scanning model,[9, 451] an mRNA sequence is scanned from left to right, testing each ATG in turn until one of them is classified as TIS; all the ATGs to the right of this ATG are skipped and classified as non-TIS. In short, exactly one prediction is made per mRNA under the ribosome scanning model. Hence, accuracy figures based on the ribosome scanning model should not be compared with models that tests every ATG. In addition, Hatzigeorgiou uses a dataset that is different from Pedersen and Nielsen. So her results cannot be directly compared to results obtained on the Pedersen and Nielsen dataset or obtained without using the ribosome scanning model.

10. Remarks

The approach of Pedersen and Nielsen[658] is interesting in that their inputs are extremely low level—just a string of nucleotides—and relies entirely on their ANN to learn high-level correlations to make prediction. Unfortunately, it is not easy to extract these correlations out of their ANN to gain more insight into sequence features that distinguish TIS from non-TIS. Nevertheless, by more extensive experiments and analysis, Pedersen and Nielsen are able to suggest that position –3 is crucial to distinguishing TIS from non-TIS.

The approach of Zien *et al.*[940, 941] and Hatzigeorgiou[335] are also very interesting in that they show us how to perform sophisticated engineerings of SVM kernel functions and ANNs. Unfortunately, it is also not easy to extract more insight into sequence features that distinguish TIS from non-TIS from their systems. Nevertheless, the improved results of the locality-improved kernel in Zien *et al.*[940] over that of standard polynomial kernels suggest that local correlations are more important than long-ranged correlations in distinguishing TIS from non-TIS.

The approach of Wong *et al.*[928] and Li *et al.*[494] is interesting in that they focus on deriving high-level understandable features first, and then use these features to distinguish TIS from non-TIS. In fact, by applying a decision tree induction method such as C4.5[693] on the selected features, highly meaningful rules such as “if up-ATG = Y and down-STOP > 0, then prediction is false TIS”; “if up3-AorG = N and down-STOP > 0, then prediction is false TIS”; and “if up-ATG = N and down-STOP ≤ 0 and up3-AorG = Y, then prediction is true TIS” are also extracted.

Finally, we summarize the TIS-recognition performance of the various methods described in this chapter in Figure 6.

I. Accuracy of some TIS recognition methods. Pedersen and Nielsen,[658] Zien *et al.*,[940, 941] and Li *et al.*[494] are directly comparable to results in Parts II and III. Hatzigeorgiou[335] is not directly comparable as she uses a different dataset and also the ribosome scanning model. Salamov *et al.*[737] is also not directly compatible as we have derived its result using the ribosome scanning model.

Classifier	Sensitivity	Specificity	Accuracy
Pedersen and Nielsen[658]	78.0%	87.0%	85.0%
Zien *et al.*[940]	69.9%	94.1%	88.1%
Zien *et al.*[941]	-	-	88.6%
Hatzigeorgiou[335]	-	-	94.0%
Salamov *et al.*[737]	-	-	88.8%
Li *et al.*[494]	84.7%	88.7%	87.7%

II. Accuracy of NB, SVM, and C4.5 reported by Wong *et al.*[928] on the Pedersen and Nielsen dataset based on 9 features selected using CFS.

Classifier	Sensitivity	Specificity	Precision	Accuracy
NB	84.3%	86.1%	66.3%	85.7%
SVM	73.9%	93.2%	77.9%	88.5%
C4.5	74.0%	94.4%	81.1%	89.4%

III. Accuracy of NB, SVM, and C4.5 reported by Liu and Wong[515] on the Pedersen and Nielsen dataset based on the 100 translated features selected using the entropy measure.

Classifier	Sensitivity	Specificity	Precision	Accuracy
NB	70.53%	87.76%	65.47%	83.49%
SVM	80.19%	96.48%	88.24%	92.45%
C4.5	74.88%	93.65%	79.51%	89.00%

Fig. 6. Accuracy results of various feature selection and machine learning methods for TIS recognition.