On Selecting Features from Splice Junctions: An Analysis Using Information Theoretic and Machine Learning Approaches

Christina L. Zheng[1]
czheng@sdsc.edu
Virginia R. de Sa[2]
desa@cogsci.ucsd.edu
Michael Gribskov[1]
gribskov@sdsc.edu
T. Murlidharan Nair[1*]
nair@sdsc.edu

[1] San Diego Supercomputer Center
[2] Department of Cognitive Science, University of California, San Diego, 9500 Gilman Dr., La Jolla CA, 92093 USA

Abstract

The computational recognition of precise splice junctions is a challenge faced in the analysis of newly sequenced genomes. This is challenging due to the fact that the distribution of sequence patterns in these regions is not always distinct. Our objective is to understand the sequence signatures at the splice junctions, not simply to create an artificial recognition system. We use a combination of a neural network based calliper randomization approach and an information theoretic based feature selection approach for this purpose. This has been done in an effort to understand regions that harbor information content and to extract features relevant for the prediction of splice junctions. The analysis using the neural network based calliper randomization approach revealed regions important in the internal representation of the network model. The calliper approach captured both correlated as well as independently important features. The feature selection approach captures features that are independently informative. The two different methods can capture features with different properties. Comparative analysis of the results using both the methods help to infer about the kind of information present in the region.

Keywords: calliper randomization, splicing, information theory, feature selection

1 Introduction

Eukaryotic genes coding for proteins usually contain introns or non-coding DNA. Splice junctions are precise points on DNA that serve as chopping points to remove non-coding DNA by a process known as splicing during which the introns are removed, and the exonic sequences on the left are joined to the exonic sequences on the right. This process is facilitated by the assembly of a complex machinery called the spliceosome [9]. Analysis of the intron-exon structure of eukaryotic genes revealed that introns characteristically begin with GU and end with AG nucleotides. Further analysis has revealed an AG/GURAGU consensus for the donor splice sites and a polypyrimidine tract followed by a CAG/A consensus for the acceptor splice sites [18, 23]. However, use of the consensus sequences for the prediction of splice sites does not provide reliable results. This may be due to the fact that almost none of the actual sites exactly fit the consensus and not all sites are of equal strength. The departure of a site from its consensus is not fully understood and may be thought to contribute towards the multiplicity of signals [20]. The co-encryption of multiple signals within the same sequences (triplet code, DNA shape code, chromatin code, gene splicing code etc.) diffuses the distinctiveness of any one signal. The complexity of signals encoded within sequences greatly increase when there is an

*Corresponding author

interaction between codes. This is seen in the case of the interaction of the gene splicing code with the triplet code. Sequences flanking the splice junction are biased in a manner to satisfy both the splicing signal requirement as well as the encryption of codons [5].

Molecular events that occur during the process of splicing are aided by the deciphering of these complex signals encoded in sequences. Understanding the process of signal encryption is of paramount importance in improving gene prediction capabilities, which is still only at about 20% accuracy at the whole-gene level [22]. These predictions rely heavily on the ability to predict gene boundaries. However, the identification of true splice sites is difficult due to the presence of a number of pseudo-splice sites (i.e. sites that match the consensus but are not actually used). Predictions can be improved by developing methods that help in extracting specific sequence signatures occurring at the boundaries of split genes as well as by differentiating these signals from other overlapping ones.

In order to extract these overlapping signals, it is important that they be analyzed over large windows of the sequence. The window must be large enough to contain all possible dependencies. Machine learning approaches such as neural networks can use large windows and decide by a process of iterative learning which positions are important in a given sequence. Neural networks are capable of extracting second and higher order correlations efficiently (which is not possible using simpler statistical methods [16]). However, one main drawback of neural network algorithms is that they provide little or no insight into what they are recognizing and are incomprehensible. Understanding the features involved in imparting knowledge to the network is one approach towards making neural networks more comprehensible, and is useful in extracting other potentially important but as yet unknown signals. Towards this a calliper randomization approach was first proposed by Nair *et al.* [10, 11]. A subsequent approach towards measuring the information content of a trained network have been to introduce "holes" in the input layer [14]. This approach has the disadvantage that it completely ignores the contribution of the signal from that position as opposed to the calliper approach, which helps to establish the relative importance of different nucleotides in that position. The calliper approach can also be used to evaluate the importance of a directed signal by introducing a specific sequence in a calliper window.

In this paper, splice junctions have been analyzed using two approaches (viz. neural networks and feature selection) that have different underlying theoretical basis, to capture features involved in their internal representation, and to understand their importance as splicing signatures. The first method consists of a neural network based calliper randomization approach [11] which helps in capturing sequence features involved in the recognition process of the black-box model. The calliper randomization approach also contributes toward the comprehensibility of the network model by revealing features that the network concentrates on during the process of learning. The second approach consists of an information theoretic feature selection method which helps in understanding the concentration of information associated within the region [7]. The method has the advantage that, besides being tolerant to inconsistencies in the training data, it is effective in eliminating both irrelevant and redundant features [7]. Our current implementation of the feature selection method evaluates the importance of each nucleotide independently and does not look at the predictive power of higher order interactions between nucleotides. This complements the machine learning approach, which does capture higher order interactions. We have tried to understand the type of information present at the splice junctions by combining the results of the analysis using neural networks with those obtained using an information theoretic approach. Other approaches thus far have not tried to distinguish the type of information encoded but rather used neural networks for predicting splice sites [2, 6]. While the two methods are not combined per se, comparative evaluation of the individual outputs helps to reveal the complexity of information encoded.

2 Materials and Method

2.1 Data

Exon/intron (E/I) sequences and intron/exon (I/E) sequences were collected from a data set at `http://industry.ebi.ac.uk/~thanaraj/MouseDataset.out` [19]. This is a curated set of data containing experimentally verified exon/intron and intron/exon boundary information. The data set also includes detailed annotation of start and end positions for untranslated exons, introns and exons. For our simulations 50 nucleotides flanking the E/I and I/E boundaries for each full-length sequence were extracted to generate the data sets. The training space for the E/I and I/E constituted 375 different sequences each. Fifteen hundred random sequences of the same length were generated for each of the sets, which reflected the marginal probability distribution contained in the E/I and I/E sequences respectively. This was then divided into training, test and validation sets for E/I and I/E. Each set contained 125 E/I or I/E sequences and its corresponding 500 random sequences. The random sequences were interspersed with the corresponding E/I and I/E sequences in a ratio of 4:1. The sequences were presented to the network by coding them in binary (A=0001; C=1000; G=0010; T=0100). The target to I/E and E/I sequence were coded as 1 and 0 for a random sequence.

2.2 Neural Network Modeling

Neural networks with different architectures were trained to recognize and classify the sequences. Separate networks were built to recognize the E/I and I/E boundaries. The network architectures consisted of 404 neurons in the input layer (sequence length times 4). The number of neurons in the hidden layer was optimized by training the networks with 3,5,7,9, and 15 neurons, and the network with the optimal performance on the test set was chosen. The number of neurons in the output layer was 1. The networks were trained using the back-propagation algorithm [17]. The training constitutes of a forward pass and a reverse pass. The updating of the weight connections between the neurons is done during the reverse pass, by propagating the errors through the weights. The objective function being minimized is the summed squared error and is defined as

$$E = \sum_{i,j} (t_{ij} - n_{ij})^2$$

where i ranges over the set of input patterns and j ranges over the set of output neurons. t_{ij} is the target corresponding to the I/E or E/I training set and $n_{ij}$is the corresponding network-calculated output of the jth neuron in the output layer for the ith pattern. Details of the algorithm can be found in several articles and books [1].

Optimal network architecture was determined by measuring the performance in terms of specificity, selectivity and the Mathews correlation coefficient (MC) [10] using the following:

$$\text{Sensitivity} = 100 \times \frac{tp}{tp + fn}$$

$$\text{Specificity} = 100 \times \frac{tn}{tn + fp}$$

$$\text{Mathews Correlation coefficient} = \frac{(tp \times tn) - (fp \times fn)}{\sqrt{(tp + fp) \times (fp + tn) \times (tn + fn) \times (fn + tp)}}$$

where *tp* is true positive, which are the I/E or E/I boundaries correctly predicted, *tn* is true negative, these are random sequences predicted as random, *fp* is false positive, these are random sequences predicted as I/E or E/I boundaries and *fn* is false negative, these are I/E or E/I predicted as random sequence.

2.3 Calliper Randomization: Understanding the Recognition Process of a Neural Network

A trained network can be manipulated to extract important features of the modeled region. In earlier work, Nair *et al.* [11, 12] manipulated neural networks to answer the question, are the regions that impart knowledge to a network the same regions that are also biologically important. This was achieved by presenting a trained network with randomized calliper window inputs of the true sequence. Randomized calliper window is any region m through n in the input sequence that has been randomized. The trained network lost its prediction capability when regions that were the main source of knowledge were randomized. Results also revealed that regions imparting knowledge to the network were the same ones that were biologically important and that such an approach could be used as a general approach to discover functionally important regions. The error function for this analysis is modified to reflect the error associated with regions being randomized and takes the following form:

$$\tilde{E}(m,n) = \sum_{i,j} (t_{ij} - \tilde{n}(m,n)_{ij})^2$$

where $\tilde{E}(m,n)$ is the error corresponding to the randomized calliper window encompassing the sequence at positions m through n. The index i range over the set of input patterns and j ranges over the set of output neurons. t_{ij} is the target and $\tilde{n}(m,n)_{ij}$ is the network-calculated output of the jth neuron in the output layer when the ith pattern is presented (randomized at positions m through n). The region corresponding to max $\tilde{E}(m,n)$ underlines the region involved in imparting maximum knowledge to the network during the learning process. This approach can be thought of as a neural network based feature selection method.

2.4 Information Theory Based Feature Analysis

The mutual information I between a category (or class) and a particular input dimension in the problem is a measure of how useful the input dimension is for determining the class. We applied this idea to the problem of finding splice junctions. Which nucleotide positions carry the most information about whether there is an I/E boundary? And which carry the most information about whether there is an E/I boundary? We ranked each nucleotide position using the mutual information between each sequence position and the class (I/E vs not I/E and E/I vs not E/I) [4]. The equations for the I/E case are shown below (those for E/I are analogous)

$$\begin{aligned}
&I(f_i;C) = H(C) - H(C|f_i) \\
&= \sum_{C=IE,notIE} -P(C)\log P(C) + \sum_{C=IE,notIE} \sum_{f_i=A,C,T,G} P(C,f_i)\log P(C|f_i) \\
&= \sum_{C=IE,notIE} \sum_{f_i=A,C,T,G} -P(C,f_i)\log P(C) + \sum_{C=IE,notIE} \sum_{f_i=A,C,T,G} P(C,f_i)\log P(C|f_i) \\
&= \sum_{C=IE,notIE} \sum_{f_i=A,C,T,G} P(f_i)P(C|f_i)\log \frac{P(C|f_i)}{P(C)} \\
&= \sum_{C=IE,notIE} \sum_{f_i=A,C,T,G} P(F_i=f_i)KL(P(C|F_i=f_i),P(C))
\end{aligned}$$

(which is the same as the Koller-Sahami [7] feature selection algorithm with $K = 0$) where f_i represents the nucleotide (A,C,T,G) at position i and C represents the class (I/E boundary, not I/E boundary) and $KL(p,q)$ is the Kullback-Leibler divergence between probability densities p and q. The probabilities are estimated from frequencies in the dataset. This algorithm picks out the input features that carry the most information about the class. The algorithm evaluates the importance of each nucleotide along the boundaries. We applied the algorithm separately for the intron/exon boundary and the exon/intron boundary.

3 Results and Discussion

The best architectures of the neural networks in both cases (I/E and E/I) were the ones with 404 neurons in the input layer, 7 neurons in the hidden layer and 1 neuron in the output layer (Figure 1). The choice of the best architecture was made to circumvent the problem of memorization by over parameterization, which results from too many neurons in the hidden layer, and to obtain the optimal predictor by analyzing the performance of the network with fewer neurons in the hidden layer. The momentum term was optimized at 0.7 and was fixed throughout the training while the learning rate was set at 0.4 and was decreased during the training process. Early stopping was used while training the network, wherein, after every epoch of training, which corresponds to the presentation of the entire training set once to the network, the weights were extracted and its prediction accuracy was determined. This was done by using these weights to determine the networks performance on a set of data (validation set) that was not presented to it at the time of training. This process helps in capturing the weights with maximum generalization capability. The weights corresponding to the minimum error for the validation set was taken as optimal and used for further analysis. A network output between 1 and 0.5 suggested that the sequence was a true boundary (I/E or E/I), while an output between 0.5 and 0 suggested that the sequence was a random one. The results of the predictions of the neural nets with the optimum architecture are given in Table 1(a) and (b).

Table 1: (a) Performance of the network with different architectures for the I/E model. (b) Performance of the network with different architectures for the E/I model.
(a) Intron/Exon

Hidden Neurons	True Positives	True Negatives	False Positives	False Negatives	%accuracy/specificity	%coverage/sensitivity	MC
3	109	478	22	16	93.92	87.20	0.813
5	99	492	8	26	94.56	79.20	0.824
7	**113**	**487**	**13**	**12**	**96.00**	**90.40**	**0.875**
9	110	485	15	15	95.20	88.00	0.800
15	111	471	29	14	93.12	88.80	0.817

(b) Exon/Intron

Hidden Neurons	True Positives	True Negatives	False Positives	False Negatives	%accuracy/specificity	%coverage/sensitivity	MC
3	108	484	16	17	94.72	86.40	0.834
5	105	479	21	20	93.44	84.00	0.795
7	**116**	**484**	**16**	**9**	**96.00**	**92.80**	**0.878**
9	101	485	15	24	93.76	80.80	0.800
15	104	485	15	21	94.24	83.20	0.817

After training, the network architecture with the best internal representation (7 hidden neurons in both cases) was used in the calliper randomization analysis of these splice junction regions. Calliper randomization helps to determine the regions in the sequences that are important in imparting knowledge to the network. Calliper randomization was done systematically using calliper lengths of 1-5, 10 and 20. The calliper window was measured in nucleotides. For this stage, the third of the dataset (test set) not used for training or early stopping was used. The results readily reveal that the network loses prediction capability when the GU at the beginning of the intron and AG at the end of the intron are randomized. These results indicate that these are the dominant signals involved in the recognition process of the E/I and I/E boundaries respectively. These are well known signals at the splice junctions, and our interest was in extracting other potentially useful signals that could also be

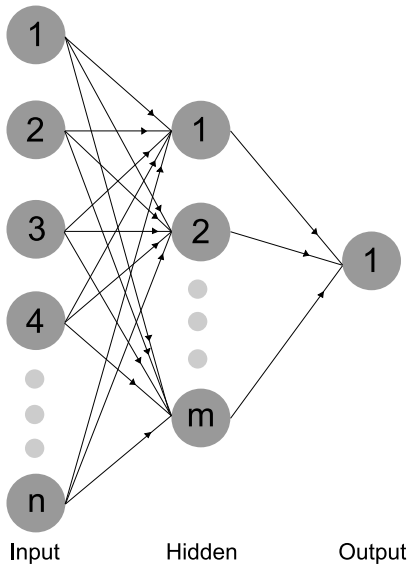

Figure 1: Architecture of the three-layered feed-forward neural network used in all simulations. The circles represent the artificial neurons that integrate input from the proceeding layer and propagate the signal to the next layer. The optimum architectures for the E/I and I/E boundaries were 404 input neurons, 7 hidden neurons, and 1 output neuron. The optimum architecture for the intron side of the I/E boundary (minus the dominant AG signals) was 126 input neurons, 5 hidden neurons, and 1 output neuron.

important in differentiating a true splice site from a false one. Towards this effort, the performance of the neural net was analyzed when different regions were randomized to different extents with varying calliper widths. Most of the signals in the case of the I/E was shown to be harbored within the intron side of the junction. I/E randomization analysis also reveals (Figure 2 I/E(a-d)) that in addition to the dominant signal, there are other nucleotides that also play a crucial role in the recognition process as indicated by the increase in error upon increasing the calliper randomized widths. With the E/I boundary there appears to be a reasonable symmetry about the GU nucleotides in the distribution of splicing signature. This is demonstrated in the error profile obtained by the calliper randomization approach (Figure 2 E/I(a-d)). In both cases the distribution of signals were biased towards the intron side, though more so in the case of the I/E boundary. Our results are consistent with the recent analysis of the sequence features by Lim and Burge [8].

In order to narrow down and capture other potentially weak splicing signals that might have been masked by the dominant signals, networks were trained after the dominant signals were eliminated (AG in the case of I/E and GU in the case of E/I). Separate networks were trained to capture the signals associated with the intron and the exon side of I/E and E/I. We were only able to build models for the intron side of I/E. None of the architectures we tried successfully captured the internal representations associated with the exon side of I/E and E/I as well as the intron side of E/I. This agrees with our earlier analysis, which revealed that most of the signal was biased towards the intron side of the I/E boundary. This also becomes relevant as the branch point and the pyrimidine rich region are positioned in the intron side of the I/E junction. The architecture of the network for the intron side of I/E was optimized to 126 neurons in the input layer, 5 neurons in the hidden layer and 1 neuron in the output layer. The modeling parameters for momentum and learning were set to 0.7 and 0.6 respectively. Early stopping was once again used to obtain the optimum weights. The performance of the network decreased when the dominant signals were removed. The overall prediction was 82%. Calliper randomization was again applied to the trained networks. Even with the dominant signals removed the calliper randomization analysis of the intron side of the I/E junction revealed that most of the signals were concentrated near the boundary. Figure 3 (a-g) reveals the relative importance

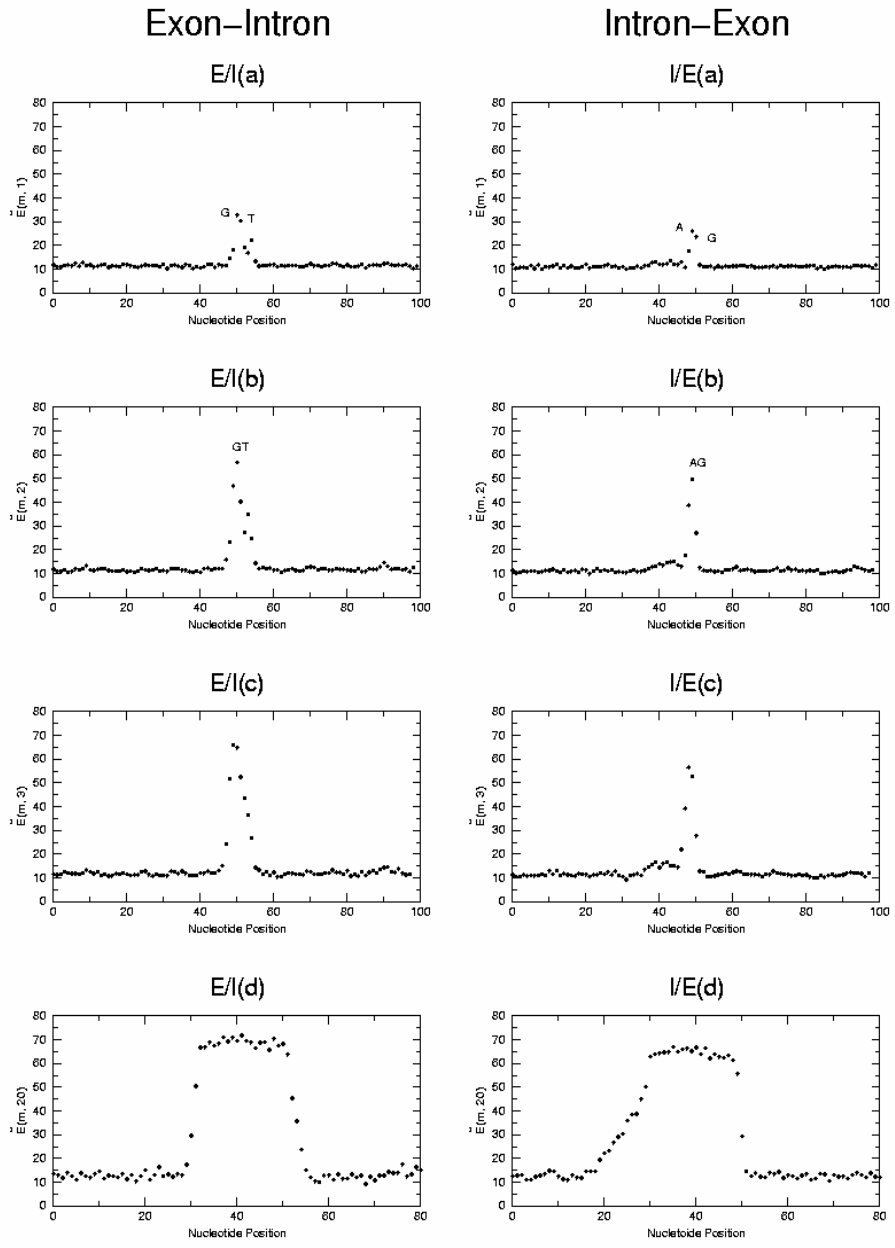

Figure 2: Neural network based calliper randomization. Performance of the neural net when varying calliper widths was randomized: (a), 1; (b), 2; (c), 3; and (d), 20. For both E/I boundaries (left) and I/E boundaries (right). $\tilde{E}(m, n)$ is the sum-squared error for the specific calliper width.

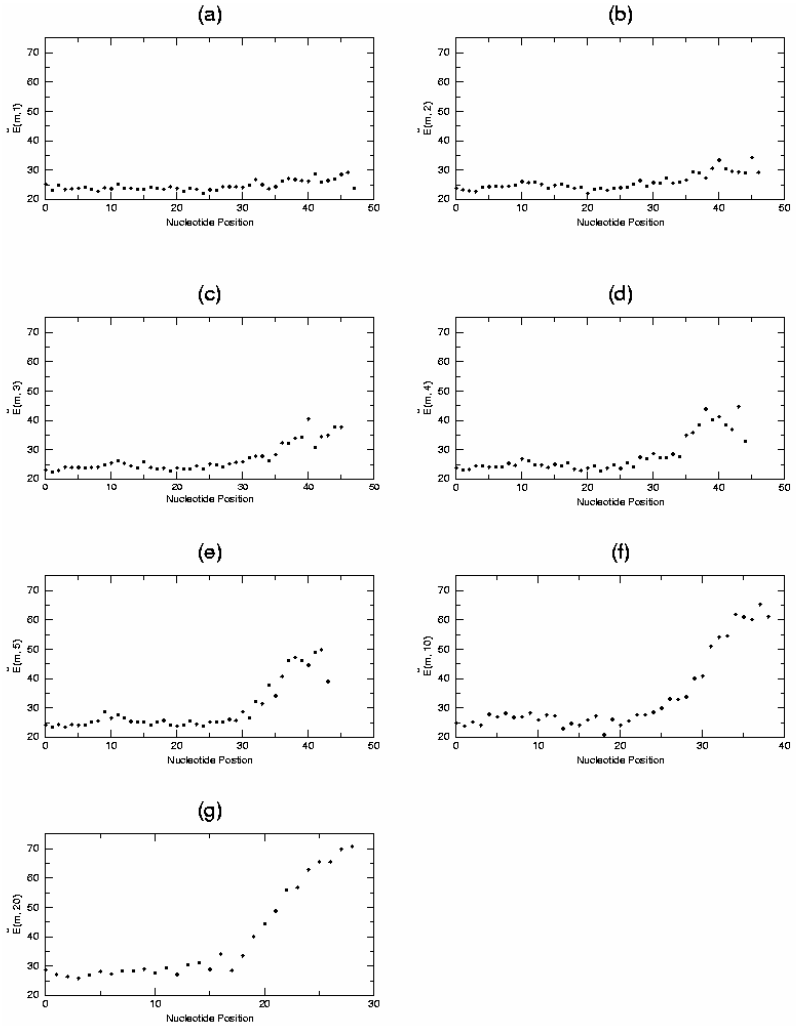

Figure 3: Neural network based calliper randomization of the intron region of the intron-exon boundary. Performance of the network when varying calliper widths were randomized:(a),1; (b),2; (c),3; (d),4; (e),5; (f),10; (g),20. $\tilde{E}(m,n)$ is the summed squared error for the specific calliper width.

of nucleotides at these positions. It is worth noting that when randomizing a single nucleotide at any position, the prediction capability is not affected. Loss in prediction capability becomes more pronounced when 2 or more nucleotides are randomized. These results further point to the dominance of signals on the intron side of the splice site for the I/E boundary.

In addition to calliper randomization, each nucleotide's positional importance was also determined using a feature selection method [7]. This method picks out the input features carrying the most information about the class. This algorithm has previously been applied to versions of the UCI splice dataset [3, 7] and in DNA microarray analysis [21], however in all the above cases, the method was used to eliminate features with insufficient information to offset the increased search space. Here we use the algorithm directly to investigate the relative benefit of each nucleotide in an E/I and I/E boundary, respectively. From figure 4, we can see that the feature selection method once again revealed the relative importance of the dominant GU and AG signals at the E/I and I/E boundaries, respectively. These results also show the importance of the nucleotides near the boundary of the I/E boundary. In addition Figure 4 depicts the commonalities independently arrived at by both the feature selection method and the calliper randomization with a window of one. It is interesting to

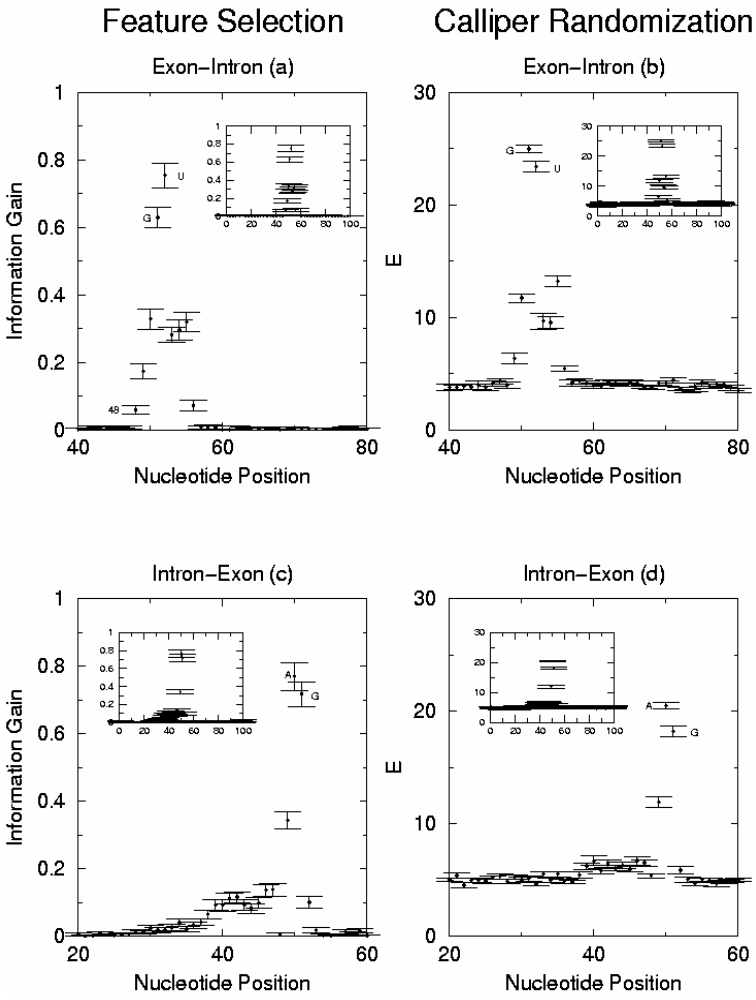

Figure 4: Comparison of feature selection with calliper randomization with calliper width of one. Error bars for the information analysis feature selection method represent bootstrapped standard deviations obtained with 500 bootstrap samples. The error bars for the calliper randomization analysis represent bootstrapped standard deviations obtained with 125 bootstrap samples using the same optimized network architectures.

make a comparative analysis of the results obtained by these two methods because both calculate the importance of individual nucleotides, but with different criteria. This then leads to a few disparate yet intriguing findings. For instance, feature selection ranked the importance of U above that of G while the calliper randomization method ranked the G above the U for the E/I boundary. Furthermore, in the I/E boundary, the A and G positions were found to impart significantly different amounts of information with the calliper randomization approach, while the feature selection approach did not rank these two positions with a significant difference. Overall the feature selection approach revealed a greater number of nucleotides to be of importance than the calliper randomization approach.

The differences found with these two very different approaches point to the importance of using these two in conjunction to study molecular sequences. Molecular sequences are known to harbor both correlated [15] and non-correlated features. Neural networks are capable of capturing such correlations while the feature selection approach we used determined independently important features. However, with lots of data the information theoretic feature selection algorithm may be used to look at higher order relationship. Presenting a randomized calliper window to a trained network will result in an

increased error if the nucleotide(s) within the window are important by themselves, *or* even if they are important due to their correlations or higher order interactions with other nucleotides (including those not in the calliper window). Looking at the neural network calliper randomization results alone, it is difficult to determine which (or both) of the two reasons are driving the error. Differences in the relative importance of individual nucleotides could indicate the importance of that nucleotide position in predictive higher-order interactions. The discrepancy found with the GU positions, could indicate that G has more of an importance in higher-order interactions than U (or simply that U is highly correlated with G and has been down-played by the neural network). As mentioned previously, the feature selection method was able to pick up a greater number of important nucleotides than the calliper randomization approach, one of which being position 48 in the E/I boundary. The fact that the calliper randomization approach did not pick position 48 as being useful may indicate a limit in the resolution of the neural network approach, or could indicate that position 48 is highly correlated with (and possibly less informative than) another position and that the neural network has chosen to ignore 48 and listen to the other signal.

In conclusion, the results presented provide evidence that valuable information can be gained about the nature of signals harbored within sequences by using the results of the two approaches in conjunction. The feature selection approach extracted independently important features based on information theory while the calliper randomization approach captured sequence features with higher order correlations, making the calliper randomization approach an alternative neural network based feature selection method. These results also lead to many unanswered questions about the underlying signal dependencies within splicing signals. More work is necessary to determine the answer to these questions but we feel that the combination of these approaches will not only be fruitful in determining useful signatures within the splice junctions but may also be used as a more general approach in studying other regulatory regions. Further, using the calliper approach as a general method to understand regions involved to imparting knowledge to a learning machine and using an ensemble of complementary learning methods [13] would help reveal more features than using any one method.

Acknowledgments

This work is supported by the National Cancer Institute funding CA 88351 and NSF career grant 0133996 to VRd. This is also part of the initial work for a proposed machine learning project (LEGEND).

References

[1] Bishop, C.M., *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[2] Brunak, S., Engelbrecht, J., and Knudsen, S., Prediction of human mRNA donor and acceptor sites from the DNA sequence, *J. Mol. Biol.*, 220:49–65, 1991.

[3] Caruana, R. and de Sa, V.R., Using feature selection to find inputs that work better as outputs, *The 8th International Conference on Artificial Neural Networks (ICANN 98)*, 299–304, 1998.

[4] Cover, T.M. and Thomas, J.A., *Elements of Information Theory*, John Wiley & Sons, 1991.

[5] Gelfand, M.S., Statistical analysis and prediction of the exonic structure of human genes, *Journal of Molecular Evolution*, 35:239–252, 1992.

[6] Hebsgaard, S.M., Korning, P.G., Tolstrup, N., Engelbrecht, J., Rouze, P., and Brunak, S., Splice site prediction in *Arabidopsis thaliana* pre-mRNA by combining local and global sequence information, *Nucleic Acids Res.*, 24:3439–3452, 1996.

[7] Koller, D. and Sahami, M., Toward optimal feature selection, *International Conference on Machine Learning*, 284–292, 1996.

[8] Lim, L.P. and Burge, C.B., A computational analysis of sequence features involved in recognition of short introns, *Proc. Natl. Acad. Sci. USA*, 98:11193–11198, 2001.

[9] Maniatis, T. and Reed, R., The role of small nuclear ribonucleoprotein particles in pre-mRNA splicing, *Nature*, 325:673–678, 1987.

[10] Matthews, B.W., Comparison of the predicted and observed secondary structure of T4 phage lysozyme, *Biochim. Biophys. Acta.*, 405:442–451, 1975.

[11] Nair, T.M., Calliper randomization: an artificial neural network based analysis of *E. coli* ribosome binding sites, *J. Biomol. Struct. Dyn.*, 15:611–617, 1997.

[12] Nair, T.M., Tambe, S.S., and Kulkarni, B.D., Application of artificial neural networks for prokaryotic transcription terminator prediction, *FEBS Lett*, 346:273–277, 1994.

[13] Paredis, J., Decision trees and neural nets: two complementary representations for learning, *Fourth International Conference on Neural Networks and their Applications*, Nimes, France, EC2, 1991.

[14] Pedersen, A.G. and Nielsen, H., Neural network prediction of translation initiation sites in eukaryotes: perspectives for EST and genome analysis, *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 5:226–233, 1997.

[15] Peng, C.K., Buldyrev, S.V., Goldberger, A.L., Havlin, S., Sciortino, F., Simons, M., and Stanley, H.E., Long-range correlations in nucleotide sequences, *Nature*, 356:168–170, 1992.

[16] Reddy, B.V. and Pandit, M.W., A statistical analytical approach to decipher information from biological sequences: application to murine splice-site analysis and prediction, *J. Biomol. Struct. Dyn.*, 12:785–801, 1995.

[17] Rumelhart, D., Hinton, G., and Williams, R., Learning representation by backpropagating errors, *Nature*, 323:533–536, 1986.

[18] Sharp, P.A., Splicing of messenger RNA precursors, *Science*, 235:766–771, 1987.

[19] Thanaraj, T.A., A clean data set of EST-confirmed splice sites from *Homo sapiens* and standards for clean-up procedures, *Nucleic Acids Res.*, 27:2627–2637, 1999.

[20] Trifonov, E.N., Interfering contexts of regulatory sequence elements, *CABIOS*, 12:423–429, 1995.

[21] Xing, E.P. and Karp, R.M., CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts, *Bioinformatics*, 17 Suppl 1:S306–S315, 2001.

[22] Zhang, M.Q., Computational prediction of eukaryotic protein-coding genes, *Nat. Rev. Genet.*, 3:698–709, 2002.

[23] Zhang, M.Q., Statistical features of human exons and their flanking regions, *Hum. Mol. Genet.*, 7:919–932, 1998.