Kernel methods in genomics and computational biology

Jean-Philippe Vert

October 17, 2005

Abstract

Support vector machines and kernel methods are increasingly popular in genomics and computational biology, due to their good performance in real-world applications and strong modularity that makes them suitable to a wide range of problems, from the classification of tumors to the automatic annotation of proteins. Their ability to work in high dimension, to process non-vectorial data, and the natural framework they provide to integrate heterogeneous data are particularly relevant to various problems arising in computational biology. In this chapter we survey some of the most prominent applications published so far, highlighting the particular developments in kernel methods triggered by problems in biology, and mention a few promising research directions likely to expand in the future.

1 INTRODUCTION

Recent years have witnessed a dramatic evolution in many fields of life science with the apparition and rapid spread of so-called high-throughput technologies, which generate huge amounts of data to characterize various aspects of biological samples or phenomena. To name just a few, DNA sequencing technologies have already provided the whole genome of several hundreds of species, including the human genome (Consortium, 2001; Venter, 2001); DNA microarrays (Schena et al., 1995), that allow the monitoring of the expression level of tens of thousands of transcripts simultaneously, opened the door to functional genomics, the elucidation of the functions of the genes found in the genomes (DeRisi et al., 1997); recent advances in ionization technology have boosted large-scale capabilities in mass spectrometry, and the rapidly growing field of proteomics, focusing on the systematic, large-scale analysis of proteins (Aebersold and Mann, 2003). As biology suddenly entered this new era characterized by the relatively cheap and easy generation of huge amounts of data, the urgent need for efficient methods to represent, store, process, analyze, and finally make sense out of these data triggered the parallel development of numerous data analysis algorithms in computational biology. Among them, kernel methods in general, and support vector machines (SVM) in particular, have quickly gained popularity for problems involving the classification and analysis of high-dimensional or complex data. Half a decade after the first pioneering papers (Mukherjee et al., 1998; Haussler, 1999; Jaakkola et al., 1999), these methods have been applied to a variety of problems in computational biology, with more than 100 research papers published in 2004 only[1]. The main reasons behind this fast development, beyond the generally good performances of SVM on real-world problems and ease of use provided by current implementations, involve (i) the particular capability of SVM to resist to high dimensional and noisy data, typically produced by various high-throughput technologies, and (ii) the possibility to process non-vectorial data, such as biological sequences, protein structures or gene networks, and to easily fuse heterogeneous data thanks to the use of kernels. More than a mere application of well-established methods to new datasets, the use of kernel methods in computational biology has been accompanied by new developments to match the specificities and the needs of the field, such as methods for feature selection in combination with the classification of high-dimensional data, the invention of string kernels to process biological sequences, or the development of methods to learn from several kernels simultaneously. In order to illustrate some of the most prominent applications of kernel methods in computational biology and the specific developments they triggered, this chapter focuses on selected applications related to the manipulation of high-dimensional data, the classification of biological sequences, and a few less developed but promising applications. This chapter is therefore not intended to be an exhaustive survey, but rather to illustrate with some examples why and how kernel methods have invaded the field of computational biology so rapidly. The interested reader will find more references in the book by Schölkopf et al. (2004) dedicated to the topic. Several kernels for structured data, such as sequences or trees, widely developed and used in computational biology, are also presented in detail in the book by Shawe-Taylor and Cristianini (2004).

2 CLASSIFICATION OF HIGH-DIMENSIONAL DATA

Several recent technologies, such as DNA microarrays, mass spectrometry or various miniaturized assays, provide thousands of quantitative parameters to characterize biological samples or phenomena. Mathematically speaking, the results of such experiments can be represented by high-dimensional vectors, and many applications involve the supervised classification of such data. Classifying data in high dimension with a limited number of training examples is a challenging task that most statistical procedures have difficulties dealing with, due in particular to the risk of overfitting the training data. The theoretical foundations of SVM and related methods, however, suggest that their use of regularization allows them to better resist to the curse of dimension than other methods. SVM were therefore naturally tested on a variety of datasets involving the classification of high-dimensional data, in particular for the analysis of tumor samples from gene expression data, and novel algorithms were developed in the framework of kernel methods to select a few relevant features for a given high-dimensional classification problem.

[1] A list of references is available at `http://cg.ensmp.fr/~vert/svn/bibli/html/biosvm.html`

2.1 Tumor Classification from Gene Expression Data

The early detection of cancer and prediction of cancer types from gene expression data have been among the first applications of kernel methods in computational biology (Mukherjee et al., 1998; Furey et al., 2000) and remain prominent. These applications have indeed potentially important impacts on the treatment of cancers, providing clinicians with an objective and possibly highly accurate information to choose the most appropriate form of treatment. In this context, SVM were widely applied and compared with other algorithms for the supervised classification of tumor samples from expression data, of typically several thousands of genes for each tumor. Examples include the discrimination between acute myeloid and acute lymphoblastic leukemia (Mukherjee et al., 1998), colon cancer and normal colon tissues (Moler et al., 2000), normal ovarian, normal non-ovarian and cancer ovarian tissues (Furey et al., 2000), melanoma, soft tissue sarcoma and clear cell sarcoma (Segal et al., 2003b), different types of soft tissue sarcomas (Segal et al., 2003a), or normal and gastric tumor tissues (Meireles et al., 2003), to name just a few. Another typical application is the prediction of the future evolution of a tumor, such as the discrimination between relapsing and nonrelapsing Wilms tumors (Williams et al., 2004), the prediction of metastatic or non-metastatic squamous cell carcinoma of the oral cavity (O'Donnell et al., 2005), or the discrimination between diffuse large B-cell lymphoma with positive or negative treatment outcome (Shipp et al., 2002).

The SVM used in these studies are usually linear hard-margin SVM, or linear soft-margin SVM with a default C parameter value. Concerning the choice of the kernel, several studies observe that nonlinear kernels tend to decrease performance (Ben-Dor et al., 2000; Valentini, 2002) compared to the simplest linear kernel, which is coherent with the intuition that the complexity of learning non-linear functions in very high dimension does not play in their favor. On the other hand, the choice of hard-margin SVM, sometimes advocated as a default method when data are linearly separable, is certainly worth questioning in more details. Indeed, the theoretical foundations of SVM suggest that in order to learn in high dimension, one should rather increase the importance of regularization as opposed to fitting the data, which corresponds to decreasing the C parameter of the soft-margin formulation. A few recent papers highlight indeed the fact that the choice of C has an important effect on the generalization performance of SVM for classification of gene expression data (Huang and Kecman, 2005).

A general conclusion of these numerous studies is that SVM generally provide good classification accuracy in spite of the large dimension of the data. For example, in a comparative study of several algorithms for multi-class supervised classification, including naive Bayes, k-nearest neighbors and decision trees, Li et al. (2004) note that "[SVM] achieve better performance than any other classifiers on almost all the datasets". However, it is fair to mention that other studies conclude that most algorithms that take into account the problem of large dimension either through regularization, or through feature selection, reach roughly similar accuracy on most datasets (Ben-Dor et al., 2000). From a practical point of view, the use of the simplest linear kernel and of the soft-margin formulation of SVM seems to be a reasonable default strategy for this application.

2.2 Feature Selection

In the classification of microarray data, it is often important, both for classification performance, biomarker identification and interpretation of results, to select only a few discriminative genes among the thousands of candidates available on a typical microarray. While the literature on feature selection is older and goes beyond the field of kernel methods, several interesting developments with kernel methods have been proposed in the recent years, explicitly motivated by the problem of gene selection from microarray data.

For example, Su et al. (2003) propose to evaluate the predictive power of a each single gene for a given classification task by the value of the functional minimized by a one-dimensional SVM, trained to classify samples from the expression of only the single gene of interest. This criterion can then be used to rank genes and select only a few with important predictive power. This procedure therefore belongs to the so-called *filter* approach to feature selection, where a criterion (here using SVM) to measure the relevance of each feature is defined, and only relevant features according to this criterion are kept.

A second general strategy for feature selection is the so-called *wrapper* approach, where feature selection alternates with the training of a classifier. The now widely-used recursive feature elimination (RFE) procedure of Guyon et al. (2002), which iteratively selects smaller and smaller sets of genes and trains SVM, follows this strategy. RFE can only be applied with linear SVM, which is nevertheless not a limitation as long as many features remain, and works as follows. Starting from the full set of genes, a linear SVM is trained and the genes with the smallest weights in the resulting linear discrimination function are eliminated. The procedure is then repeated iteratively starting from the set of remaining genes, and stops when a desired number of genes is reached.

Finally, a third strategy for feature selection, called *embedded* approach, combines the learning of a classifier and the selection of features in a single step. A kernel method following this strategy has been implemented in the joint classifier and feature optimization (JCFO) procedure of Krishnapuram et al. (2004). JCFO is roughly speaking a variant of SVM with a Bayesian formulation, in which sparseness is obtained both for the features and the classifier expansion in terms of kernel by appropriate choices of prior probabilities. The precise description of the complete procedure to train this algorithm, involving an expectation-maximization (EM) iteration, would go beyond the scope of this chapter and the interested reader is referred to the original publication for further practical details.

Generally speaking, and in spite of these efforts to develop clever algorithms, the effect of feature selection on the classification accuracy of SVM is still debated. Although very good results are sometimes reported, for example for the JCFO procedure (Krishnapuram et al., 2004), several studies conclude that feature selection, for example with procedures like RFE, do not actually improve the accuracy of SVM trained on all genes (Ambroise and McLachlan, 2002; Ramaswamy et al., 2001). The relevance of feature selection algorithms for gene expression data is therefore currently still a research topic, that practitioners should test and assess case by case.

2.3 Other High-Dimensional Data in Computational Biology

While early applications of kernel methods to high-dimensional data in genomics and bioinformatics mainly focused on gene expression data, a number of other applications have flourished more recently, some being likely to expand quickly as major applications of machine learning algorithms. For example, studies focusing on tissue classification from data obtained by other technologies, such as methylation assays, to monitor the patterns of cytosine methylation in the upstream regions of genes (Model et al., 2001), or array comparative genomic hybridization (CGH), to measure gene copy number changes in hundreds of genes simultaneously (Aliferis et al., 2002) are starting to accumulate. A huge field of application that still barely caught the interest of the machine learning community is *proteomics*, that is, the quantitative study of the protein content of cells and tissues. Technologies such as tandem mass spectromtetry, to monitor the protein content of a biological sample, are now well developed, and classification of tissues from these data is a future potential application of SVM (Wu et al., 2003; Wagner et al., 2003). Applications in toxicogenomics (Steiner et al., 2004), chemogenomics (Bao and Sun, 2002; Bock and Gough, 2002) and analysis of single nucleotide polyphormisms (Yoon et al., 2003; Listgarten et al., 2004) are also promising applications for which the capacity of SVM to classify high-dimensional data has only started to be exploited.

3 SEQUENCE CLASSIFICATION

The various genome sequencing projects have produced huge amounts of sequence data that need to be analyzed. In particular, the urgent need for methods to automatically process, segment, annotate and classify various sequence data has triggered the fast development of numerous algorithms for strings. In this context, the possibility offered by kernel methods to process any type of data, as soon as a kernel for the data to be processed is available, has been quickly exploited to offer the power of state-of-the-art machine learning algorithms to sequence processing.

Problems that arise in computational biology consist in processing either sets of sequences of a fixed length, or sets of sequences with variable lengths. From a technical point of view the two problems slightly differ: while there are natural ways to encode fixed-length sequences as fixed-length vectors, making them amenable to processing by most learning algorithms, manipulating variable-length sequences is less obvious. In both cases, many successful applications of SVM have been reported, combining ingenious developments of string kernels, sometimes specifically adapted to a given classification task, with the power of SVM.

3.1 Kernels for Fixed-Length Sequences

Problems involving the classification of fixed-length sequences appear typically when one wants to predict a property along a sequence, such as the local structure or solvent accessibility along a protein sequence. In that case, indeed, a common approach is to use a moving window, that is, to predict the property at each position independently from the others, and to base the prediction only on the nearby sequence contained in a small window around the site of interest. More formally, this requires the construction of predictive models that take a sequence of fixed length as input to predict the property of interest, the length of the sequences being exactly the width of the window.

To fix notations, let us denote by p the common length of the sequences, and by $x = x_1 \ldots x_p$ a typical sequence, where each x_i is a letter from the alphabet, e.g., an amino-acid. The most natural way to transform such a sequence into a vector of fixed length is to first encode each letter itself into a vector of fixed length k, and then to concatenate the codes of the successive letters to obtain a vector of size pk for the whole sequence. A simple code for letters is the following so-called sparse encoding: denoting by a the size of the alphabet, the i-th letter of the alphabet is encoded as a vector of dimension a containing only zeros, except for the i-th dimension that is set to 1. For example, in the case of nucleotide sequences with alphabet (A, C, G, T), the codes for A, C, G and T would respectively be $(1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0)$ and $(0, 0, 0, 1)$ and the code for the sequence of length 3 AGT would be $(1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1)$. Several more evolved codes for single letters have also been proposed. For example, if one has a prior matrix of pairwise similarities between letters, such as widely-used similarity matrices between amino-acids, it is possible to replace the 0/1 sparse encoding of a given letter by the vector of similarity with other letters; hence the A in the previous example could for instance be represented by the vector $(1, 0, 0.5, 0)$ to emphasize one's belief that A and G share some similarity. This is particularly relevant for biological sequences where mutations of single letters to similar letters are very common. Alternatively, instead of using a prior matrix of similarity, one can automatically align the sequence of interest to similar sequences in a large sequence database, and encode each position by the frequency of each letter in the alignment. As a trivial example, if our previous sequence AGT was found to be aligned to the following sequences: AGA, AGC, CGT, ATT, then it could be encoded by the vector $(0.8, 0.2, 0, 0, 0, 0, 0.8, 0.2, 0.2, 0.2, 0, 0.6)$, corresponding to the respective frequencies of each letter at each position.

In terms of kernel, it is easy to see that the inner product between sparsely encoded sequences is the number of positions with identical letter. In this representation, any linear classifier, such as that learned by a SVM, associates a weight to each feature, that is, to each letter at each position, and the score of a sequence is the sum of the scores of its letters. Such a classifier is usually referred to as a position-specific score matrix in bioinformatics. Similar interpretations can be given for other letter encodings. An interesting extension of these linear kernels for sequences is to raise them to some small power d; in that case, the dimension of the feature space used by kernel methods increases, and the new features correspond to all products of d original features. This is particularly appealing for the sparse encoding, because a product of d binary factors is a binary variable equal to 1 if and only if all factors are 1, meaning that the features created by the sparse encoding to the power d exactly indicate the simultaneous presence of up to d particular letters at d particular positions. The trick to take a linear kernel to some power is therefore a convenient way to create a classifier for problems that involve the presence of several particular letters at particular positions.

A first limitation of these kernels is that they do not contain any information about the order of the letters: they are for example left unchanged if the letters in all sequences are shuffled according to any given permutation. Several attempts to include ordering information have been proposed. For example, Rätsch et al. (2005) replace the local encoding of single letters by a local encoding of k consecutive letters; Zien et al. (2000) propose an ingenious variant to the polynomial kernel in order to restrict the feature space to products of features at nearby positions only.

A second limitation of these kernels is that the comparison of two sequences only involves the comparison of features at identical positions. This can be problematic in the case of biological sequences, where insertion of deletions of letters are common, resulting in possible shifts within a window. This problem led Meinicke et al. (2004) to propose a kernel which incorporates a comparison of features at nearby positions, using the following trick: if a feature f (e.g., binary or continuous) appears at position i in the first sequence, and a feature g appears at position j in the second sequence, then the kernel between the two sequences is increased by $K_0(f,g)\exp\left(-(i-j)^2/s\right)$, where $K_0(.,.)$ is a basic kernel between the features such as the simple product. When σ is chosen very large, then one recovers the classical kernels obtained by comparing only identical positions ($i=j$); the important point here is that for smaller values of σ, features can contribute positively even though they might be located at different positions on the sequences.

The applications of kernels for fixed-length sequences to solve problems in computational biology are already numerous. For example, they have been widely used to predict local properties along protein sequences using a moving window, such as secondary structure (Hua and Sun, 2001a; Guermeur et al., 2004), disulfide bridges involving cysteines (Passerini and Frasconi, 2004; Chen et al., 2004), phosphorylation sites (Kim et al., 2004), interface residues (Yan et al., 2004; Res et al., 2005), or solvent accessibility (Yuan et al., 2002). Another important field of application is the annotation of DNA, using fixed-length windows centered on a candidate point of interest as an input to a classifier to detect translation initiation sites (Zien et al., 2000; Meinicke et al., 2004), splice sites (Degroeve et al., 2005; Rätsch et al., 2005), or binding sites of transcription factors (O'Flanagan et al., 2005; Sharan and Myers, 2005). The recent interest in short RNA such as antisense oligonucleotides or small interfering RNAs for sequence-specific knockdown of messenger RNAs has also resulted in several works involving classification of such sequences, which have typically a fixed length by nature (Camps-Valls et al., 2004; Teramoto et al., 2005). Another important application field for these methods is immunoinformatics, including the prediction of peptides that can elicit an immune response (Dönnes and Elofsson, 2002; Bhasin and Raghava, 2004), or the classification of immunoglobulins collected from sain or ill patients (Zavaljevski et al., 2002; Yu et al., 2005). In most of these applications, SVM lead to comparable if not better prediction accuracy than competing state-of-the-art methods such as neural networks.

3.2 Kernels for Variable-Length Sequences

Many problems in computational biology involve sequences of different lengths. For example, the automatic functional or structural annotation of genes found in sequenced genomes requires the processing of amino-acid sequences with no fixed length. Learning from variable-length sequences is a more challenging problem than learning from fixed-length sequences, because there is no natural way to transform a variable-length string into a vector. For kernel methods, this issue boils down to the problem of defining kernels for variable-length strings, a topic that has deserved a lot of attention in the last few years and has given rise to a variety of ingenious solutions summarized in this section.

The most common approach to make a kernel for strings, as for many other types of data, is to design explicitly a set of numerical features that can be extracted from strings, and then to form a kernel as a dot product between the resulting feature vectors. As an example, Leslie et al. (2002) represent a sequence by the vector of counts of occurrences of all possible k-mers in the sequence, for a given integer k, effectively resulting in a vector of dimension a^k, where a is the size of the alphabet. As an example, the sequence $AACGTCACGAA$ over the alphabet (A, C, G, T) is represented by the 16-dimensional vector $(2, 2, 0, 0, 1, 0, 2, 0, 1, 0, 0, 1, 0, 1, 0, 0)$ for $k = 2$, where the dimensions are the counts of occurrences of each 2-mer $AA, AC, ..., TG, TT$ lexicographically ordered. The resulting spectrum kernel between this sequence and the sequence $ACGAAA$, defined as the linear product between the two 16-dimensional representation vectors, is equal to 9. It should be noted that although the number of possible k-mers easily reaches the order of several thousands as soon as k is equal to 3 or 4, classification of sequences by SVM in this high-dimensional space results in fairly good results. A major advantage of the spectrum kernel is its fast computation; indeed, the set of k-mers appearing in a given sequence can be indexed in linear time in a trie structure, and the inner product between two vectors is linear with respect to the non-zero coordinates, i.e., at most linear in the total lengths of the sequences. Several variants to the basic spectrum kernel have also been proposed, including for example kernels based on counts of k-mers appearing with up to m mismatches in the sequences (Leslie et al., 2004).

Another natural approach to vector representation for variable-length strings is to replace each letter by one or several numerical features, such as physico-chemical properties of amino-acids, and then to extract features from the resulting variable-length numerical time series using classical signal processing techniques such as Fourier transforms (Wang et al., 2004) or autocorrelation analysis (Zhang et al., 2003a). For example, if $h_1, \ldots, h_n$ denote n numerical features associated to the successive letters of a sequence of length n, then the autocorrelation function r_j for a given $j > 0$ is defined by

$$r_j = \frac{1}{n-j} \sum_{i=1}^{n-j} h_i h_{i+j}.$$

One can them keep a fixed numbers of these coefficients, for example $r_1, \ldots, r_J$, and create a J-dimensional vector to represent each sequence.

A completely different approach for kernel design is to derive them from probabilistic models. Indeed, before the interest on string kernels grew, a number of ingenious probabilistic models had been defined to represent biological sequences or families of sequences, including for example Markov and hidden Markov models for protein sequences, or stochastic context-free grammars for RNA sequences (Durbin et al., 1998). Several authors have therefore explored the possibility to use such models to make kernels, starting with the seminal work of Jaakkola et al. (2000) that introduced the *Fisher kernel*. The Fisher kernel is a general method to extract a fixed number of features from any data x for which a parametric probabilistic model P_θ is defined. Here, θ represents a continuous d-dimensional vector of parameters for the probabilistic model, such as transition and emission probabilities for a hidden Markov model, and each P_θ is a probability distribution. Once a particular parameter θ_0 is chosen to fit a given set of objects, for example by maximum likelihood, then a d-dimensional feature vector for each individual object x can be extracted by taking the gradient in the parameter space of the log-likelihood of the point:

$$\phi(x) = \nabla_\theta \log P_\theta(x).$$

The intuitive interpretation of this feature vector, usually referred to as the Fisher score in statistics, is that it represents how changes in the d parameters affect the likelihood of the point x. In other word, one feature is extracted for each parameter of the model; the particularities of the data point are seen from the eyes of the parameters of the probabilistic model. The Fisher kernel is then obtained as the dot product of these d-dimensional vectors, eventually multiplied by the inverse of the Fisher information matrix to render it independent of the parametrization of the model.

A second line of thoughts to make a kernel out of a parametric probabilistic model is to use the concept of covariance kernels (Seeger, 2002), that is, kernels of the form:

$$K(x, x') = \int P_\theta(x) P_\theta(x') d\mu(\theta),$$

where $d\mu$ is a prior distribution on the parameter space. Here, the features correspond to the likelihoods of the objects under all distributions of the probabilistic model; objects are considered similar when they have large likelihoods under similar distributions. An important difference with the kernels seen so far is that here, no explicit extraction of finite-dimensional vectors can be performed. Hence for practical applications one must chose probabilistic models that allow the computation of the integral above. This was carried by Cuturi and Vert (2005) who present a family of variable-length Markov models for strings and an algorithm to perform the integral over parameters and models in the same time, resulting in a string kernel with linear complexity in time and memory with respect to the total length of the sequences.

Alternatively, many probabilistic models for biological sequences, such as hidden Markov models, involve a hidden variable that is marginalized over to obtain the probability of a sequence, i.e., can be written as

$$P(x) = \sum_h P(x, h).$$

For such distributions, Tsuda et al. (2002) introduced the notion of *marginalized kernel*, obtained by marginalizing a kernel for the complete variable over the hidden variable. More precisely, assuming that a kernel for objects of the form (x, h) is defined, the marginalized kernel for observed objects x is given by

$$K(x, x') = \sum_{h,h'} K\left((x, h), (x', h')\right) P(h|x)P(h'|x').$$

In order to motivate this definition with a simple example, let us consider a hidden Markov model with two possible hidden states, to model sequences with two possible regimes, such as introns/exons in eukaryotic genes. In that case the hidden variable corresponding to a sequence x of length n is a binary sequence h of length n describing the states along the sequence. For two sequences x and x', if the correct hidden states h and h' were known, such as the correct decomposition into introns and exons, then it would make sense to define a kernel $K\left((x, h), (x', h')\right)$ taking into account the specific decomposition of the sequences into two regimes; for example, the kernel for complete data could be a spectrum kernel restricted to the exons, i.e., to positions with a particular state. Because the actual hidden states are not known in practice, the marginalization over the hidden state of this kernel using an adequate probabilistic model can be interpreted as an attempt to apply the kernel for complete data by guessing the hidden variables. As for the covariance kernel, marginalized kernels can often not be expressed as inner products between feature vectors, and require computational tricks to be computed. Several beautiful examples of such kernels for various probabilistic models have been worked out, including hidden Markov models for sequences (Tsuda et al., 2002; Vert et al., 2006), stochastic context-free grammars for RNA sequences (Kin et al., 2002), or random walk models on graphs for molecular structures (Kashima et al., 2004).

Following a different line of thought, Haussler (1999) introduced the concept of *convolution kernels* for objects that can be decomposed into subparts, such as sequences or trees. For example, the concatenation of two strings x_1 and x_2 results in another string $x = x_1x_2$. If two initial string kernels K_1 and K_2 are chosen, then a new string kernel is obtained by convolution of the initial kernels following the equation:

$$K(x, x') = \sum_{x=x_1x_2, x'=x'_1x'_2} K_1(x_1, x'_1)K_2(x_2, x'_2).$$

Here the sum is over all possible decompositions of x and x' into two concatenated subsequences. The rational behind this approach is that it allows the combination of different kernels adapted to different parts of the sequences, such as introns/exons or gaps/aligned residues in alignment, without knowing the exact segmentation of the sequences. Besides proving that the convolution of two kernels is a valid kernel, Haussler (1999) gives several examples of convolution kernels relevant for biological sequences; for example, he shows that the joint probability $P(x, x')$ of two sequences under a pair HMM model is a valid kernel, under mild assumptions. This work is extended by Vert et al. (2004) where a valid convolution kernel based on the alignment of two sequences is proposed. This kernel, named *local alignment kernel*, is a close relative of the widely used Smith-Waterman local alignment score (Smith and Waterman, 1981), and gives excellent results on the problem of detecting remote homologs of proteins.

Finally, another popular approach to design features and therefore kernels for biological sequences is to "project" them onto a fixed dictionary of sequences or motifs, using classical similarity measures, and to use the resulting vector of similarities as feature vector. For example, Logan et al. (2001) represent each sequence by a 10,000-dimensional vector indicating the presence of 10,000 motifs of the BLOCKS database; similarly, Ben-Hur and Brutlag (2003) use a vector that indicates the presence or absence of about 500,000 motifs in the eMOTIF database, requiring the use of a trie structure to compute efficiently the kernel without explicitly storing the 500,000 features; and Liao and Noble (2003) represent each sequence by a vector of sequence similarities with a fixed set of sequences.

These kernels for variable-length sequences have been widely applied, often in combination with SVM, to various classification tasks in computational biology. Examples including the prediction of protein structural or functional classes from their primary sequence (Ding and Dubchak, 2001; Jaakkola et al., 2000; Vert et al., 2004; Karchin et al., 2002; Cai et al., 2003), the prediction of the subcellular localization of proteins (Hua and Sun, 2001b; Park and Kanehisa, 2003; Matsuda et al., 2005), the classification of transfer RNA (Kin et al., 2002) and non-coding RNA (Karklin et al., 2005), the prediction of pseudo-exons and alternatively spliced exons (Zhang et al., 2003b; Dror et al., 2005), the separation of mixed plant-pathogen EST collections (Friedel et al., 2005), the classification of mammalian viral genomes (Rose et al., 2005), or the prediction of ribosomal proteins (Lin et al., 2002).

This short review of kernels developed for the purpose of biological sequence classification, besides highlighting the dynamism of research in kernel methods resulting from practical needs in computational biology, naturally raises the practical question of which kernel to use for a given application. Although no clear answer has emerged yet, some lessons can be learned from early studies. First, there is certainly no kernel universally better than others, and the choice of kernel should depend on the targeted application. Intuitively, a kernel for a classification task is likely to work well if it is based on features relevant to the task; for example, a kernel based on sequence alignments, such as the local alignment kernel, gives excellent results on remote homology detection problems, while a kernel based on the global content of sequences in short subsequences, such as the spectrum kernel, works well for the prediction of subcellular localization. Although some methods for systematic selection and combination of kernels are starting to emerge (see next section), empirical evaluation of different kernels on a given problem seems to be the most common way to chose a kernel. Another important point to notice, besides the classification accuracy obtained with a kernel, is its computational cost. Indeed, practical applications often involve datasets of thousands or tenth of thousands of sequences, and the computational cost of a method can become a critical factor in this context, in particular in an online setting. The kernels presented above differ a lot in their computational cost, ranging from fast linear-time kernels like the spectrum kernel, to slower kernels like the quadratic-time local alignment kernel. The final choice of kernel for a given application often results from a trade-off between classification performance and computational burden.

4 OTHER APPLICATIONS AND FUTURE TRENDS

Besides the important applications mentioned in the previous sections, several other attempts to import ideas of kernel methods in computational biology have emerged recently. In this section we highlight three promising directions that are likely to develop quickly in the near future: the engineering of new kernels, the development of methods to handle multiple kernels, and the use of kernel methods for graphs in systems biology.

4.1 More Kernels

The power of kernel methods to process virtually any sort of data as soon as a valid kernel is defined has recently been exploited for a variety of data, besides high-dimensional data and sequences. For example, Vert (2002) derives a kernel for phylogenetic profiles, that is, a signature indicating the presence or absence of each gene in all sequenced genomes. Several recent works have investigated kernels for protein 3D structures, a topic that is likely to expand quickly with the foreseeable availability of predicted or solved structures for whole genomes (Dobson and Doig, 2003; Borgwardt et al., 2005). For smaller molecules, several kernels based on planar or 3D structures have emerged, with many potential applications in computational chemistry (Kashima et al., 2004; Mahé et al., 2005; Swamidass et al., 2005). This trend to develop more and more kernels, often designed for specific data and applications, is likely to continue in the future because it has proved to be a good approach to obtain efficient algorithms for real-world applications. A nice by-product of these efforts, which is still barely exploited, is the fact that any kernel can be used by any kernel methods, paving the way to a multitude of applications such as clustering (Qin et al., 2003) or data visualization (Komura et al., 2005).

4.2 Integration of Heterogeneous Data

Operations on kernels provide simple and powerful tools to integrate heterogeneous data or multiple kernels; this is particularly relevant in computational biology, where biological objects can typically be described by heterogeneous representations, and the availability of a large number of possible kernels for even a single representation raises the question of choice or combination of kernels. Suppose for instance that one wants to perform a functional classification of genes based on their sequences, expression over a series of experiments, evolutionary conservation, and position in an interaction network. A natural approach with kernel methods is to start by defining one or several kernels for each sort of a data, that is, string kernels for the gene sequences, vector kernels to process the expression profiles, etc... The apparent heterogeneity of data types then vanishes as one simply obtains a family of kernel functions $K_1, ..., K_p$. In order to learn from all data simultaneously, the simplest approach is to define an integrated kernel as the sum of the initial kernels:

$$K = \sum_{i=1}^{p} K_i.$$

The rational behind this sum is that if each kernel is a simple dot product, then the sum of dot products is equal to the dot product of the concatenated vectors. In other words, taking a sum of kernels amounts to putting all features of each individual kernel together; if different features in different kernels are relevant for a given problem, then one expects the kernel method trained on the integrated kernel to pick those relevant features. This idea was pioneered by Pavlidis et al. (2002) where gene expression profiles and gene phylogenetic profiles are integrated to predict the functional classes of genes, effectively integrating evolutionary and transcriptional information.

An interesting generalization of this approach is to form a convex combination of kernels, of the form:

$$K = \sum_{i=1}^{p} w_i K_i,$$

where the w_i are nonnegative weights. Lanckriet et al. (2004) propose a general framework, based on semidefinite programming, to optimize the weights and learn a discrimination function for a given classification task simultaneously. Promising empirical results on gene functional classification show that by integrating several kernels, better results than each individual kernel can be obtained.

Finally, other kernel methods can be used to compare and search correlation between heterogeneous data. For example, Vert and Kanehisa (2003) propose to use a kernelized version of canonical correlation analysis (CCA) to compare gene expression data, on the one hand, with the position of genes in the metabolic network, on the other hand. Each type of data is first converted into a kernel for genes, the information about gene positions in the metabolic network being encoded with the so-called diffusion kernel (Kondor and Vert, 2004). These two kernels define embeddings of the set of genes into two Euclidean spaces, in which correlated directions are detected by CCA. It is then shown that the directions detected in the feature space of the diffusion kernel can be interpreted as clusters in the metabolic network, resulting in a method to monitor the expression patterns of metabolic pathways.

4.3 Kernel Methods in Systems Biology

Another promising field of research where kernel methods can certainly contribute is *systems biology*, which roughly speaking focuses on the analysis of biological systems of interacting molecules, in particular biological networks.

A first avenue of research is the reconstruction of biological networks from high-throughput data. For example, the prediction of interacting proteins to reconstruct the interaction network can be posed as a binary classification problem – given a pair of proteins, do they interact or not?–, and can therefore be tackled with SVM as soon as a kernel between *pairs* of proteins is defined. As the primary data available to make the interaction prediction are about each single protein, it is natural to try to derive kernels for pairs of protein from kernel for single proteins. This has been carried out for example by Bock and Gough (2001) who characterize each protein by a vector, and concatenate two such individual vectors to represent a protein pair. Observing that there is usually no order in a protein pair, Martin et al. (2005) and Ben-Hur and Noble (2005) propose to define a kernel between pairs (A, B) and (C, D) by the equation:

$$K_p\left((A, B), (C, D)\right) = K_i(A, C)K_i(B, D) + K_i(A, D)K_i(B, C),$$

where K_i denotes a kernel for individual protein and K_p the resulting kernel for pairs of proteins. The rationale behind this definition is that in order to match the pair (A, B) with the pair (C, D), one can either try to match A with C and B with D, or to match A with D and B with C. Reported accuracies on the problem of protein interaction prediction are very high, confirming the potential of kernel methods in this fast-moving field.

A parallel approach to network inference from genomic data has been investigated by Yamanishi et al. (2004), who show that learning the edges of a network can be carried out by first mapping the vertices, e.g., the genes, onto a Euclidean space, and then connecting the pairs of points which are close to each other in this embedding. The problem then becomes that of learning an optimal embedding of the vertices, a problem known as distance metric learning that recently caught the attention of the machine learning community and for which several kernel methods exist (Vert and Yamanishi, 2005).

Finally, several other emerging application in systems biology, such as inference on networks (Tsuda and Noble, 2004) or classification of networks (Middendorf et al., 2004), are likely to be subject to increasing attention in the future, due to the growing interest and amount of data related to biological networks.

5 CONCLUSION

This brief survey, although far from being complete, highlights the impressive advances in the applications of kernel methods in computational biology in the last 5 years. More than a just importing well-established algorithms to a new application domain, biology has triggered the development of new algorithms and methods, ranging from the engineering of various kernels to the development of new methods for learning from multiple kernels or for feature selection. The widespread diffusion of easy-to-use SVM softwares, and the ongoing integration of various kernels and kernel methods in major computing environments for bioinformatics, are likely to foster again the use of kernel methods in computational biology, as long as they will provide state-of-the-art methods for practical problems. Many questions remain open, regarding for example the automatic choice and integration of kernels, the possibility to incorporate prior knowledge in kernel methods, and the extension of kernel methods to more general kernels that positive definite, suggesting that theoretical developments are also likely to progress quickly in the near future.

References

Aebersold, R. and Mann, M. (2003). Mass spectrometry-based proteomics. *Nature*, 422(6928):198–207.

Aliferis, C., Hardin, D., and Massion, P. (2002). Machine learning models for lung cancer classification using array comparative genomic hybridization. In *Proceedings of the 2002 American Medical Informatics Association (AMIA) Annual Symposium*, pages 7–11.

Ambroise, C. and McLachlan, G. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Natl. Acad. Sci. USA*, 99(10):6562–6566.

Bao, L. and Sun, Z. (2002). Identifying genes related to drug anticancer mechanisms using support vector machine. *FEBS Lett.*, 521:109–114.

Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M., and Yakhini, Z. (2000). Tissue classification with gene expression profiles. *J. Comput. Biol.*, 7(3-4):559–583.

Ben-Hur, A. and Brutlag, D. (2003). Remote homology detection: a motif based approach. *Bioinformatics*, 19(Suppl. 1):i26–i33.

Ben-Hur, A. and Noble, W. S. (2005). Kernel methods for predicting protein-protein interactions. *Bioinformatics*, 21(Suppl. 1):i38–i46.

Bhasin, M. and Raghava, G. P. S. (2004). Prediction of CTL epitopes using QM, SVM and ANN techniques. *Vaccine*, 22(23-24):3195–3204.

Bock, J. R. and Gough, D. A. (2001). Predicting protein-protein interactions from primary structure. *Bioinformatics*, 17(5):455–460.

Bock, J. R. and Gough, D. A. (2002). A new method to estimate ligand-receptor energetics. *Mol Cell Proteomics*, 1(11):904–910.

Borgwardt, K. M., Ong, C. S., Schnauer, S., Vishwanathan, S. V. N., Smola, A. J., and Kriegel, H.-P. (2005). Protein function prediction via graph kernels. *Bioinformatics*, 21(Suppl. 1):i47–i56.

Cai, C., Wang, W., Sun, L., and Chen, Y. (2003). Protein function classification via support vector machine approach. *Math. Biosci.*, 185(2):111–122.

Camps-Valls, G., Chalk, A., Serrano-Lopez, A., Martin-Guerrero, J., and Sonnhammer, E. (2004). Profiled support vector machines for antisense oligonucleotide efficacy prediction. *BMC Bioinformatics*, 5(135):135.

Chen, Y., Lin, Y., Lin, C., and Hwang, J. (2004). Prediction of the bonding states of cysteines using the support vector machines based on multiple feature vectors and cysteine state sequences. *Proteins*, 55(4):1036–1042.

Consortium, I. H. G. S. (2001). Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921.

Cuturi, M. and Vert, J.-P. (2005). The context-tree kernel for strings. *Neural Network.*

Degroeve, S., Saeys, Y., De Baets, B., Rouze, P., and Van de Peer, Y. (2005). SpliceMachine: predicting splice sites from high-dimensional local context representations. *Bioinformatics*, 21:1332–1338.

DeRisi, J. L., Iyer, V. R., and Brown, P. O. (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278(5338):680–686.

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

Dobson, P. and Doig, A. (2003). Distinguishing enzyme structures from non-enzymes without alignments. *J. Mol. Biol.*, 330(4):771–783.

Dönnes, P. and Elofsson, A. (2002). Prediction of MHC class I binding peptides, using SVMHC. *BMC Bioinformatics*, 3(1):25.

Dror, G., Sorek, R., and Shamir, R. (2005). Accurate identification of alternatively spliced exons using support vector machine. *Bioinformatics*, 21(7):897–901.

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press.

Friedel, C. C., Jahn, K. H. V., Sommer, S., Rudd, S., Mewes, H. W., and Tetko, I. V. (2005). Support vector machines for separation of mixed plant-pathogen EST collections based on codon usage. *Bioinformatics*, 21:1383–1388.

Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., and Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914.

Guermeur, Y., Lifschitz, A., and Vert, R. (2004). A kernel for protein secondary structure prediction. In Schölkopf, B., Tsuda, K., and Vert, J., editors, *Kernel Methods in Computational Biology*, pages 193–206. MIT Press.

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

Haussler, D. (1999). Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, UC Santa Cruz.

Hua, S. and Sun, Z. (2001a). A novel method of protein secondary structure prediction with high segment overlap measure: Support vector machine approach. *J. Mol. Biol.*, 308(2):397–407.

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

Huang, T. M. and Kecman, V. (2005). Gene extraction for cancer diagnosis by support vector machines-An improvement. *Artif. Intell. Med.*

Jaakkola, T., Diekhans, M., and Haussler, D. (2000). A discriminative framework for detecting remote protein homologies. *J. Comput. Biol.*, 7(1,2):95–114.

Jaakkola, T. S., Diekhans, M., and Haussler, D. (1999). 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. AAAI Press.

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

Karklin, Y., Meraz, R. F., and Holbrook, S. R. (2005). Classification of non-coding RNA using graph representations of secondary structure. *Pac. Symp. Biocomput.*, pages 4–15.

Kashima, H., Tsuda, K., and Inokuchi, A. (2004). Kernels for graphs. In Schölkopf, B., Tsuda, K., and Vert, J., editors, *Kernel Methods in Computational Biology*, pages 155–170. MIT Press.

Kim, J. H., Lee, J., Oh, B., Kimm, K., and Koh, I. (2004). Prediction of phosphorylation sites using SVMs. *Bioinformatics*, 20(17):3179–3184.

Kin, T., Tsuda, K., and Asai, K. (2002). Marginalized kernels for RNA sequence data analysis. In Lathtop, R., Nakai, K., Miyano, S., Takagi, T., and Kanehisa, M., editors, *Genome Informatics 2002*, pages 112–122. Universal Academic Press.

Komura, D., Nakamura, H., Tsutsumi, S., Aburatani, H., and Ihara, S. (2005). Multidimensional support vector machines for visualization of gene expression data. *Bioinformatics*, 21(4):439–444.

Kondor, R. and Vert, J.-P. (2004). Diffusion kernels. In Schölkopf, B., Tsuda, K., and Vert, J., editors, *Kernel Methods in Computational Biology*, pages 171–192. MIT Press.

Krishnapuram, B., Carin, L., and Hartemink, A. (2004). Joint classifier and feature optimization for comprehensive cancer diagnosis using gene expression data. *J. Comput. Biol.*, 11(2-3):227–242.

Lanckriet, G. R. G., De Bie, T., Cristianini, N., Jordan, M. I., and Noble, W. S. (2004). A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635.

Leslie, C., Eskin, E., and Noble, W. (2002). The spectrum kernel: a string kernel for svm protein classification. In Altman, R. B., Dunker, A. K., Hunter, L., Lauerdale, K., and Klein, T. E., editors, *Proceedings of the Pacific Symposium on Biocomputing 2002*, pages 564–575. World Scientific.

Leslie, C. S., Eskin, E., Cohen, A., Weston, J., and Noble, W. S. (2004). Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–476.

Li, T., Zhang, C., and Ogihara, M. (2004). A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20(15):2429–2437.

Liao, L. and Noble, W. (2003). Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *J. Comput. Biol.*, 10(6):857–868.

Lin, K., Kuang, Y., Joseph, J. S., and Kolatkar, P. R. (2002). Conserved codon composition of ribosomal protein coding genes in Escherichia coli, Mycobacterium tuberculosis and Saccharomyces cerevisiae: lessons from supervised machine learning in functional genomics. *Nucl. Acids Res.*, 30(11):2599–2607.

Listgarten, J., Damaraju, S., Poulin, B., Cook, L., Dufour, J., Driga, A., Mackey, J., Wishart, D., Greiner, R., and Zanke, B. (2004). Predictive models for breast cancer susceptibility from multiple single nucleotide polymorphisms. *Clin. Cancer Res.*, 10(8):2725–2737.

Logan, B., Moreno, P., Suzek, B., Weng, Z., and Kasif, S. (2001). A study of remote homology detection. Technical Report CRL 2001/05, Compaq Cambridge Research laboratory.

Mahé, P., Ueda, N., Akutsu, T., Perret, J.-L., and Vert, J.-P. (2005). Graph kernels for molecular structure-activity relationship analysis with support vector machines. *J. Chem. Inf. Model.*, 45(4):939–51.

Martin, S., Roe, D., and Faulon, J.-L. (2005). Predicting protein-protein interactions using signature products. *Bioinformatics*, 21(2):218–226.

Matsuda, A., Vert, J.-P., Saigo, H., Ueda, N., Toh, H., and Akutsu, T. (2005). A novel representation of protein sequences for prediction of subcellular location using support vector machines. *Protein Sci.*, 14(11).

Meinicke, P., Tech, M., Morgenstern, B., and Merkl, R. (2004). Oligo kernels for datamining on biological sequences: a case study on prokaryotic translation initiation sites. *BMC Bioinformatics*, 5(169).

Meireles, S., Carvalho, A., Hirata, R., Montagnini, A., Martins, W., Runza, F., Stolf, B., Termini, L., Neto, C., Silva, R., Soares, F., Neves, E., and Reis, L.

(2003). Differentially expressed genes in gastric tumors identified by cDNA array. *Cancer Lett.*, 190(2):199–211.

Middendorf, M., Ziv, E., Adams, C., Hom, J., Koytcheff, R., Levovitz, C., Woods, G., Chen, L., and Wiggins, C. (2004). Discriminative topological features reveal biological network mechanisms. *BMC Bioinformatics*, 5(181).

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

Moler, E. J., Chow, M. L., and Mian, I. S. (2000). Analysis of molecular profile data using generative and discriminative methods. *Physiol. Genomics*, 4(2):109–126.

Mukherjee, S., Tamayo, P., Mesirov, J. P., Slonim, D., Verri, A., and Poggio, T. (1998). Support vector machine classification of microarray data. Technical Report 182, C.B.L.C. A.I. Memo 1677.

O'Donnell, R. K., Kupferman, M., Wei, S. J., Singhal, S., Weber, R., O'Malley, B., Cheng, Y., Putt, M., Feldman, M., Ziober, B., and Muschel, R. J. (2005). Gene expression signature predicts lymphatic metastasis in squamous cell carcinoma of the oral cavity. *Oncogene*, 24(7):1244–51.

O'Flanagan, R. A., Paillard, G., Lavery, R., and Sengupta, A. M. (2005). Non-additivity in protein-DNA binding. *Bioinformatics*, 21(10):2254–63.

Park, K.-J. and Kanehisa, M. (2003). Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics*, 19(13):1656–1663.

Passerini, A. and Frasconi, P. (2004). Learning to discriminate between ligand-bound and disulfide-bound cysteines. *Protein Eng. Des. Sel.*, 17(4):367–373.

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

Qin, J., Lewis, D. P., and Noble, W. S. (2003). Kernel hierarchical gene clustering from microarray expression data. *Bioinformatics*, 19(16):2097–2104.

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J., Poggio, T., Gerald, W., Loda, M., Lander, E., and Golub, T. (2001). Multiclass cancer diagnosis using tumor gene expression signatures. *Proc. Natl. Acad. Sci. USA*, 98(26):15149–15154.

Rätsch, G., Sonnenburg, S., and Schölkopf, B. (2005). RASE: recognition of alternatively spliced exons in C.elegans. *Bioinformatics*, 21(Suppl. 1):i369–i377.

Res, I., Mihalek, I., and Lichtarge, O. (2005). An evolution based classifier for prediction of protein interfaces without using protein structures. *Bioinformatics*, 21(10):2496–501.

Rose, J. R., Turkett, W. H., J., Oroian, I. C., Laegreid, W. W., and Keele, J. (2005). Correlation of amino acid preference and mammalian viral genome type. *Bioinformatics*.

Schena, M., Shalon, D., Davis, R., and Brown, P. (1995). Quantitative monitoring of gene expression patterns with a complimentary DNA microarray. *Science*, 270:467–470.

Schölkopf, B., Tsuda, K., and Vert, J.-P. (2004). *Kernel Methods in Computational Biology*. MIT Press.

Seeger, M. (2002). Covariance kernels from bayesian generative models. In *Adv. Neural Inform. Process. Syst.*, volume 14, pages 905–912.

Segal, N. H., Pavlidis, P., Antonescu, C. R., Maki, R. G., Noble, W. S., DeSantis, D., Woodruff, J. M., Lewis, J. J., Brennan, M. F., Houghton, A. N., and Cordon-Cardo, C. (2003a). Classification and subtype prediction of adult soft tissue sarcoma by functional genomics. *Am. J. Pathol.*, 163(2):691–700.

Segal, N. H., Pavlidis, P., Noble, W. S., Antonescu, C. R., Viale, A., Wesley, U. V., Busam, K., Gallardo, H., DeSantis, D., Brennan, M. F., Cordon-Cardo, C., Wolchok, J. D., and Houghton, A. N. (2003b). Classification of clear-cell sarcoma as a subtype of melanoma by genomic profiling. *J. Clin. Oncol.*, 21(9):1775–1781.

Sharan, R. and Myers, E. W. (2005). A motif-based framework for recognizing sequence families. *Bioinformatics*, 21 Suppl 1:i387–i393.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Shipp, M. A., Ross, K. N., Tamayo, P., Weng, A. P., Kutok, J. L., Aguiar, R. C. T., Gaasenbeek, M., Angelo, M., Reich, M., Pinkus, G. A., Ray, T. S., Koval, M. A., Last, K. W., Norton, A., Lister, T. A., Mesirov, J., Neuberg, D. S., Lander, E. S., Aster, J. C., and Golub, T. R. (2002). Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nat. Med.*, 8(1):68–74.

Smith, T. and Waterman, M. (1981). Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197.

Steiner, G., Suter, L., Boess, F., Gasser, R., de Vera, M. C., Albertini, S., and Ruepp, S. (2004). Discriminating different classes of toxicants by transcript profiling. *Environ. Health Perspect.*, 112(12):1236–48.

Su, Y., Murali, T., Pavlovic, V., Schaffer, M., and Kasif, S. (2003). RankGene: identification of diagnostic genes based on expression data. *Bioinformatics*, 19(12):1578–1579.

Swamidass, S. J., Chen, J., Bruand, J., Phung, P., Ralaivola, L., and Baldi, P. (2005). Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity. *Bioinformatics*, 21(Suppl. 1):i359–i368.

Teramoto, R., Aoki, M., Kimura, T., and Kanaoka, M. (2005). Prediction of siRNA functionality using generalized string kernel and support vector machine. *FEBS Lett.*, 579(13):2878–82.

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

Tsuda, K. and Noble, W. (2004). Learning kernels from biological networks by maximizing entropy. *Bioinformatics*, 20:i326–i333.

Valentini, G. (2002). Gene expression data analysis of human lymphoma using support vector machines and output coding ensembles. *Artif. Intell. Med.*, 26(3):281–304.

Venter, J. C. e. a. (2001). The sequence of the human genome. *Science*, 291(5507):1304–1351.

Vert, J.-P. (2002). A tree kernel to analyze phylogenetic profiles. *Bioinformatics*, 18:S276–S284.

Vert, J.-P. and Kanehisa, M. (2003). Extracting active pathways from gene expression data. *Bioinformatics*, 19:238ii–234ii.

Vert, J.-P., Saigo, H., and Akutsu, T. (2004). Local alignment kernels for biological sequences. In Schölkopf, B., Tsuda, K., and Vert, J., editors, *Kernel Methods in Computational Biology*, pages 131–154. MIT Press.

Vert, J.-P., Thurman, R., and Noble, W. S. (2006). Kernels for gene regulatory regions. In *Adv. Neural. Inform. Process Syst.*

Vert, J.-P. and Yamanishi, Y. (2005). Supervised graph inference. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Adv. Neural Inform. Process. Syst.*, volume 17, pages 1433–1440. MIT Press, Cambridge, MA.

Wagner, M., Naik, D., and Pothen, A. (2003). Protocols for disease classification from mass spectrometry data. *Proteomics*, 3(9):1692–1698.

Wang, M., Yang, J., Liu, G.-P., Xu, Z.-J., and Chou, K.-C. (2004). Weighted-support vector machines for predicting membrane protein types based on pseudo-amino acid composition. *Protein Eng. Des. Sel.*, 17(6):509–516.

Williams, R., Hing, S., Greer, B., Whiteford, C., Wei, J., Natrajan, R., Kelsey, A., Rogers, S., Campbell, C., Pritchard-Jones, K., and Khan, J. (2004). Prognostic classification of relapsing favorable histology Wilms tumor using cDNA microarray expression profiling and support vector machines. *Genes Chromosomes Cancer*, 41(1):65–79.

Wu, B., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., Ward, D., Williams, K., and Zhao, H. (2003). Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. *Bioinformatics*, 19(13):1636–1643.

Yamanishi, Y., Vert, J.-P., and Kanehisa, M. (2004). Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, 20:i363–i370.

Yan, C., Dobbs, D., and Honavar, V. (2004). A two-stage classifier for identification of protein-protein interface residues. *Bioinformatics*, 20(Suppl. 1):i371–i378.

Yoon, Y., Song, J., Hong, S., and Kim, J. (2003). Analysis of multiple single nucleotide polymorphisms of candidate genes related to coronary heart disease susceptibility by using support vector machines. *Clin. Chem. Lab. Med.*, 41(4):529–534.

Yu, C., Zavaljevski, N., Stevens, F. J., Yackovich, K., and Reifman, J. (2005). Classifying noisy protein sequence data: a case study of immunoglobulin light chains. *Bioinformatics*, 21(Supp 1):i495–i501.

Yuan, Z., Burrage, K., and Mattick, J. (2002). Prediction of protein solvent accessibility using support vector machines. *Proteins*, 48(3):566–570.

Zavaljevski, N., Stevens, F., and Reifman, J. (2002). Support vector machines with selective kernel scaling for protein classification and identification of key amino acid positions. *Bioinformatics*, 18(5):689–696.

Zhang, S.-W., Pan, Q., Zhang, H.-C., Zhang, Y.-L., and Wang, H.-Y. (2003a). Classification of protein quaternary structure with support vector machine. *Bioinformatics*, 19(18):2390–2396.

Zhang, X. H.-F., Heller, K. A., Hefter, I., Leslie, C. S., and Chasin, L. A. (2003b). Sequence information for the splicing of human pre-mRNA identified by support vector machine classification. *Genome Res.*, 13(12):2637–2650.

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