1 Support vector machine applications in computational biology

William Stafford Noble[1]
Department of Genome Sciences
University of Washington
Seattle, WA, USA
noble@gs.washington.edu

During the past three years, the support vector machine learning algorithm has been extensively applied within the field of computational biology. The algorithm has been used to detect patterns within and among biological sequences, to classify genes and patients based upon gene expression profiles, and has recently been applied to several new biological problems. This chapter reviews the state of the art with respect to SVM applications in computational biology.

1.1 Introduction

The support vector machine (SVM) algorithm (Boser et al., 1992; Vapnik, 1998) is a classification algorithm that provides state-of-the-art performance in a wide variety of application domains, including handwriting recognition, object recognition, speaker identification, face detection and text categorization (Cristianini and Shawe-Taylor, 2000). During the past three years, SVMs have been applied very broadly within the field of computational biology, to pattern recognition problems including protein remote homology detection, microarray gene expression analysis, recognition of translation start sites, functional classification of promoter regions, prediction of protein-protein interactions, and peptide identification from mass spectrometry data. The purpose of this chapter is to review these applications, summarizing the state of the art.

1. Formerly William Noble Grundy, see `www.gs.washington.edu/noble/name-change.html`

Two main motivations suggest the use of SVMs in computational biology. First, many biological problems involve high-dimensional, noisy data, for which SVMs are known to behave well compared to other statistical or machine learning methods. Second, in contrast to most machine learning methods, kernel methods like the SVM can easily handle non-vector inputs, such as variable length sequences or graphs. These types of data are common in biology applications, and often require the engineering of knowledge-based kernel functions. Much of this review consists of explaining these kernels and relating them to one another.

This review assumes that the reader has a basic familiarity with support vector machines, including the notion of a kernel function and the mapping from input space to feature space. Background information can be found in Cristianini and Shawe-Taylor (2000); Burges (1998) and at `www.kernel-machines.org`. The chapter is organized by application domain, beginning in Section 1.2 with perhaps the most intensively studied application, the recognition of subtle similarities among protein sequences. Section 1.3 reviews other protein and gene classification tasks, and Section 1.4 looks at problems that involve recognizing patterns within a protein or DNA sequence. Section 1.5 reviews the many applications of SVMs to the analysis of DNA microarray expression data. Section 1.6 describes three approaches to learning from heterogeneous biological data. Finally, the paper closes with a description of several applications that do not fit neatly into the previous categories, followed by a brief discussion.

1.2 Protein remote homology detection

Over the past 25 years, researchers have developed a battery of successively more powerful methods for detecting protein sequence similarities. This development can be broken into four stages. Early methods looked for pairwise similarities between proteins. Among such algorithms, the Smith-Waterman dynamic programming algorithm (Smith and Waterman, 1981) is among the most accurate, whereas heuristic algorithms such as BLAST (Altschul et al., 1990) and FASTA (Pearson, 1985) trade reduced accuracy for improved efficiency.

In the second stage, further accuracy was achieved by collecting aggregate statistics from a set of similar sequences and comparing the resulting statistics to a single, unlabeled protein of interest. Profiles (Gribskov et al., 1990) and hidden Markov models (HMMs) (Krogh et al., 1994; Baldi et al., 1994) are two methods for representing these aggregate statistics. For a given false positive rate, these family-based methods allow the computational biologist to infer nearly three times as many homologies as a simple pairwise alignment algorithm (Park et al., 1998).

In stage three, additional accuracy was gleaned by leveraging the information in large databases of unlabeled protein sequences. Iterative methods such as PSI-BLAST (Altschul et al., 1997) and SAM-T98 (Karplus et al., 1998) improve upon profile-based methods by iteratively collecting homologous sequences from a large database and incorporating the resulting statistics into a single model. All of

The Fisher kernel

the resulting statistics, however, are generated from positive examples, i.e., from sequences that are known or posited to be evolutionarily related to one another.

In 1999, Jaakkola *et al.* ushered in stage four of the development of homology detection algorithms with a paper that garnered the "Best Paper" award at the annual Intelligent Systems for Molecular Biology conference. Their primary insight was that additional accuracy can be obtained by modeling the difference between positive and negative examples. Because the homology task requires discriminating between related and unrelated sequences, explicitly modeling the difference between these two sets of sequences yields an extremely powerful method. The algorithm described in that paper is called *SVM-Fisher*.

The SVM-Fisher method (Jaakkola et al., 1999, 2000) couples an iterative HMM training scheme with the SVM. For any given family of related proteins, the HMM provides a kernel function. First, the HMM is trained on positive members of the training set using the standard Baum-Welch training routine. The training is iterated, adding to the training set at each round similar sequences from a large unlabelled database. After training, the gradient vector of any sequence—positive, negative or unlabeled—can be computed with respect to the trained model. As in the Baum-Welch training algorithm for HMMs, the forward and backward matrices are combined to yield a count of observations for each parameter in the HMM. As shown in (Jaakkola et al., 1999), the counts can be converted into components of a gradient vector $\vec{\mathbf{U}}$ via the following equation:

$$\vec{\mathbf{U}}_{ij} = \frac{E_j(i)}{e_j(i)} - \sum_k E_j(k), \tag{1.1}$$

where $E_j(i)$ is the number of times that amino acid i is observed in state j, and $e_j(i)$ is the emission probability for amino acid i in state j. Although these gradients can be computed for every HMM parameter, the SVM-Fisher method uses only the gradient components that correspond to emission probabilities in the match states. Furthermore, a more compact gradient vector can be derived using a mixture decomposition of the emission probabilities. Each sequence vector summarizes how different the given sequence is from a typical member of the given protein family. Finally, an SVM is trained on a collection of positively and negatively labeled protein gradient vectors. By combining HMMs and SVMs, SVM-Fisher offers an interpretable model, a means of incorporating prior knowledge and missing data, and excellent recognition performance.

Indeed, the SVM-Fisher method yields results that improve significantly upon the previous state of the art. The standard benchmark for this classification task comes from the Structural Classification of Proteins (SCOP) (Murzin et al., 1995), which provides protein superfamily labels based upon human interpretation of three-dimensional protein structures (see Figure 1.1). The original experiment compared SVM-Fisher to BLAST and to the SAM-T98 iterative HMM method

(Hughey and Krogh, 1996), and a subsequent experiment included a comparison to PSI-BLAST (Leslie et al., 2002). In each case, SVM-Fisher performs significantly better than previous methods. Subsequent work by Karchin et al. (2002) demon-

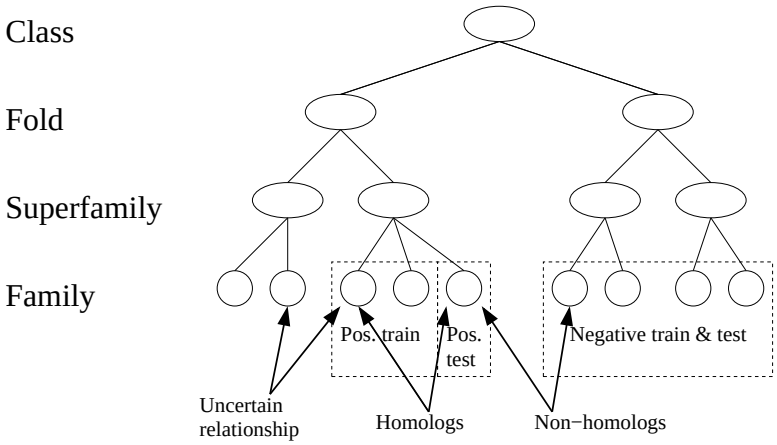

Figure 1.1 The SCOP hiearchy of protein domains. SCOP is a hand-curated database that is arranged hierarchically according to protein three-dimensional structure. The three primary levels of the hierarchy—family, superfamily and fold—correspond to varying degrees of similarity. Proteins within a single family show clear evolutionary relationships, typically evidenced by more than 30% pairwise identities at the sequence level, while members of a superfamily may have low sequence identity, but have structural and functional features that suggest a common evolutionary origin. Finally, proteins belong to the same fold if they have the same major secondary structures in the same arrangement and with the same topological connections. Proteins placed together in the same fold category may not have a common evolutionary origin. The figure illustrates how a SCOP-based benchmark is created. All but one family within a given superfamily constitute the positive training set, and the held-out family constitutes the positive test set. Negative examples are drawn from outside of the training set fold.

strates the successful application of the SVM-Fisher methodology to the recognition of a large, pharmaceutically important class of proteins, the G-protein coupled receptors.

Recently, the Fisher kernel framework was elegantly generalized by Tsuda et al. (2002). They describe a general method for deriving a kernel from any latent variable model, such as an HMM. The kernel assumes the availability of the hidden variables, which are estimated probabilistically. The resulting *joint kernel* can be converted to a *marginalized kernel* by taking its expectation with respect to the hidden variables. The Fisher kernel, it turns out, is a special case of marginalized kernels. The framework is demonstrated by using a small HMM-based marginalized kernel to characterize a single family of bacterial proteins

Composition kernels

Subsequent to the introduction of the Fisher kernel, many different kernels have been applied to the problem of protein remote homology. Ding and Dubchak (2001) define one of the simplest such kernels, a composition-based kernel function that characterizes a given protein via the frequency with which various amino acids occur therein. In this work, each protein is characterized by a simple vector of letter frequencies. Each protein sequence is represented via six different alphabets, corresponding to amino acids, predicted secondary structure, hydrophobicity, normalized van der Waals volume, polarity and polarizability. A single protein is represented by the letter frequencies across each of these alphabets, for a total of 125 features.

The focus of this work is not the kernel function but the machinery for making multi-class predictions. The most common means of training an SVM for an n-class problem is the *one-vs-others method*: n SVMs are trained, one per class, using members of all other classes as negative examples. The final classification of a test example is the class corresponding to the SVM that yields the discriminant with largest absolute value. Ding and Dubchak introduce a method called the *unique one-vs-others* method, which performs additional SVM optimizations in order to sort out disagreements among SVMs trained using the standard, one-vs-others method, and they show that their method leads to significant improvement in test set accuracy. The work also shows that an SVM out-performs a similarly trained neural network on this task.

A similar composition kernel is used by Cai et al. (2001) to recognize broad structural classes of proteins (all-α, all-β, α/β and $\alpha+\beta$). On this task, the SVM yields better discrimination performance than a neural network method and a method previously developed by the same authors.

Motif kernels

A significant drawback to the composition kernel is the simplicity of the protein representation. Logan et al. (2001) propose a richer representational scheme, in which features correspond to motifs in a pre-existing database. The BLOCKS database (Henikoff and Henikoff, 1991) contains weight matrix motifs derived from protein multiple alignments. Because these motifs occur in regions that are highly conserved, they tend to correspond to functionally important regions of the proteins. This observation motivates using motifs as features for an SVM. Logan *et al.* use the BLIMPS tool (Wallace and Henikoff, 1992) to compare 10,000 BLOCKS motifs to each protein in the SCOP database. The resulting scores are used to map each

protein into a 10,000-dimensional space. On a small collection of SCOP families, this motif kernel performs better than an HMM method and comparably with the Fisher-SVM kernel.

Recently, a different motif kernel was described by Ben-hur and Brutlag (2003). This kernel uses the eBLOCKS database (`motif.stanford.edu/eblocks`), which contains close to 500,000 motifs. Rather than representing each motif via a weight matrix, eBLOCKS uses discrete sequence motifs. For example, the 6-mer motif `[AS].DKF[FILMV]` contains three types of sites: the first position matches either `A` or `S`, the second position matches any amino acid, and the third position matches only the amino acid `D`. Thus, this motif would match the following example sequences: `ACDKFF`, `SRDKFI` and `SADKFV`. Because the motif database is so large, a simple vector representation is computationally infeasible. Ben-Hur and Brutlag therefore demonstrate how to compute the corresonding kernel values efficiently using a trie data structure. Tested on a SCOP benchmark (Liao and Noble, 2002), the motif kernel provides a significant improvement in performance over previously described kernels.

Pairwise comparison kernels

One appealing characteristic of the Fisher kernel is its ability to incorporate prior knowledge that is built into the profile HMM framework, including a simple model of molecular evolution. An alternative evolutionary model is implicit in pairwise sequence comparison algorithms, such as the Smith-Waterman (Smith and Waterman, 1981) dynamic programming algorithm and its heuristic approximation, BLAST (Altschul et al., 1990). Like the HMM, these algorithms assume that molecular evolution primarily proceeds via mutations and small-scale insertions and deletions. Furthermore, through extensive application over more than two decades of research, pairwise sequence comparison algorithms have been exhaustively analyzed and optimized. For example, the distribution of scores produced by these algorithms can be well characterized and used to compute a p-value or E-value associated with each observed score.

Liao and Noble (2002, 2003) describe a simple method for generating a kernel from the scores produced by a pairwise sequence comparison algorithm. These algorithms have the form of a kernel function, in the sense that they measure the similarity between a pair of objects being classified; however, the scores themselves are not positive semi-definite and so cannot be used as kernels. Therefore, Liao and Noble employ the empirical kernel map (Tsuda, 1999) to convert the scores to a valid kernel. This procedure is illustrated in Figure 1.2. The matrix on the left is an m by m matrix of Smith-Waterman scores, corresponding to all pairs of proteins in a training set. Each row in this matrix can be used as a vector representation of the corresponding protein. A standard kernel function is then used to compute the similarity between these vectors. Thus, each entry in the matrix on the right in Figure 1.2 is simply the scalar product of two rows from the matrix on the left. Because the procedure uses a standard kernel function, the empirical kernel map guarantees a valid kernel matrix. Furthermore, the empirical kernel map offers an easy way to incorporate prior knowledge directly into the kernel. For example, a sequence kernel based on the Smith-Waterman or BLAST

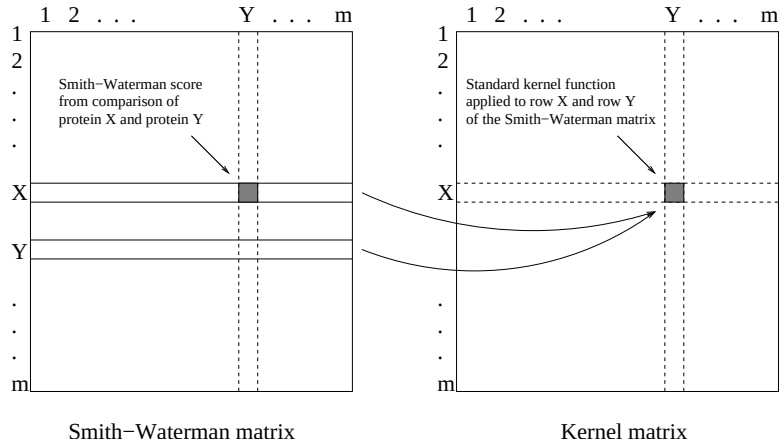

Figure 1.2 An empirical kernel map derived from the Smith-Waterman sequence comparison algorithm. Each matrix contains m rows and columns, corresponding to the proteins in the training set. Each entry in the matrix on the left is the Smith-Waterman score of the corresponding proteins. Each entry on the right is the result of applying a standard kernel function (e.g., dot product, polynomial or radial basis) to the two corresponding rows from the Smith-Waterman matrix.

algorithm benefits from their implicit model of molecular evolution as well as from two decades of empirical optimization of the algorithm's parameters. In conjunction with an SVM classifier, the Smith-Waterman empirical kernel map yields a powerful method—called SVM-pairwise—for detection of subtle protein sequence similarity, performing significantly better than the Fisher kernel on the data set used in that paper (Liao and Noble, 2002).

One drawback to the SVM-pairwise algorithm is its efficiency; however, several variants of the algorithm address this issue. The computation of the kernel matrix requires pre-computation of all pairwise sequence comparison scores in the training set. For the Smith-Waterman algorithm, each such computation is $O(p^2)$, where p is the length of the protein sequences. This step can be sped up by a factor of p by using the heuristic BLAST algorithm instead, at a small loss in accuracy (Liao and Noble, 2002). The second step of the kernel computation—calculation of the empirical kernel map—is also expensive, requiring $O(m)$ time for each kernel value, where m is the number of proteins in the training set. For some families of proteins, the value of m can become quite large, on the order of 10,000. This step can be sped up by using a smaller *vectorization set* of proteins in the empirical kernel map, where the vectorization set defines the columns in the left-hand matrix in Figure 1.2. For example, using a vectorization set consisting only of the positive training examples leads to a significant time savings, again at a relatively small decrease in performance (Liao and Noble, 2003).

String kernels

String kernels comprise another class of kernels for protein remote homology detection. Like the BLAST and Smith-Waterman algorithms, string kernels operate

directly on pairs of proteins; however, string kernels are positive semi-definite functions and hence do not require the empirical feature map. The most general types of string kernels are pair HMM and convolution kernels (Watkins, 1999; Haussler, 1999; Lodhi et al., 2002). However, these kernels are expensive to compute and have not been applied to protein classification.

Leslie, Eskin and Noble describe a simple string kernel—the *spectrum kernel*—that is more efficient to compute. This kernel is, in a sense, a generalization of the composition kernel mentioned earlier, in which the composition is computed with respect to length-k substrings, called k-mers. For example, for $k = 5$ and an alphabet of size 20, each vector consists of $5^{20} = 9.5 * 10^{13}$ elements, each corresponding to a single 5-mer. The kernel can be computed efficiently using a trie data structure. On the SCOP benchmark used by Jaakkola et al. (1999), the spectrum kernel using $k = 3$ provides performance comparable to that of the HMM-based Fisher kernel. An alternate version of the spectrum kernel based upon suffix trees and suffix links was subsequently described by Vishwanathan and Smola (2003). For computing individual kernel values, the suffix tree implementation is faster by a factor of $O(k)$. However, this difference disappears for the computation of a complete matrix of m^2 kernel values: the trie-based spectrum kernel method allows for efficient construction of the full matrix in one pass of the algorithm, and this computation is as fast as calculating m^2 individual kernel values with the suffix tree method.

The spectrum kernel has also been generalized to allow for a more accurate model of molecular evolution. Mutations in the protein sequence are modeled using a *mismatch kernel* (Leslie et al., 2003b), in which matches between k-mers are allowed to contain at most M mismatches. Thus, for $M = 1$, a feature corresponding to a k-mer such as `VTWTA` would match sequences such as `VTATA`, `VCWTA`, or `VTWTK`. Further flexibility, including deletions of amino acids and more accurate modeling of mutations, are modeled using a collection of string kernel functions introduced by Leslie and Kuang (2003). These generalizations also use the trie data structure, and have a running time that does not depend upon the size of the alphabet.

The efficiencies of the various kernels functions for protein remote homology detection are summarized in Table 1.1. With respect to the quality of the results produced by these various kernels, conclusions are difficult to draw. There are two primary SCOP benchmarks, one that includes in the training set additional non-SCOP homologs identified via an HMM (Jaakkola et al., 1999) and one that uses only SCOP domains (Liao and Noble, 2002). The SVM-Fisher method performs well on its original benchmark (Jaakkola et al., 1999) but less well when non-SCOP homologs are removed from the training set (Liao and Noble, 2002), presumably because the HMMs are consequently under-trained. The SVM-pairwise algorithm performs better than SVM-Fisher on the non-homology benchmark (Liao and Noble, 2002); however, performing SVM-pairwise on the Jaakkola benchmark is not practical due to the $O(m^3)$ running time of the empirical kernel map. Published results indicate that the discrete motif method outperforms SVM-pairwise on the non-homology benchmark (Ben-hur and Brutlag, 2003); however, subsequent

Table 1.1 Efficiency of kernels for protein sequence comparison. Each entry in the first table is the running time required to compute an m by m matrix of kernel values. Variables are defined in the second table. For simplicity, all proteins are assumed to be of approximately the same length p.

Kernel	Complexity	Cite
SVM-Fisher	$O(s^2mp + sm^2)$	(Jaakkola et al., 1999)
SVM-pairwise	$O(vmp^2 + vm^2)$	(Liao and Noble, 2003)
spectrum	$O(pm^2)$	(Leslie et al., 2002)
mismatch	$O(k^M \ell^M pm^2)$	(Leslie et al., 2003b)
gappy, substitution, wildcard	$O(c_K pm^2)$	(Leslie and Kuang, 2003)
weight matrix motif	$O(\ell pqm^2)$	(Logan et al., 2001)
discrete motif	$O(pqm^2)$	(Ben-hur and Brutlag, 2003)

Variable definitions

p	length of one protein
m	number of proteins in training set
s	number of states in profile HMM
v	number of proteins in vectorization set
k	k-mer (substring) length
M	number of allowed mismatches
ℓ	size of alphabet
c_K	constant that is independent of alphabet size
q	number of motifs in database

experiments using a larger E-value threshold show the two methods performing comparably. Finally, although the spectrum kernel does not perform as well as SVM-Fisher (Leslie et al., 2002), its variants (mismatch, gappy, substitution and wildcard) are comparable to SVM-Fisher on the homology benchmark (Leslie et al., 2003b; Leslie and Kuang, 2003) and (for the mismatch kernel) comparable to SVM-pairwise on the non-homology benchmark (Leslie et al., 2003a).

1.3 Classification of genes and proteins

The recognition of remote homology relationships among proteins is a multi-class classification problem, in which the classes are defined by similarities of protein 3D structure. There are, however, numerous other ways in which proteins and their corresponding genes can be placed into biologically interesting categories. SVMs have been applied to the recognition of several such types of categories.

Functional classification of promoter regions

In addition to the primary amino acid sequence, the functional role of a protein can sometimes be determined by analyzing the DNA sequence that occurs upstream of the corresponding gene. This region contains the switching mechanism that controls when the gene is turned on and off; i.e., when and how frequently the gene is translated into a protein sequence. Pavlidis et al. (2001a) demonstrate the application of the Fisher kernel to the problem of classifying genes according to the characteristics of their switching mechanisms. This work thus assumes that genes with similar switching mechanisms are likely to operate in response to the same environmental stimulation and hence are likely to have similar or related functional roles. The Fisher kernel is derived from a motif-based hidden Markov model, constructed using Meta-MEME (Grundy et al., 1997). In this model, each motif corresponds to one transcription factor binding site. The method is used successfully to predict membership in two groups of co-regulated genes in yeast.

Prediction of protein function from phylogenetic profiles

Protein function can also be determined via sequence comparison with other species. Vert describes an elegant kernel function that operates on phylogenetic profiles (Vert, 2002b). A phylogenetic profile is a bit string representation of a protein, in which each bit corresponds to one species for which the complete genome is available (Pellegrini et al., 1999). A bit is 1 if the protein has a close homolog in that species, and 0 otherwise. Thus, the phylogenetic profile captures (part of) the evolutionary history of a given protein. Two proteins that have similar phylogenetic profiles likely have similar functions, via a kind of guilt by association. Say that in every genome that protein A is observed, we also observe protein B, and vice versa. Given enough complete genomes, the probability of such consistent co-occurrence happening by chance is extremely small.

Vert's phylogenetic profile kernel uses a simple Bayesian tree model to capture the evolutionary relationships among sequences. The tree defines a joint probability distribution, and the corresponding feature space contains one dimension for each possible evolutionary history. The tree kernel is a weighted sum over these histories. Vert demonstrates how to compute this kernel in linear time. For predicting

yeast protein functional classes, an SVM trained using the tree kernel performs significantly better than an SVM trained using a simple, dot product kernel from the same data set.

Prediction of subcellular localization

Hua and Sun (2001b) use SVMs to perform protein classification with respect to subcellular localization. Here, the label of each protein corresponds to the region of the cell in which it typically resides, including for prokaryotes, the cytoplasm, the periplasm, and the exterior of the cell, and for eukaryotes the nucleus, cytoplasm, mitochondria and the exterior of the cell. In this work, the kernel function is a simple, 20-feature composition kernel. The SVM is shown to produce more accurate classifications than competing methods, including a neural network, a Markov model and an algorithm specifically designed for this task (Chou and Elrod, 1999).

Distinguishing between benign and pathologic human immunoglobulin light chains

Zavaljevski and Reifman (2002) describe the application of an SVM to a clinically important, binary protein classification problem. The class of human antibody light chain proteins is large and is implicated in several types of plasma cell diseases. In particular, Zavaljevski, Steven and Reifman use SVMs to classify the κ family of human antibody light chains into benign or pathogenic categories. The data set consists of 70 protein sequences. Significantly, these proteins are aligned to one another, in a multiple alignment of width 120. This alignment suggests a simple vectorization, in which each binary feature represents the occurrence of a particular amino acid at a particular position in the alignment. In order to reduce the size of the resulting feature vector, the authors compress the amino acid to an alphabet of size seven, based upon biochemical similarities.

In addition to making accurate predictions, the SVM is used in this context to identify positions in the alignment that are most discriminative with respect to the benign/pathogenic distinction. This identification is accomplished via *selective kernel scaling*, in which a scaling factor is computed for each alignment position and is subsequently incorporated into the kernel computation. The scale factors are computed in two different fashions: first, by measuring the degree of conservation in a reference alignment of 14 prototypical human κ light chains, and second, by computing a normalized sensitivity index based upon the output of the SVM. The latter method is iterative and is related to the recursive feature elimination method described below (Guyon et al., 2002). The resulting classifier yields an accuracy of around 80%, measured using leave-one-out cross-validation, which compares favorably with the error rate of human experts. Furthermore, the kernel scaling technique confirms the importance of three previously identified positions in the alignment.

1.4 Prediction along the DNA or protein strand

In addition to classifying invidual gene or protein sequences, SVMs have been applied to a number of task that involve searching for a particular pattern within a single sequence.

Translation start sites

An early such application involved the recognition of translation start sites in

DNA. These positions mark the beginnings of protein-coding genes; hence, an accurate recognizer for this task is an integral part of automatic gene-finding methods. Zien et al. (2000) compare SVMs to a previously described neural network approach to this problem. A fixed-length window of DNA is encoded in redundant binary form (four bits per base), and the SVM and neural network are trained on the resulting vectors. Using a simple polynomial kernel function, the SVM improves upon the neural network's error rate (15.4% down to 13.2%). Furthermore, Zien *et al.* demonstrate how to encode prior knowledge about the importance of local interactions along the DNA strand. This locality-improved kernel reduces the error still further to 11.9%.

Splice sites

A similar application is described by Degroeve et al. (2002). Here, rather than recognizing the starts of genes, the SVM learns to recognize the starts of introns. Training and testing are performed on sequences from *Arabidopsis thaliana*. Once again, the data is collected in fixed-length windows and is encoded in redundant binary form. The emphasis in this work is feature selection: the authors would like to determine which positions around the splice site provide the most information. They therefore propose a wrapper-based feature selection method, removing features one at a time using the following selection criterion:

$$\underset{k}{\text{argmax}} \left(\sum_{j=1}^{l} y_i \times \left(\sum_{i=1}^{l} \alpha_i y_i K({x_i}^k, {x_j}^k) + b \right) \right), \tag{1.2}$$

where y_j is the label ($+1$ or -1) of example j, b is the SVM bias term, and ${x_j}^k$ is instance x_j with feature k set to its mean value. Three SVM methods (using linear, polynomial and RBF kernels) are compared to a similar method based upon a weight matrix, or naive Bayes classifier. The experiments do not show a clear superiority of any method. Indeed, in no case does feature selection improve performance relative to using the entire window of 100 bases. All methods, not surprisingly, indicate that the most important features are those closest to the splice site, though the methods do not agree on which specific sites are most relevant.

Signal peptide cleavage sites

Signal peptides are molecular bar codes at the end of a protein sequence that help to direct the protein to a particular location in the cell. Vert (2002a) describes an SVM approach to recognizing the position at which a signal peptide is cleaved from the main protein once it reaches its location. This application is thus similar to recognizing translation starts and splice sites, except that it is performed on proteins rather than DNA sequences. The recognition of signal peptides is important for the development of new drugs.

However, the emphasis in Vert's paper is not the signal peptide application *per se*, but the description of a general class of kernels derived from probabilistic models. The primary aim is to describe a kernel that defines two objects as "close" when they share rare common substructures. Here, "rarity" is defined with respect to a particular naive Bayes probabilistic model. In general, for any probability density p

on X and any set of substructures $V \subset P(S)$, the kernel $K_{p,V}$ is defined as follows:

$$K_{p,V}(x,y) = \frac{p(x)p(y)}{|V|} \sum_{T \epsilon V} \frac{\delta(x_T, y_T)}{p(x_T)}, \tag{1.3}$$

for any two realizations $(x,y)\epsilon A^{2S}$, where $\delta(x_T, y_T)$ is 1 if $x_T = y_T$, 0 otherwise.

Previous research has successfully applied a simple weight matrix model to the recognition of signal peptide cleavage sites (von Heijne, 1986). Accordingly, Vert demonstrates how to derive from a weight matrix a kernel based upon co-occurrences of rare substrings. The resulting SVM yields dramatically better recognition performance than the simple weight matrix approach. For example, at a false positive rate of 3%, the weight matrix method retrieves 46% of true positives, whereas the SVM method retrieves 68%.

Functional RNAs in prokaryotes

The three previous methods aim at recognizing specific sites in a DNA or protein sequence. In contrast, Carter et al. (2001) have demonstrated the application of SVMs to the problem of recognizing functional RNAs in genomic DNA. With respect to a typical protein-coding gene, RNA is an intermediate between the repository of genetic information (the DNA strand) and the functional product (the protein). Functional RNAs (fRNAs), in contrast, are RNA molecules that have a functional role in the cell and do not code for a protein molecule. Recognizing these RNAs in the DNA strand is difficult because they are typically short and lack the many constraints imposed upon genes that encode proteins. However, because theses gene are so short, they can be recognized effectively using a fixed-width sliding window. This is the approach used by Carter, Dubchak and Holbrook. Each window is encoded using two types of features: compositional features (frequencies of nucleotides and dinucleotides) and structural features (occurrences of six structural motifs associated with fRNAs). The SVM performs well, with leave-one-out error rates of approximately 0.7% to 16.8%, depending upon the organism. However, the SVM is compared to a neural network, which performs slightly better. The comparison is somewhat unfair because the neural network employs a structured network that builds in prior knowledge about the two different classes of inputs, whereas the SVM kernel treats all the inputs uniformly. Thus, this application provides a clear opportunity for engineering an SVM kernel.

Secondary structure

Finally, Hua and Sun (2001a) have demonstrated how to predict the secondary structure at each location along a protein strand. Secondary structure elements fall into three categories: helix, sheet or coil. Accordingly, this is a multi-class recognition problem, which Hua and Sun address in a straightforward fashion. The protein sequence is encoded in redundant binary fashion, using an 11-amino acid sliding window. An RBF kernel is used, and three separate SVMs are trained, one per secondary structure element. The final classification of a given amino acid is the label associated with the SVM that assigns the discriminant score that is farthest from zero. The resulting classifier achieves a per-residue accuracy of 73.5% on a standard data set, which is comparable to existing methods based upon neural networks.

1.5 Microarray gene expression analysis

All of the SVM applications described thus far have involved the analysis of biosequences. There is, however, an entirely different type of data, the analysis of which has received considerable attention recently [see (Knudsen, 2002) for a useful overview]. A *microarray* measures the number of copies of messenger RNA (mRNA) in a given sample of cells. The technology comes in two primary forms. The first technique involves affixing known DNA strands (called *probes*) to a 1 cm^2 glass slide. A fluorescently labeled sample of mRNA is then washed over the slide, and mRNAs that match the probes on the slide bind there. Subsequently, the dye is fluoresced under a microscope, and the intensity at each spot is measured. Each spot on the slide corresponds to a known gene; hence, each spot intensity indirectly indicates how many copies of that gene's mRNA exist in the sample. The second technique is similar to the first, except that the substrate is a silicon chip, and the probes are synthesized photolithographically on the surface of the silicon. Because synthesizing long sequences is expensive, many (between 20 and 40) spots are created for each gene, each spot containing copies of a relatively short (25 nucleotide) probe sequence. Again, the spot intensities are measured via fluorescence. The overall signal for a given gene is computed by combining the measurements from the corresponding spots. Using either technology, the end result is a collection of on the order of 10,000 measurements of gene activity per experiment. The microarray is appealling because of its ability to produce data in a high-throughput fashion. However, the data itself is quite noisy. Consequently, many research groups have resorted to the use of clustering and pattern recognition techniques to interpret their microarray data.

1.5.1 Gene classification

The first application of SVMs to microarray data involved the classification of yeast genes into functional categories (Brown et al., 2000). The microarray data was collected from several previous studies (DeRisi et al., 1997; Spellman et al., 1998; Chu et al., 1998) and had previously been analyzed using hierarchical clustering (Eisen et al., 1998). The data set consists of 79 glass slide microarray experiments, each measuring the activity of all approximately 6000 yeast genes. Based upon the previously published analysis, Brown *et al.* selected five functional classes from the MIPS Yeast Genome Database (Mewes et al., 2000)—tricarboxylic-acid pathway, respiration chain complexes, cytoplasmic ribosomal proteins, proteasome and histones— and measured the ability of the SVM to recognize members of each of these classes.

The SVM yields very good performance on this task. In comparison with a collection of traditional machine learning techniques, including Fisher's linear discriminant, C4.5, Parzen windows and MOC1, the SVM using either an RBF or third degree polynomial kernel is always the best performing method. Furthermore, the

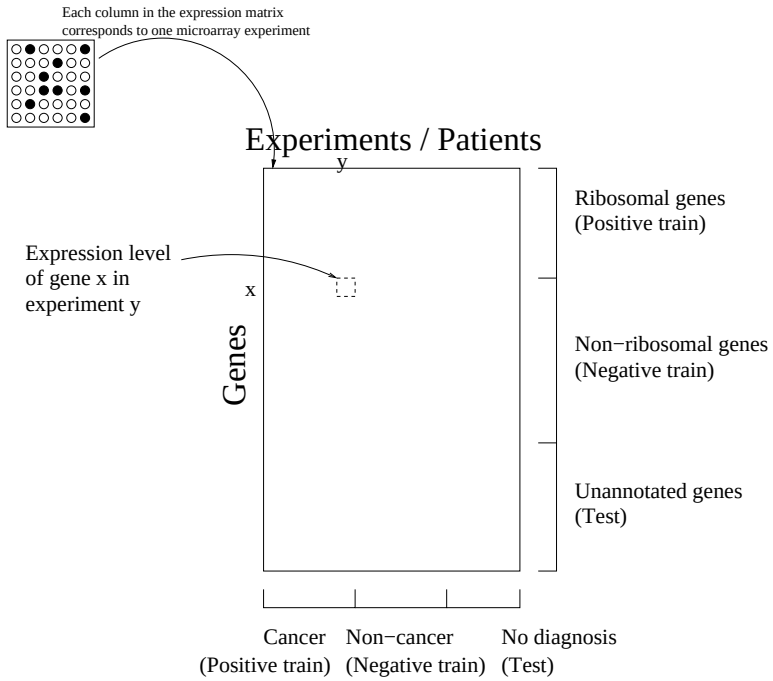

Figure 1.3 Classification tasks with microarray gene expression data. Data from many separate microarray experiments are collected into a single matrix, indexed by gene (row) and experiment (column). Classification can be performed along either dimension of this matrix: gene functional classification along the row dimension or diagnostic or prognostic patient classification along the column dimension.

study demonstrates that the SVM can be used both to make predictions for previously unannotated genes and to identify genes in the training set that have been mislabelled. Finally, an analysis of the mistakes made by the SVM shows that the learning algorithm's behavior is in many cases explainable due to noise or known biological anomalies. For example, some of the false negative examples in the TCA class turn out to be post-translationally modified, meaning that the regulation of these genes occur after the mRNA has been translated into a protein. In such cases, microarray data cannot be expected to provide useful insights.

1.5.2 Tissue classification

A more popular application of SVMs to the analysis of microarray data involves transposing the matrix of expression values. Rather than classifying each gene according to its profile across multiple experiments, the SVM learns to classify experiments. In this type of study, one experiment typically corresponds to one patient, and the classification label corresponds to a diagnosis. As such, the dimensionality of the problem is unusual: typically, a data set contains tens of experiments (examples) and thousands of genes (features).

Acute myeloid and acute lymphoblastic leukemia

The first application of a supervised learning algorithm to a tissue classification task was performed by Golub et al. (1999). They use a collection of 38 training samples and 34 test samples to train a simple learning algorithm called "weighted voting" to recognize the distinction between two forms of leukemia. This algorithm uses a feature selection metric, the signal-to-noise ratio $P(j)$, defined as follows:

$$P(j) = \left| \frac{\mu_1(j) - \mu_{-1}(j)}{\sigma_1(j) + \sigma_{-1}(j)} \right|, \tag{1.4}$$

where j is the gene index, μ_i is the mean of class 1 for gene j, μ_{-1} is the mean of class -1 for gene j, and σ_1 and σ_{-1} are the corresponding per-class standard deviations. This metric is closely related to the Fisher criterion score used in Fisher's linear discriminant (Duda and Hart, 1973).

Subsequently, Mukherjee et al. (1999) demonstrated the application of the SVM to this learning task. Because of the high dimensionality of the examples, a linear kernel is applied. Using the signal-to-noise ratio as a feature selection method, Mukherjee *et al.* improve upon the accuracy of the weighted voting method, reducing the error rate from 6% (2 errrors out of 34) to 0%. Note, however, that the method lacks a principled means of setting *a priori* the number of selected features. Without feature selection, the SVM makes 1 error, and with the number of features set too low (49 genes out of 7129), the number of errors is again 2.

Mukherjee et al. (1999) also describe a technique for assigning confidence values to the SVM predictions. The method assumes that the probability of a particular class, given a particular example, is approximately equal to the probability of the class given the corresponding SVM discriminant value. Discriminant values are estimated using leave-one-out cross-validation, and their distribution is estimated using an SVM-based, non-parametric density estimation algorithm (Mukherjee and

Vapnik, 1999). Introducing confidence levels results in 100% accuracy and between 0 and 4 rejects, depending upon the number of features selected.

Colon cancer

In work carried out concurrently, Moler et al. (2000) describe the application of SVMs to the recognition of colon cancer tissues. The data set consists of 40 colon cancer tumor and 22 normal colon tissues (Alon et al., 1999). This work describes a general, modular framework for the analysis of gene expression data, including generative, Bayesian methods for unsupervised and supervised learning, and the SVM for discriminative supervised learning.

The SVM is used in two ways, first to identify outlier or mislabeled training examples. An unsupervised naive Bayes class discovery method identifies four classes in the entire data set, and a multi-class (one-vs-all) linear SVM is trained and tested on all 1988 genes via leave-one-out cross-validation on these four classes. The authors claim that examples that are always support vectors are of particular interest: if these examples are consistently assigned to their labeled class, then they are considered unambiguous; if the examples are inconsistently assigned, then they may be mislabeled. Overall, the results suggest that the data can be divided into three subtypes (clearly tumor, mainly non-tumor and heterogeneous), which the authors claim may be of clinical significance.

The second SVM application involves recognition of tumor versus non-tumor tissues. A feature selection metric, the naive Bayes relevance (NBR) score, is proposed, which is based on the probability of a class given the observed value of the feature, under a Gaussian model. The performance of the SVM using various numbers of selected genes is compared to the performance of a naive Bayes classifier using the same genes. In every case, the SVM performs better than naive Bayes.

Ovarian cancer

In a similar set of experiments, Furey *et al.* apply linear SVMs with feature selection to three cancer data sets. The first data set consists of 31 tissues samples, including cancerous ovarian, normal ovarian and normal non-ovarian tissue. The others sets are the AML/ALL and colon cancer sets mentioned above. Following Golub et al. (1999), the signal-to-noise ratio is used to select genes for input to the classifier. The SVM successfully identifies a mislabeled sample in the ovarian set, and is able to produce a perfect classification. However, this classification is fragile with respect to the SVM parameter settings (softness of the margin and number of genes selected for input). Overall, the SVM provides reasonably good performance across multiple data sets, although the experiments also demonstrate that several perceptron-based algorithms perform similarly.

Soft tissue sarcoma

Segal et al. (2003b) use the SVM to develop a genome-based classification scheme for clear cell sarcoma. This type of tumor displays characteristics of both soft tissue sarcoma and melanoma. A linear SVM is trained to recognize the distinction between melanoma and soft tissue sarcoma, using 256 genes selected via a t-test. In a leave-one-out setting, the classifier correctly classifies 75 out of 76 examples. Subsequently, the trained classifier is applied to five previously unseen clear cell sarcoma examples, and places all five within the melanoma class. Thus, SVM analysis of gene expression profiles supports the classification of clear cell sarcoma as a distinct genomic subtype of melanoma.

In related work, Segal et al. (2003a) use SVMs to investigate the complex histopathology of adult soft tissue sarcomas. Here, the data set consists of 51 samples that have been classified by pathologists into nine histologic subtypes. The SVM, again using a t-test for feature selection, successfully recognizes the four subtypes for which molecular phenotypes are already known. Among the remaining samples, a combination of SVMs and hierarchical clustering uncovers a well-separated subset of the malignant fibrous hystiocytoma subtype, which is a particularly controversial subtype.

Recursive feature elimination

All of the methods described thus far for cancer classification rely upon a score (either the signal-to-noise ratio, NBR score or t-test) for selecting which genes to give to the SVM classifier. A significant drawback to these scores is that they treat each gene independently, thereby ignoring any significant gene-gene correlations that may occur in the data. Guyon et al. (2002) propose an SVM-based learning method, called SVM recursive feature elimination (SVM-RFE) that addresses this issue. The motivating idea is that the orientation of the separating hyperplane found by the SVM can be used to select informative features: if the plane is orthogonal to a particular feature dimension, then that feature is informative, and vice versa. Specifically, given an SVM with weight vector $\vec{w} = \sum_k \alpha_k y_k \vec{x}_k$, the ranking criterion for feature i is $c_i = (w_i)^2$. This criterion suggests the following wrapper-based learning method:

1. Initialize the data set to contain all features.
2. Train an SVM on the data set.
3. Rank features according to the criterion c.
4. Eliminate the lowest-ranked feature.
5. If more than one feature remains, return to step 2.

In practice, the algorithm is sped up by removing half of the features in step 4.

The SVM-RFE algorithm is tested on the AML/ALL and colon cancer data sets. For the leukemia data set, SVM-RFE identifies two genes that together yield zero leave-one-out error. In addition, several other classification algorithms, including the weighted voting algorithm, are applied to the data using the genes selected by SVM-RFE. The results show that the selection of genes is more important than the particular learning algorithm employed.

Gene selection

SVM-RFE has the dual goals of producing a good discriminator and reducing the number of genes to a manageable number. If we eliminate the first goal, then we are left with the problem of gene ranking. Identifying genes that exhibit predictive power in discriminating between two classes of samples is often the primary goal of a microarray study. Su et al. (2003) describe a tool called RankGene that produces gene rankings. One of the ranking metrics available in RankGene is the discriminant of a one-dimensional SVM trained on a given gene.

Multi-class classification

Many tissue classification analyses have been hampered somewhat by the dearth of useful, publically available gene expression data sets. Yeang et al. (2001) addressed this issue by producing a data set of 190 samples from 14 tumor classes.

This collection was later expanded by to include 308 samples, including 90 normal tissue samples (Ramaswamy et al., 2001). The initial study compares six different supervised learning methods: weighted voting, k-nearest neighbor and the SVM, each trained for multi-class classification using both a one-versus-all and an all-pairs approach. The signal-to-noise ratio is used for feature selection for the weighted voting and k-nearest neighbor, but feature selection is not applied to the SVM algorithm. Nonetheless, the one-versus-all SVM algorithm trained using all genes performs better than the all-pairs SVM and better than any of the other classifiers trained using 20, 40, 50, 100 or 200 genes. The second, larger study does apply SVM-RFE, but the best performance is again obtained by the one-versus-all SVM trained using all genes.

At this stage, the diagnosis and prognosis of cancer using microarray assays is still the subject of both hype and controversy. For example, an important and occasionally overlooked characteristic of these studies is the risk of introducing selection bias by choosing discriminative genes prior to performing cross-validation. Ambroise and McLachlan (2002) demonstrate that this bias occurs in several published studies, including in the SVM-RFE analysis performed by Guyon et al. (2002). A re-analysis of the colon cancer and leukemia data sets, taking into account the selection bias, shows that feature selection does not actually improve discrimination performance relative to an SVM trained from all of the genes. This result agrees with the results reported by Ramaswamy et al. (2001). Despite the controversy, a microarray assay is already in clinical trial in the Netherlands for determining whether breast cancer patients will receive adjuvant treatment (chemotherapy, tamoxifen or radiation) after surgery (Schubert, 2003), and at least five additional clinical trials are set to begin soon (Branca, 2003). Ironically, the Dutch microarray screen is based, in part, on a (non-SVM based) microarray analysis (van't Veer et al., 2002) that has been demonstrated independently to suffer from selection bias (Tibshirani and Efron, 2002).

1.6 Data fusion

Now that the human genome is more or less completely sequenced, more interest is being paid to the problem of data fusion, of integrating heterogeneous biological data. For example, for a given gene we might know the protein it encodes, that protein's similarity to other proteins, the mRNA expression levels associated with the given gene under hundreds of experimental conditions, the occurrences of known or inferred transcription factor binding sites in the upstream region of that gene, and the identities of many of the proteins that interact with the given gene's protein product. Each of these distinct data types provides one view of the molecular machinery of the cell.

Summing kernel matrices

Several efforts have been made at performing biological data fusion in the context of SVM learning. Pavlidis et al. (2001b, 2002) trained SVMs to recognize functional categories of yeast genes, using a combination of microarray gene expression data

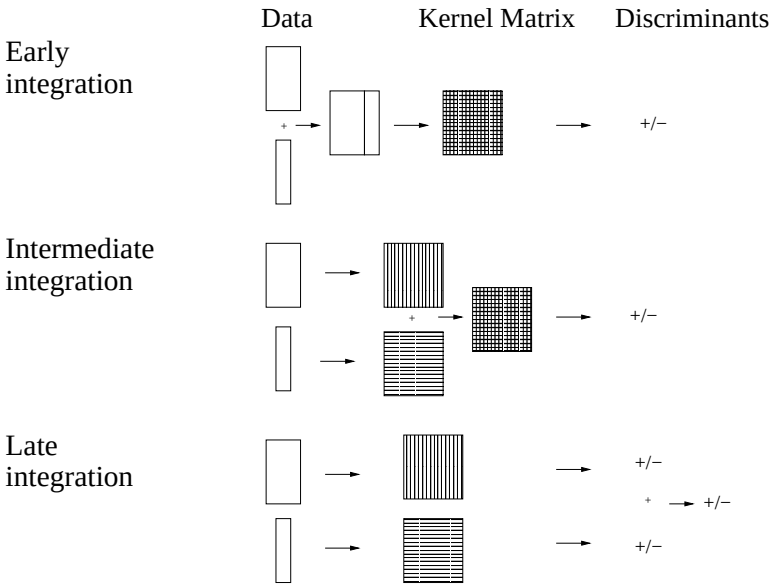

Figure 1.4 Three methods for learning from heterogeneous data with a support vector machine. In early integration, the two types of data are concatenated to form a single set of input vectors. In intermediate integration, the kernel values are computed separately for each data set and then summed. In late integration, one SVM is trained on each data type, and the resulting discriminant values are summed.

and phylogenetic profiles. In this case, both types of data are fixed-length, real-valued vectors, so a standard third-degree polynomial kernel is employed. Pavlidis *et al.* compare three different techniques for combining these two types of data (see Figure 1.4): early integration, in which the two vectors are simply concatenated, intermediate integration, in which two kernels are computed separately and then added, and late integration, in which two SVMs are trained and their discriminant scores are added. Intermediate integration provides the best results, presumably because it trades off making too many independence assumptions (in late integration) versus allowing too many dependencies (in early integration). The paper also presents some heuristic techniques for choosing scaling factors to be applied to each kernel function.

Kernel canonical correlation analysis

Another form of data fusion was performed by Vert and Kanehisa (2003). This approach integrates gene expression profiles with prior knowledge of a metabolic network. The network represents pathways of proteins that operate upon one another in the cell. Vert and Kanehisa hypothesize that gene expression patterns that are well measured (i.e., that correspond to actual biological events, such as the activation or inhibition of a particular pathway) are more likely to be shared by genes that are close to one another in the metabolic network. Accordingly, the expression data and the metabolic network are encoded into kernel functions, and these functions are combined in feature space using canonical correlation analysis (Bach and Jordan, 2002). Using yeast functional categories, an SVM trained from the combined kernel performs significantly better than an SVM trained only on expression data.

Semi-definite programming

Recently, Lanckriet et al. (2003) have described a new method for integrating heterogeneous genomic data. Similar to the work of Pavlidis et al. (2001b, 2002), the method involves summing a collection of kernel matrices, one per data set. In this case, however, each matrix is weighted, and Lanckriet *et al.* demonstrate how to optimize simultaneously the hyperplane selection and the selection of kernel weights. The result is a convex optimization problem that can be solved with semi-definite programming techniques. The paper demonstrates the utility of these techniques by solving the problem of predicting membrane proteins from heterogeneous data, including amino acid sequences, hydropathy profiles, gene expression data and known protein-protein interactions. An SVM algorithm trained from all of these data performs significantly better than the SVM trained on any single type of data and better than existing algorithms for membrane protein classification. Furthermore, the algorithm is robust to noise: when a randomly generated data set is included in the mix, the corresponding kernel function receives a weight close to zero, and the overall performance of the discriminator is essentially unchanged.

Expectation-maximization for missing data

Finally, Tsuda et al. (2003) describe a different type of data fusion algorithm. This approach applies a variant of the expectation-maximization algorithm (Dempster et al., 1977) to the problem of infering missing entries in a kernel matrix by using a second kernel matrix from an alternate data source. The method is demonstrated using two kernel matrices derived from two different types of bacterial protein sequences (16S rRNA and gyrase subunit B). The quality of the resulting matrix

is evaluated by using the matrix to perform unsupervised learning. The results suggest that this approach may prove useful in a supervised context as well.

1.7 Other applications

Cancer classification from methylation data

Model et al. (2001) describe a classification task very similar to the cancer classification tasks described above. The primary difference is that, in this case, the data comes from a methylation assay, rather than a microarray gene expression profile. Methylation is a molecular modification of DNA, in which a methyl group is added to the nucleotide cytosine. Methylation patterns in the upstream regions of genes are thought to be a major factor in gene regulation. Model *et al.* have developed a high-throughput method for collecting methylation data, and have used it to collect data from leukemia patients, 17 with AML and 8 with ALL. Each methylation pattern contains measurements from 81 positions along the DNA strand. The computational experiment consists of training a linear SVM to differentiate between AML and ALL. Many feature selection methods are employed, including principle components analysis, the signal-to-noise ratio, the Fisher criterion score, the t-test, and a method called backward elimination. The latter is essentially identical to the SVM-RFE algorithm of Guyon et al. (2002) and appears to have been invented independently. For this task, SVM-RFE does not outperform the linear feature selection methods. Instead, feature selection via the Fisher criterion score provides the best results.

Prediction of developmental age of *Drosophila* embryos

Perhaps one of the most unusual learning tasks is described by Myasnikova et al. (2002). They are interested in characterizing gene expression changes in *Drosophila* during development, and they measure these changes in a gene-specific fashion using fluorescent dyes and light microscopy of *Drosophila* embryos. In order to precisely and efficiently analyze the resulting data, they need an automatic method for determining the developmental age of a *Drosophila* embryo. To solve this problem, they use support vector regression (Drucker et al., 1997).

The data set consists of 103 embyros for which the precise developmental age is known. A microphotograph of each embryo is reduced, using previously developed techniques, to a table of values in which each row corresponds to a single cell, and columns represent the x and y coordinates of the nucleus and the expression levels of three genes in that cell. The resulting regression estimator appears to perform well, though no comparison to other algorithms is performed. The authors also demonstrate how factor analysis, performed on a data set of labeled and unlabeled examples, can be used to reduce the number of features to 3, thereby significantly increasing the speed of the regression estimation with no accompanying loss in accuracy.

Prediction of protein-protein interactions

Bock and Gough (2001) apply SVMs to the very important problem of predicting protein-protein interactions. This task fits cleanly into a binary discrimination framework: given a pair of proteins, the SVM predicts whether or not they interact. A critical question is how best to represent the protein pairs, and Bock and Gough

derive a set of features characterizing the charge, hydrophobicity, and surface tension at each amino acid in a given protein. Protein pairs are represented simply as the concatenation of the corresponding vectors. The SVM performs impressively, achieving an accuracy better than 80% in a cross-validated test. However, the experiment suffers from a significant flaw: the negative examples are generated randomly. Therefore, it is not clear whether the SVM is learning to differentiate between interating and non-interacting proteins pairs, or to differentiate between real and simulated protein pairs. Further experiments will need to be performed in order to validate these results.

Indeed, a subsequent experiment addressing this same problem shows the SVM performing comparably to a simple Bayesian technique (Gomez et al., 2003). The SVM's drawback, in this work, is that the training set is extremely large, and the SVM is consequently quite slow relative to the simpler method.

Peptide identification from mass spectrometry data

In tandem mass spectrometry, a sample of unknown proteins is enzymatically digested into relatively short strings of amino acids, called peptides. These peptides are size selected via mass spectrometry, fragmented via ionization, and the fragments are measured by a second mass spectrometer. The final spectrum contains peaks corresponding to all or most of the substrings in a single peptide. It is possible to infer the original peptide from the spectrum, using only the known masses of the amino acids. In practice, however, performing this task *de novo* is too difficult, and successful algorithms like Sequest (Eng et al., 1994) use an auxiliary database of known proteins. Sequest performs a simulation of tandem mass spectrometry on each peptide in the database, searching for a theoretical spectrum that matches the observed spectrum.

Anderson et al. (2003) apply the SVM to the problem of interpreting Sequest output. The algorithm produces a large number of false positives, and the SVM's task is to learn to differentiate true from false positives. Thus, the input to the classifier is a pair of spectra—observed and theoretical—and the output is a prediction—true positive or false positive. The input spectra are represented by a collection of thirteen parameters, reflecting the quality of the observed spectrum, the similarity of the observed and theoretical spectrum, and the difference between this match and the next-best match found by Sequest. The SVM uses a quadratic kernel function, and achieves error rates of 7-14%, depending upon the quality of the instrument used to generate the data. This compares favorably with QScore, a previously published, non-learning based probabilistic algorithm that addresses the same task (Moore et al., 2002). The same SVM has been subsequently used to construct an assay of the ubiquitin system (Gururaja et al., 2003), which is responsible for targeting proteins for degradation.

1.8 Discussion

Clearly, the application of support vector machine learning in computational biology is a popular and successful undertaking. The appeal of this approach is due in part

to the power of the SVM algorithm, and in part to the flexibility of the kernel approach to representing data. In particular, the kernel framework accommodates in a straightforward fashion many different types of data—vectors, strings, trees, graphs, etc.—that are common in biology. Also, kernels provide an easy way to incorporate biological knowledge and unlabeled data into the learning algorithm. A kernel matrix derived from a particular experiment can thus summarize the relevant features of the primary data, encapsulate biological knowledge, and serve as input to a wide variety of subsequent data analyses.

Finally, as an avenue for future research, the kernel approach to learning allows for a principled way to perform *transduction* (Gammerman et al., 1998). A transductive learning task is one in which the (unlabeled) test data is available to the algorithm *a priori*. In the post-genomic era, many computational biology tasks are transductive because the entire complement of genes or proteins in a given organism is known. Exploiting the finite nature of these learning tasks may lead to improved recognition performance in many biological domains.

References

U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 96:6745–6750, 1999.

S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. A basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.

C. Ambroise and G. J. McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences of the United States of America*, 99(10):6562–6566, 2002.

D. C. Anderson, W. Li, D. G. Payan, and W. S. Noble. A new algorithm for the evaluation of shotgun peptide sequencing in proteomics: support vector machine classification of peptide MS/MS spectra and SEQUEST scores. *Journal of Proteome Research*, 2(2):137–146, 2003.

Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

P. Baldi, Y. Chauvin, T. Hunkapiller, and M. A. McClure. Hidden Markov models of biological primary sequence information. *Proceedings of the National Academy of Sciences of the United States of America*, 91(3):1059–1063, 1994.

A. Ben-hur and D. Brutlag. Remote homology detection: a motif based approach. *Bioinformatics*, 2003. To appear.

J. R. Bock and D. A. Gough. Predicting protein-protein interactions from primary structure. *Bioinformatics*, 17:455–460, 2001.

B. E. Boser, I. M. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, July 1992. ACM Press.

M. Branca. Putting gene arrays to the test. *Science*, 300:238, 2003.

M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. Sugnet, T. S. Furey, Jr. M.

Ares, and D. Haussler. Knowledge-based analysis of microarray gene expression data using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America*, 97(1):262–267, 2000.

C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

Y.-D. Cai, X.-J. Liu, X.-B. Xu, and G.-P. Zhou. Support vector machines for predicting protein structural class. *BMC Bioinformatics*, 2(3), 2001.

R. J. Carter, I. Dubchak, and S. R. Holbrook. A computational approach to identify genes for functional RNAs in genomic sequences. *Nucleic Acids Research*, 29(19): 3928–3938, 2001.

K. C. Chou and D. Elrod. Protein subcellular location prediction. *Protein Engineering*, 12:107–118, 1999.

S. Chu, J. DeRisi, M. Eisen, J. Mulholland, D. Botstein, P. Brown, and I. Herskowitz. The transcriptional program of sporulation in budding yeast. *Science*, 282:699–705, 1998.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, UK, 2000.

S. Degroeve, B. De Baets, Y. Van de Peer, and P. Rouz. Feature subset selection for splice site prediction. *Bioinformatics*, 18:S75–S83, 2002.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–22, 1977.

J. L. DeRisi, V. R. Iyer, and P. O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278:680–686, 1997.

C. Ding and I. Dubchak. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17(4):349–358, 2001.

H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In *Advances in Neural Information Processing Systems*, pages 155–161, Cambridge, MA, 1997. MIT Press.

R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

M. Eisen, P. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95:14863–14868, 1998.

J. K. Eng, A. L. McCormack, and J. R. Yates. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *Journal of the American Society for Mass Spectrometry*, 5:976–989, 1994.

A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In G. F. Cooper and S. Moral, editors, *Proceedings of the Fourteenth Conference on*

Uncertainty in Artificial Intelligence, pages 148–155, San Francisco, CA, 1998. Morgan Kaufmann.

T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

S. M. Gomez, W. S. Noble, and A. Rzhetsky. Learning to predict protein-protein interactions. *Bioinformatics*, 2003. To appear.

M. Gribskov, R. Lüthy, and D. Eisenberg. Profile analysis. *Methods in Enzymology*, 183:146–159, 1990.

W. N. Grundy, T. L. Bailey, C. P. Elkan, and M. E. Baker. Meta-MEME: Motif-based hidden Markov models of protein families. *Computer Applications in the Biosciences*, 13(4):397–406, 1997.

T. Gururaja, W. Li, W. S. Noble, D. G. Payan, and D. C. Anderson. Multiple functional categories of proteins identified in an in vitro cellular ubiquitin affinity extract using shotgun peptide sequencing. *Journal of Proteome Research*, 2003. To appear.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

D. Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, University of California, Santa Cruz, Santa Cruz, CA, July 1999.

S. Henikoff and J. G. Henikoff. Automated assembly of protein blocks for database searching. *Nucleic Acids Research*, 19(23):6565–6572, 1991.

S. Hua and Z. Sun. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *Journal of Molecular Biology*, 308:397–407, 2001a.

S. Hua and Z. Sun. Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, 17(8):721–728, 2001b.

R. Hughey and A. Krogh. Hidden Markov models for sequence analysis: Extension and analysis of the basic method. *Computer Applications in the Biosciences*, 12 (2):95–107, 1996.

T. Jaakkola, M. Diekhans, and D. Haussler. Using the Fisher kernel method to detect remote protein homologies. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 149–158, Menlo Park, CA, 1999. AAAI Press.

T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1-2): 95–114, 2000.

R. Karchin, K. Karplus, and David Haussler. Classifying G-protein coupled receptors with support vector machines. *Bioinformatics*, 18:147–159, 2002.

K. Karplus, C. Barrett, and R. Hughey. Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, 14(10):846–56, 1998.

S. Knudsen. *A Biologist's Guide to Analysis of DNA Microarray Data*. Wiley, New York, 2002.

A. Krogh, M. Brown, I. Mian, K. Sjolander, and D. Haussler. Hidden Markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology*, 235:1501–1531, 1994.

G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A framework for genomic data fusion and its application to membrane protein prediction. Submitted for publication, 2003.

C. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for SVM protein classification. *Proceedings of the Pacific Symposium on Biocomputing*, 2002.

C. Leslie, E. Eskin, J. Weston, and W. S. Noble. Mismatch string kernels for discriminative protein classification. Submitted for publication, 2003a.

C. Leslie, E. Eskin, J. Weston, and W. S. Noble. Mismatch string kernels for SVM protein classification. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems*. MIT Press, 2003b.

C. Leslie and R. Kuang. Fast kernels for inexact string matching. In *Conference on Learning Theory*, 2003. To appear.

L. Liao and W. S. Noble. Combining pairwise sequence similarity and support vector machines for remote protein homology detection. In *Proceedings of the Sixth Annual International Conference on Computational Molecular Biology*, pages 225–232, 2002.

L. Liao and W. S. Noble. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *Journal of Computational Biology*, 2003. To appear.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

B. Logan, P. Moreno, B. Suzek, Z. Weng, and S. Kasif. A study of remote homology detection. Technical report, Cambridge Research Laboratory, June 2001. http://www.hpl.hp.com/techreports/Compaq-DEC/CRL-2001-5.html.

H. W. Mewes, D. Frishman, C. Gruber, B. Geier, D. Haase, A. Kaps, K. Lemcke, G. Mannhaupt, F. Pfeiffer, C Schüller, S. Stocker, and B. Weil. MIPS: a database for genomes and protein sequences. *Nucleic Acids Research*, 28(1):37–40, 2000.

F. Model, P. Adorján, A. Olek, and C. Piepenbrock. Feature selection for DNA methylation based cancer classification. *Bioinformatics*, 17(Suppl 1):S157–S164, 2001.

E. J. Moler, M. L. Chow, and I. S. Mian. Analysis of molecular profile data using

generative and discriminative methods. *Physiol Genomics*, 4:109–126, 2000.

R. E. Moore, M. K. Young, and T. D. Lee. Qscore: An algorithm for evaluating SEQUEST database search results. *Journal of the American Society for Mass Spectrometry*, 13(4):378–386, 2002.

S. Mukherjee, P. Tamayo, D. Slonim, A. Verri, T. Golub, J. Mesirov, and T. Poggio. Support vector machine classification of microarray data. Technical Report AI Memo 1677, Massachusetts Institute of Technology, 1999.

S. Mukherjee and V. Vapnik. Multivariate density estimation: An SVM approach. Technical Report AI Memo 1653, Massachusetts Institute of Technology, 1999.

A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–540, 1995.

E. Myasnikova, A. Samsonova, M. Samsonova, , and J. Reinitz. Support vector regression applied to the determination of the developmental age of a *Drosophila* embryo from its segmentation gene expression patterns. *Bioinformatics*, 18:S87–S95, 2002.

J. Park, K. Karplus, C. Barrett, R. Hughey, D. Haussler, T. Hubbard, and C. Chothia. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *Journal of Molecular Biology*, 284(4):1201–1210, 1998.

P. Pavlidis, T. S. Furey, M. Liberto, and W. N. Grundy. Promoter region-based classification of genes. *Proceedings of the Pacific Symposium on Biocomputing*, pages 151–163, 2001a.

P. Pavlidis, J. Weston, J. Cai, and W. N. Grundy. Gene functional classification from heterogeneous data. In *Proceedings of the Fifth International Conference on Computational Molecular Biology*, pages 242–248, 2001b.

P. Pavlidis, J. Weston, J. Cai, and W. S. Noble. Learning gene functional classifications from multiple data types. *Journal of Computational Biology*, 9(2):401–411, 2002.

W. R. Pearson. Rapid and sensitive sequence comparisions with FASTP and FASTA. *Methods in Enzymology*, 183:63–98, 1985.

M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 96(8):4285–4288, 1999.

S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T. R. Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences of the United States of America*, 98(26):15149–54, 2001.

C. M. Schubert. Microarray to be used as routine clinical screen. *Nature Medicine*,

9(1):9, 2003.

N. H. Segal, P. Pavlidis, C. R. Antonescu, R. G. Maki, W. S. Noble, J. M. Woodruff, J. J. Lewis, M. F. Brennan, A. N. Houghton, and C. Cordon-Cardo. Classification and subtype prediction of soft tissue sarcoma by functional genomics and support vector machine analysis. *American Journal of Pathology*, 2003a. To appear.

N. H. Segal, P. Pavlidis, W. S. Noble, C. R. Antonescu, A. Viale, U. V. Wesley, K. Busam, H. Gallardo, D. DeSantis, M. F. Brennan, C. Cordon-Cardo, J. D. Wolchok, and A. N. Houghton. Classification of clear cell sarcoma as melanoma of soft parts by genomic profiling. *Journal of Clinical Oncology*, 21:1775–1781, 2003b.

T. Smith and M. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol Biol Cell*, 9:3273–3297, 1998.

Y. Su, T. M. Mural, V. Pavlovic, M. Schaffer, and S. Kasif. RankGene: Identification of diagnostic genes based on expression data. *Bioinformatics*, 2003. To appear.

R. J. Tibshirani and B. Efron. Pre-validation and inference in microarrays. *Statistical Applications in Genetics and Molecular Biology*, 1(1):1–18, 2002.

K. Tsuda. Support vector classification with asymmetric kernel function. In M. Verleysen, editor, *Proceedings ESANN*, pages 183–188, 1999.

K. Tsuda, S. Akaho, and K. Asai. The em algorithm for kernel matrix completion with auxiliary data. *Journal of Machine Learning Research*, 4:67–81, May 2003.

K. Tsuda, T. Kin, and K. Asai. Marginalized kernels for biological sequences. *Bioinformatics*, 18:S268–S275, 2002.

L. J. van't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536, 2002.

V. N. Vapnik. *Statistical Learning Theory*. Adaptive and learning systems for signal processing, communications, and control. Wiley, New York, 1998.

Jean-Philippe Vert. Support vector machine prediction of signal peptide cleavage site using a new class of kernels for strings. In Russ B. Altman, A. Keith Dunker, Lawrence Hunter, Kevin Lauerdale, and Teri E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing*, pages 649–660. World Scientific, 2002a.

Jean-Philippe Vert and Minoru Kanehisa. Graph-driven features extraction from microarray data using diffusion kernels and kernel CCA. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.

Jean-Phlippe Vert. A tree kernel to analyze phylogenetic profiles. *Bioinformatics*, 18:S276–S284, 2002b.

S. V. N. Vishwanathan and A. J. Smola. Fast kernels for string and tree matching. In *Advances in Neural Information Processing Systems*, Cambridge, MA, 2003. MIT Press. To appear.

G. von Heijne. A new method for predicting signal sequence cleavage sites. *Nucleic Acids Research*, 14:4683–4690, 1986.

J. C. Wallace and S. Henikoff. PATMAT: a searching and extraction program for sequence, pattern and block queries and databases. *CABIOS*, 8:249–254, 1992.

C. Watkins. Dynamic alignment kernels. In A. J. Smola, P. Bartlett, B. Schölkopf, and C. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 1999.

C. Yeang, S. Ramaswamy, P. Tamayo, S. Mukherjee, R. R. Rifkin, M. Angelo, M. Reich, E. Lander, J. Mesirov, and T. Golub. Molecular classification of multiple tumor types. *Bioinformatics*, 17 Suppl 1.:S316–S322, 2001.

N. Zavaljevski and J. Reifman. Support vector machines with selective kernel scaling for protein classification and identification of key amino acid positions. *Bioinformatics*, 18(5):698–696, 2002.

A. Zien, G. Rätch, S. Mika, B. Schölkopf, T. Lengauer, and K.-R. Müller. Engineering support vector machine kernels that recognize translation initiation sites. *Bioinformatics*, 16(9):799–807, 2000.