New Methods for Splice Site Recognition*

S. Sonnenburg[1], G. Rätsch[2], A. Jagota[3], and K.-R. Müller[1,4]

[1] Fraunhofer FIRST, Kekuléstr. 7, 12489 Berlin, Germany
[2] Australian National University, Canberra, ACT 0200, Australia
[3] University of California at Santa Cruz, CA 95064, USA
[4] University of Potsdam, August-Bebel-Str. 89, 14482 Potsdam, Germany

Abstract. Splice sites are locations in DNA which separate protein-coding regions (exons) from noncoding regions (introns). Accurate splice site detectors thus form important components of computational gene finders. We pose splice site recognition as a classification problem with the classifier learnt from a labeled data set consisting of only local information around the potential splice site. Note that finding the correct position of splice sites without using global information is a rather hard task. We analyze the genomes of the nematode Caenorhabditis elegans and of humans using specially designed support vector kernels. One of the kernels is adapted from our previous work on detecting translation initiation sites in vertebrates and another uses an extension to the well-known Fisher-kernel. We find excellent performance on both data sets.

1 Introduction

Splice sites are locations in DNA at the boundaries of exons (which code for proteins) and introns (which do not). The more accurately a splice site can be located, the easier and more reliable it becomes to locate the genes – hence the coding regions – in a DNA sequence. For this reason, splice site "detectors" are valuable components of state-of-the-art gene finders [4, 13, 21, 14, 5]. Furthermore, since ever-larger chunks of DNA are to be analyzed by gene finders the problem of accurate splice site recognition has never been more important.

Although present-day splice site detectors are reported to perform at a fairly good level [13, 12, 7], several of the reported performance numbers should be interpreted with caution, for a number of reasons. First of all, these results were based on small data sets of a limited number (one or two) organisms. Now that large, complex genomes have been fully sequenced, these results will need to be re-evaluated. Second, issues in generating negative examples (decoys) were, if recognized, not adequately documented.[1] Third, the results are expected to be highly dependent on the chosen window size. (The window defines the extent of the context around a site's boundary used during training and classification.) Since the different studies [13, 12, 7] chose different window sizes, and we choose a fourth different one; unfortunately no pair of these studies is directly comparable. Finally, some works [12] highlighted their accuracy ($\sim$error-rate) results. These can paint an overly optimistic picture on this problem (also [2]).

Support vector machines (SVMs) (e.g. [20, 11, 15]) with their strong theoretical roots are known to be excellent algorithms for solving classification problems. To date, they have been applied only to a handful of Bioinformatics problems (see e.g. [10, 22, 8, 19]). In this paper we apply SVMs to two binary classification problems, the discrimination of donor sites (those at the exon-intron boundary) from decoys for these sites, and the discrimination of acceptor sites (those at the intron-exon boundary) from decoys for these sites. We evaluate our SVMs on two

* We thank for valuable discussions with A. Zien, K. Karplus and T. Furey. G.R. would like to thank UC Santa Cruz for warm hospitality. This work was partially funded by DFG under contract JA 379/9-2, JA 379/7-2, MU 987/1-1, and NSF grant CCR-9821087. This work was supported by an award under the Merit Allocation Scheme on the National Facility of the Australian Partnerschip for Advanced Computing.

[1] To our knowledge, on the splice site recognition problem, only the work of [13] explicitly documented the care it exercised in the design of the experiments.

data sets: (a) on the IP-data data set (a relatively old data set of human splice sites with weak decoys) where the SVM method outperforms nine other methods, including a recent one [12] and (b) on splice sites extracted from the *complete* C. Elegans genome [1]. Also here SVM methods with a good kernel are able to achieve remarkably high accuracies. Apart from pure bioinformatics and experimental insights we obtain a better understanding about the important issue of whether and when SV-kernels from probabilistic models are preferable over specially engineered kernels. Note that both kernel choices are intended to incorporate prior knowledge into SVM learning.

2 Basic Ideas of the Methods

The essence of SVMs is that a very rich feature space is used for discrimination purposes while at the same time the complexity of the overall classifier is controlled carefully. This allows to give guarantees for high performance on unseen data. The key is a good choice of the so-called support vector kernel k which implicitely defines the feature space in which one classifies. Three particularly successful options for the kernel choice in DNA analysis exist: (a) available biological prior knowledge is directly engineered into a polynomial-type kernel, for example the so-called "locality improved (LI) kernel" as proposed in [22] or a probabilistic model that encodes the prior knowledge, e.g. an HMM is extracted from the data and is used for constructing a kernel: (b) the so called fisher-kernel [9] or its recent extension (c) the TOP kernel [19]. In some cases, one can reinterpret (a) in the probabilistic context. For instance the locality improved kernel corresponds to a higher order Markov model of the DNA sequence [19, 18].

Support Vector Machines For a given data set $\mathbf{x}_i \in \mathbb{R}^n$ $(i = 1, \ldots, N)$ with respective labels y_i, a SVM classifier yields $f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^N y_i \alpha_i \, \mathrm{k}(\mathbf{x}, \mathbf{x}_i) + b\right)$ given an input vector $\mathbf{x}$. For learning the parameters α_i a quadratic program is to be solved by (cf. [20]): $\max_{\boldsymbol{\alpha}} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \, \mathrm{k}(\mathbf{x}_i, \mathbf{x}_j)$ such that $\alpha_i \in [0, C]$, $i = 1, \ldots, N$ and $\sum_{i=1}^N \alpha_i y_i = 0$.

The Locality Improved Kernel [22] is obtained by comparing the two sequences locally, within a small window of length $2l + 1$ around a sequence position, where we count matching nucleotides. This number is then multiplied with weights $p_{-l}, \ldots, p_{+l}$ increasing linearly from the boundaries to the center of the window. The resulting weighted counts are taken to the d_1^{th} power, where d_1 reflects the order of local correlations (within the window) that we expect to be of importance: $\text{win}_p(\mathbf{x}, \mathbf{y}) = (\sum_{j=-l}^{+l} p_j \, \text{match}_{p+j}(\mathbf{x}, \mathbf{y}))^{d_1}$. Here, $\text{match}_{p+j}(\mathbf{x}, \mathbf{y})$ is 1 for matching nucleotides at position $p+j$ and 0 otherwise. The window scores computed with win_p are summed over the whole length of the sequence. Correlations between up to d_2 windows are taken into account by finally using the SV-kernel $k_{\text{LI}}(\mathbf{x}, \mathbf{y}) = \left(\sum_{p=1}^l \text{win}_p(\mathbf{x}, \mathbf{y})\right)^{d_2}$.

Fisher and TOP Kernel A further highly successful idea is to incorporate prior knowledge via probabilistic models $p(\mathbf{x}|\hat{\boldsymbol{\theta}})$ of the data (e.g. HMMs) into SVM-kernels [9]. This so-called *Fisher Kernel* (FK) is defined as $\mathrm{k}_{\text{FK}}(\mathbf{x}, \mathbf{x}') = \boldsymbol{s}(\mathbf{x}, \hat{\boldsymbol{\theta}})^\top Z^{-1}(\hat{\boldsymbol{\theta}}) \boldsymbol{s}(\mathbf{x}', \hat{\boldsymbol{\theta}})$, where $\boldsymbol{s}$ is the Fisher score $\boldsymbol{s}(\mathbf{x}, \hat{\boldsymbol{\theta}}) = \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}, \hat{\boldsymbol{\theta}})$ and where Z is the Fisher information matrix: $Z(\boldsymbol{\theta}) = \mathrm{E}_{\mathbf{x}}\left[\boldsymbol{s}(\mathbf{x}, \boldsymbol{\theta}) \boldsymbol{s}(\mathbf{x}, \boldsymbol{\theta})^\top \middle| \boldsymbol{\theta}\right]$ (for further details see [9]). Our recent extension of the FK uses the Tangent vectors Of Posterior log-odds (TOP) leading to the TOP kernel $\mathrm{k}_{\text{TOP}}(\mathbf{x}, \mathbf{x}') = \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\mathbf{x})^\top \boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\mathbf{x}')$ [19], where $\boldsymbol{f}_{\hat{\boldsymbol{\theta}}}(\mathbf{x}) := (v(\mathbf{x}, \hat{\boldsymbol{\theta}}), \partial_{\theta_1} v(\mathbf{x}, \hat{\boldsymbol{\theta}}), \ldots, \partial_{\theta_p} v(\mathbf{x}, \hat{\boldsymbol{\theta}}))^\top$ with $v(\mathbf{x}, \boldsymbol{\theta}) = \log(P(y = +1|\mathbf{x}, \boldsymbol{\theta})) - \log(P(y = -1|\mathbf{x}, \boldsymbol{\theta}))$ [19]. (We do not use Z^{-1} but a diagonal matrix for FK & TOP such that the variance in each coordinate is 1.) The essential difference between both kernels is that the TOP kernel is explicitely designed [19] for discrimination tasks. In fact, on protein family classification experiments we have shown that it performs significantly better than the Fisher kernel. As probabilistic models we employ Hidden Markov Models (HMMs), with several biologically motivated architectures (description below). We use the implementation described in [17] of the Baum-Welch algorithm [6] for training.

3 Data sets, experiments and results

Two data sets are analysed with different purposes. For the first IP benchmark set the goal is a comparison to other machine learning methods and we will see that our approaches easily outperform existing entries. For C. elegans we cannot compare to existing algorithms for systematic reasons (see section 1) and since our results on the IP data tell us that SVMs with our kernels are the method of choice, we focus in the second part of this section on the evaluation of different SVM kernels: locality improved vs. probabilistic model based kernels. Whereas the IP data is a fixed benchmark, for C. elegans, the decoys were chosen to be windows of the same length as the true sites from -25 to +25 of the site with two additional constraints: (i) the decoy windows were limited to those near the true sites and (ii) the decoy windows were forced to contain the conserved dinucleotide (GT or AG) centered in the same location in the window as in the true sites (donor and acceptor, respectively). This made the decoys not only harder than random ones from throughout the genome but also modeled the use of a splice site detector in a gene finder somewhat more realistically since it is more likely that a gene finder invokes a splice site detector in the proximity of true sites than at an arbitrary place in the genome (vast amounts of intergenic regions are already filtered out before any splice site prediction needs to be done). Not surprisingly, the performance reported in [13], where the decoys were similarly constructed, though in the proximity -40 and +40, was significantly poorer than in [12].

IPData is a benchmark set of human splice site data from the UC Irvine machine learning repository [3]. It contains 765 acceptor sites, 767 donor sites and 1654 decoys; the latter are however of low quality as they do not have a true site's consensus dint centered except by chance. The task is a donor/acceptor classification given a position in the middle of a window of 60 DNA letters as input.

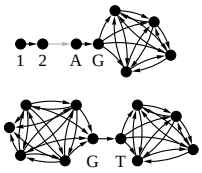

Fig. 1: *Acceptor and Donor Model*

In our experiments, first a careful model selection of the hyper-parameters of the HMMs and SVM is performed (cf. [11]). This is done separately on each of ten random (train, test) split of the data of size (2000,1186) (a single same-sized split was used in [12]). As HMM architecture we used (a) a combination of a linear model and a fully connected model for the acceptor sites (cf. Fig. 1, upper), (b) a combination of two fully connected model for the donor sites (cf. Fig. 1, lower) and (c) a fully connected model for modeling decoys. (These architectures can be biologically motivated.) The corresponding number of states in the components, as well as the regularization parameter of the SVM, are found by 10-fold cross validation.

For our comparison we computed the plain HMM, SVMs with locality improved kernel and with FK and TOP kernel (based on the HMMs for each class). Each classifier was then evaluated on the test set and results are averaged over the 10 runs and the standard deviation is given in Table 1 (shows errors on each of the classes).

Comparing our classifiers we observe that SVMs with TOP and FK (total error 5.4% and 5.3%) cannot improve the HMM, which performs quite well (6.0%), but has a quite large error in classifying the acceptor sites. The SVM with locality improved kernel does not suffer from this problem and achieves the best total error of of 3.7%.

We observe that the SVM methods outper-

System	Neither	Donor	Acceptor
HMM	2.6±0.5%	1.0±0.4%	2.4±0.7%
LI-SVM	2.0±0.3%	0.8±0.2%	0.9±0.3%
TOP-SVM	2.2±0.4%	1.5±0.4%	1.7±0.3%
FK-SVM	2.1±0.4%	1.6±0.5%	1.6±0.4%
NN-BRAIN	n.d.	2.6%	4.3%
BRAIN	4.0%	5.0%	4.0%
KBANN	4.6%	7.6%	8.5%
BackProp	5.3%	5.7%	10.7%
PEBLS	6.9%	8.2%	7.6%
Perceptron	4.0%	16.3%	17.4%
ID3	8.8%	10.6%	14.0%
COBWEB	11.8%	15.0%	9.5%
Near. Neigh.	31.1%	11.7%	9.1%

Table 1: Test-set errors on the IPData data set. All except the first 4 results are cited from [12], Table 6. (n.d.=not documented)

form all other documented methods on the IP data set (taken from [12]). These include not only the BRAIN algorithms of [12] published recently, but also established machine learning methods such as nearest-neighbor classifiers, neural networks and decision trees. The SVM achieves test-set errors that are half of the best other methods, but only if the kernel is suitable.[2]

The C. Elegans data set was derived from the Caenorhabditis Elegans genome [1], specifically from the chromosome and GFF files at http://genome.wustl.edu/gsc/C_elegans. From these files, windows of -50 to +50 around the true sites were extracted.[3] This resulted in 74,455 acceptor sites and 74,505 donor sites, each of which we clipped to -25 to +25 (i.e. length 50) with the consensus dinucleotide centered. For the decoys, we extracted, from the -50/+50 windows around each site, all windows `w` (except the true site's window) of length 50 as the true site windows, with the consensus dint `GT` or `AG` centered in `w` in the same offset as in a true site's window. This resulted in 122,154 acceptor decoys and 177,061 donor decoys. The complete data is available at our web-site `http://mlg.anu.edu.au/~raetsch/splice` related to this paper. In this paper we will only use subsets of at most 25000 examples.

In our study, we consider the classification of C. elegans acceptor sites only. We expect similar results on the donor sites. As probabilistic models for true acceptor sites we use the HMM described in Fig. 1 (upper). Since we assume that the decoy splice sites are close to a true splice site (in fact, in our data they are never further away then 25 base pairs), we can exploit this in our probabilistic model for the decoys. We propose to append the previosly obtained HMM for the true sites to a linear strand of length 25 (as in Fig. 2). Then we allow all new states $1', 2', \ldots, 25'$ and all states in the positive model (except the first state) to be the starting states of this model. Hence, true sites not centered are detected as decoys. (Only the emission probabilities of the new states and the start state distribution are optimized.)

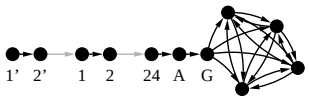

Fig. 2: *The HMM architecture for modeling C. elegans decoys*

For training of HMMs and SVMs we use 100, 1000 and 10000 examples. For simplicity we use additional 5000 examples for model selection (to select number of states, regularization constant; not possible in practice, but makes the comparison easier). This is done on each of 5 realizations and then the best classifiers are chosen and evaluated on the test set (10000 examples) leading to the results in Table 2. Our first result reveals that the test error of the SVM classifier decreased consistently as the training set size was increased. This means that although the larger data set certainly contains redundant information, the SVMs can still extract some additional insights for the classification. We conjecture that there is useful statistical signal away from the conserved portion of the sites that the SVM classifier is eventually picking up from the larger training set. Also observe that the locality improved kernel starts with a very good result on 100 examples and then cannot gain much from more examples. We conjecture that it profited from the weak prior information incoded in the kernel, but then only improves slowly. The TOP kernel, however, starts with a poor result (possibly due to numerical problems) and improves much with more examples. The HMM reaches a plateau at 1000 examples (using 100.000 examples the HMMs achieve 2.4%), whereas TOP and FK SVMs can improve when seeing more patterns (prelimary results show error rates even below 2%).

	HMM	Loc.Imp.	FK	TOP
100	10.7±2.8%	**7.6±1.0%**	9.4±4.2%	20.8±3.0%
1000	**2.8±0.1%**	5.2±0.1%	3.5±0.2%	4.6±0.4%
10000	2.6±0.2%	3.9±0.2%	2.5±0.3%	**2.3±0.1%**

Table 2: *Test errors of our 4 methods on 100-10000 examples*

[2] If one uses RBF kernels, one gets worse results than the BRAIN method [11].

[3] We thank David Kulp and others at University of California, Santa Cruz for preparing these datasets and David Haussler for granting us permission to use them. As an extra step, we verified their extracted sites by matching them to the chromosome DNA sequences.

Figure 3 shows the ROC plot of the performance of our four classifiers on the C. elegans *acceptor* data set of size 10,000 on a test set of size 10,000. The predictive performance was plotted as a function of the classification threshold swept to vary the trade-off between false positives and false negatives. From the perspective of gene finding as well as researchers wanting to locate the sites, it is important to keep the false negative rate as low as possible. But since the number of true negatives (non-sites) when scanning even the regions of the genome in the proximity of the true sites will vastly outnumber the true sites, it is also important to keep the false positive rate down. Since we cannot keep both down simultaneously, we should look at the performance of the classifier at least at two extremes – at low false positive rate and at low false negative rate.

We see that TOP- and FK-SVM classifier achieves a simultaneous 1% false-positive rate (i.e., a sensitivity of 0.99) and a 5% and 8% false-negative rate (i.e., a specificity of about 0.95 and 0.92), respectively. While conclusive comparisons are inadvisable owing to experiments having been done on different data sets, some comparisons with the results of [13] are still helpful. In [13] a similar methodology as ours was applied to similar data sets, in particular, the procedure to construct the decoys is similar (although, as already indicated above, in the proximity -40 and +40 of the site instead of -25 and +25). The result in [13] could achieve a simultaneous 1% false-positive rate and 20% false-negative rate, which is worse than our result.

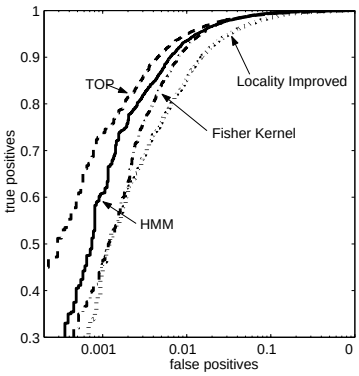

Fig. 3: *ROC Curves*

We would also like to highlight an interesting outcome concerning the issue of SV-kernel choice. The experiments show that the locality improved kernel, where biological knowledge has been directly engineered into the kernel geometry, works very nicely. Nevertheless this approach can be outperformed by a SV kernel derived from a probabilistic model like fisher or TOP kernel. The important point is, however, that this additional improvement holds only for very problem-specific probabilistic models, like the specially taylored negative and positive HMMs used here (cf. section 2). Already as stand-alone classifiers those HMMs perform very competitive. If less fine-tuned HMMs are used as a basis for discriminative training with FK or TOP kernel, the performance decreases considerably (cf. splice site recognition results in [10]). So the incorporation of detailed biological prior knowledge makes the difference in performance.

4 Conclusions

In this paper we successfully applied SVMs to the problem of splice site detection. The key for obtaining our excellent results was a smart inclusion of prior knowledge into SVMs, more precisely into their kernels. A general problem in assessing classification performance on bioinformatics data is that, while there is a lot of publically available molecular data, at present there are few standardized data sets to evaluate new classifiers with. (We contribute to overcoming this problem by making all data and detailed results publically available on the previously mentioned website.) Another issue is that the problems we address here (as well as many other classification problems in bioinformatics) involve separating signal (one class) from noise (the other class). The noise class is generally far denser than the signal class. Both due to this imbalance, and because the noise class is ill-defined, classifiers have to be designed and evaluated with special care.

Our first set of experiments used the well-known but small IP benchmark data set and showed that our SVMs compare favourably over existing results. For the C. Elegans study, we

could not find any preprocessed benchmark data, and therefore could not compare the performance directly against existing methods (except HMMs). Therefore, we decided to study more closely the SVM-based learning itself, and in particular the quality of probabilistic vs. engineered kernels. Clearly, including biological prior knowledge like in locality improved kernels gives an excellent performance which cannot be surpassed by a *straight forward* probabilistic kernel (e.g. a first order Markov model as used in [10]). However, if we use sophisticated probabilistic modeling like in specific HMMs that are fine-tuned for splice site recognition, then an additional discriminative training on top of the probabilistic model provides a further improvement.

Future research will focus on the construction of better probabilistic models and SV kernels. We furthermore plan to train our classifiers on larger problems (we used only 10.000 out of 180.000 examples), for which some additional practical problems have to be solved.[4] And finally we would like to apply our insights to splice site detection on the complete human genome.

References

1. Genome sequence of the Nematode Caenorhabditis elegans. *Science*, 282:2012–2018, 1998.
2. P. Baldi, S. Brunak, Y. Chauvin, C.A.F. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.
3. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
4. C. Burge and S. Karlin. Prediction of complete gene structures. *J. Mol. Biol.*, 268:78–94, 1997.
5. A.L. Delcher, D. Harmon, S. Kasif, O. White, and S.L. Salzberg. Improved microbial gene identification with GLIMMER. *Nucleic Acids Research*, 27(23):4636–4641, 1999.
6. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.
7. D. Cai et al. Modeling splice sites with Bayes networks. *Bioinformatics*, 16(2):152–158, 2000.
8. M.P.S. Brown et al. Knowledge-based analysis by using SVMs. *PNAS*, 97:262–267, 2000.
9. T.S. Jaakkola, M. Diekhans, and D. Haussler. *J. Comp. Biol.*, 7:95–114, 2000.
10. T.S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In M.S. Kearns et al., editor, *Adv. in Neural Inf. Proc. Systems*, volume 11, pages 487–493, 1999.
11. K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.
12. S. Rampone. Recognition of splice junctions on DNA. *Bioinformatics*, 14(8):676–684, 1998.
13. M.G. Reese, F. H. Eeckman, D. Kulp, and D. Haussler. *J. Comp. Biol.*, 4:311–323, 1997.
14. S. Salzberg, A.L. Delcher, K.H. Fasman, and J. Henderson. *J. Comp. Biol.*, 5(4):667–680, 1998.
15. B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
16. A.J. Smola and J. MacNicol. Scalable kernel methods. Unpublished Manuscript, 2002.
17. S. Sonnenburg. *Hidden Markov Model for Genome Analysis*. Humbold University, 2001. Proj. Rep.
18. S. Sonnenburg. New methods for splice site recognition. Master's thesis, 2002. Forthcoming.
19. K. Tsuda, M. Kawanabe, G. Rätsch, S. Sonnenburg, and K.R. Müller. A new discriminative kernel from probabilistic models. In *Adv. in Neural Inf. proc. systems*, volume 14, 2002. In press.
20. V.N. Vapnik. *The nature of statistical learning theory*. Springer Verlag, New York, 1995.
21. Y. Xu and E. Uberbacher. Automated gene identification. *J. Comp. Biol.*, 4:325–338, 1997.
22. A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K.-R. Müller. Engineering svm kernels that recognize translation initiation sites. *Bioinformatics*, 16(9):799–807, 2000.

[4] When using the TOP or FK kernel, one has to handle vectors of length about 5000 per example. This leads to quite large matrices, which are more difficult to handle. We plan on following the approach of [16] to overcome this problem with a more efficient algorithm and also more computing power.