

Bayesian prediction of tissue-regulated splicing using RNA sequence and cellular context

Hui Yuan Xiong^{1,†}, Yoseph Barash^{1,2,†} and Brendan J. Frey^{1,2,*}

¹Department of Electrical and Computer Engineering, University of Toronto, Toronto, M5S3G4 and ²Banting and Best Department of Medical Research, Centre of Cellular and Biomolecular Research, University of Toronto, Toronto, M5S3E1, Canada

Associate Editor: Ivo Hofacker

ABSTRACT

Motivation: Alternative splicing is a major contributor to cellular diversity in mammalian tissues and relates to many human diseases. An important goal in understanding this phenomenon is to infer a ‘splicing code’ that predicts how splicing is regulated in different cell types by features derived from RNA, DNA and epigenetic modifiers.

Methods: We formulate the assembly of a splicing code as a problem of statistical inference and introduce a Bayesian method that uses an adaptively selected number of hidden variables to combine subgroups of features into a network, allows different tissues to share feature subgroups and uses a Gibbs sampler to hedge predictions and ascertain the statistical significance of identified features.

Results: Using data for 3665 cassette exons, 1014 RNA features and 4 tissue types derived from 27 mouse tissues (<http://genes.toronto.edu/wasp>), we benchmarked several methods. Our method outperforms all others, and achieves relative improvements of 52% in splicing code quality and up to 22% in classification error, compared with the state of the art. Novel combinations of regulatory features and novel combinations of tissues that share feature subgroups were identified using our method.

Contact: frey@psi.toronto.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on May 3, 2011; revised on July 7, 2011; accepted on July 23, 2011

1 INTRODUCTION

Alternative splicing enables individual genes to generate different transcripts, by selectively including or excluding RNA sequences. High-throughput sequencing shows that over 90% of human genes are alternatively spliced, mostly in a tissue-dependent manner (Pan *et al.*, 2008; Wang *et al.*, 2008). The importance of alternative splicing is evidenced by numerous examples of genes whose functions are switched depending on which alternative transcript (isoform) is expressed, plus analyses showing that a large fraction of human disease mutations affect splice site selection (Wang and Cooper, 2007). These results underscore the importance of accounting for splicing regulation when modeling gene expression.

For over two decades, researchers have sought to define splicing regulatory models in the form of a mapping from genomic features

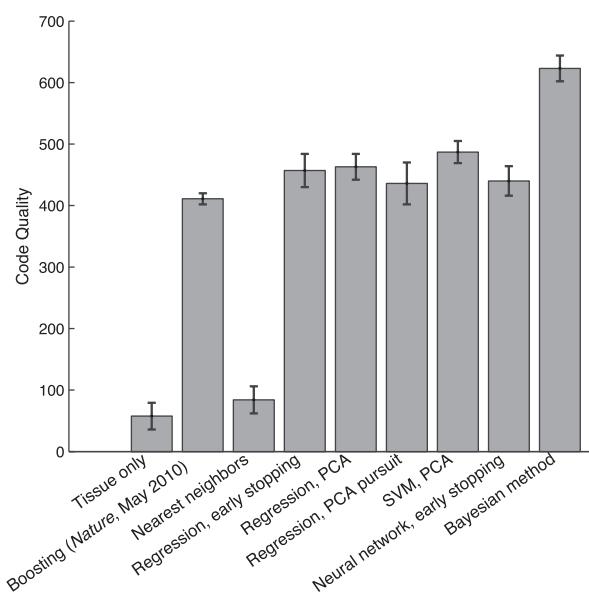


Fig. 1. A comparison of methods for predicting tissue-regulated splicing in mouse, using the metric of ‘code quality’ measured in bits (see main text).

and cellular conditions to predicted abundances of alternative transcripts (Blencowe, 2006; Chan and Black, 1997; Hartmann and Valcárcel, 2009; Lim and Sharp, 1998). In the words of Wang and Burge (2008), ‘An important long-term goal in the community is to determine a ‘splicing code’: A set of rules that can predict the splicing pattern of any primary transcript sequence’.

Recently, we described the assembly of a mouse splicing code that can be used to predict the regulatory properties of previously uncharacterized exons, predict regions in the unspliced transcript that when mutated led to changes in splicing patterns, and reveal novel regulatory mechanisms (Barash *et al.*, 2010a). Our purpose here is to (i) describe a dataset and evaluation method that researchers can use to improve and extend splicing codes; (ii) introduce a Bayesian technique that uses hidden variables to model relationships between features and splicing changes within a network; and (iii) benchmark several machine learning methods.

Figure 1 compares our Bayesian technique to several other methods in terms of ‘code quality’, which is the amount of genome-wide splicing variability accounted for by RNA sequence features (see below). The result labeled ‘tissue only’ indicates how much splicing variability is accounted for by tissue type, i.e. that different tissues have different overall levels of splicing. Most of the methods

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

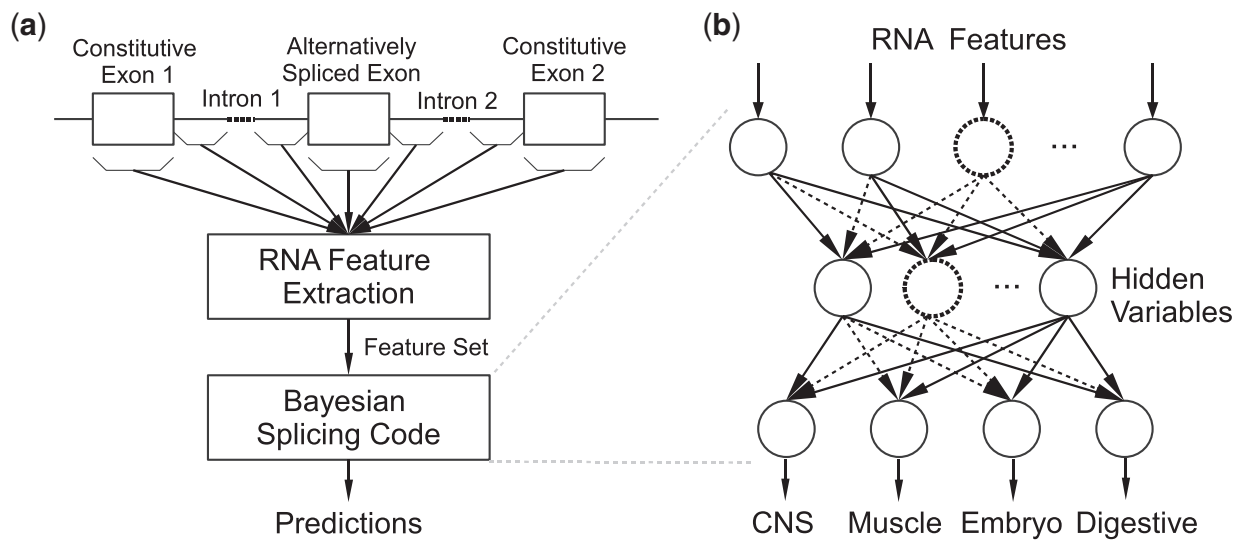


Fig. 2. (a) A 'splicing code' uses features extracted from primary RNA to predict splicing patterns. (b) A Bayesian neural network uses RNA features (top) to determine hidden variables (middle) that are used to predict tissue-dependent splicing (bottom). Inference involves adding and deleting features, hidden variables and connections (solid and dashed lines).

we examined were able to account for significantly more variability by relating splicing levels with features in the unspliced RNA, such as potential binding sites of factors that regulate splicing. The result labeled 'boosting' was obtained in our original work on the splicing code (Barash *et al.*, 2010a). A simple nearest neighbours method, which uses previously profiled exons with similar sequence features to make predictions, accounts for very little additional variability, if any. This result suggests that the features governing splicing operate in a combinatorial fashion. Methods based on principal components analysis (PCA), multinomial regression and regularized neural networks perform similarly to the original method. The SVM outperforms the original method. The Bayesian technique described below performs significantly better than all other techniques.

Next, we describe our publicly available dataset, review a log-likelihood-based measure of splicing code quality and explain how inference of the splicing code can be formulated as statistical inference. We describe our Bayesian technique and several other approaches, before providing details about how they compare in terms of code quality and classification accuracy. We investigate properties of the Bayesian method such as how it benefits by sharing hidden variables when making predictions for different tissues, and how the number of selected hidden variables differs between tissues. We examine novel combinations of regulatory features that are elucidated by our method and conclude by discussing promising directions for further research in the area.

2 LEARNING REGULATORY MODELS

We are not really interested in exactly how only a single transcript is spliced at a particular point in time within a particular cell. This knowledge would neither provide an understanding of how splicing works nor enable us to predict what would happen under different conditions. A more compelling scientific goal is to infer a network model that summarizes related splicing patterns, accounts for variability in cellular conditions and how they influence splicing,

and predicts how splicing will occur for novel RNA sequences. We expect such a model to provide predictions in the form of probabilities, because we can measure cellular conditions in only an approximate manner and also because the underlying biochemical processes are stochastic. By peering inside the inferred network, researchers can predict regulatory mechanisms.

Taking a predictive modeling approach, we seek a network model that can be applied to a comprehensive set of RNA sequences to make accurate probabilistic predictions for how splicing will occur under different cellular conditions, using features derived from RNA, DNA, epigenetic modifiers, etc. Here, we use only RNA features (Fig. 2a). This can be viewed as a problem of statistical inference, where we assume that the set of RNA sequences and cellular conditions is an unbiased sample from a distribution of interest, such as all alternatively spliced exons in a group of mouse tissues. Given a set of corresponding RNA sequences, cellular conditions and splicing patterns, statistical inference is used to infer a predictive model. Previously, we used this approach to produce a model that can make predictions for previously unseen transcripts and verified several novel predicted regulatory mechanisms by mutating sequences in a minigene reporter (Barash *et al.*, 2010a).

Data used to infer the code: Using data from Fagnani *et al.* (2007) and a preprocessing method described in Barash *et al.* (2010b), we generated a new, publicly available evaluation protocol that can be used to infer predictive splicing models and compare different inference methods. RNA feature vectors and splicing patterns for different tissues are provided for several training sets and test sets at <http://genes.toronto.edu/wasp>.

The set of RNA sequences was obtained by mining EST and cDNA libraries and identifying 3665 cassette-type mouse exons, i.e. exons that are sometimes included and sometimes excluded in the spliced transcripts extracted from tissues and cell lines. The fraction of isoforms including each cassette exon was profiled across 27 mouse tissues. Predicting fractions of isoforms is difficult because many unobserved variables contribute to exon-specific regulation.

We found it was easier to predict, for each exon and tissue type, the direction of change of the fraction of isoforms including the exon, relative to other tissues. In Barash *et al.* (2010b), we introduced a preprocessing method that estimates such relative changes while taking into account different noise sources. Splicing changes were estimated for four cellular conditions, approximately corresponding to cells from central nerve system (CNS) tissues, muscle tissues, whole embryos plus embryonic cell lines and digestive tissues. For each exon and tissue type, we generated three real-valued, positive prediction targets q^{inc} , q^{exc} and q^{nc} corresponding to probabilities that the exon is more likely to be included in the given tissue relative to other tissues, more likely to be excluded or more likely to exhibit no change relative to other tissues (Supplementary Fig. S1). These targets need not add up to one; their sum relates to the confidence in the observed splicing pattern and can be estimated using a probability model (Barash *et al.*, 2010b). However, for simplicity, we normalized them in the current dataset s.t. $q^{\text{inc}} + q^{\text{exc}} + q^{\text{nc}} = 1$. In each tissue, $\sim 10\%$ of exons exhibit increased inclusion or exclusion ($q^{\text{inc/exc}} > 0.9$) and the entropy of the mean target distribution is 0.63 bits per tissue.

For each exon in the dataset, 1014 features were extracted from the exon and its two flanking exons, windows of 300 nt of intronic sequence adjacent to those exons, and other regions of the unspliced transcript (Barash *et al.*, 2010a). Introns with < 300 nt were padded with blanks. Examples of features include region-specific counts of short sequence motifs, scores for potential RNA-binding protein binding sites, exon and intron lengths, secondary structure probabilities and whether or not exon inclusion or exclusion introduces a premature termination codon. Feature values may be binary, integer, discrete but non-ordinal, or real-valued, and real-valued features may be sparse or densely distributed (Supplementary Fig. S2). Many groups of features are highly correlated, such as those derived using slightly different literature-curated definitions of binding sites.

The feature vectors form a 3665×1014 matrix, with rows corresponding to exons and columns corresponding to features, and the tissue-dependent targets form a 3665×12 matrix, with each row containing q^{inc} , q^{exc} and q^{nc} for each of the four tissue types.

Training sets and test sets: The number of features is large relative to the number of examples, so methods that are not regularized will likely overfit the data. The situation is made worse by the fact that the targets are sparse, having only ~ 2400 bits of information per tissue. Care needs to be taken to avoid overfitting the model in such a way that generalization is not possible. For example, when gradient descent is used to train a multinomial regression model, the training log-likelihood continues to increase while the test log-likelihood quickly peaks and then rapidly drops below the log-likelihood achieved by a naive guessing scheme (Supplementary Fig. S3). To ensure that reported results are unbiased, we obtain them using held out test data. Also, care was taken to remove redundancies between training cases and test cases by checking for exon sequence similarity (Barash *et al.*, 2010a). We use five-fold cross-validation and to estimate confidence intervals we repeat the procedure six times using different randomly generated data partitions. For each partition, five models are constructed and the corresponding five test set performances are summed together to obtain an unbiased estimate of performance.

A complete set of feature vectors and targets (q 's) are provided for training sets and test sets at <http://genes.toronto.edu/wasp>. In these datasets, for each partition, every exon is used once for testing and four times for training. For each fold and each partition, the preprocessing stage used to convert microarray measurements to targets (Barash *et al.*, 2010b) is constructed independently using only training exons. Then, measurements of test exons are preprocessed to produce the test targets for that partition and fold. This procedure is designed to avoid reporting performance estimates that are biased by using test data to develop the preprocessing stage. Note that the targets for the same exon may differ in different folds.

Measuring code quality using relative log-likelihood: The feature vectors and targets (q 's) described above are used to train a model. For each exon and tissue type, the model outputs predictions in the form of three probabilities p^{inc} , p^{exc} and p^{nc} , that are meant to be similar to the training targets q^{inc} , q^{exc} and q^{nc} . The code quality for tissue t is measured thus (Barash *et al.*, 2010a):

$$\mathcal{H}_t = \sum_{e \in \text{Exons}} \sum_{s \in \{\text{inc}, \text{exc}, \text{nc}\}} q_{t,e}^s \log \left(\frac{p_{t,e}^s}{\bar{q}^s} \right), \quad (1)$$

where $p_{t,e}^s$ and $q_{t,e}^s$ are the model prediction and the target for exon e , splicing change s and tissue t . \bar{q}^s is the average of $q_{t,e}^s$ across all tissues and exons, $\bar{q}^s = \sum_{t,e} q_{t,e}^s / \sum_{t,e} \sum_{s'} q_{t,e}^{s'}$, and corresponds to the prediction made by a naive guesser that ignores the RNA feature vector and tissue type. Note that if $\sum_s q_{t,e}^s > \sum_s q_{t,e'}^s$, then exon e counts more than exon e' toward the code quality, \mathcal{H}_t .

Code quality can be viewed as a difference of two Kullback-Leibler (KL) divergences: $\mathcal{H}_t = \sum_e D_{\text{KL}}(\mathbf{q}_{t,e} \| \bar{\mathbf{q}}) - D_{\text{KL}}(\mathbf{q}_{t,e} \| \mathbf{p}_{t,e})$. These two KL divergences measure, in bits of information, how much the predictions from the model and the naive guesser inform us about the splicing patterns. The difference between them is the amount of additional information provided by the model, beyond naive guessing. When the predictions perfectly match the targets, the highest possible code quality is obtained. It equals the entropy of the targets minus the cross-entropy between the targets and the naive guesser: ~ 2.5 bits/exon. A negative code quality implies that the prediction is worse than naive guessing; this can occur if the predictions are overly confident and sometimes wrong. The net code quality is obtained by summing over tissues: $\mathcal{H} = \sum_{t \in \text{Tissues}} \mathcal{H}_t$.

Code quality can be alternatively interpreted as the improvement in log-likelihood, calculated using partial data counts:

$$\mathcal{H} = \mathcal{L} - \mathcal{L}^{\text{Naive}}, \quad (2)$$

$$\mathcal{L} = \sum_t \sum_e \sum_s q_{t,e}^s \log p_{t,e}^s, \quad \mathcal{L}^{\text{Naive}} = \sum_t \sum_e \sum_s q_{t,e}^s \log \bar{q}^s.$$

Inferring a model influences only the first term, \mathcal{L} , so the code quality \mathcal{H} can be optimized using likelihood-based methods.

3 ALGORITHMS

3.1 Bayesian Neural Network

To account for combinatorial interactions between RNA features, we consider models with hidden variables that are non-linear functions of combinations of subsets of features (Rumelhart *et al.*, 1986). Hidden variables are used to predict tissue-dependent splicing changes, as shown in Figure 2b. Since it is not known beforehand

which features should be connected to each hidden variable, an exact search over all possible models is not feasible. Also, as explained above, overfitting is a concern because the data are limited, sparse and noisy. To address these issues, we take a Bayesian approach (Bishop, 2006; MacKay, 1992; Neal, 1996) where the computational task is to sample from a posterior distribution over models. Predictions for novel test cases are made by averaging the predictions from the sample of models, and important features can be identified by checking to see if they are used more frequently than expected at random under the prior distribution of models.

Model architecture: We use a two layer network (Fig. 2b) that receives as input F RNA features x_1, \dots, x_F and uses them to determine the values of up to N hidden variables h_1, \dots, h_N , which are used to determine the prediction probabilities p_t^{inc} , p_t^{exc} and p_t^{nc} for each tissue t . Parameters are used to account for how input features influence hidden variables and how hidden variables influence prediction probabilities. A parameter value of zero indicates an absent connection. If all parameters connecting a hidden variable to the outputs are zero, then the hidden variable is effectively absent from the network. The algorithm described below is used to search over network structures and parameter values.

The influence of feature x_f on hidden variable h_i is accounted for by the real-valued parameter $w_{f,i}$. The hidden variables process the sum of weighted features using a non-linear sigmoid function:

$$h_i = 1 / (1 + e^{-\sum_{f=1}^F w_{f,i} x_f}). \quad (3)$$

The outputs of the hidden variables are used to compute the prediction probability p_t^s for each splicing pattern s (inclusion, exclusion or no change) and each tissue t as follows:

$$p_t^s = e^{\sum_{i=1}^N v_{t,i}^s h_i} / \left(\sum_{s' \in \{\text{inc}, \text{exc}, \text{nc}\}} e^{\sum_{i=1}^N v_{t,i}^{s'} h_i} \right). \quad (4)$$

Here, the influence of hidden variable h_i on the prediction for splicing change s in tissue t is accounted for by the parameter $v_{t,i}^s$.

Depending on the connectivity (non-zero weights), each feature may be used by more than one hidden variable and each hidden variable may be used to predict splicing in more than one tissue. These properties enable the model to account for combinations of features that are tissue specific or shared across different tissues. To test whether allowing the model to share hidden regulatory variables across tissues is important, we also tried a model without sharing of hidden variables, i.e. where one model was trained for each tissue.

Prior distribution over models: We use a prior distribution that allows flexibility in the number of hidden variables and the connectivity in the network. The prior distribution over each parameter has a ‘spike and slab’ form (Ishwaran and Rao, 2005), which enables connections to be shut off. The input features are connected to hidden variables independently with Bernoulli probability $1 - \alpha$ and the parameters for connected variables have standard normal distributions. The number of hidden variables n_t used to predict splicing in each tissue is Poisson distributed with an expected number of hidden variables λ . Non-zero hidden-to-output parameters have multivariate standard normal distributions. Under the Bernoulli and Poisson priors, the hidden variables are exchangeable and their connections to different input features are independent, as are their connections to different tissues. We truncate the Poisson distribution using a maximum number of allowed hidden variables, N , which controls the sharing of hidden variables between

tissues. If N is large compared with λ , sharing occurs infrequently, whereas if N is small, different tissues are likely to share hidden variables.

Based on initial experiments using validation data, we set $\alpha = 0.1$ to encourage sparse use of RNA features, and we set $\lambda = 10$ and $N = 30$ to encourage moderate sharing of hidden variables and so that on the order of 10 hidden variables are used to predict each tissue. Finally, to facilitate traversing the space of possible models using Gibbs sampling as described below, we discretize the parameters, so that $w_{f,i} \in \{-5.0, -4.9, \dots, 4.9, 5.0\}$ and $v_{t,i}^s \in \{-5.0, -4.8, \dots, 4.8, 5.0\}$. We found that the performance of our method is robust to the above choices (Section 2 in Supplementary Material and Fig. 4).

Markov Chain Monte Carlo sampling: We use Gibbs sampling to sample from the posterior distribution over models. In each iteration, we first sample the $w_{f,i}$ ’s in sequence from their posterior distributions while fixing all other parameters:

$$w_{f,i} \sim P(w_{f,i}) \prod_e \prod_t \prod_s p_{t,e}^s(w, v)^{q_{t,e}^s}. \quad (5)$$

$P(w_{f,i})$ is the spike-and-slab prior distribution over $w_{f,i}$, and the likelihood $p_{t,e}^s(w, v)$ is computed using Equations (3) and (4). Next, for each hidden variable i and tissue t , we jointly sample $v_{t,i} = (v_{t,i}^{\text{inc}}, v_{t,i}^{\text{exc}}, v_{t,i}^{\text{nc}})$ from its posterior distribution:

$$v_{t,i} \sim P(v_{t,i}) \prod_e \prod_t \prod_s p_{t,e}^s(w, v)^{q_{t,e}^s}, \quad (6)$$

where $P(v_{t,i})$ is the spike-and-slab prior on $v_{t,i} = (v_{t,i}^{\text{inc}}, v_{t,i}^{\text{exc}}, v_{t,i}^{\text{nc}})$.

Initially, all parameters are set to zero and in each iteration of Gibbs sampling, the features and hidden variables are processed in random order. The parameters (w ’s and v ’s) are recorded after each iteration up to a maximum number of 2000 iterations. The initial 150 samples are not used when making predictions, because we found that at least that many iterations were needed for mixing (Supplementary Fig. S5). Using the feature vector for a test exon e' , predictions are made using each model in this ensemble of models, and the predictions are averaged together to make a final prediction.

3.2 Other methods included in the benchmark

We examined several popular methods, including the boosting method used in Barash *et al.* (2010a). For methods that use validation data to set a regularization parameter, we set aside 1/4 of the training data for validation using code quality. Using the selected regularization parameter, the model was re-trained using all training data.

The simplest method was k nearest neighbors (Bishop, 2006): For a test exon e' , the q -values corresponding to the k training exons $e \in \mathcal{E}$ whose feature vectors were closest in L2 to the test feature vector were averaged to make a prediction: $p_t^s = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} q_{t,e}^s$. k was chosen using validation data, as described above.

We examined three regularized multinomial regression methods (Bishop, 2006). The first method was early stopping, where the parameters were initialized to small random values and then batch gradient descent with a learning rate of 0.1 was used to adjust the parameters using training data, until the code quality of validation data reached its maximum. The second method used principal component analysis (PCA) to reduce the feature vector dimension from F to k , where k was chosen using validation data. The

third method, named PCA pursuit, constructed a feature vector by recursively selecting PCA features that gave the largest increase in validation code quality, until the validation code quality reached its maximum. For each of the above methods, 10 training runs were applied and the validation data was used to select the best model.

We also tried several variations of the support vector machine (SVM) (Schölkopf and Smola, 2002). For each tissue type, three one-versus-all SVMs were trained using the L2 kernel for each of the labels inc, exc and nc, which were obtained by thresholding the q 's at 0.5. The resulting triplets of real-valued discriminants were used as inputs for multinomial regression, to predict the three probabilities $p_{t,e}^{inc}, p_{t,e}^{exc}, p_{t,e}^{nc}$. Results were poor when we trained SVMs using all 1014 features, so PCA was used to project them onto a subspace with 40 dimensions, which was selected using validation data.

To test whether a non-Bayesian version of the model described in the previous section could give good results, we trained a fully connected network with 10 hidden variables (which equals the expected number of hidden variables λ) using early stopping (Bishop, 2006; Rumelhart *et al.*, 1986). The parameters were initialized to small random values and batch gradient descent with a learning rate of 0.1 was used to train the network until the validation code quality reached a maximum. The best of 10 trained models was selected using validation data.

We also examined naive Bayes, where the features are assumed to be independent given the splicing class (inc, exc, nc) and tissue type. For each tissue, every non-binary feature was binarized by searching for a threshold that maximized its mutual information with the q -distributions. Then, for each splicing class, the Bernoulli probability of every binary feature was independently estimated using the training data. Given a test feature vector, Bayes' rule was used to compute the posterior probability of each class.

It is often a good idea to average the predictions from quite different methods, since they may err in different ways. Based on initial experiments, we combined the predictions from the best non-Bayesian methods, including the SVM and multinomial regression using high-variance PCA features as inputs, plus the neural network and multinomial regression trained using early stopping.

4 RESULTS

Comparisons of test code quality: All methods were evaluated using held out test data as described above and the test code quality for several methods is plotted in Figure 1 (see below for further details). The Bayesian method achieved a relative improvement of 52% over the previously published result obtained using boosting (Barash *et al.*, 2010a), and significantly outperforms the other methods. While most other methods performed reasonably well, a notable exception is the nearest neighbor method. This may be due to a large fraction of irrelevant features and subgroups of features that operate combinatorially, which simple nearest neighbor methods are not well-suited to dealing with.

Table 1 gives a breakdown of code quality according to tissue type.¹ For each tissue and direction of regulation, bold font is used to indicate the highest code quality, regular font is used to indicate values that are significantly lower than the highest value ($P < 0.021$,

¹Naive Bayes is not shown, because it achieves a negative code quality due to extreme posterior probabilities induced by making a highly inaccurate feature independence assumption.

Table 1. Comparison of splicing test accuracy for different methods, measured using code quality (bits)

| Method | Tissue type | | | |
|--------------------------------|-----------------|----------------|----------------|----------------|
| | CNS | Muscle | Embryo | Digest |
| Boosting (<i>Nature</i> 2010) | 198 ± 7 | 71 ± 1 | 78 ± 2 | 64 ± 3 |
| Nearest neighbors | 19 ± 15 | 21 ± 2 | 20 ± 4 | 24 ± 6 |
| Regression, early stop* | 202 ± 23 | 103 ± 10 | 79 ± 7 | 73 ± 7 |
| Regression, PCA* | 210 ± 18 | 98 ± 5 | 90 ± 3 | 65 ± 4 |
| Regression, PCA pursuit | 192 ± 30 | 95 ± 7 | 92 ± 3 | 57 ± 8 |
| SVM, PCA* | 223 ± 12 | 95 ± 7 | 97 ± 7 | 72 ± 4 |
| Neural net, early stop* | 196 ± 12 | 100 ± 10 | 77 ± 9 | 67 ± 8 |
| Avg predictions from* | 232 ± 14 | 114 ± 5 | 102 ± 4 | 82 ± 4 |
| Bayesian method | 263 ± 13 | 129 ± 4 | 126 ± 3 | 105 ± 8 |
| Without sharing | 240 ± 16 | 112 ± 3 | 104 ± 3 | 76 ± 7 |

± indicates 1 SD; top performances are shown in bold; * denotes methods that were bagged together (predictions averaged).

t -test), and bold font is used for the other values to indicate that they might be comparable to the highest value. Using the best result published to date as a baseline (Barash *et al.*, 2010a), the Bayesian technique achieves improvements ranging from 30% in CNS tissues to almost 80% in muscle tissues. Interestingly, when the Bayesian method was applied to each tissue separately so that hidden variables were not shared across tissues, lower code qualities were obtained. Later, we explore how hidden variables are successfully shared across tissues.

Among all tissues, most methods achieved their highest code quality for CNS tissues. One reason for this is that there is a higher amount of tissue-specific splicing variability in CNS tissues, so the splicing information capacity is higher. Out of those exons exhibiting tissue-variable splicing ($q^{inc} \geq 0.99$ or $q^{exc} \geq 0.99$ in at least one tissue), 51% exhibit changes in CNS tissues, compared with 23, 32 and 25% in muscle, embryonic and digestive tissues, which is consistent with human RNA-Seq analysis (Pan *et al.*, 2008; Wang *et al.*, 2008). Another reason is that many of the RNA features were derived from previous work, which concentrated on the regulation in brain and muscle tissues. So, the inference problem is more straightforward for those tissues. A third reason is that splicing in certain tissues may be controlled by relatively easily inferred mechanisms, e.g. well-studied regulators such as Fox contribute to the regulation of splicing in both muscle and CNS tissues.

Averaging the predictions from the best non-Bayesian methods led to an increase in code quality over the individual predictors, but the Bayesian method performed significantly better. We wondered if the other methods had anything complementary to offer to the Bayesian method, so we included the Bayesian method in the average predictor. Interestingly, there was no significant improvement beyond the stand-alone Bayesian method's performance, even when we adjusted the relative weighting of the methods using test data (the Bayesian method weight was 0.94).

Comparisons of classification accuracy: The log-likelihood based measure of code quality described above takes into account how accurately each method can assess its confidence in its predictions. A different, but related task is to apply a threshold to each method's prediction probabilities and measure classification accuracy. We defined two binary classification tasks for each tissue type,

Table 2. Comparison of splicing test accuracy, measured using area under the ROC curve

| Method | Tissue type | | | |
|--------------------------------|-------------------|-------------------|-------------------|-------------------|
| | CNS | | Muscle | |
| | Inclusion | Exclusion | Inclusion | Exclusion |
| Boosting (<i>Nature</i> 2010) | 76.1 ± 0.6 | 60.3 ± 1.3 | 70.7 ± 0.5 | 60.7 ± 0.5 |
| Nearest neighbors | 71.7 ± 1.0 | 53.8 ± 1.2 | 60.0 ± 1.8 | 53.1 ± 1.3 |
| Regression, early stop* | 75.7 ± 0.8 | 61.3 ± 0.8 | 74.8 ± 1.0 | 62.6 ± 1.0 |
| Regression, PCA* | 77.1 ± 0.6 | 61.3 ± 0.8 | 73.4 ± 0.4 | 63.5 ± 0.5 |
| Regression, PCA pursuit | 76.6 ± 1.2 | 61.1 ± 1.7 | 73.5 ± 1.2 | 62.8 ± 1.0 |
| SVM, PCA* | 77.0 ± 1.1 | 61.1 ± 0.9 | 72.5 ± 0.6 | 61.8 ± 1.3 |
| Naive Bayes | 75.4 ± 0.6 | 62.1 ± 1.1 | 73.3 ± 0.5 | 62.9 ± 0.8 |
| Neural net, early stop* | 75.6 ± 0.7 | 60.9 ± 1.0 | 74.2 ± 1.1 | 62.6 ± 1.6 |
| Avg predictions from * | 77.3 ± 0.7 | 62.1 ± 0.7 | 74.7 ± 0.6 | 63.4 ± 0.7 |
| Bayesian method | 79.1 ± 0.5 | 63.1 ± 0.6 | 77.0 ± 0.5 | 63.2 ± 0.8 |
| Without sharing | 77.5 ± 0.7 | 62.9 ± 0.8 | 75.6 ± 0.5 | 63.6 ± 0.8 |

| Method | Tissue type | | | |
|--------------------------------|-------------------|-------------------|-------------------|-------------------|
| | Embryo | | Digestive | |
| | Inclusion | Exclusion | Inclusion | Exclusion |
| Boosting (<i>Nature</i> 2010) | 53.9 ± 0.9 | 69.6 ± 0.4 | 63.9 ± 1.2 | 63.8 ± 1.1 |
| Nearest neighbors | 52.7 ± 2.1 | 59.0 ± 0.9 | 55.3 ± 1.0 | 56.4 ± 1.0 |
| Regression, early stop* | 54.8 ± 1.6 | 68.8 ± 0.7 | 64.9 ± 0.6 | 64.5 ± 1.3 |
| Regression, PCA* | 55.0 ± 1.4 | 70.3 ± 1.1 | 64.0 ± 0.9 | 64.5 ± 0.9 |
| Regression, PCA pursuit | 55.5 ± 2.8 | 70.3 ± 1.1 | 63.3 ± 0.6 | 63.9 ± 0.9 |
| SVM, PCA* | 56.5 ± 1.5 | 69.8 ± 1.0 | 63.6 ± 1.0 | 64.1 ± 1.1 |
| Naive Bayes | 53.4 ± 1.0 | 68.6 ± 0.9 | 64.7 ± 1.1 | 65.0 ± 0.6 |
| Neural net, early stop* | 54.3 ± 1.1 | 69.0 ± 1.7 | 64.3 ± 0.5 | 63.6 ± 1.6 |
| Avg predictions from* | 55.7 ± 1.2 | 70.7 ± 1.2 | 65.6 ± 0.6 | 65.2 ± 1.1 |
| Bayesian method | 56.8 ± 0.6 | 73.0 ± 1.1 | 68.5 ± 0.5 | 68.2 ± 0.7 |
| Without sharing | 55.3 ± 1.0 | 71.3 ± 1.0 | 66.8 ± 0.6 | 66.0 ± 0.8 |

± indicates 1 SD; top performances are shown in bold; * denotes methods that were bagged together (predictions averaged).

corresponding to identifying exons exhibiting increased inclusion or exclusion. For each tissue type, ambiguous exons with $0.1 < q < 0.9$ were removed from the analysis. This screening retained on average 91, 81, 81 and 72% of the exons for classification in CNS, muscle, embryonic and digestive tissues. Then, for each tissue type, q^{inc} and q^{exc} were thresholded at 0.5 to define positive and negative examples. For the Bayesian method, we found that predictions for increased inclusion in CNS and muscle tissues and increased exclusion in embryonic tissues are the most accurate, whereas predictions for the reversed effect in those tissues are significantly less accurate (ROC curves are plotted in Supplementary Fig. S6).

Table 2 summarizes the classification accuracies for all methods in terms of the area under the ROC curve. The Bayesian method is the only consistent top performer. As before, when hidden variables are not shared across tissues, performance drops. Classification results are mostly consistent with the code quality results in Table 1; a notable exception is Naive Bayes, whose extreme probabilities give poor code quality but reasonable classification results.

The relative improvement in classification error of the Bayesian method over the original boosting method (Barash *et al.*, 2010a)

ranges from 22% for exon inclusion in muscle tissues to 7% for exon exclusion in CNS tissues. These improvements correspond to correctly classifying an additional 187 and 92 exons. Interestingly, these are larger sets of exons than examined in most studies of splicing regulation, cf. (Zhang *et al.*, 2010).

Differences in performance for different tissues and regulatory effects may suggest different types of regulatory mechanisms. Increased inclusion was easiest to predict in CNS and muscle tissues, whereas increased exclusion was easiest to predict in embryonic tissues. All methods performed comparably well on predicting increased exclusion in muscle tissues.

Analysis of hidden variable connectivity: The significant improvement of the Bayesian neural network over other methods along with the fact that it explicitly selects features and hidden variables, serves as a strong incentive to probe into the inferred model structure and how various features are used to predict splicing regulation. An advantage of the Bayesian approach is that it does not place all bets on one model, but instead provides a distribution over models so that hypotheses about selected features and model structures can be tested statistically. In the analyses reported below, the distribution over models was approximated using an ensemble of 60 000 models obtained from 2000 samples taken from each Gibbs sampling run for each of the 30 different training sets (5-fold cross-validation using six different random data partitions).

First, we examined how frequently each hidden variable was used to make predictions for each tissue type. Figure 3a indicates whether or not each hidden variable (row) was connected to the predictor for each tissue (color coded) after each iteration of Gibbs sampling (column), for one of the training sets. During the first ~50 iterations, hidden variables are infrequently used for predictions, because beneficial parameter values and feature combinations have not yet been learnt. Later, hidden variables are more frequently used and their connectivity becomes more stable. However, the plot supports the arguments made above that the Bayesian method benefits from broad exploration of connectivity and parameter settings. Often, connections are made briefly before being discarded. In other cases, the connection between a hidden variable and tissue is more stable and lasts for hundreds of Markov chain Monte Carlo (MCMC) iterations. In some of those cases, other tissues attempt to benefit by using the stable hidden variable. For example, from samples 260 to 500, hidden variable 4 is primarily used to predict splicing in CNS tissues, but is sometimes also used to predict splicing in muscle and digestive tissues, and to a lesser degree in embryonic tissues.

Figure 3b plots the distribution of the inferred number of hidden variables for each tissue, along with the prior distribution. All tissue types use fewer hidden variables than expected under the prior, with the exception of CNS, which uses more. Two possible explanations are that the regulation of splicing in CNS tissues is more complex than in other tissues, and that the dataset (features and/or splicing patterns) is biased toward having more information about splicing regulation in CNS tissues.

We next asked how frequently hidden variables were connected to the output variables for multiple tissues. Figure 3c plots the distribution of the inferred number of hidden variables that are used to make predictions for different numbers of tissues, along with the prior distribution. The number of hidden variables connected to a single tissue matches the prior quite closely. The number of hidden variables connected to two or three tissues is lower than expected

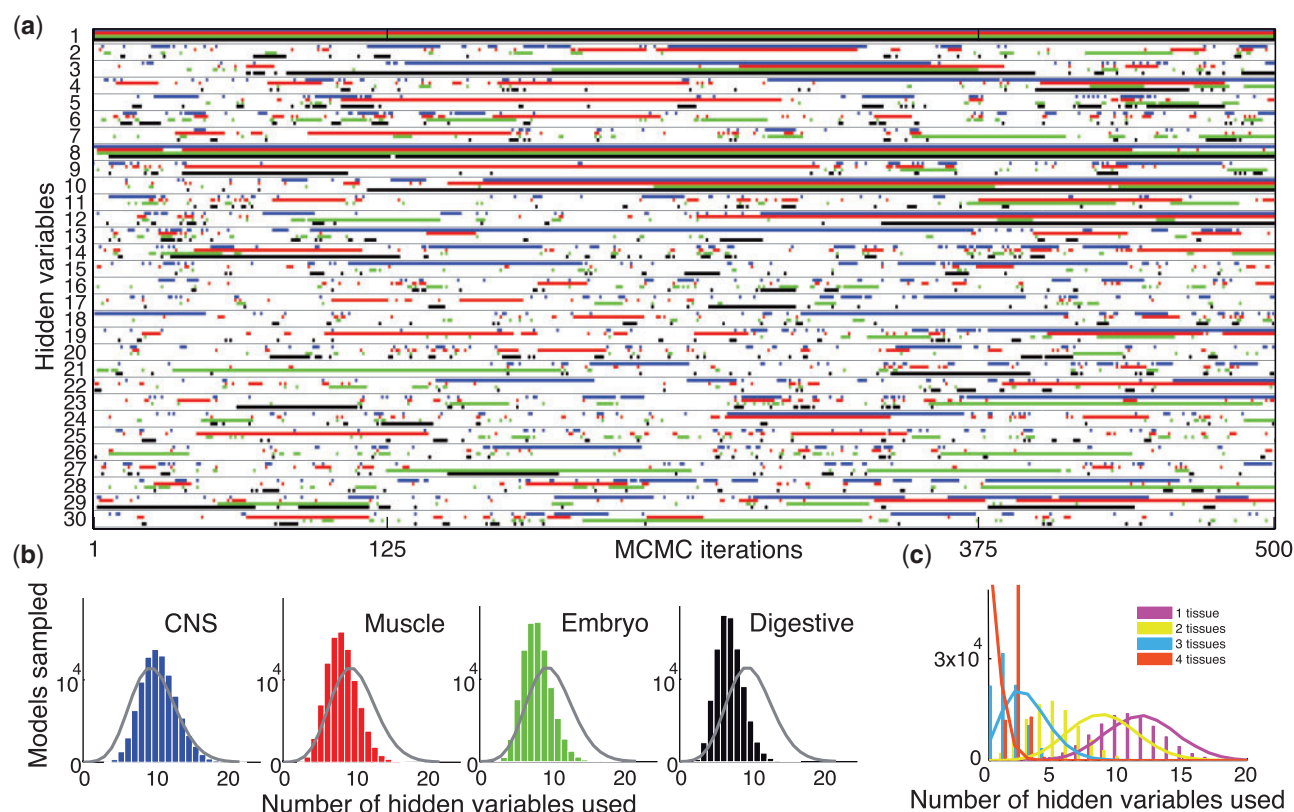


Fig. 3. (a) Use of different hidden variables (rows) by different tissues (colors), after each iteration of Gibbs sampling (columns). Blue, central nervous system (CNS) tissues; red, muscle tissues; green: embryonic tissues; black: digestive tissues. (b) Distribution of the number of hidden variables used to make predictions for different tissue types, *a priori* (solid curves) and *a posteriori* (bars). (c) Distribution of the number of hidden variables used to make predictions for different numbers of tissues, *a priori* (solid curves) and *a posteriori* (bars).

under the prior, but substantial nonetheless. For example, in over $\sim 87\%$ of the models, at least three hidden variables were connected to two tissues. Interestingly, the number of hidden variables connected to four tissues is substantially higher than expected under the prior.

Analysis of selected features and their connectivity: It was shown above that the number of hidden variables connected to all tissue types is significantly higher than expected under the prior. Upon examination, we found that those hidden variables tend to be connected to features measuring conservation levels in the up- and downstream introns, the presence of secondary structures, the strength of splice site junctions, exon length and the introduction of premature termination codons upon exon inclusion or exclusion.

We explored features that were selected by the Bayesian method and compared them to previously published results. The 10 most commonly used features were the same as those reported in Barash *et al.* (2010a), but the 50 most commonly used features had a lower overlap of $\sim 70\%$. These features include measurements of junction score, exon length and conservation, along with potential binding sites of regulators such as Fox, (n)PTB and Cugbp. The Bayesian method selected an overlapping but somewhat different set of motifs from the collection derived using conservation analysis (Yeo *et al.*, 2007). Slightly fewer previously defined motifs were used, especially those for the neural-specific regulators Nova1/2

(Licatalosi *et al.*, 2008). In contrast to previous work (Barash *et al.*, 2010a), the Bayesian method was able to achieve significantly higher prediction accuracy by using alternative definitions of motifs and detecting a larger number of combinations of simpler and less specific motifs through the use of hidden variables.

To demonstrate that the Bayesian method benefits from less frequently selected features, we identified the 40 most frequently selected features and discarded the remaining features. We re-applied the Bayesian learning procedure to one of the six training sets using the reduced feature set and found that the test code quality decreased by 16%.

We next asked whether we could identify relationships between features, which may correspond to functional modules in the regulation of alternative splicing. Hidden variables are not identifiable and indeed the MCMC procedure often deactivates previously useful hidden variables and activates new ones (Fig. 3a), so methods for analyzing static model structures are inappropriate. Instead, we examined the frequency with which pairs of most frequently used features were co-wired to the same hidden unit, the identity of which could vary across the 600 000 models in the ensemble. To avoid the problem of feature degeneracy, we labeled each feature using manually defined feature categories, such as 'Fox motif in the upstream intron', which includes alternative definitions of Fox motifs located anywhere in the upstream intron.

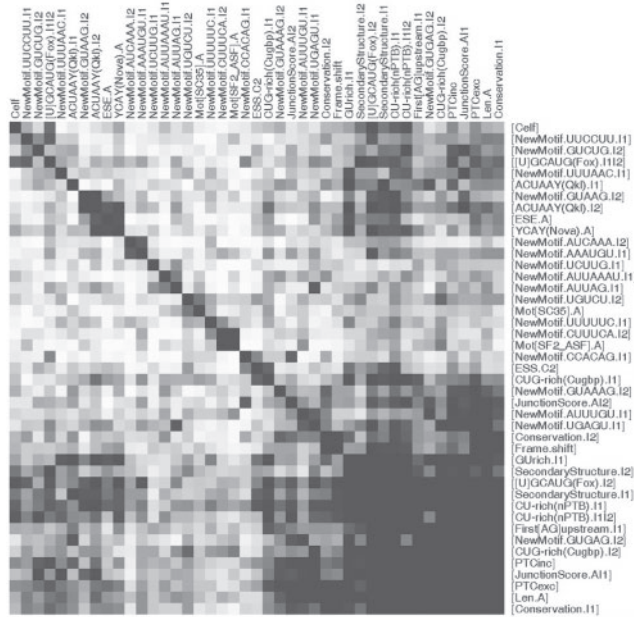


Fig. 4. Correlation of usage for pairs of frequently selected feature categories. White, insignificant correlation; dark grey, high-significance correlation ($p < 10^{-40}$, Fisher exact test).

We then computed the statistical significance of the overlap between pairs of feature categories. Figure 4 shows the resulting symmetric matrix of $-\log_{10}P$ -values, with rows and columns rearranged using clustering. The highly related features in the lower right part of the matrix primarily correspond to features that are used by frequently connected hidden units (e.g. hidden variables 1 and 10 in Fig. 3a). These include transcript structure features such as exon length, plus motifs corresponding to commonly active regulators, such as Fox, (n)PTB and Cugbp. In the upper right part of the matrix, there are less commonly used features, which are sometimes co-wired via frequently connected hidden variables, but which also sometimes form separate modules. The presence of an exonic splicing enhancer (ESE) in the alternative exon is co-wired via frequently connected hidden variables, but is also co-wired jointly with Quaking-like (Qkl) binding sites in the downstream intron and Nova binding sites in the alternative exon. Other novel relations include the conservation-based motif cluster UUUAAC and the strength of the junction between the alternative exon and the flanking intron, and GU-rich motifs in the upstream intron and Qkl motifs.

Scope for improvements from larger datasets: To direct future research, it is useful to predict whether or not additional gains in performance can be achieved through the use of larger datasets. While it is generally true that increasing the amount of training data can only improve test performance, returns will diminish as performance closes in on the maximum achievable level. To explore how sensitive the achieved test code quality is to the amount of training data, we trained the Bayesian method using seven differently sized training sets that were obtained by subsampling the original training data. In Figure 5, we plot test code quality against the training set size (log-scale). For all tissues, there is no evidence that

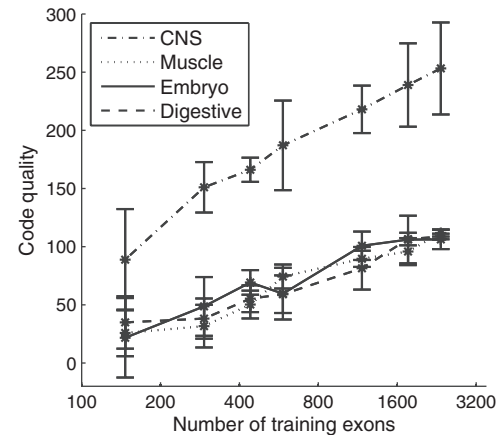


Fig. 5. The effect of the number of training cases on test code quality.

the method is near to a maximum code quality, suggesting that larger datasets can be used to achieve significantly higher code qualities.

5 CONCLUSIONS

Deriving a splicing code that uses combinations of RNA features to predict how splicing will occur under different cellular contexts is critical to understanding gene regulation. We introduced a novel Bayesian method that uses hidden variables within a network architecture to model non-linear relationships between putative regulatory features and splicing changes. We compiled a freely available benchmarking dataset along with a methodology for evaluating different techniques.

Our method achieved relative improvements of 52% in test code quality and up to 22% in test classification accuracy, compared with the state of the art (Barash *et al.*, 2010a). It correctly classified hundreds of additional tissue-dependent splicing changes, and outperformed all other methods that we tested. Even when predictions from four of the best other methods were combined, the Bayesian technique performed significantly better. Using a sample of models from the posterior distribution, the importance of individual RNA features and their pairwise combinations were assessed. These feature combinations were mostly consistent with the previous results, and also included novel predictions.

A promising future direction is to profile more cassette exons, since we found that this will likely lead to significant improvements. It would also be useful to account for other kinds of alternative splicing such as alternative splice sites and mutually exclusive exons. Ultimately, we would like to be able to predict the relative abundances of entire transcripts.

Our methodology and the Bayesian technique can be applied to datasets profiling larger numbers of tissues, different species, and different types of alternative splicing. When we applied the Bayesian technique to splicing patterns derived from RNASeq data for 16 human tissues, along with the RNA feature definitions that we used for mouse, classification rates ranging from 67% in thyroid gland to 83% in whole brain were obtained.

A multispecies splicing code with shared and species-specific regulatory subprograms can be inferred using matched tissue data

and by feeding the feature vectors and target splicing patterns for all species into the learning algorithm.

It was found that using large numbers of features improved code quality, pointing to the importance of further exploring new feature types, such as those derived using *in vivo* (Licatalosi et al., 2008) or *in vitro* (Ray et al., 2009) RNA binding data, DNA, chromatin structure and histone modifications (Luco et al., 2011).

We believe that the method developed in this study, along with the accompanying benchmark dataset, will help push the envelope of our ability to predict splicing outcomes, with possible applications ranging from analyzing transcripts of genes with low expression to disease-specific mutation analysis.

ACKNOWLEDGEMENT

We thank Ben Blencowe, Geoffrey Hinton, Yann LeCun and Radford Neal for discussions.

Funding: Canadian Institutes for Health Research Operating Grant MOP-106690 (to B.J.F.); Genome Canada and Ontario Genomics Institute Grants (to B.J.F.); Natural Sciences and Engineering Research Council (NSERC) Grant SMFSU 379968-09 (to B.J.F.); Canadian Foundation for Innovation and Ontario Research Fund Grant 203788 (to B.J.F.). B.J.F. is a Fellow of the Canadian Institute for Advanced Research and an NSERC E.W.R. Steacie Fellow.

Conflict of interest: None declared.

REFERENCES

- Barash, Y. et al. (2010a) Deciphering the splicing code. *Nature*, **465**, 53–59.
- Barash, Y. et al. (2010b) Model-based detection of alternative splicing signals. *Bioinformatics*, **26**, i325.
- Bishop, C.M. (2006) *Pattern Recognition and Machine Learning*. Springer, NY.
- Blencowe, B. (2006) Alternative splicing: new insights from global analyses. *Cell*, **126**, 37–47.
- Chan, R. and Black, D. (1997) The polypyrimidine tract binding protein binds upstream of neural cell-specific c-src exon n1 to repress the splicing of the intron downstream. *Mol. Cell. Biol.*, **17**, 4667.
- Fagnani, M. et al. (2007) Functional coordination of alternative splicing in the mammalian central nervous system. *Genome Biol.*, **8**, R108.
- Hartmann, B. and Valcárcel, J. (2009) Decrypting the genome's alternative messages. *Curr. Opin. Cell Biol.*, **21**, 377–386.
- Ishwaran, H. and Rao, J. (2005) Spike and slab gene selection for multigroup microarray data. *J. Am. Stat. Assoc.*, **100**, 764–780.
- Licatalosi, D. et al. (2008) Hits-clip yields genome-wide insights into brain alternative RNA processing. *Nature*, **456**, 464–469.
- Lim, L.P. and Sharp, P.A. (1998) Alternative splicing of the fibronectin EIIIB exon depends on specific TGCATG repeats. *Mol. Cell. Biol.*, **18**, 3900–3906.
- MacKay, D. (1992) A practical Bayesian framework for backpropagation networks. *Neural Comput.*, **4**, 448–472.
- Neal, R. (1996) *Bayesian Learning for Neural Networks*, Vol. 118. Springer, NY.
- Pan, Q. et al. (2008) Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nat. Genet.*, **40**, 1413–1415.
- Ray, D. et al. (2009) Rapid and systematic analysis of the RNA recognition specificities of RNA-binding proteins. *Nat. Biotechnol.*, **27**, 667–670.
- Rumelhart, D. et al. (1986) Learning representations by back-propagating errors. *Nature*, **323**, 533–536.
- Schölkopf, B. and Smola, A. (2002) *Learning With Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, MA.
- Wang, Z. and Burge, C. (2008) Splicing regulation: From a parts list of regulatory elements to an integrated splicing code. *RNA*, **14**, 802.
- Wang, G. and Cooper, T. (2007) Splicing in disease: disruption of the splicing code and the decoding machinery. *Nat. Rev. Genet.*, **8**, 749–761.
- Wang, E. et al. (2008) Alternative isoform regulation in human tissue transcriptomes. *Nature*, **456**, 470–476.
- Yeo, G. et al. (2007) Discovery and analysis of evolutionarily conserved intronic splicing regulatory elements. *PLoS Genet.*, **3**, e85.
- Zhang, C. et al. (2010) Integrative modeling defines the nova splicing-regulatory network and its combinatorial controls. *Science*, **329**, 439.
- Luco, R.F. et al. (2011) Epigenetics in alternative pre-mRNA splicing. *Cell*, **144**, 16.