Published online 16 February 2012 *Nucleic Acids Research, 2012, Vol. 40, No. 11* ***4765–4773***
doi:10.1093/nar/gks154

Exploiting mid-range DNA patterns for sequence classification: binary abstraction Markov models

Samuel S. Shepard[1], Andrew McSweeny[1,2], Gursel Serpen[3] and Alexei Fedorov[1,*]

[1]Department of Medicine, [2]Program in Bioinformatics and Proteomics/Genomics, University of Toledo, Health Science Campus, Toledo, OH 43614 and [3]Department of Electrical Engineering and Computer Science, University of Toledo, Toledo, OH 43606, USA

Received September 15, 2011; Revised January 18, 2012; Accepted January 25, 2012

ABSTRACT

Messenger RNA sequences possess specific nucleotide patterns distinguishing them from non-coding genomic sequences. In this study, we explore the utilization of modified Markov models to analyze sequences up to 44 bp, far beyond the 8-bp limit of conventional Markov models, for exon/intron discrimination. In order to analyze nucleotide sequences of this length, their information content is first reduced by conversion into shorter binary patterns via the application of numerous abstraction schemes. After the conversion of genomic sequences to binary strings, homogenous Markov models trained on the binary sequences are used to discriminate between exons and introns. We term this approach the Binary Abstraction Markov Model (BAMM). High-quality abstraction schemes for exon/intron discrimination are selected using optimization algorithms on supercomputers. The best MM classifiers are then combined using support vector machines into a single classifier. With this approach, over 95% classification accuracy is achieved without taking reading frame into account. With further development, the BAMM approach can be applied to sequences lacking the genetic code such as ncRNAs and 5′-untranslated regions.

INTRODUCTION

The application of gene prediction algorithms to whole-genome sequencing data produces hundreds of 'hypothetical' genes with low similarity to well-known genes and poor EST coverage. The quantities and boundaries of hypothetical genes annotated using different prediction algorithms have little overlap. Moreover, updated versions of the same gene-finder often produce drastically different results. For example, human annotated genes for the Build 35 (1) Build 36 (2007) and Build 37 (2009) genomic assemblies have little overlap in their hypothetical gene subsets. Therefore, there is a continuing need for improvement in gene prediction. For eukaryotes, a primary component of gene prediction is the discrimination of exonic and intronic sequences, which is the focal point of this article. While other factors such as identifying splicing junctions and promoters are important, they are beyond the scope of the current study.

Markov models (MMs) are widely used in bioinformatics, particularly for gene prediction (2,3). The *order* of a MMs is the number of contiguous nucleotides used to obtain the probability of occurrence of the next nucleotide in a sequence. Generally, the order of the MM ranges from order 0 (single nucleotide analysis with no memory) to about 5 (3,4). Larger order MMs tend to be more accurate. In order to train a MM of order 5, the frequencies of all possible 6-mer oligonucleotides within the training sequences must be determined. This is feasible with a training set comprising several hundreds of genes. For higher order MMs the size of the training set must be drastically increased. For example, MMs of order 9 (10-mer oligonucleotides) may require more than 10 000 genes in training set. Due to such an exponential growth in the training set, the limit of the order of MM usable for gene prediction is near 7. Thus, contemporary MMs are limited to analyzing short-range (1–8 bp) DNA patterns for sequence classification. However, longer mid-range (10–44 bp) sequence patterns are abundant and functionally important in the genomes of higher eukaryotes (5). These mid-range patterns are not adequately leveraged for sequence classification if one analyzes only the composition of 1–8 nt long oligonucleotides using conventional MM approaches.

In this study, we tailor homogeneous Markov chain algorithms to use mid-range genomic patterns for sequence classification. Specifically, our approach utilizes longer oligonucleotides (from 10 to 44 bp) within Markov model exon/intron classifiers by first abstracting the

*To whom correspondence should be addressed. Tel: +1 419 383 5270; Fax: +1 419 383 3102; Email: alexei.fedorov@utoledo.edu

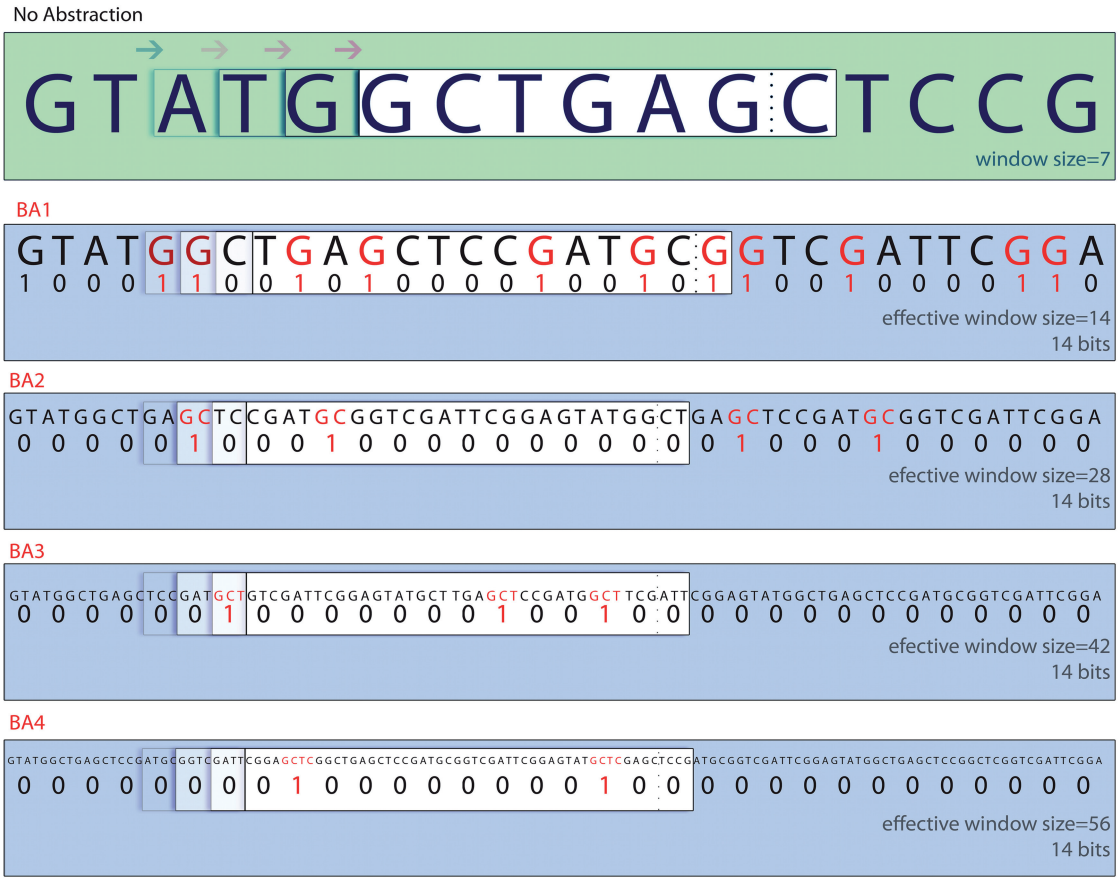

Figure 1. The abstraction of nucleotides into binary sequences starting at abstraction level 0 (no abstraction) up to binary abstraction level 4 (BA4). A sliding window for the homogeneous MM algorithm shows the information being processed on the binary level as well as the effective nucleotide coverage.

nucleotide information content into representative binary strings. Figure 1 explains various levels of nucleotide to binary conversions. The process for conversion of nucleotide letters into 0 or 1 digits is specified by an 'abstraction scheme.' The smallest level of abstraction (reducing the information present by about a half) occurs when each nucleotide is converted into a binary digit (zero or one). Abstraction schemes for single nucleotides are said to be at *binary abstraction level 1* or BA1 (second row in Figure 1). An example BA1 abstraction scheme would be to convert all purines to 0s and pyrimidines to 1s. After the conversion of exons and introns into binary strings, they are processed with the usual homogenous MM training algorithm and tested for the exon/intron discrimination ability. We refer to such Markov models trained with binary representations of nucleotide sequences as Binary Abstraction Markov Models (BAMM).

The highest abstraction level investigated in this study, BA4, shown in Figure 1, reduces each of 256 4-mer oligonucleotide words to a single 0 or 1. For example, the sequence fragment 'ATGCGGATAACC' with a specific abstraction scheme (ATGC → 0; GGAT → 1; AACC → 0; etc) can be converted into a three-digit-long '010' binary string. When using binary sequences as opposed to nucleotide sequences, the alphabet is reduced to from four characters to two. Therefore, longer MM orders are permissible, given the reduced number of possible words for a given order. A training set of several hundred genes is sufficient for use of 11-bit long binary strings (e.g. 11000101100) required for MM order 10. Such 11-bit strings can correspond to 11-mer oligonucleotides if converted at the BA1 abstraction level, and oligonucleotides up to length 44 at BA4.

There are an astronomical number of possible abstraction schemes that can convert nucleotide sequences into binary ones (for the BA4 abstraction level, $2^{256} \sim 10^{77}$ abstraction schemes are possible.) Using optimization algorithms on a supercomputer, we found optimal schemes for exon/intron discrimination at abstraction levels BA1 through BA4. We show that the best BA4 abstraction scheme (BA4-best) is more informative in terms of discrimination (87% accuracy) than a conventional order 5 homogenous MM that uses nucleotide sequences (84% accuracy).

We last demonstrate that it is possible to effectively combine several abstraction schemes into a very powerful exon/intron discriminator using a support vector machine (SVM) approach. This SVM-unified method achieves ~95% accuracy in exon/intron classification. Furthermore, this is achieved without the

reading-frame information required by an inhomogenous MM. After describing the development and use of BAMM in more detail, we will discuss the biological meaning of mid-range genomic patterns and approaches to utilize them in gene finders.

MATERIALS AND METHODS

BAMM

BAMM can be trained with much longer nucleotide sequence patterns than traditional MMs. This is achieved by converting nucleotide sequences into binary ones with the use of abstraction schemes. The binary sequence is then used as input for the Markov chain algorithm. The nucleotide information analyzed can effectively cover 11–44 bp (depending on the given parameters) and can emphasize specific aspects or distinguishing characteristics of the nucleotide information according to abstraction scheme applied. In this section, we discuss the algorithm and give the necessary nomenclature. More details on the mathematics and scoring of binary sequences using homogeneous Markov chains are given in Supplementary File S2.

The binary abstraction process. Let BAp be the binary-abstraction of nucleotides on the p-mer level. For $p = 1$ (an 'abstraction level' of 1 or 'BA1' for short) single nucleotides are converted to 0 or 1 according to a given *abstraction scheme*. An example BA1 scheme would be: 'if G then convert to 1, else 0'. The abstraction scheme is used to generate binary sequences from a set of training nucleotide data and build a Markov chain. After the Markov model has been built, it can be used to classify sequences that have been converted using the same abstraction scheme. At the BA1 level, four different nucleotides can be assigned to 0 or 1—allowing for a total of $2^{(4-1)} = 8$ unique abstraction schemes. At BA2, 16 (or 4^2) different dinucleotides can be abstracted to 0 or 1, so the total number of abstraction schemes increases to $2^{(4^2-1)} = 32,768$ schemes. For BA3, 64 possible triplets can be assigned to binary digits and the number of possible unique abstraction schemes expands to $2^{(4^3-1)} = 9.22\times10^{18}$. Accordingly, BA4 contains a total of 5.78×10^{76} unique abstraction schemes.

Datasets and databases

Exons, introns and untranslated regions were obtained from our exon–intron database (EID) (6). The EID was constructed from build 37 of the human genome. The coding sequences (CDS), UTR and intron datasets were pre-processed by removing redundant intron and exon sequences that had duplicate 100 bp 5′ or 3′ sequences, as well as sequences for hypothetical or unnamed genes.

All training and test sets were randomly selected from the filtered intron, exon or UTR subsets, respectively. Training data was used to build our model, while test data was used to measure the accuracy of our models. [Different sequences from a previous version of our database, as seen in ref. (7), were used to find the best abstraction schemes, hence we did not create an additional validation set.] The random sampling algorithm for generating datasets requires a threshold for the total number of nucleotides (typically 6 Mb for training sets and 3 Mb for test) and then randomly selects sequences from the database until that threshold is surpassed. A minimum sequence length can be specified; we used a minimum length of 45 bp for all datasets and ensured the training and test datasets had no overlap. Dataset details such as sequence count, nucleotide total and average sequence length are listed in Supplementary Table S2. Fewer 5′-UTR exons were available for sampling and hence the 5′-UTR exon test set was about half the usual size. Additionally, sequences containing non-canonical bases such as 'N' were excluded. Zero to two nucleotides were randomly removed from the beginning of coding exons to ensure that reading frame was not a factor in our BA3 analysis. The distribution of phases for human introns is not uniform: about 46% are in-frame, 32% are in phase 1, and ~22% are in phase 2 (8,9).

Context-dependent abstraction schemes

Context-dependent binary abstraction Markov models (CDBAMM) specify abstraction schemes based on the context of adjacent nucleotides windows. They are described in detail within ref. (7).

Among the immense number of possible context-dependent abstraction schemes (CDAS) we considered two with straightforward biological interpretations. The first one relates to single nucleotide insertions and deletions. Small insertions or deletions (indels) predominantly result in homonucleotide runs such as AAA or GGGG. The size and frequency of such runs should be different in coding exons versus the neighboring non-coding sequences (introns) due to the restrictions in the reading frame shifts for exons. We examined different context-dependent rules for the assignment of a binary digit to a nucleotide or dinucleotide, depending on the adjacent bases. The best CDAS for exon/intron discrimination presented in the 'Results' section is known as the DUP or duplication CDAS. The DUP scheme scans a sequence using two adjacent dinucleotide windows, generating a 1 when these windows are identical. For example, the sequence AGTCTCGGAATCGC can be separated into dinucleotide windows AG, TC, TC, GA, AA, TC and GC. The sequence would be converted to 01000, as the second window (TC) has the same sequence as the third window (TC).

The second considered subgroup of CDAS is based on the overabundance of purine (R) to pyrimidine (Y) alternations in the genomes of multicellular eukaryotes (5). The CDAS of this type for exon/intron discrimination is named 'YR' in Table 1. The YR scheme is exactly like DUP except that purine–pyrimidine parity is tested instead of nucleotide equality. For the same example sequence as before, non-overlapping dinucleotides windows AG, TC, TC, GA, AA, TC and GC would be converted to 010100, since the second (TC = YY) and third (TC = YY) as well as fourth (GA = RR) and fifth (AA = RR) windows now match in terms of their purine–pyrimidine composition. After a binary sequence

Table 1. A diverse selection of abstraction schemes are shown with their original accuracies versus their SVM optimized accuracies

Abstraction rule	Original			SVM-optimized		
	%EX	%IN	*M*	%EX	%IN	*M*
BA1-best	76%	78%	0.77	93%	72%	0.79
BA2-best	75	85	0.80	93	79	0.84
BA3-best	77	87	0.81	94	81	0.86
BA4-best	79	88	0.83	95	82	0.87
A priori 3	74	71	0.72	90	78	0.83
Pos. splicing	71	82	0.76	94	72	0.80
GT-rich (BA3)	66	83	0.73	94	72	0.80
Dupl. method	76	85	0.80	94	76	0.83
YR method	79	66	0.72	94	70	0.78
$\log_2$(AMI)	n/a	n/a	n/a	70	89	0.78
Nt. MM5	84	82	0.83	93	78	0.84

The SVM used a non-homogeneous polynomial kernel of degree 3 with normalization. The homogeneous Markov model of order 5 and the log average mutual information is also listed for comparison. Accuracies are listed as the percent correctly predicted exon, introns or *M*-value (which combines exon and intron accuracy, see 'Methods' section). Without SVM utilization, there was no pre-set decision boundary between introns & exons for AMI, making classification tests not applicable (n/a).

is generated via the context-dependent abstraction process, the homogeneous Markov chain algorithm can be employed for discriminating exons and introns.

Untrained variables

For certain choices of parameters and abstraction schemes, CDAS as well as non-CDAS models might start experiencing a significant number of *empty probabilities* (denoted $P_{\emptyset}$) when evaluating the test dataset. An empty probability is an untrained variable within the Markov chain (see Supplementary File S2). Having a large enough training set will usually ensure none of these $P_{\emptyset}$ will occur, but sufficient size is not always possible. There are two easy methods of addressing empty probabilities without lowering the Markov model order: ignoring any sequence in the test set with empty probabilities, or, ignoring the $P_{\emptyset}$ themselves during Markov chain computation. We prefer the latter method for CDBAMM and BAMM, although selecting an appropriate Markov model order (usually $k < 14$ for BAMM and $k < 7$ for CDBAMM) will minimize $P_{\emptyset}$, depending on the particular training set size and abstraction scheme. Unfortunately, even with a smaller Markov model order and a larger training set (12 Mb per group), empty probabilities may still occur because of the nature of the specified abstraction scheme. In such anomalous cases, ignoring the $P_{\emptyset}$ may cause some mild to severe inaccuracy in the model. In Table 1 all models produce zero to negligible (<0.01% over all variable *instances*) empty probabilities with the given parameters.

Measuring abstraction scheme accuracy

As described in ref. (7), optimized abstraction schemes at the BA1 and BA2 levels were obtained through an exhaustive search of all possible schemes. The BA3-best abstraction scheme was found using a binary particle swarm optimization (10) algorithm. The BA4-best abstraction scheme was jdentified using a hill-climbing algorithm, requiring optimized C code and extensive resources from the Ohio Supercomputer Center. For each possible abstraction scheme, a measure of overall accuracy, *M*, was used (Equation 1), combining sensitivity (SN) and specificity (SP) into one measure of classification accuracy. The metric is based on Euclidean distance from ideal SN and SP, using the axes for an ROC curve, as in (11). More details can be found in Supplementary File S2.

$$SN = \text{exon accuracy},\ SP = \text{intron accuracy}$$
$$M\text{-value} = 1 - \sqrt{\frac{(SN-1)^2 + (SP-1)^2}{2}} \quad (1)$$

Markov chains used as binary classifiers often have positive and negative as the two possible classifications. Here, a sequence classified as an exon is considered a positive. Therefore, the proportion of exons correctly classified (exon accuracy) is equal to sensitivity (SN) while the proportion of introns correctly classified (intron accuracy) is equal to specificity (SP). *M* is in the unit interval of [0, 1] with larger values indicating better prediction accuracy.

SVMs and model combination

The SVM was implemented using the SHOGUN interface (12) for Octave (a free open-source version of Matlab, see http://www.octave.org). As was recommended in the accompanying literature (13), a grid search was performed of the parameters on the Gaussian and then sigmoid kernels before selecting the polynomial kernel for our data domain. The optimal configuration for the SVM was a non-homogeneous polynomial kernel, of degree 3, with normalization turned on.

The SVM was trained and tested with whole sequence score values only and not with primary sequence data itself (such as the binary or nucleotide sequences). Although the dataset groups (exons and introns) were very similar in terms of total number of nucleotides, they differed widely in terms of number of sequences per group, as exons are much shorter than introns. The larger number of exon scores in the training set could bias the SVM toward exon detection versus intron detection and thus accuracy could be lost. Of the possible solutions mentioned in the literature for dealing with such an 'unbalanced dataset' (14), we chose to *down-sample* or reduce the number of exons to match the number of introns.

To train the SVM we used BAMM sequence scores (log-likelihoods) where the classifier was both trained and tested on the same dataset (one only has *a priori* knowledge of the identities of the training sequences in a real world application). In machine-learning terminology, each sequence in the self-tested training set will produce a data point with a vector of fields (scores) that correspond to a particular set of classifiers scores (e.g. BA3-best, BA2-best, etc.) for that sequence. Therefore, the dimensionality of the data is the number of classifiers used while the number of data points is with respect to the number of sequences classified. All 1165 introns were used along with the first [down-sampled] 1165

exons found in the training set. Since the training set is a random dataset, the selection of the first sequences should not bias the results in any way. This 1:1 ratio for the down-sampling produced the most balanced SVM training.

After the SVM was trained, a similar procedure was applied to test the overall accuracy of the SVM on the sequence data. First, the classifiers applied to the training of the SVM were used to score each sequence in the test dataset. (No down-sampling was needed for *testing* sequence scores.) Next, suppose two classifiers (e.g. BA3MM10 and AP3) produced scores of +1 and +2, respectively for a particular exon in the test set, then the SVM would score the exon based on the score vector (+1, +2) for that exon. If the SVM result was greater than zero, the sequence would be classified as exonic. This process continued until all sequences had been classified as positive or negative according to the SVM. The usual *M*-value and exon/intron accuracy could then be estimated for the chosen set of classifiers. In short, SVM are trained and tested using the predictions of other classifiers—such as BAMM—although it is not limited to classification based on these score values alone. Other data can be added, such as sequence length, splice site scores, etc. This sort of SVM methodology (on different types of biological sequence signals) is used in gene finding systems like mGene (15).

Finally, for Supplementary Table S3 K classifiers out of N analysis $[\binom{N}{K}]$ was performed to see which combination was the best. For each abstraction scheme, the optimal Markov model *order* was selected, given the individual SVM-optimized model. The best model combination was selected of the total $\binom{N}{K}$ classifiers for each fixed K. The nine abstraction schemes listed in Table 1 were used, plus the log average mutual information (16) of the sequences as a 10th SVM field value. Using a custom Octave script we tested every combination for $K = 1or9$ (10 combinations), $K = 2or8$ (45 combinations), $K = 3or7$ (120), $K = 4or6$ (210), $K = 5$ (252), and the single test for all 10 classifiers.

RESULTS

Finding best abstraction schemes

Individual approaches for identifying the best abstraction schemes at each abstraction level were chosen based on the number of possible schemes. In trivial situations, when single nucleotides (BA1 level) or dinucleotides (BA2 level) have been assigned to binary digits, the best schemes were found using exhaustive searches of all possible nucleotide-to-binary abstraction schemes. Since $2^{64} > 10^{18}$ abstraction schemes were possible for the 3-mer abstraction level (BA3); locating the 'best' abstraction scheme required a special computational approach to traverse the search space. Binary particle swarm optimization algorithm (17) was used to find a good optimum in a feasible amount of time. Similarly, due to the enormity of the 4-mer abstraction space, we performed three optimization trials utilizing a hill-climbing method on the Glenn supercomputer as described in ref. (7). The best map of these trials was selected to be the BA4-best map. Analysis of the BA4-best abstraction scheme showed that it could not be extended from the BA2-best scheme. In other words, the BA4-best scheme possesses additional information compared with BA2-best and thus supercomputer use could not have been avoided. Supplementary Table S1 shows each best optimal scheme for the four abstraction levels (BA1 to BA4). Interestingly, a search for best abstraction schemes for exon/intron classification at BA3 and BA4 levels did not result in the finding of several well-separated local maxima, rather all explorations led to a single prominent maxima plateau in the abstraction scheme space. In addition to the BA1-best, BA2-best, BA3-best and BA4-best abstraction schemes, five more schemes were created and examined for the exon/intron classification. Three of these, at the BA3 level, were designed using the biological rationale detailed in ref. (7). The first is an *a priori* abstraction scheme (AP3)—with 1 assigned to triplets more frequent in exons and 0 to triplets more abundant in introns of the training set (see Supplementary Table S1). The second is a triplet abstraction scheme sensitive to GT-rich patterns. In this scheme, a triplet is assigned to 1 if there are 2 or more G+T nucleotides, otherwise to 0. The third abstraction scheme was based on splicing potential information (POS SP, for POSitive SPlicing potential abstraction scheme, see Supplementary Table S1) and assigned 1s to triplets abundant in splicing enhancers while 0s to triplets predominant in splicing silencers (18). Finally, two context-dependent BAMM abstraction schemes sensitive to duplications (DUP) and purine–pyrimidine patterns were created as described in ref. (7). The unoptimized results on exon/intron discrimination for each of these nine abstraction schemes are presented in Table 1 under the Original column. For all BAMM models in Table 1 Markov order 10 was used. For comparison, we used two well-known exon/intron classifiers: the homogeneous nucleotide Markov model of order 5 and the log average mutual information (AMI) of the sequences (SVM was used to build a classifier for the AMI approach as well).

SVM optimization

A SVM machine-learning algorithm was used to optimize the exon/intron prediction accuracies of the individual models as shown in Table 1. These results are presented side-by-side with the unoptimized prediction accuracies for comparison. The SVM itself used a polynomial kernel of degree 3 to transform the data into a feature space. The non-homogeneous kernel option was also used with data normalization turned on for better accuracy and faster convergence. Accuracies were listed as the percent of correctly predicted exon, introns and as the *M*-value (which combines exon and intron accuracy, see the 'Methods' section). For each and every scheme some increase in *M*-value was reached due to the optimization process: a minimum of +0.02 for the BA1-best map to a maximum of +0.11 for the *a priori* 3 (AP3) BAMM abstraction scheme. This demonstrates the ability of the SVM to improve the accuracy of single classifiers. For comparison with a standard model, the

homogeneous nucleotide MM of order 5 achieved an exon accuracy of 84% with an intron accuracy of 82% and $M = 0.83$ while its SVM optimized value increased only slightly to $M = 0.84$.

The SVM optimization technique typically came with a trade-off (with the exception of AP3 and the purine–pyrimidine model). For the optimization to work, a few points of intron accuracy needed to be 'spent' in order to 'earn' a few extra points of exon accuracy. This trade-off between sensitivity and specificity is a common phenomena when trying to optimize a classifier without adding new information. For example, for the GT-rich model (a BA3 scheme), 11 percentage points of intron accuracy are lost in order to gain 28 points in exon accuracy. Clearly this trade-off is desirable. In the case of AP3 and the YR model, both intron and exon accuracy increase.

Table 2. The prediction accuracy of all 10 classifiers combined under a SVM polynomial kernel of degree 3

Order of MM	%Exon	%Intron	*M*
2	94.7	93.9	0.943
3	95.5	93.9	0.947
4	96.0	93.8	0.948
5	96.0	94.7	0.953
6	96.2	94.2	0.951
7	96.1	93.8	0.948
8	96.1	93.6	0.947
9	96.1	92.6	0.941
10	95.8	92.7	0.941

Accuracies are listed as the percent of correctly predicted exons, introns or *M*-value (which combines exon and intron accuracy, see 'Methods' section).

Combining multiple abstraction schemes using SVM

Using multiple sources of evidence increases predictive power in sequence classification. For example, the 'Statistical Combiner' program uses the predictions of multiple gene-finders to do gene prediction (19). Similarly, each BAMM classifier score was used as a field in a vector in order to compute an overall SVM classification (normalized, inhomogeneous polynomial kernel of degree 3). The optimized solution is robust in that it maximizes the distance between the two prediction classes (20). Table 2 shows the results of combining all nine abstraction schemes plus the log average mutual information field (as listed individually in Table 1) and varying by the MM order used for the BAMM classifiers. From Table 2, the highest performing SVM prediction is based on Markov order 5 data at an *M*-value of 0.953 for all 10 models. The individual optimized classifiers themselves peak at different MM orders and decrease beyond these orders, hence Markov order 5 is a reasonable value for the combined peak accuracy. Next the optimal MM orders were chosen for each of the nine BAMM maps using individual SVM optimized results and then combined all 10 models as before. This yielded an accuracy of $M = 0.953$ as well. In Supplementary Table S3 the best results are shown for each choice of K out of N classifiers and combined under SVM. The highest result was $M = 0.957$ for $K = 8$ models (ba2mm10, ba3mm9, GTm m2, POSmm5, AP3mm4, DUPw1j1mm7, YRw1j1m5 and AMI $\log_2$). $M > 0.95$ is achievable with only six models (ba2mm10, GTmm2, POSmm5, AP3mm4, DUPw1j1mm7, YRw1j1mm5), although the aggregation of larger numbers of models results expedites the identification of more accurate combinations of classifiers.

Using reading frame information

While it was our purpose to investigate datasets where reading frame was unknown or mixed, long open reading frames are an important property of coding exons sometimes exploited in gene finding algorithms in the early rounds of *ab initio* self-training (21). In ref. (7) BA3 schemes were formulated based on codon usage and bias patterns as well as for stop codons. (Codon bias is relative to the amino acid and the first two codon positions while codon usage is relative to the codon frequency over the whole table of triplets.) The best result was a taa/tag/tga-abstraction scheme at MM12 with an exon accuracy of 94% and an intron accuracy of 93% (*M*-value of 0.93) using an in-frame only dataset, however, accuracy dramatically decreases when using the other two reading frames ($M = 0.59$ and 0.58, respectively). Codon usage bias abstraction schemes accuracies (data not shown) fluctuate a little less with respect to reading frame, but are less accurate overall. Interestingly, many models, like the BA3-best abstraction scheme (M from 0.84 to 0.82 for three fixed reading frame datasets) and the AP3 ($M \sim 0.75$) abstraction scheme, operate in a near frame-independent manner.

Using abstract patterns in UTR data

In addition to discriminating introns versus coding exons, sequence classification of introns versus untranslated exons instead was also explored. UTRs are more difficult to find than coding sequences because they lack the structure and periodicity of the genetic code. However, splicing signals should still be retained for proper mRNA processing to occur.

Supplementary Table S4 shows the difficulty of using triplet abstraction schemes trained on CDS exons to predict both 5′- and 3′-UTR exons.

For 5′-UTR exons, the most accurate CDS-trained model is the positive splicing potential abstraction scheme at $M = 0.81$. A CDS-trained homogenous nucleotide Markov model of order 5 exceeds the tested unoptimized BAMM classifiers at $M = 0.84$ when predicting for introns and 5′-UTR exons. However, using our SVM optimization strategy, the individual BA3-best abstraction scheme rises to $M = 0.84$ as well. Using SVM to combine different models, the value further increases to $M = 0.88$ with an exon accuracy of 89% and an intron accuracy of about 88% [BA1MM10 + BA2MM10 + BA4MM10 + GT-rich abstraction scheme + positive splicing potential model + the YR CDAS + $\log_2(AMI)$, all with an SVM polynomial kernel of degree 3].

Unlike 5′-UTR exons, the oligonucleotide composition of 3′-UTR exons are much more similar to introns than to

coding sequences. Hence, neither BAMM nor ordinary MM are effective for 3′-UTR /intron discrimination using CDS-trained data [for details see (7)].

Given the compositional differences of 5′-UTR regions from other genomic segments, it is likely that that mid-range abstract patterns can be used for sequence discrimination of 5′-UTR exons but would require UTR training datasets and custom abstraction schemes.

DISCUSSION

Comparing BAMM to other algorithms

This study should not be regarded entirely as a research paper, but a one containing an important methodological component. As a tool, BAMM is still in its infancy and requires some effort to be applied by others. Currently we are working to extend BAMM into the form of an easy-to-use public tool or, at the least, to collaborate to integrate it with existing methods.

The results reveal that genomic oligonucleotide segments can be assigned to binary digits (0 or 1) such that their arrangement along the sequence is highly informative for determination of its functionality. The concept of binary abstraction is introduced as a new paradigm for processing nucleotide sequence data before utilizing it for gene prediction. We believe that the usage of BAMM to help classify genomic segments has a very promising future in the development of powerful gene prediction methods.

The accuracy of exon/intron discrimination in *de novo* gene prediction has been reviewed by (22) for several modern algorithms. The highest reported accuracy was 82.7% exon-specificity obtained by the CONTRAST algorithm. Accuracy of these algorithms depends on the species under consideration. The human genome has a mosaic isochore structure, making gene prediction more difficult than with shorter, more homogeneous genomes like those of *Drosophila*, *Caenorhabditis elegans* or *Arabidopsis*. According to ref. (23), a review of state-of-art *ab initio* and comparative gene finding approaches, the highest accuracy (85%, fig 16.1) for human exon predictions belongs to the JIGSAW program, which combines all available evidence (*ab initio* sequence analysis, similarity from related genomes, and expression data). In ref. (15) an SVM-based gene prediction system called mGene is presented, and, according to the authors, achieved the highest exon accuracy recorded (88%) on nematode genes. Measuring the accuracy of gene finders for comparison is therefore a difficult task and subject to many variables.

For our purposes, it is important to note that full gene-finders such as GeneMark (21,24) parse a sequence into various segments (exons, introns and intergenic regions) according to start and splice sites. Correctly identified 5′ and/or 3′ exonic splice sites are then used to measure accuracy. Our method, at this point, only classifies existing whole exons and introns with the splice sites already assumed. Thus, a direct comparison of BAMM with programs that produce a sequence parse is difficult at this stage of development. Instead, Table 1 shows direct comparison of the SVM-optimized performance of our BAMM algorithms with two well-known phase-independent alternatives for exon/intron discrimination: a homogeneous nucleotide Markov model of order 5 (MM5) and the average mutual information (AMI) of the nucleotide sequences (16). The exon/intron discrimination accuracy for the AMI algorithm (Table 1, 78%) corresponds well to the accuracy of this algorithm presented by its authors (76.1% for a 108 bp window and 80.7% for a 162 bp window (16). The accuracy for MM5 (Table 1, 84%) is much higher than the accuracy for 'hexamer' usage of 74.2% reported by Grosse and co-authors (2000) (16) since the decision boundary is calculated using a SVM and has splice sites assumed. These results provide an unbiased comparison of the various BAMM approaches with both the MM5 and AMI algorithms, demonstrating a performance advantage of certain BAMM abstraction schemes over other approaches. Similar results were obtained in ref. (7) using a different sampling of an earlier genome build demonstrating that our comparisons are independent on gene sampling. All in all, the 95% exon prediction accuracy for SVM-combining of independent BAMM schemes is competitive with modern gene prediction approaches and merits further development, testing and verification. The main disadvantage of BAMM is the computationally intensive identification of the best abstraction schemes for a new genome. Hence, we have published our human abstraction schemes on our web-site in order to provide a good starting point for finding new species-specific schemes.

Deciphering genomic signals

It is counterintuitive that a considerable reduction in the genomic sequence information can be used to produce better exon/intron classification accuracy than the use of the original nucleotide sequences. Possible explanations for the power of BAMM require further elaboration.

The human genome has prominent long-range non-randomness—the GC-isochore structure (25,26). In general, genes located within GC-rich chromosomal segments have short introns and substantial codon usage bias. Not only the introns but exons are GC-rich. Conversely, human genes located inside AT-rich isochores are lower in GC-content for exons and introns and often have an inverted codon usage bias. Many gene-finder programs do not separate chromosomal regions with distinct properties and as a result their training sets contain sequences from all parts of the genome. Thus, the oligonucleotide frequencies generated from the training data could vary significantly from oligonucleotide frequencies derived from a particular genomic locus. While occasional errors in gene-prediction are unavoidable due to the mosaic structure of the genome, sequence abstraction may counteract fluctuations in oligonucleotide frequencies. For example, a BA1 abstraction scheme converting purines to 1s and pyrimidines to 0s is not necessarily sensitive to GC-compositional variation since GC-rich and AT-rich regions may have the same purine/pyrimidine composition.

A specific subset of oligonucleotides, exonic and intronic splicing enhancers and silencers, are critical for splicing. Many of the remaining oligonucleotides may be less informative to pre-mRNA processing. Exonic splicing enhancers are binding sites for a group of proteins, belonging to the SR-family (27). These proteins bind to pre-mRNAs at the RNA-binding domain, and bind other proteins at the RS-domain, comprised of a series of alternating arginine and serine residues. These interactions result in SR-proteins forming a net over many exons that is essential for proper splicing. More than a hundred documented cases exist where a single mutation within a splicing enhancer converts an exon into an intron (28). Therefore, the arrangement of splicing enhancers is essential for exon/intron discrimination by spliceosomal machinery. Some of our computer-selected abstraction schemes (such as the BA3-best scheme) may effectively detect functional sequences (like exonic splicing enhancers) that are hidden by the informational noise of the rest of oligonucleotides. In addition, we designed a specific abstraction scheme using available but limited data on putative exonic splicing enhancers and silencers (Positive splicing scheme, Table 1). This scheme demonstrates good exon/intron discrimination power with an accuracy (M-value) of 80%, however, it is significantly behind BA3-best and BA4-best computer-selected abstraction schemes with percent M-values of 86 and 87%, respectively.

An essential step in the discrimination between exon/intron segments by spliceosomal machinery is based on the density and quality of splicing enhancer/silencer signals. While designing our computational experiments, we hypothesized that increasing the order of BAMM as high as 10 might considerably improve the accuracy in exon/intron classification by covering an oligonucleotide range that could overlap several exon and intron splicing enhancers/silencers. Using order 10 BAMM, we sought regularities in the arrangement of splicing enhancers/silencers in the 10–44 nt range (that corresponds to BA1–BA4 abstraction levels). However, we observed only a minor increase in the exon/intron discrimination ability when the order of the BAMM models was raised from 2 to 5 (see Table 2) and negligible improvement in classification beyond order 5. We interpret these results to mean that no important arrangement in putative splicing enhancers/silencers in the range of 20–40 bp are detectable using BAMM. Presumably, the discrimination between exon/intron segments by spliceosomal machinery is based on the density and quality of splicing enhancer/silencer signals.

Each of the abstraction schemes identified in our experiments leveraged different compositional properties of exonic and intronic sequences to perform classification. It was therefore of interest to use multiple abstraction schemes, combining their individual predictive power to build a single classifier. For this purpose we used a support vector machine approach, which allowed us to raise individual SVM optimized accuracies (e.g. M-value = 87% for the single BA4-best scheme) up to 95.6% using a combination of 8 different abstraction schemes. At a whole sequence classification level, this value is among the best of modern exon/intron computational discriminators, though it may not be the true upper limit for the combined classifier approach.

Final remarks

Thus far we have studied only regular abstraction schemes from 1 to 4 nucleotide levels (BA1–BA4), while context-dependent schemes (such as the DUP abstraction scheme, Table 1) have not been thoroughly investigated. There are many possible context-dependent abstraction schemes that could be developed to reflect different features of nucleotide sequences, such as RNA-folding properties. Only a very small group of context-dependent abstraction schemes relating to small insertions/deletions were studied. Examination of context-dependent schemes and identification of optimal BA5 abstraction schemes are possible future directions for study. Additionally, we have considered only homogeneous models where reading frames were disregarded. For the BA3 level of abstraction, where triplets are converted to binary digits, the transition to an inhomogeneous Markov model is straightforward. The preliminary data for utilizing reading frame information for BA3 are reported in ref. (7). On the other hand, utilization of the reading frame information is not trivial for abstraction levels where dinucleotides or 4-mers are converted to a single bit. Various approaches to account for shifts between abstraction frames and protein-coding reading frames are currently in development. Nonetheless, homogenous Markov models are the one and only choice for nucleotide sequences that do not possess protein-coding reading frames, such as ncRNAs and untranslated exonic regions of mRNA, presenting an additional area for the future investigation of BAMM.

Availability

Source code for algorithms, datasets and related files can be found online at our BAMM website http://bpg.utoledo.edu/bamm/.

SUPPLEMENTARY DATA

Supplementary data are available at NAR Online: Supplementary Tables 1–4, Supplementary methods.

ACKNOWLEDGEMENTS

We wish to thank the Ohio Supercomputer Center for their assistance with the use and operation of the Glenn Supercomputer. We are grateful to the Mark Borodovsky lab at Georgia Tech for their guidance with Markov models and of *a priori* abstraction schemes. We thank Vadim Filatov, Dinom LLC for his guidance in the development of context-dependent abstraction schemes. We greatly appreciate Craig Zirbel, of Bowling Green State University, for his insightful suggestion of using SVMs to combine classifiers. Peter Bazeley, of the University of Toledo, provided technical support as well as advice on optimization methods. Jared Rosenberg provided graphic design assistance for Figure 1. Finally we would like to recognize Sadik Khuder, of the University of Toledo for his counsel with many issues relating to statistics.

FUNDING

National Science Foundation (0643542, to A.F.). Funding for open access charge: NSF MCB-0643542.

Conflict of interest statement. None declared.

REFERENCES

1. Consortium,I.H.G. (2004) Finishing the euchromatic sequence of the human genome. *Nature*, **431**, 931–945.
2. Do,J.H. and Choi,D.-K. (2006) Computational approaches to gene prediction. *J. Microbiol.*, **44**, 137–144.
3. Guigo,R. (1998) DNA composition, codon usage and exon prediction. *Informatica*. Academic Press, pp. 53–79.
4. Borodovsky,M. and Mcininch,J. (1993) GENMARK: parallel gene recognition for both DNA strands. *Computers Chemistry*, **17**, 123–133.
5. Fedorova,L. and Fedorov,A. (2011) Mid-range inhomogeneity of eukaryotic genomes. *Scientific World J.*, **11**, 842–854.
6. Shepelev,V. and Fedorov,A. (2006) Advances in the exon-intron database (EID). *Brief Bioinform.*, **7**, 178–185.
7. Shepard,S.S. (2010) The characterization and utilization of middle-range sequence patterns within the human genome. *Ph.D. Thesis*.. University of Toledo / OhioLINK.
8. Fedorov,A., Suboch,G., Bujakov,M. and Fedorova,L. (1992) Analysis of nonuniformity in intron phase distribution. *Nucleic Acids Res.*, **20**, 2553–2557.
9. Ruvinsky,A., Eskesen,S.T., Eskesen,F.N. and Hurst,L.D. (2005) Can codon usage bias explain intron phase distributions and exon symmetry? *J. Mol. Evol.*, **60**, 99–104.
10. Kennedy,J. and Eberhart,R.C. (1997) *A discrete binary version of the particle swarm algorithm*. In Proceedings of Systems, Man, and Cybernetics, 1997. IEEE International Conference on Computational Cybernetics and Simulation, Vol. 5. IEEE Press, Piscataway, NJ, pp. 4–8.
11. Sboner,A., Eccher,C., Blanzieri,E., Bauer,P., Cristofolini,M., Zumiani,G. and Forti,S. (2003) A multiple classifier system for early melanoma diagnosis. *Artif. Intell. Med.*, **27**, 29–44.
12. Sonnenburg,S., Bernhard,B.S., Bennett,P. and Parrado-hernández,E. (2006) Large scale multiple kernel learning. *J. Mach. Learn. Res.*, **7**, 2006.
13. Hsu,C.W., Chang,C.C. and Lin,C.J. (2003) A practical guide to support vector classification. Department of Computer Science and Information Engineering, National Taiwan University Taipei, Taiwan.
14. Provost,F. (2000) Machine learning from imbalanced data sets 101. In *Proceedings of the AAAI-2000 Workshop on Imbalanced Data Sets*. Austin, Texas.
15. Schweikert,G., Zien,A., Zeller,G., Behr,J., Dieterich,C., Ong,C.S., Philips,P., De Bona,F., Hartmann,L., Bohlen,A. *et al.* (2009) mGene: accurate SVM-based gene finding with an application to nematode genomes. *Genome Res.*, **19**, 2133–2143.
16. Grosse,I., Buldyrev,S.V., Stanley,H.E., Holste,D. and Herzel,H. (2000) Average mutual information of coding and noncoding DNA. *Pac. Symp. Biocomput.*, 614–23.
17. Lee,S., Park,H. and Jeon,M. (2007) Binary particle swarm optimization with bit change mutation. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, **E90-A**, 2253–2256.
18. Bechtel,J.M., Rajesh,P., Ilikchyan,I., Deng,Y., Mishra,P.K., Wang,Q., Wu,X., Afonin,K.A., Grose,W.E., Wang,Y. *et al.* (2008) Calculation of splicing potential from the alternative splicing mutation database. *BMC Res. Notes*, **1**, 4.
19. Allen,J.E., Pertea,M. and Salzberg,S.L. (2004) Computational gene prediction using multiple sources of evidence. *Genome Res.*, **14**, 142–148.
20. Han,J. and Kamber,M. (2006) *Data Mining: Concepts and Techniques*, 2nd edn. Morgan Kaufmann, San Francisco, CA.
21. Ter-Hovhannisyan,V., Lomsadze,A., Chernoff,Y.O. and Borodovsky,M. (2008) Gene prediction in novel fungal genomes using an ab initio algorithm with unsupervised training. *Genome Res.*, **18**, 1979–1990.
22. Flicek,P. (2007) Gene prediction: compare and CONTRAST. *Genome Biol.*, **8**, 233.
23. Picardi,E. and Pesole,G. (2010) Computational methods for ab initio and comparative gene finding. *Methods Mol. Biol.*, **609**, 269–284.
24. Lukashin,A.V. and Borodovsky,M. (1998) GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Res.*, **26**, 1107–1115.
25. Costantini,M., Cammarano,R. and Bernardi,G. (2009) The evolution of isochore patterns in vertebrate genomes. *BMC Genomics*, **10**, 146.
26. Bernardi,G. (2000) Isochores and the evolutionary genomics of vertebrates. *Gene*, **241**, 3–17.
27. Shepard,P.J. and Hertel,K.J. (2009) The SR protein family. *Genome Biol.*, **10**, 242.
28. Bechtel,J.M., Rajesh,P., Ilikchyan,I., Deng,Y., Mishra,P.K., Wang,Q., Wu,X., Afonin,K.A., Grose,W.E., Wang,Y. *et al.* (2008) The alternative splicing mutation database: a hub for investigations of alternative splicing using mutational evidence. *BMC Res. Notes*, **1**, 3.