Splice Site Prediction using Multiple Sequence Alignment

Ross Bayer and Konstantin Davydov
Collaborators: Marina Sirota, Sam Gross, Serafim Batzoglou

Introduction

Computational prediction of genes is currently an area of active research. Since only 2% of the entire human genome codes for proteins, ruling out the 98% of the genome which does not directly result in protein production would be of great value to genomic research. While genes in simple prokaryotic organisms like bacteria are relatively easy to identify (since they begin with a start codon[1] and terminate with a stop codon), the situation in eukaryotic organisms, such as mammals, is more complicated. Only certain parts of a gene (known as "exons") are actually transcribed into proteins, while other subsequences (known as "introns") are removed before the protein transcription process.

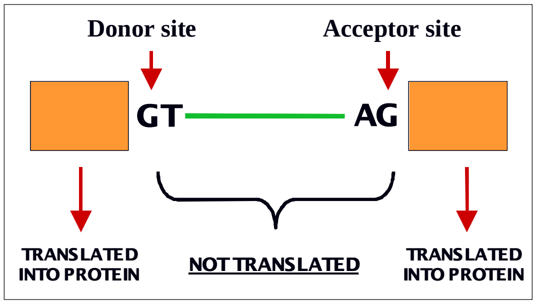

Fig 1. Splice Sites

Splicing refers to the machinery which removes these introns from the sequence, and splice sites are the locations in the sequence which indicate to the splicing machinery that splicing should occur[2]. Since gene prediction in these more complex organisms can no longer depend upon such a simple strategy as looking at start and stop codons (since introns can contain stop codons which will not actually terminate the gene), we need an accurate method of predicting splice sites, i.e. modeling intron/exon behavior, in order to accurately predict the likelihood of a region being a gene.

Splice Site Recognition

Splice sites fall into two categories: donor sites at the 5' end of an intron and acceptor sites at the 3' end of an intron (see Fig. 1). These sites display certain characteristic patterns, e.g. 99% of donor sites begin with GT and acceptor sites tend to end with AG. However, not all locations with base pairs GT or AG are necessarily splice sites. Some occurrences of GT or AG occur outside of a gene or inside an exon. These are typically called decoys, as they do not in fact indicate the presence of a splice site (see Fig. 2). Nonetheless, the clear presence of patterning within the data makes this classification task (between genuine splice site and decoy) amenable to machine learning methods.

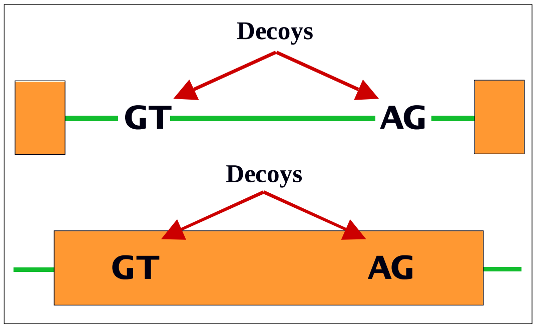

Fig 2. Decoys

[1] A codon is a DNA triplet of three base pairs. Each such codon is mapped to an amino-acid when proteins are transcribed.
[2] A less formal definition is that splice sites mark the boundaries between exons and introns.

Traditional models have typically been based on Hidden Markov Models, though the very strong independence assumptions leave much to be desired, especially for the modeling of long-range interaction effects which biologists generally believe are present. Support Vector Machines have also been applied to the problem with some success, however only using features from the particular sequence of interest. We extend this approach to also use features from multiple aligned sequences, in particular: mouse, rat, chicken, dog, fugu, zebrafish, and chimpanzee (see Fig. 3).

Human	GGCCT**AG**TAT
Mouse	GGCCA**AG**CCG
Rat	AGCCA**AG**CGC
Chicken	-GCCC**AG**G--
Dog	CGCCG**AG**ATA
Fugu	NNCCC**AG**GGT
Zebrafish	.....**AG**GCT
Chimp	GGCCT**AG**TAA

Fig 3. Multiple Alignment of Species

Sequence alignment is a thoroughly studied field of research which does a good job of comparing homologous sequences from different genomes. We can use such alignment data as a source for extracting additional features. This information can be quite useful since functionally important patterns are more conserved over the course of evolution. Furthermore, having several sequences with different evolutionary distances from human (e.g. zebrafish and chimp) will be beneficial too, as it provides more information about the evolutionary history.

Machine Learning Methodology

We used John Platt's Sequential Minimal Optimization algorithm for Support Vector Machines, as implemented by an appropriate SVM package for Matlab (LIBSVM)[3]. The domain of the features was the base pair (A, T, C, or G) or alignment information ('-' for gap, 'N' for no available information, '.' for unaligned) which we represented with 7 values, where the value corresponding to the base pair is 1 and the other values are 0 (i.e. A is represented as [1 0 0 0 0 0 0], T is represented as [0 1 0 0 0 0 0], C as [0 0 1 0 0 0 0], etc). We have one feature for each location in the range spanning 3 positions before a suspected donor site to 37 positions after for each sequence in the multiple alignment. In the case of acceptor sites, the corresponding range was from 6 positions before to 3 positions after for each species. These specific ranges were chosen based on biological considerations.

Separate SVMs were trained for the two tasks of discriminating between donor sites and decoy donor sites, and between acceptor sites and decoy acceptor sites. A quadratic kernel was chosen in order to model interaction effects (possibly long-range) between the various base pairs. In addition, different penalties were used in the cases of misclassification of positive examples and misclassification of negative examples, since in this field of research, false negatives are much more damaging than false positives. This ratio was adjusted to be 1:1000 in line with a best approximation of the ratio of true splice sites to decoys within the actual human genome.

Data

In collaboration with Serafim Batzoglou's computational biology research group, we obtained the full genome multiple alignment (in FASTA format) of human, mouse, rat, chicken, dog, fugu, zebrafish,

[3] See http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

and chimpanzee from the UCSC browser. In addition, we obtained human gene annotation files (in GTF format[4]) which label exons within well-studied genes. This was used as the source for supervised learning. These annotation files were used to extract known splice sites from the given alignments using human as our reference species. These splice site locations were then used to generate positive examples for our SVM training by scanning through the alignment file for each of the chromosomes. Particular attention was paid to the case of splice sites on the negative strand, in which case the corresponding sequence data had to be reversed and complemented for the format to be comparable.

In order to generate negative examples (decoys), a high number of random positions within the genome were chosen, each likely with extremely high probability to not be a splice site (a random position has probability of about 0.000009 of being a splice site). The features for these random positions were then extracted from the multiple alignment, but only those which happened to fall upon an AG or a GT were kept in order to isolate decoys.

The ratio of positive to negative examples was adjusted to be approximately 1:1, bearing in mind that a higher false negative penalty was used. Training for each SVM model was done on a randomly selected subset of the data in which each example had 70% chance of being included in the subset, and testing for cross-validation purposes was performed on the remaining examples excluded from the training set (approximately 30% of the data).

SVM Input

The resulting input to the SVM consisted of the *label* matrix and a *features* matrix. The *label* matrix was a vector of labels, where a +1 corresponded to a positive label (splice site) and a –1 corresponded to a negative label (decoy). The *features* matrix consisted of 2296 features (41 positions × 7 letters × 8 species) in the case of the donor site model, and 560 features in the case of the acceptor site model (10 positions instead of 41).

Computational Challenges

There were several computational challenges involved in this data generation process. Due to the prohibitively large sizes of the sequence alignment files involved (several GB per each of the 24 chromosomes), extracting features from these files had to be performed in a very careful fashion. Firstly, inefficient random access would lead to vast slow-downs due to repeated seeks. Secondly, reading in large portions of the file at any time could use considerable amounts of memory and in the worst-case scenario result in thrashing.

The approach taken to alleviate these difficulties was to calculate beforehand all the positions within the chromosome that would be analyzed and potentially have features extracted. As mentioned, this was done based on the GTF annotation files, followed by generation of random positions for negative examples. These positions (and positive/negative status) were all stored within a single vector, which was then sorted in increasing order.

This allowed scanning of the alignment file to be done in sequential order. Since the positions were known in advance, seeking could be done to each correct position directly, avoiding the unnecessary overhead of reading in large buffers, and speeding up file traversal time. In addition, the features for each training example were built up incrementally species by species. In other words, all examples had their features for human populated, then all examples had

[4] See http://genes.cs.wustl.edu/GTF21.html.

their features for mouse populated, and so on. This could be done since the length of each chromosome was known in advance, allowing direct calculation of the correct location within the file. This species optimization ensured that the entire file traversal was also in strictly sequential order, reducing the total seek time to the minimum possible. This strategy allowed for minimal total access to the alignment file, and made dealing with such huge files quite practical.

Results

We trained two separate SVMs, one to recognize acceptor splice sites and one to recognize donor splice sites, and ran the resulting SVM on the test data. The number of training examples was varied gradually, for which the corresponding test set accuracy is plotted below (see Fig. 4). As expected, as we increased the number of training examples, the test set accuracy rapidly increased and then leveled off. For the largest number of training examples experimented on (1,875 examples), we achieved a test set accuracy of about 99.9% for both the donor and the acceptor models. In general, the acceptor model performed better than the donor model. One possible explanation for this is that since the donor SVM used considerably more features, it was more prone to over-fitting for low training set sizes, potentially leading to a higher generalization error. Overall, however, the results were very impressive and suggest that this approach to splice-site recognition is an extremely fruitful avenue of exploration.

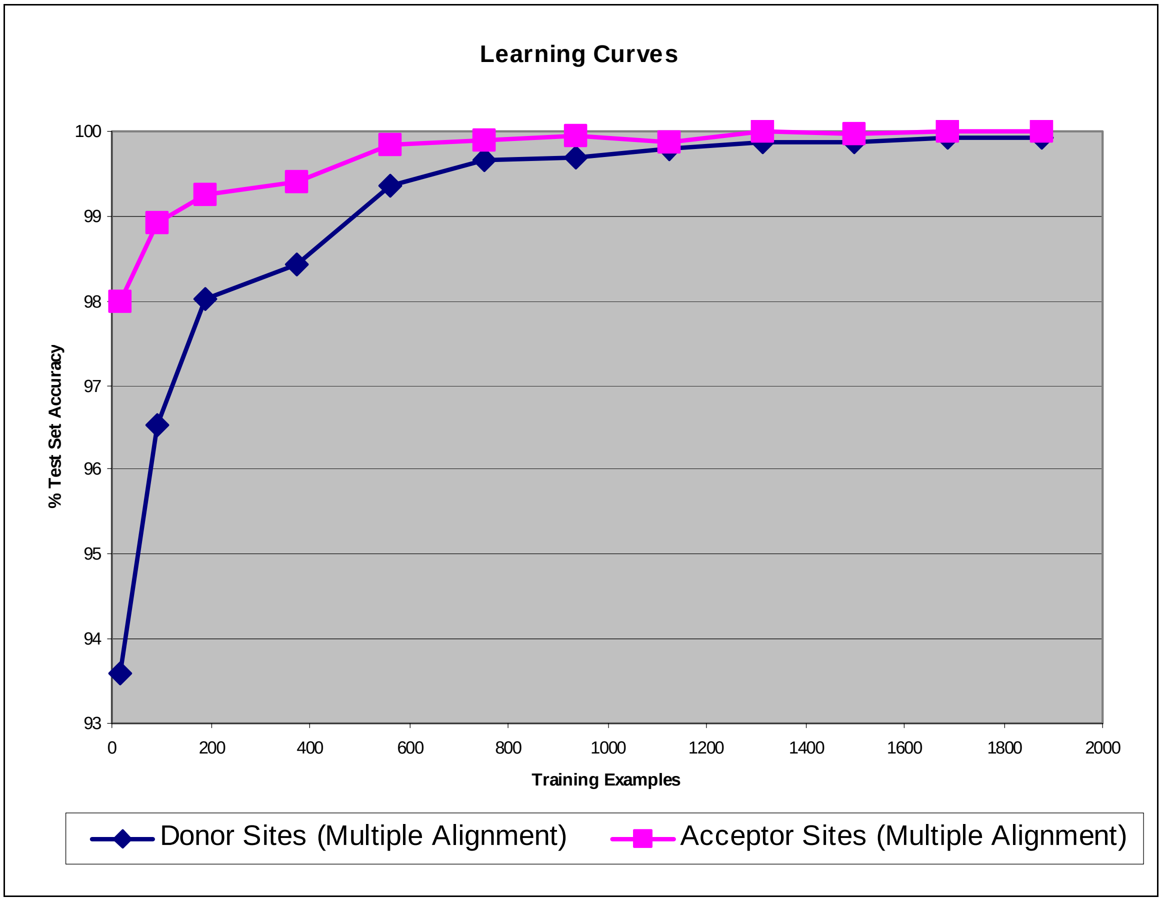

Fig 4. Learning Curve Results