BIOINFORMATICS

Vol. 20 Suppl. 1 2004, pages i69–i76
DOI: 10.1093/bioinformatics/bth932

Splice site identification by idl*BNs*

Robert Castelo* and Roderic Guigó

Grup de Recerca en Informàtica Biomèdica, Institut Municipal d'Investigació Mèdica, Universitat Pompeu Fabra, Centre de Regulació Genòmica, Psg. Marítim 37–49, 08003 Barcelona, Spain

Received on January 15, 2004; accepted on March 1, 2004

ABSTRACT

Motivation: Computational identification of functional sites in nucleotide sequences is at the core of many algorithms for the analysis of genomic data. This identification is based on the statistical parameters estimated from a training set. Often, because of the huge number of parameters, it is difficult to obtain consistent estimators. To simplify the estimation problem, one imposes independent assumptions between the nucleotides along the site. However, this can potentially limit the minimum value of the estimation error.

Results: In this paper, we introduce a novel method in the context of identifying functional sites, that finds a reasonable set of independence assumptions supported by the data, among the nucleotides, and uses it to perform the identification of the sites by their likelihood ratio. More importantly, in many practical situations it is capable of improving its performance as the training sample size increases. We apply the method to the identification of splice sites, and further evaluate its effect within the context of exon and gene prediction.

Contact: rcastelo@imim.es

Supplementary information: The datasets built specifically for this paper as well as the full set of results are available at http://genome.imim.es/datasets/splidlbns2004

1 INTRODUCTION

Genome information is decoded through the processing of signals in the primary nucleotide sequence. Sequence signals involved in a common functionality often show some degree of similarity. Therefore, numerous methods have been developed to locate functional sites in genomic sequences, based on the sequence patterns characteristically correlated with the sought functionality. Typically, sequence patterns are probability distributions that assign high probabilities to sequences that resemble the functional sites.

Given a genomic sequence s that forms a candidate functional site, one of the most popular, and simple, probabilistic methods to decide whether s belongs to a set $\mathcal{S}$ of real sites, or belongs to a set $\mathcal{N}$ of false sites, is the ratio of the likelihood that $s \in \mathcal{S}$, over the likelihood that $s \in \mathcal{N}$. This ratio is implemented by assuming that the statistical properties of the functional sites in $\mathcal{S}$ and $\mathcal{N}$ can be summarized through two sets of parameters $\theta_{\mathcal{S}}$ and $\theta_{\mathcal{N}}$, and hence we write the likelihood ratio as

$$\lambda = \frac{L_{\mathcal{S}}(s \mid \theta_{\mathcal{S}})}{L_{\mathcal{N}}(s \mid \theta_{\mathcal{N}})}. \tag{1}$$

The value λ is regarded as the likelihood-ratio statistic and, when we consider λ as sufficiently small, we discard s as a real functional site (thus, $s \in \mathcal{N}$). In fact, Neyman and Pearson (1928) showed that this is the most powerful test for such a decision problem. However, in practice the sets of parameters $\theta_{\mathcal{S}}$ and $\theta_{\mathcal{N}}$ are unknown, and can only be estimated through some training data of confirmed real and false, functional sites. This implies that we should select $\theta_{\mathcal{S}}$ and $\theta_{\mathcal{N}}$ as those that maximize the likelihood functions $L_{\mathcal{S}}(s \mid \theta_{\mathcal{S}})$ and $L_{\mathcal{N}}(s \mid \theta_{\mathcal{N}})$.

Since the parameters in $\theta_{\mathcal{S}}$ and $\theta_{\mathcal{N}}$ correspond to probabilities of observing a particular genomic sequence forming the (real or false) functional site, the number of them, for which we need to obtain the maximum-likelihood estimators (MLE), grows exponentially in the number of nucleotides of s. This makes infeasible to simply count occurrences of each different site in the training data in order to obtain an MLE. A standard approach, commonly known as position weight matrix (PWM), has been used to assume that given a functional site of length n, the nucleotides occur independently within the site, and thus only $3 \times n$ parameters are required. However, the assumption that the nucleotides occur independently may not hold, which in turn, can introduce a positive bias in the estimation of each parameter. A positive bias increases the expectation of the estimation error as this equals the variance plus the squared bias of the parameter.

Ideally, when we have an unbiased estimator, $\hat{\theta}$, it converges to its true value θ as the sample size increases, because the variance decreases at the same time. In this case, one says that the estimator is consistent. Therefore, PWMs might be performing an inconsistent estimation and in order to alleviate this problem, first-order Markov models (FMMs) are used, under which the occurrence of each nucleotide depends on the occurrence of the nucleotide in the previous position. This makes all the positions marginally dependent while the number of required parameters remains moderate, $n \times 4 \times 3$.

*To whom correspondence should be addressed.

The FMM affords a substantial improvement over the PWM when dependencies are present in the biological signal, but still some of the conditional independencies implied by the first-order Markov assumption can introduce a positive bias. A case in point is the prediction of 5′ (donor) splice sites. The molecular recognition of these functional sites during splicing is mediated by base pair interactions between the sequences at the site and at the 5′ end of the so-called U1 snRNP—a ribonucleoprotein complex. The probability distribution of the nucleotides estimated independently at each position along 5′ splice sites nicely reflects this complementarity: the most frequent nucleotide at each position along the site is the complement of the interacting nucleotide in the U1 snRNP. However, because of the staking of the double-stranded DNA, the energy of the interaction between a pair of nucleotides also depends on their nearest neighbor nucleotides. That is, the probability of a nucleotide at a given position in the 5′ site is not independent of the nucleotide(s) occurring upstream of the site. To capture adjacent, as well as non-adjacent, dependencies between positions in sequence patterns, a number of methods have been developed (Agarwal and Bafna, 1998; Burge, 1998; Cai *et al*., 2000; Dash and Gopalakrishnan, 2001; Barash *et al*., 2003; Yeo and Burge, 2003).

We follow the approach to modeling (non-)adjacent dependencies by using Bayesian networks, exploiting recent results (Castelo and Kočka, 2003) in structure learning of these models, which permit reducing the bias in the estimation problem as the sample size increases. We apply this method to the identification of splice sites, showing a significant improvement over some of the currently existing methods. We further assess the effect of integrating such an improvement within exon and gene prediction.

2 METHODS

In this section, we first review the asymptotic form of the classification error probabilities. Then, we describe the implementation of the likelihood ratio by *idl*BNs and finally, we study two different ways to combine the scores of the identified splice sites in order to predict an exon.

2.1 Asymptotic form of the error probabilities

The decision to classify a functional site as real or false, using the likelihood ratio, is taken by setting a threshold t that determines what sites are discarded as real functional sites because the likelihood-ratio statistic λ is smaller than t, and what sites are accepted as real functional sites, because $\lambda \geq t$. Depending on the threshold t we face different classification error probabilities, and the performance of any classifier is bounded by the asymptotic form of these error probabilities.

Assume that the real, and false, functional sites are independent and identically distributed (iid) samples from probability distributions $P_{\theta_\mathcal{S}}$ and $P_{\theta_\mathcal{N}}$ with probability mass functions (pmf) $p(s \mid \theta_\mathcal{S})$ and $p(s \mid \theta_\mathcal{N})$, respectively. The error probability of classifying a real site as a false one is $\alpha = P_{\theta_\mathcal{S}}(\lambda < t)$ and classifying a false site as a real one is $\beta = P_{\theta_\mathcal{N}}(\lambda \geq t)$. The probability of effectively discarding false functional sites as such is $1 - \beta$, commonly known as the power of the test (in this case a likelihood-ratio test). The asymptotic bound on the error β of misclassifying a false functional site is given by the so-called Stein's Lemma[1] (Cover and Thomas, 1991, Ch. 6, § 12.8) in the following way:

$$\lim_{N\to\infty} \frac{1}{N} \log \beta = -D(P_{\theta_\mathcal{S}} || P_{\theta_\mathcal{N}}), \tag{2}$$

where N is the sample size and $D(P_{\theta_\mathcal{S}} || P_{\theta_\mathcal{N}})$ is the Kullback–Leibler (KL) divergence defined by Kullback and Leibler (1951):

$$\mathrm{D}(P_{\theta_\mathcal{S}} || P_{\theta_\mathcal{N}}) = \sum_s p(s \mid \theta_\mathcal{S}) \log \frac{p(s \mid \theta_\mathcal{S})}{p(s \mid \theta_\mathcal{N})}. \tag{3}$$

The KL divergence is a measure of the difference between two probability distributions. It is a non-negative quantity that is equal to 0 when the distributions are identical, and grows proportionally to the difference between them. Therefore, the asymptotic bound (2) tells us that the error probability β, of misclassifying a false functional site, has an exponential decay $2^{-ND(P_{\theta_\mathcal{S}} || P_{\theta_\mathcal{N}})}$, i.e. in the sample size N, bounded by the difference between the two probability distributions from which the real, and false, functional sites are drawn.

However, the parameter sets $\theta_\mathcal{S}$ and $\theta_\mathcal{N}$ are unknown and, in practice, we have to work with sets $\hat{\theta}_\mathcal{S}$ and $\hat{\theta}_\mathcal{N}$ of parameters estimated from datasets of confirmed real, and false, functional sites. In this situation, it can be shown (Castelo, 2004, submitted for publication) that the asymptotic bound (2) on β becomes:

$$\begin{aligned}\lim_{N\to\infty} \frac{1}{N} \log \beta = &-D(P_{\theta_\mathcal{S}} || P_{\theta_\mathcal{N}}) + D(P_{\theta_\mathcal{S}} || P_{\hat{\theta}_\mathcal{S}}) \\ &+ D(P_{\theta_\mathcal{N}} || P_{\hat{\theta}_\mathcal{N}}) - \sum_s (p(s \mid \theta_\mathcal{S}) \\ &+ p(s \mid \theta_\mathcal{N})) \log \frac{p(s \mid \theta_\mathcal{N})}{p(s \mid \hat{\theta}_\mathcal{N})}.\end{aligned} \tag{4}$$

This bound consists of the baseline term we had in (2), plus two terms corresponding to the KL divergence between the true parameters $\theta_\mathcal{S}$ and $\theta_\mathcal{N}$, and the estimated parameters $\hat{\theta}_\mathcal{S}$ and $\hat{\theta}_\mathcal{N}$. The latter two terms increase the error β proportionally to the difference between the true and the estimated distributions. The last term in (4) depends also on the difference between $\theta_\mathcal{N}$ and $\hat{\theta}_\mathcal{N}$ and in the limit of the size of the sample all these three additional terms will cancel when $\hat{\theta}_\mathcal{S} \to \theta_\mathcal{S}$ and $\hat{\theta}_\mathcal{N} \to \theta_\mathcal{N}$. Therefore, a biased estimation of the sets of parameters bounds the minimum size of the error probability β of classifying a false functional site as real.

[1] Although this result seems to be originally written by Chernoff (1952).

2.2 Inclusion-driven learned Bayesian networks

A Bayesian network (Pearl, 1988) for categorical data can be defined as a statistical model consisting of a family $\{P_\theta \mid \theta \in \Theta_G\}$ of multinomial probability distributions that convey a set of conditional independence (CI) restrictions summarized in an acyclic digraph, or DAG, G. The DAG G is formed by the pair (V, E) of sets, where $V \equiv V(G)$ are the vertices and $E \equiv E(G)$ are the directed edges. The vertices in V index a vector of categorical random variables $\mathbf{X}_V = \{X_1, \ldots, X_n\}$ and the fundamental idea is that the lack of an edge between two vertices i and j represents some particular CI restriction between X_i and X_j that holds throughout the entire family $\{P_\theta \mid \theta \in \Theta_G\}$. When a distribution P_θ only contains CI restrictions that can be represented by a DAG, one says that P_θ is a DAG-distribution.

By the chain rule of probability, and the CI restrictions specified in G, we obtain the following unique factorization of the pmf of $\mathbf{X}_V$:

$$p(x_1, \ldots, x_n \mid G, \theta) = \prod_i^n p\left[x_i \mid \mathbf{x}_{\mathrm{pa}(i)}, \theta_i\right], \qquad (5)$$

where i is a vertex from G, indexing a random variable X_i that takes values x_i and pa(i) are the parent vertices of i in G (source vertices of the directed edges pointing to i), indexing a set of random variables $\mathbf{X}_{\mathrm{pa}(i)}$ that take values $\mathbf{x}_{\mathrm{pa}(i)}$.

Given a training dataset, the structure of the Bayesian network, i.e. the DAG G, can be automatically learned by using a scoring metric and a search procedure. In this work, we have used the so-called BDeu scoring metric, which is constructed by making some assumptions about the training data and integrating out the parameters θ in (5) for each observation. The reader may find a thorough description of the BDeu scoring metric in the paper by Heckerman *et al*. (1995). An important feature of the BDeu scoring metric is that it is consistent, i.e. in the limit of the size of the training data sampled from a distribution P_θ, the BDeu scoring metric assigns the highest score to every[2] DAG G for which $P_\theta \in \{P_\theta \mid \theta \in \Theta_G\}$ with smallest dimension. In this context, the dimension refers to the number of parameters required by G.

Since the number of possible DAGs G grows exponentially in the number of vertices a search procedure is required. In a stepwise manner, starting from the DAG with no edges, the search procedure creates a set of candidate (neighbor) DAGs following some particular search policy, and ranks them using the scoring metric. Then selects the one that maximizes the score, and starts again from the selected DAG until a stopping criterion is met.

Among the many algorithms developed during the last decade for structure learning there is a class of them, called inclusion-driven structure learning algorithms, that has been introduced recently (Kočka and Castelo, 2001; Chickering, 2002; Castelo and Kočka, 2003) and which are optimal in the following sense. Under the assumption that the data are sampled from a DAG-distribution and in the limit of the size of the sample, they learn a correct DAG structure when using a consistent scoring metric (Castelo and Kočka, 2003, Th. 3.4). In this work, we have used the HCMC algorithm (Kočka and Castelo, 2001; Castelo and Kočka, 2003), but we shall distinguish a Bayesian network learned by any algorithm of this kind as an inclusion-driven learned Bayesian network, or *idl*BN[3].

Once an *idl*BN is learned by the HCMC algorithm for a set of true sites, and another one for a set of false sites, we use them to estimate the probabilities corresponding to the margins of the dependencies. Let θ_{ijk} be the parameter specifying the probability of observing the value k taken by variable X_i when the variables forming the parent set of vertex i in a DAG G take the values indexed by j in the corresponding product space. We perform a Bayesian estimation using a Dirichlet prior specified with hyperparameters N'_{ijk}, and the probability θ_{ijk} corresponds to the expectation (DeGroot, 1970):

$$\mathrm{E}[\theta_{ijk} \mid G, D, N'_{ijk}] = \frac{N'_{ijk} + N_{ijk}}{N'_{ij} + N_{ij}}, \qquad (6)$$

where N_{ijk} are the sufficient statistics of a dataset D, $N'_{ij} = \sum_k N'_{ijk}$ and $N_{ij} = \sum_k N_{ijk}$. We assume an uninformative prior $N'_{ijk} = 1/(q_i r_i)$, where q_i is the number of different configurations of values in the product space induced by the parent set of the variable X_i (in the corresponding DAG), and r_i is the number of values that X_i can take. Such an uninformative prior acts as a flattening constant (sometimes known as pseudocount) that smooths the estimation when the counts N_{ijk} are close, or equal to zero. Using these estimates in log-scale, we build the corresponding implementation of the log-likelihood ratio in a similar way as a PWM is built, but taking into account the dependencies among the positions determined by the learned DAGs of the sets of true, and false, functional sites.

Assuming that the probability distributions $P_{\theta_\mathcal{S}}$ and $P_{\theta_\mathcal{N}}$ of real, and false, functional sites were DAG-distributions, the corresponding *idl*BNs obtained from the training data will approach a correct DAG underlying the corresponding distribution as the training data size grows. Then, by the chain rule of probability and the DAG factorization in (5), the KL divergence between the distributions built from estimated and true parameters will approach 0, and thus the error probability bound in (4) will approach the optimal error bound (2).

[2] Two or more different DAGs can specify the same CI restrictions, and thus be (Markov) equivalent.

[3] For an easier parsing and recall of the term we propose to pronounce it as *idealBN*.

2.3 Exon prediction from flanking signals

In many gene prediction programs, as for instance, `geneid`, an exon is usually scored by pairing flanking signals and combining their scores, jointly with other features as the protein-coding bias which we shall not treat here. Given an exon sequence s with flanking signals s_5 and s_3, a straightforward way of combining the scores is by adding them up:

$$\mathrm{sc}_E(s) = \mathrm{sc}_5(s_5) + \mathrm{sc}_3(s_3)\,, \tag{7}$$

where $\mathrm{sc}_5(s_5)$ and $\mathrm{sc}_3(s_3)$ refer to the scores from the $5'$ and $3'$ flanking signals, respectively. This is actually how `geneid` [Parra *et al.* 2000, Expression (6)] computes part of the score of an exon (other parts involve the use of codons and homology). By substituting terms in expression (7),

$$\mathrm{sc}_E(s) = \log\left[\frac{L_{\mathcal{S}5}(s_5 \mid \theta_{\mathcal{S}5})}{L_{\mathcal{N}5}(s_5 \mid \theta_{\mathcal{N}5})} \cdot \frac{L_{\mathcal{S}3}(s_3 \mid \theta_{\mathcal{S}3})}{L_{\mathcal{N}3}(s_3 \mid \theta_{\mathcal{N}3})}\right], \tag{8}$$

the exon score in (7) can be interpreted as a likelihood ratio for an exon (LRE) (Parra *et al.*, 2000, pp. 513), where the likelihoods of a real and a false exon are respectively

$$\begin{aligned} L_{\mathcal{E}}(s \mid \theta_{\mathcal{E}}) &= L_{\mathcal{S}5}(s_5 \mid \theta_{\mathcal{S}5}) \cdot L_{\mathcal{S}3}(s_3 \mid \theta_{\mathcal{S}3})\,, \\ L_{\mathcal{F}}(s \mid \theta_{\mathcal{F}}) &= L_{\mathcal{N}5}(s_5 \mid \theta_{\mathcal{N}5}) \cdot L_{\mathcal{N}3}(s_3 \mid \theta_{\mathcal{N}3})\,. \end{aligned} \tag{9}$$

Assuming that the likelihoods $L_{\mathcal{S}5}$ and $L_{\mathcal{S}3}$ of the $5'$ and $3'$ splice sites assign high probabilities to real splice sites, and that a real exon should have two real splice sites, the likelihood $L_{\mathcal{E}}$ of a real exon in (9) is maximum when the exon is real. Analogously, by assuming that the likelihoods $L_{\mathcal{N}5}$ and $L_{\mathcal{N}3}$ assign high probabilities to false splice sites, the likelihood $L_{\mathcal{F}}$ of a false exon is maximum when both $5'$ and $3'$ splice sites are false. However, this does not necessarily imply that $L_{\mathcal{F}}$ is maximum when the exon is false, as an exon is already false with just one false splice site. In such a case, $L_{\mathcal{F}}$ will not be maximum, $L_{\mathcal{E}}$ will not be as low as it should for a false exon, and therefore the exon score will be larger than it should, leading to a potential misclassification.

Consider three models of a false exon denoted by $\mathcal{F}^{5,3}$, where both splice sites are false; $\mathcal{F}^5$ where only the $5'$ splice site is false; and $\mathcal{F}^3$ where only the $3'$ splice site is false. The corresponding likelihood functions can be built as:

$$\begin{aligned} L_{\mathcal{F}^{5,3}}(s \mid \theta_{\mathcal{F}}^{5,3}) &= L_{\mathcal{N}5}(s_5 \mid \theta_{\mathcal{N}5}) \cdot L_{\mathcal{N}3}(s_3 \mid \theta_{\mathcal{N}3})\,, \\ L_{\mathcal{F}^5}(s \mid \theta_{\mathcal{F}}^{5}) &= L_{\mathcal{N}5}(s_5 \mid \theta_{\mathcal{N}5}) \cdot L_{\mathcal{S}3}(s_3 \mid \theta_{\mathcal{S}3})\,, \\ L_{\mathcal{F}^3}(s \mid \theta_{\mathcal{F}}^{3}) &= L_{\mathcal{S}5}(s_5 \mid \theta_{\mathcal{S}5}) \cdot L_{\mathcal{N}3}(s_3 \mid \theta_{\mathcal{N}3})\,. \end{aligned} \tag{10}$$

Note that $L_{\mathcal{F}^5}$ and $L_{\mathcal{F}^3}$ are functions of parameters in $\theta_{\mathcal{E}}$, and this prevents us from averaging over $\mathcal{F}^{5,3}$, $\mathcal{F}^5$ and $\mathcal{F}^3$ in order to obtain a likelihood for a false exon. Instead, we use these models to define the posterior probability of a real exon:

$$p(\mathcal{E} \mid s) = \frac{L_{\mathcal{E}}(s \mid \theta_{\mathcal{E}})\,p(\mathcal{E})}{\sum_{M \in \{\mathcal{E}, \mathcal{F}^{5,3}, \mathcal{F}^5, \mathcal{F}^3\}} L_M(s \mid \theta_M)\,p(M)}\,, \tag{11}$$

where $p(M)$ are the prior probabilities of each model, which we assume uniform, i.e. $p(M) = 1/4$. In this way, $1 - p(\mathcal{E} \mid s)$ gives us a probability that the sequence s is a false exon with either two false splice sites or only one. So now, we propose to correct the LRE (7) multiplying it by the odds of the exon having two real flanking signals against at least one being false:

$$\begin{aligned} \mathrm{sc}_{\mathrm{E}}(s) = \log \frac{L_{\mathcal{S}5}(s_5 \mid \theta_{\mathcal{S}5}) L_{\mathcal{S}3}(s_3 \mid \theta_{\mathcal{S}3})}{L_{\mathcal{N}5}(s_5 \mid \theta_{\mathcal{N}5}) L_{\mathcal{N}3}(s_3 \mid \theta_{\mathcal{N}3})} \\ + \log \frac{p(\mathcal{E} \mid s)}{1 - p(\mathcal{E} \mid s)}\,, \end{aligned} \tag{12}$$

expressed in logarithmic scale and we shall denote it as the corrected likelihood ratio (CLR).

3 RESULTS

Our goal in this section is 2-fold: first and foremost, to show that *idl*BNs improve splice site, exon and gene prediction, and second, to show that the way scores are combined (LRE versus CLR) is crucial to take full advantage of a better signal identification.

3.1 The data

In the experiments we have used three different datasets. One is the set of 19 174 human annotations in the reference sequence (RefSeq) dataset (UCSC version hg15) based on NCBI Build 33 (April 10, 2003)[4]. The other two are used exclusively for testing purposes and correspond to the Burset and Guigó (1996) dataset of 570 human genes (BG-570), and the Rogic *et al.* (2001) dataset of 195 genes (HMR-195), including human (103), mouse (82) and rat (10) genes.

In order to assess splice site identification, we have extracted from the RefSeq genes two sets of non-redundant canonical donor and acceptor sites. The non-redundancy has been enforced by selecting unique stretches of DNA containing the donor and the acceptor site, where the donor stretch had 39 bp (the first 3 bp in the exon) and the acceptor stretch had 23 bp. Afterwards, from the donor context we selected 9 bp (3 bp exon $+gt+$ 4 bp intron) to form every donor site. This left a total of 124 727 donor sites and 130 220 acceptor sites. We created two datasets of these same sizes with false (decoy) donor and acceptor sites by sampling uniformly from coding regions, and two more datasets by sampling from intronic regions of the RefSeq genes. All the false sites matched the corresponding minimum consensus (gt, ag). Non-redundancy was enforced in the same way as for the

[4] http://www.genome.ucsc.edu/goldenPath/10april2003/database/refGene.txt.gz

true sites. We shall refer to these datasets as the ACCDON datasets.

In order to assess exon and gene prediction, we proceed as follows. From the RefSeq data, we have filtered out annotations with either the same name, or in a loose chromosome chunk (`chr_random`) or those that overlap another annotation (leaving one copy arbitrarily chosen). From this latter set, we translated each gene into protein, and discard those that did not start and end, with a start and stop codons, and those which had in-frame stop codons.

Further, we obtained the corresponding proteins from the BG-570 and HMR-175 genes and performed BLASTP against the previously filtered RefSeq proteins. We considered only those heat shock proteins (HSPs) that had an E-value smaller than or equal to 10^{-5} and from these, select those that showed an identity greater than or equal to 60%. There were 855 proteins matching these criteria, and the corresponding genes were removed from the RefSeq genes, leaving a total of 13 225 genes, to which we shall refer as the NOBGRORS dataset[5]. In this manner, we ensured that our training set of genes did have neither of the test genes nor those which might be highly similar to them.

From the BG-570 and HMR-195 genes, we extracted all the internal exons (2110). From each gene with n internal exons, we have randomly sampled the same amount of false exons thrice. First, ensuring that both splice sites are false; second, for every true acceptor site, we chose randomly a downstream false donor site; and third, for every true donor site we chose randomly an upstream false acceptor site. We shall refer to these datasets as the BGROIEXONS datasets. All data are available in the online Supplementary information.

3.2 Donor and acceptor site prediction

We have performed a 10-fold cross-validation along increasing sizes of the training datasets of donor and acceptor sites, starting on 1000, 5000 and 10 000, and continuing with sizes that increase in 10 000 sites up to 100 000. The rest of the sites not contained in each training dataset form the test dataset. The 10-fold cross-validation is done by sampling each of the 12 sizes, 10 times from the ACCDON datasets, where half of the false sites belong to coding regions and half to intronic ones. Finally, the results are averaged over the 10 samples for each size.

We have assessed the *idl*BN, PWM, FMM, MDD (Burge, 1998) and MAXENT (Yeo and Burge, 2003) methods. The latter two have not been trained in our data, but have been used through the scoresplice webserver[6], which scores donor sites with the MDD and MAXENT methods, and acceptor sites with the MAXENT method only. This means that, with respect to MDD and MAXENT, we will not appreciate the effect of an increasing training sample size, and that some of the sites in the test datasets could be included in those used for training the MDD and MAXENT methods currently implemented in the scoresplice webserver. We think, however, the comparison still provides an idea of how competitive the *idl*BN method is, with respect to MDD and MAXENT.

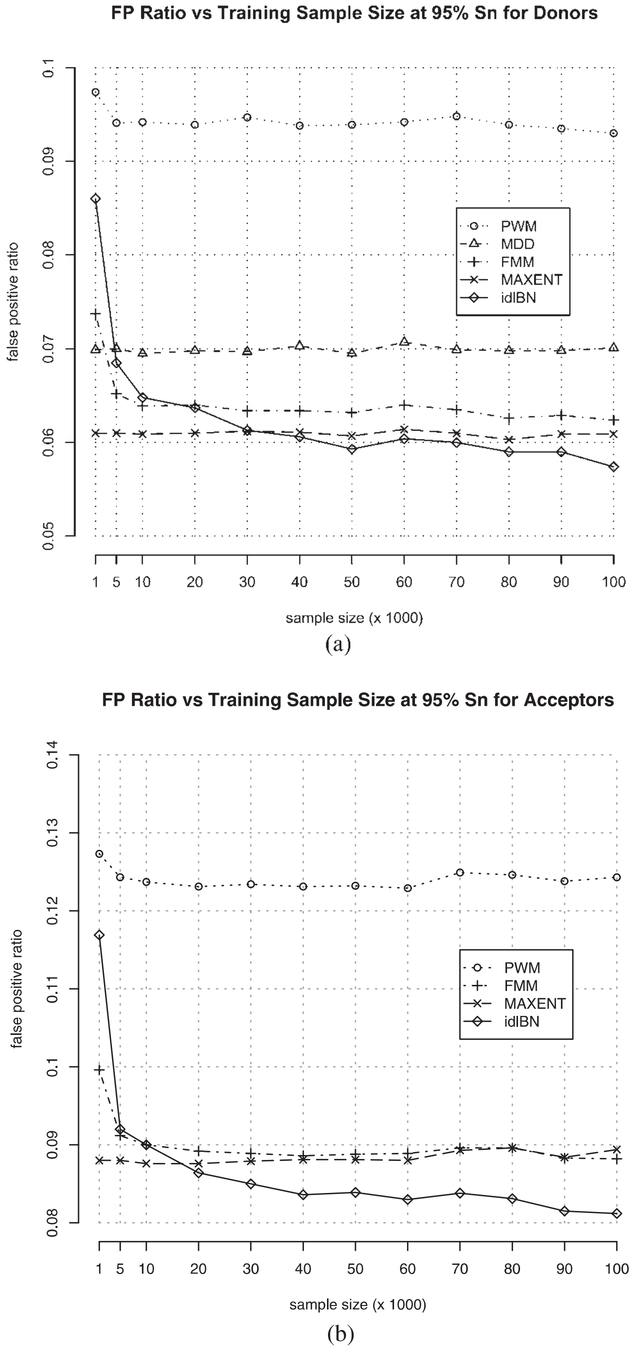

Fig. 1. False positive ratio versus the training sample size for donor **(a)** and acceptor **(b)** signals.

In Figure 1, we have plotted the false positive ratio, at a sensitivity level of 95%, as a function of the training sample

[5] **NO** **B**urset, **G**uigó, or **RO**gic **R**ef**S**eq genes.

[6] http://genes.mit.edu/burgelab/maxent/Xmaxentscan_scoreseq.html

Table 1. AUC values for internal exon prediction by LRE and CLR

False exon	Exon	AUC values			*idl*BN gain (%)	
(CLR gain)	score by	PWM	FMM	*idl*BN	PWM	FMM
FA–FD	LRE	0.9831	0.9854	**0.9898**	0.7	0.4
(0.0%)	CLR	0.9843	0.9856	0.9892	0.5	0.4
TA–FD	LRE	0.9101	0.9332	0.9602	5.5	2.9
(2.0%)	CLR	0.9395	0.9517	**0.9672**	2.9	1.6
FA–TD	LRE	0.9606	0.9669	0.9672	0.7	0.0
(0.4%)	CLR	0.9653	**0.9712**	0.9711	0.6	0.0

Gain percentage of CLR over LRE and *idl*BNs over PWM and FMM are specified. Largest AUC values are in boldface.

size for donor (a) and acceptor (b) sites. For both types of signals we observe the following. The PWM method is the one that performs worst at a substantial distance from the rest. The differences among the other methods, FMM, MDD, MAXENT and *idl*BN, lie in a range of ∼1% of the false positive ratio, where the method introduced in this paper, the splice site identification by *idl*BNs, achieves in both types of signals the best false positive ratio, after 20 000 to 30 000 training sites. Besides this fact, the *idl*BN method reduces much more rapidly the false positive ratio as the training sample size increases. This shows empirically how not only the variance, but also the bias is clearly reduced with a larger sample size. This is remarkable as donor and acceptor sites are data not necessarily sampled from DAG-distributions.

3.3 Internal exon prediction

We assess internal exon prediction based exclusively on the identification of the splice sites that determine the exon boundaries. We test the two exon scoring methods discussed in Subsection 2.3, the LRE from Expression (8) and the CLR from Expression (12).

We have built PWMs, FMMs and *idl*BNs for donor and acceptor sites from the NOBGRORS dataset, and test the six combinations of splice site and exon scoring methods, in the BGROIEXONS sets of internal true and false exons. We computed receiver operating characteristic (ROC) curves formed by the entire range of sensitivity values (using intervals of 5%) and the false positive ratio. In order to summarize these results succintly, we have calculated[7] the Area Under the ROC curve (AUC). The AUC takes values between 0.5 and 1.0 where 0.5 means that the the two scores distributions do not differ, and a value of 1.0 indicates that the scores distributions do not overlap.

In Table 1 we show the AUC values, organized to show what combination of scoring methods was used, and over what type of false internal exon, where FA–FD indicates that both splice sites are false, TA–FD indicates a true acceptor and a false donor, and FA–TD indicates a false acceptor and a true donor. Moreover, we have included the proportional gains in AUC for CLR with respect to LRE, and *idl*BNs with respect to PWMs and FMMs.

We see that the use of *idl*BNs in splice site identification improves exon prediction in all three different types of false exons, ranging from 0.4% up to 5.5%, except when FMMs are used and the false exons have a true donor site. In this latter case, there is a gain (LRE) and a loss (CLR) smaller than 0.1%. This is surprising as there is a clear gain of 1.6% up to 2.9% when the false exon has a true acceptor (TA–FD), and we are currently investigating this situation. In general, the proportional gains may seem small but the gains of FMM over PWM are not larger than $2.5\% = [(0.9332/0.9101) - 1] * 100$, while it is generally observed that FMMs afford a substantial improvement over PWMs (e.g. Fig. 1).

In the first column of Table 1 we have included the average gain in using CLR instead of LRE when combining the scores of the splice sites. When the false exon has both splice sites false (FA–FD) there is no gain, which makes sense as the false exon is already correctly modeled with LRE. However, in the other two cases, TA–FD and FA–TD, we obtain 2.0 and 0.4%, respectively. The positive effect of using CLR is also supported by the observation that the gains in using *idl*BNs instead of PWMs or FMMs, are smaller when in all three cases CLR is used, which means that it definitely helps in predicting exons.

3.4 Gene prediction

The splice site identification by *idl*BNs can be incorporated in any prediction algorithm that uses the likelihood ratio. We have used it within a particular gene prediction program, `geneid v1.1`. The scoring scheme for exons in `geneid` is by LRE, but we have also implemented the exon score by CLR. The resulting version is by now only a prototype and not yet publicly available. The default parameter file, uses a PWM for donor sites (9 bp), a FMM for acceptor sites (27 bp), a second order Markov model for start sites (20 bp) and any codon matching a stop codon is considered as a candidate stop site. These models within this default parameter file were trained using different datasets downloaded on August 2000. For the donor and acceptor sites the Intron–Exon database (Saxonov *et al*., 2000) was used, and for the start sites and the coding-bias statistics (which are implemented as a Markov model of order 5), the Human Transcript Database[8] was used. When using this parameter file only, we will refer to as the `default` configuration.

We show results on gene prediction for two well-known benchmark datasets, the BG-570 and the HMR-175. We have

[7] Using the `pAUCi` function from the Bioconductor `ROC` library at http://www.bioconductor.org/repository/devel/package/html/ROC.html

[8] Available on http://www.hgsc.bcm.tmc.edu/HTDB

Table 2. Comparison of the different configurations of `geneid` for the BG-570 and the HMR-175 gene datasets

configuration	SNn		SPn		CCn		SNe		SPe		SNSPe		SNg		SPg		SNPg		MG		WG	
	cb	ncb	cb	ncb	cb	ncb	cb	ncb	cb	ncb	cb	ncb	cb	ncb	cb	ncb	cb	ncb	cb	ncb	cb	ncb
BG-570																						
Default v1.1	0.90	0.53	**0.89**	0.63	0.88	0.51	0.67	0.42	0.70	0.33	0.69	0.38	0.14	0.02	0.12	0.01	0.13	0.01	0.03	**0.09**	**0.12**	0.31
PWM-LRE	0.92	0.45	0.86	0.62	0.87	0.46	0.67	0.36	0.67	0.32	0.67	0.34	0.16	0.03	0.13	0.02	0.14	0.03	**0.02**	0.13	0.16	0.36
PWM-CLR	0.91	0.46	0.88	0.65	0.88	0.49	0.67	0.36	0.67	0.35	0.67	0.36	0.15	0.02	0.13	0.01	0.14	0.02	0.03	0.13	0.14	**0.30**
FMM-LRE	**0.93**	0.50	0.87	0.67	0.88	0.52	0.71	0.41	0.70	**0.37**	0.70	0.39	0.21	0.04	0.17	0.02	0.19	0.03	**0.02**	0.10	0.18	0.36
FMM-CLR	0.92	0.47	0.88	**0.72**	**0.89**	**0.54**	**0.73**	0.38	0.70	0.43	0.71	**0.41**	**0.22**	0.02	**0.19**	0.02	**0.20**	0.02	0.03	0.12	0.17	0.27
*idl*BN-LRE	**0.93**	**0.56**	0.87	0.61	0.88	0.52	0.72	**0.47**	0.71	0.34	**0.72**	**0.41**	0.21	**0.06**	0.17	**0.04**	0.19	**0.05**	0.03	0.11	0.19	0.40
*idl*BN-CLR	0.92	0.54	**0.89**	0.64	**0.89**	0.52	0.72	0.45	**0.72**	**0.37**	**0.72**	**0.41**	0.21	0.05	0.18	0.03	0.19	0.04	0.03	0.11	0.16	0.35
HMR-195																						
Default v1.1	**0.92**	0.30	**0.87**	0.70	**0.88**	0.44	0.70	0.28	**0.71**	0.44	**0.70**	0.36	0.18	0.00	0.15	0.00	0.16	0.00	**0.01**	0.27	0.18	**0.30**
PWM-LRE	**0.92**	0.30	0.84	0.60	0.86	0.38	0.69	0.29	0.65	0.32	0.67	0.31	0.21	0.00	0.15	0.00	0.18	0.00	**0.01**	0.32	0.23	0.44
PWM-CLR	0.90	0.35	0.86	0.58	0.86	0.39	0.68	0.35	0.68	0.31	0.68	0.33	0.18	0.00	0.15	0.00	0.17	0.00	**0.01**	0.30	**0.17**	0.46
FMM-LRE	0.91	0.34	0.86	0.67	0.87	0.44	0.68	0.34	**0.71**	0.38	0.69	0.36	0.19	**0.01**	0.15	**0.01**	0.17	**0.01**	**0.01**	0.27	**0.17**	0.42
FMM-CLR	**0.92**	0.29	0.86	**0.71**	0.87	0.43	**0.72**	0.30	0.69	**0.45**	**0.70**	**0.37**	0.22	**0.01**	0.17	0.00	0.20	0.00	**0.01**	0.32	0.20	0.39
*idl*BN-LRE	0.91	0.34	0.86	0.69	0.86	**0.45**	0.69	0.32	**0.71**	0.40	**0.70**	0.36	0.20	**0.01**	0.16	0.00	0.18	0.00	**0.01**	0.26	0.19	0.36
*idl*BN-CLR	0.91	**0.39**	0.86	0.63	0.87	0.44	0.71	**0.38**	0.69	0.36	**0.70**	**0.37**	**0.23**	**0.01**	**0.18**	0.00	**0.21**	0.00	**0.01**	**0.23**	0.19	0.39

Best values are in boldface.

trained PWMs, FMMs and *idl*BNs from the NOBGRORS dataset, which excludes genes that are similar to those in BG-570 and HMR-175. Note that, however, the default parameter file of `geneid` was trained on sequences that may include some of those within BG-570 and HMR-175. The three different types of models we trained (PWMs, FMMs and *idl*BNs) were used for start sites (20 bp), stop sites (15 bp), acceptor sites (23 bp) and donor sites (9 bp). Regarding the coding-bias statistics, we have used those from the default parameter file, and also we have considered performing gene prediction without using the coding-bias statistics to more clearly assess the improvement in signal identification.

The accuracy measures taken are: sensitivity and specificity at nucleotide (SNn, SPn), exon (SNe, SPe) and gene (SNg, SPg) level; the correlation coefficient (CCn) at nucleotide level; the proportion of genes totally missed in the prediction (MG); and the proportion of predicted genes totally wrong (WG). The sensitivity and specificity are averaged at exon (SNSPe) and gene (SNSPg) levels. These are standard measures used in the literature (Burset and Guigó, 1996).

A particular feature of `geneid` is that permits incorporating our prior odds for an exon [Parra *et al*. 2000, Expression (8)] in the so-called exon weight, which forces the program to make a smaller or larger amount of predictions according to smaller or larger values of this parameter. This allows the program to be more specific or more sensible.

Our main purpose here is to show to which extent a better splice site identification can improve gene prediction with `geneid`, and hence we have run the program with a wide range of different exon weights for each dataset and picked the run that achieves the best CCn in order to compare the best performances between the different configurations. We show this comparison in Table 2, and the reader may find the full set of results in the online Supplementary Information.

Table 2 shows the results separately for each test dataset and, within each accuracy measure, we include the result when using coding-bias statistics (cb) and when not using them (ncb). When coding-bias statistics are not used there is an improvement of 5–6% in the average sensitivity and specificity at exon level (SNSPe) between *idl*BN-LRE and PWM-LRE, which becomes smaller by 2% when CLR is used. This difference decreases more when comparing with FMMs, and specially FMM-CLR. A similar situation occurs with the CCn values where even FMM-CLR reaches a higher value than *idl*BNs in the BG-570 dataset. At gene level the performance is extremely poor in general, showing how important is the use of the coding-bias. The default configuration performs similarly to FMMs and *idl*BNs in the HMR-195 dataset, and slightly worse in the BG-570 dataset.

When coding-bias is used, the CCn values do not differ very much, and in fact the default configuration achieves the best value for the HMR-195 dataset and is only 1% smaller than FMM-CLR or *idl*BN-CLR in the BG-570 dataset. In this same dataset, a similar situation occurs with the SNSPe, which is, though, ∼3% worse in the BG-570 dataset. At gene level, there is a more clear difference in favor of FMMs and *idl*BNs. In the case of the BG-570 dataset, it reaches a 5% improvement in the average sensitivity and specificity (SNSPg) between PWMs and *idl*BNs-FMMs (LRE) and up to 7% between the default configuration and FMM-CLR. In the case of the HMR-195 dataset, *idl*BN-CLR achieves the highest SNSPg showing an improvement up to 5%.

4 CONCLUSION

In this paper we have introduced a novel method for the identification of functional sites, and concretely, splice sites (identification by *idl*BNs), which implements the likelihood ratio by first learning the structure of a Bayesian network from each corresponding training dataset by an inclusion-driven learning algorithm. In our case, we have used the HCMC algorithm (Kočka and Castelo, 2001; Castelo and Kočka, 2003). Bayesian networks were used previously for this purpose but the advantage of *idl*BNs is that their learning process is theoretically grounded (Castelo and Kočka, 2003, see Th. 3.4) and therefore optimal under the circumstances reviewed in this paper and more thoroughly investigated by Chickering (2002) and Castelo and Kočka (2003). We have observed through the experiments with donor and acceptor sites that the ratio of false positives decreases as the training sample size increases, at a higher rate than using PWMs or FMMs.

The use of *idl*BNs for the more general purpose of the discovery of DNA binding sites is likely to be successful within Bayesian network based procedures like the one from Barash *et al*. (2003) that greatly rely on the Bayesian network structure. However, the extent of the improvement will always depend on how much the training data is away from the assumption of being sampled from a DAG-distribution.

We have analyzed the problem of modeling properly a false exon and proposed a correction to the likelihood ratio, CLR, which clearly contributed to a better exon and gene prediction. In this latter task, the improvement afforded by *idl*BNs that we observed when predicting donor and acceptor sites diminished to the point where *idl*BNs and FMMs perform similarly where FMMs win in the BG-570 dataset, and *idl*BNs win in the HMR-195 dataset. From this, we may conclude that the complexity of the gene prediction problem does not make a straightforward job to carry an improvement in signal identification to an improvement in gene identification. This, in turn, implies that the potential improvement can greatly depend on the gene prediction program, and therefore it may be interesting to deploy *idl*BNs in other ones.

REFERENCES

Agarwal,P. and Bafna,V. (1998) Detecting non-adjoining correlations with signals in DNA. *Proceedings of the 2nd International Conference on Research in Computational Molecular Biology*, ACM Press, New York, pp. 2–8.

Barash,Y., Elidan,G., Friedman,N. and Kaplan,T. (2003) Modeling dependencies in protein–DNA binding sites. In Miller, W.(ed.), *Proceedings of the 7th International Conference on Research in Computational Molecular Biology*. ACM Press, New York, pp. 28–37.

Burge,C. (1998) Modeling dependencies in pre-mRNA splicing signals. In Salzberg,S., Searls,D. and Kasif,S. (eds), *Computational Methods in Molecular Biology*, Elsevier Science, Amsterdam, pp. 129–164.

Burset,M. and Guigó,R. (1996) Evaluation of gene structure prediction programs. *Genomics*, **34**, 353–367.

Cai,D., Delcher,A., Kao,B. and Kasif,S. (2000) Modeling splice sites with Bayes networks. *Bioinformatics*, **16**, 152–158.

Castelo,R. and Kočka,T. (2003) On inclusion-driven learning of Bayesian networks. *J. Mach. Learn. Res.*, **4**: 527–574.

Chernoff,H. (1952) A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Ann. Math. Stat.*, **23**, 493–507.

Chickering,D. (2002) Optimal structure identification with greedy search. *J. Mach. Learn. Res.*, **3**, 507–554.

Cover,T. and Thomas,J. (1991) *Elements of Information Theory*. Wiley Interscience, New York.

Dash,D. and Gopalakrishnan,V. (2001) Modeling DNA splice regions by learning Bayesian networks. *Technical report*, Center for Biomedical Informatics, University of Pittsburgh, Pittsburgh, PA.

DeGroot,M.H. (1970) *Optimal Statistical Decisions*. McGraw-Hill, New York.

Heckerman,D., Geiger,D. and Chickering,D. (1995) Learning Bayesian networks: the combination of knowledge and statistical data. *Mach. Learn.*, **20**, 194–243.

Kočka,T. and Castelo,R. (2001) Improved learning of Bayesian networks. In Breese, J. and Koller, D. (eds), *Proceedins of the Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, pp. 269–276.

Kullback,S. and Leibler,R. (1951) On information and sufficiency. *Ann. Math. Stat.*, **22**, 79–86.

Neyman,J. and Pearson,E. (1928) On the use and interpretation of certain test criteria for purposes of statistical inference. *Biometrika*, **20**, 175–240, 264–299.

Parra,G., Blanco,E. and Guigó,R. (2000) GeneID in *Drosophila*. *Genome Res.*, **10**, 511–515.

Pearl,J. (1988) *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA.

Rogic,S., Mackworth,A. and Ouellette,F. (2001) Evaluation of gene-finding programs on mammalian sequences. *Genome Res.*, **11**, 817–832.

Saxonov,S., Daizadeh,I., Fedorov,A. and Gilbert,W. (2000) Eid: the Exon–Intron Database—an exhaustive database of protein-coding intron-containing genes. *Nucleic Acids Res.*, **28**, 185–190.

Yeo,G. and Burge,C. (2003) Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals. In Miller, W. (ed.), *Proceedings of the 7th International Conference on Research in Computational Molecular Biology*, ACM Press, New York, NY, pp. 322–331.