BIOINFORMATICS

Vol. 00 no. 00 2007
Pages 1–8

Gene Selection via the BAHSIC Family of Algorithms

Le Song[1,2], Justin Bedo[1],
Karsten M. Borgwardt[3], Arthur Gretton[4], Alex Smola[1]

1 National ICT Australia, 2 University of Sydney, Australia, 3 Department "Institute for Informatics", Ludwig-Maximilians-University, Munich, Germany, 4 Max Planck Institute for Biological Cybernetics, Tübingen, Germany

ABSTRACT

Motivation Identifying significant genes among thousands of sequences on a microarray is a central challenge for cancer research in bioinformatics. The ultimate goal is to detect the genes that are involved in disease outbreak and progression. A multitude of methods have been proposed for this task of feature selection, yet the selected gene lists differ greatly between different methods. To accomplish biologically meaningful gene selection from microarray data, we have to understand the theoretical connections and the differences between these methods. In this article, we define a kernel-based framework for feature selection based on the Hilbert-Schmidt Independence Criterion and backward elimination, called BAHSIC. We show that several well-known feature selectors are instances of BAHSIC, thereby clarifying their relationship. Furthermore, by choosing a different kernel, BAHSIC allows us to easily define novel feature selection algorithms. As a further advantage, feature selection via BAHSIC works directly on multiclass problems.

Results In a broad experimental evaluation, the members of the BAHSIC family reach high levels of accuracy and robustness when compared to other feature selection techniques. Experiments show that features selected with a linear kernel provide the best classification performance in general, but if strong non-linearities are present in the data then nonlinear kernels can be more suitable.

Availability: Accompanying homepage is
http://www.dbs.ifi.lmu.de/∼borgward/BAHSIC

Contact: kb@dbs.ifi.lmu.de

1 INTRODUCTION

Gene selection from microarray data is clearly one of the most popular topics in bioinformatics. To illustrate this, the database for "Bibliography on Microarray Data Analysis" (**?**) has grown from less than 100 articles in 2000 to 1690 papers in January 2007. What are the reasons for this huge interest in feature selection?

There are two main reasons for this popularity, the first biological, the second statistically motivated. First, by selecting genes from a microarray that result in good separation between healthy and diseased patients, one hopes to find the significant genes affected by the disease, or even causing it. This is a central step towards understanding the underlying biological process.

Second, classifiers on microarray data tend to overfit due to the low number of patients and the high number of observed genes. This means that they achieve high accuracy levels on the training data, but do not generalise to new data. The underlying problem is that if sample size is much smaller than the number of genes, one can distinguish different classes of patients based on the noise present in these measurements, rather than on distinct biological characteristics of their gene expression levels. Via feature selection, one aims to reduce the number of genes by removing meaningless features.

Although feature selection on microarrays is popular, gene selection methods suffer from several problems. First of all, they lack robustness. In **?**, prognostic cancer gene lists selected from microarrays differ significantly between different methods, and even for different subsets of the same microarray datasets. The authors conclude that thousands of samples are needed for robust gene selection. Given that clinical studies almost exclusively deal with comparatively low sample sizes, this is a very pessimistic view of clinical microarray data analysis. At the other end of the spectrum are recent results of sparse decoding (**??**), which suggest that for a very well defined family of inverse problems, asymptotically only $n(1 + \log d)$ observations are needed to recover n features accurately from d dimensions.

Besides small sample size and high dimensionality, another crucial problem arises from the plethora of feature selection methods for microarray data. Each approach is endowed with its own theoretical analysis, and the connections between them are so far poorly understood (**?**). This makes it difficult to explain why different algorithms generate different prognostic gene lists on the same set of cancer microarray data. A unifying framework for feature selection algorithms would help to understand these relations and to clarify which feature selection algorithms are most helpful for gene selection.

In this paper, we present such a unifying framework called BAHSIC. BAHSIC defines a class of backward (BA) elimination feature selection algorithms that make use of I) kernels and II) the Hilbert-Schmidt Independence Criterion (HSIC) (**?**). We show that BAHSIC includes several well-known feature selection methods, namely Pearson's correlation coefficient (**??**), t-test (**?**), signal-to-noise ratio (**?**), Centroid (**??**), Shrunken Centroid (**??**) and ridge regression (**?**).

By choosing different kernels, one may define new types of feature selection algorithm. We show that several well-known feature selection methods merely differ in their choice of kernel. Furthermore, BAHSIC can be extended in a principled fashion to multiclass and regression problems, in contrast to most competing methods which are exclusively geared towards two-class problems.

In a broad experimental evaluation, we compare feature selection methods that are instances of BAHSIC to several competing approaches, with respect to both the robustness of the selected features and the resulting classification accuracy. Our unified framework assists

us in explaining how the kernel used by a particular feature selector determines which genes are preferred. Our experiments show that features selected with a linear kernel provide the best classification performance in general, but if strong non-linearities are present in the gene expression data then nonlinear kernels can be more suitable.

2 FEATURE SELECTION AND BAHSIC

The problem of feature selection can be cast as a combinatorial optimisation problem. We denote by $\mathcal{S}$ the full set of features, which in our case corresponds to expression levels of various genes. We use these features to predict a particular outcome, for instance the presence of cancer: clearly, only a subset $\mathcal{T}$ of features will be relevant. Suppose the relevance of a feature subset to the outcome is quantified by $\mathcal{Q}(\mathcal{T})$ and it is computed by restricting the data to the dimensions in $\mathcal{T}$. Feature selection can then be formulated as:

$$\mathcal{T}_0 = \arg\max_{\mathcal{T}\subseteq\mathcal{S}} \mathcal{Q}(\mathcal{T}) \qquad \text{s.t.} \quad |\mathcal{T}| \leq t \tag{1}$$

where $|\cdot|$ computes the cardinality of a set and t upper bounds the number of selected features. Two important aspects of problem (1) are the choice of the criterion $\mathcal{Q}(\mathcal{T})$ and the selection algorithm. We therefore begin with a description of our criterion, and later introduce the feature selection algorithm based on this criterion.

To describe our feature selection criterion, we begin with the simple example of linear dependence detection, which we then generalise to the detection of more general kinds of dependence. Consider spaces $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} \subset \mathbb{R}^l$, on which we jointly sample observations (x, y) from a distribution $\Pr_{xy}$. We may define a covariance matrix

$$\mathcal{C}_{xy} = \mathsf{E}_{xy}(xy^\top) - \mathsf{E}_x(x)\mathsf{E}_y(y^\top), \tag{2}$$

which contains all second order dependence between the random variables. A statistic that efficiently summarises the content of this matrix is its Hilbert-Schmidt norm: denote by σ_i the singular values of $\mathcal{C}_{xy}$, then the square of this norm is

$$\|\mathcal{C}_{xy}\|^2_{\mathrm{HS}} := \sum_i \sigma_i^2.$$

This quantity is zero if and only if there exists no *second order dependence* between x and y. This statistic is limited in several respects, however, of which we mention two: first, dependence can exist in forms other than that detectable via covariance (and even when a second order relation exists, the full extent of the dependence between x and y may only be apparent when nonlinear effects are included). Second, the restriction to subsets of $\mathbb{R}^d$ excludes many interesting kinds of variables, such as strings and class labels. We wish therefore to generalise the notion of covariance to nonlinear relationships, and to a wider range of data types.

We now define $\mathcal{X}$ and $\mathcal{Y}$ more broadly as two domains from which we draw samples (x, y) as before: these may be real valued, vector valued, class labels, strings (**?**), graphs (**?**), and so on (see **?** for further examples in bioinformatics). We define a (possibly nonlinear) mapping $\phi(x) \in \mathcal{F}$ from each $x \in \mathcal{X}$ to a feature space $\mathcal{F}$, such that the inner product between the features is given by a kernel function $k(x, x') := \langle \phi(x), \phi(x') \rangle$: $\mathcal{F}$ is called a reproducing kernel Hilbert space (RKHS).[1] Likewise, let $\mathcal{G}$ be a second RKHS on $\mathcal{Y}$ with kernel $l(\cdot, \cdot)$ and feature map $\psi(y)$. We may now define a cross-covariance operator between these feature maps, which is analogous to the covariance matrix in (2): this is a linear operator $\mathcal{C}_{xy} : \mathcal{G} \longmapsto \mathcal{F}$ such that

$$\mathcal{C}_{xy} = \mathsf{E}_{xy}[(\phi(x) - \mu_x) \otimes (\psi(y) - \mu_y)], \tag{3}$$

where $\otimes$ is the tensor product (see **??** for more detail). The square of the Hilbert-Schmidt norm of the cross-covariance operator (HSIC), $\|\mathcal{C}_{xy}\|^2_{\mathrm{HS}}$, is then used as our feature selection criterion $\mathcal{Q}(\mathcal{T})$. HSIC was shown in **?** to be expressible in terms of kernels as

$$\begin{aligned} &\mathrm{HSIC}(\mathcal{F}, \mathcal{G}, \Pr_{xy}) = \|\mathcal{C}_{xy}\|^2_{\mathrm{HS}} \\ &= \mathsf{E}_{xx'yy'}[k(x, x')l(y, y')] + +\mathsf{E}_{xx'}[k(x, x')]\mathsf{E}_{yy'}[l(y, y')] \\ &- 2\mathsf{E}_{xy}[\mathsf{E}_{x'}[k(x, x')]\mathsf{E}_{y'}[l(y, y')]]. \end{aligned} \tag{4}$$

Given a sample $Z = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ of size m drawn from $\Pr_{xy}$, an empirical estimator of HSIC was shown in **?** to be

$$\mathrm{HSIC}(\mathcal{F}, \mathcal{G}, Z) = (m-1)^{-2}\mathsf{Tr}(\mathbf{KHLH}), \tag{5}$$

where $\mathbf{K}, \mathbf{L} \in \mathbb{R}^{m\times m}$ are the kernel matrices for the data and the labels respectively, and $\mathbf{H}_{ij} = \delta_{ij} - m^{-1}$ centres the data and the label features. See **?** for a different interpretation of a related criterion used in independence testing.

We now describe two theorems from **?** which support our using HSIC as a feature selection criterion. The first (**?**, Theorem 3) shows that the empirical HSIC converges in probability to its population counterpart with rate $1/\sqrt{m}$. This implies that if the empirical HSIC is large, then given sufficient samples it is very probable that the population HSIC is also large; likewise, a small empirical HSIC likely corresponds to a small population HSIC. Moreover, the same features should consistently be selected to achieve high dependence if the data is repeatedly drawn from the same distribution. The second result (**?**, Theorem 4) states that when $\mathcal{F}, \mathcal{G}$ RKHSs with universal (**?**) kernels k, l on respective compact domains $\mathcal{X}$ and $\mathcal{Y}$, then $\mathrm{HSIC}(\mathcal{F}, \mathcal{G}, \Pr_{xy}) = 0$ if and only if x and y are independent. In terms of our microarray setting, using a universal kernel such as the Gaussian RBF kernel or the Laplace kernel, HSIC is zero if gene expression levels and class labels are independent; clearly we want to reach the opposite result, namely strong dependence between expression levels and class labels. Hence we try to select genes that maximise HSIC.

BAHSIC Having defined our feature selection criterion, we now describe an algorithm that conducts feature selection on the basis of this dependence measure. Using HSIC, we can perform both forward and backward selection of the features. In particular, when we use a linear kernel on both the data and labels, forward selection and backward selection are equivalent: the objective function decomposes into individual coordinates, and thus feature selection can be done without recursion in one go.

[1] A note on the nonlinear mapping: if $\mathcal{X} = \mathbb{R}^d$, then this could be as simple as a set of polynomials of order up to t in the components of x, with kernel $k(x, x') = (\langle x, x' \rangle + c)^t$. Other kernels, like the Gaussian, correspond to infinitely large feature spaces. We need never evaluate these feature representations explicitly, however.

In the case of more general kernels, forward selection is computationally more efficient, however backward elimination in general yields better features, since the quality of the features is assessed within the context of all other features. Hence we present the backward elimination (BA) version of our algorithm here.

Our feature selection algorithm (BAHSIC) appends the features from $\mathcal{S}$ to the end of a list $\mathcal{S}^\dagger$ so that the elements towards the end of $\mathcal{S}^\dagger$ have higher relevance to the learning task. The feature selection problem in (1) can be solved by simply taking the last t elements from $\mathcal{S}^\dagger$. Our algorithm produces $\mathcal{S}^\dagger$ recursively, eliminating the least relevant features from $\mathcal{S}$ and adding them to the end of $\mathcal{S}^\dagger$ at each iteration.

Algorithm 1 Feature Selection via Backward Elimination

Input: The full set of features $\mathcal{S}$
Output: An ordered set of features $\mathcal{S}^\dagger$

1: $\mathcal{S}^\dagger \leftarrow \varnothing$
2: **repeat**
3: $\quad \sigma_0 \leftarrow \arg\max_\sigma \ \mathrm{HSIC}(\sigma, \mathcal{S}),\ \ \sigma \in \Xi$
4: $\quad i \leftarrow \arg\max_i \ \mathrm{HSIC}(\sigma_0, \mathcal{S} \setminus \{i\}),\ \ i \in \mathcal{S}$
5: $\quad \mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$
6: $\quad \mathcal{S}^\dagger \leftarrow \mathcal{S}^\dagger \cup \{i\}$
7: **until** $\mathcal{S} = \varnothing$

Step 3 of the algorithm optimises over all possible choices of kernel parameters in the set Ξ. Note that Ξ is chosen such that the kernels are bounded. If we have no prior knowledge regarding the nature of the nonlinearity in the data, then optimising over Ξ is essential: it allows us to adapt to the scale of the nonlinearity present in the (feature-reduced) data. If we have prior knowledge about the type of nonlinearity, we can use a kernel with fixed parameters for BAHSIC. In this case, step 3 can be omitted since there will be no parameter to tune. For faster elimination of features, we can choose a group of features at step 4 and delete them in one shot at step 5.

3 FEATURE SELECTORS THAT ARE INSTANCES OF BAHSIC

In this section we will show that several feature selection criteria are special cases of BAHSIC, and thus BAHSIC is capable of finding and exploiting dependence of a much more general nature (for instance, dependence between data and labels with graph and string values).

We first define the symbols used in the following sections. Let $\mathbf{X}$ be the full data matrix with each row a sample and each column a feature, $\mathbf{x}$ be a column of $\mathbf{X}$, and x_i be the entries in $\mathbf{x}$. Let $\mathbf{y}$ be the vector of labels with entries y_i. When the labels are multidimensional, we express them as a matrix $\mathbf{Y}$, with each row a datum and each column a dimension. The kth column of $\mathbf{Y}$ is then $\mathbf{Y}(k)$.

Suppose the number of data points is m. We denote the mean of a particular feature of the data as $\bar{x}$, and its standard deviation as s_x. For two-class data, let the number of the positive and negative samples be m_+ and m_-, respectively ($m = m_+ + m_-$). In this case, denote the mean of the samples from the positive and the negative classes by $\bar{x}_+$ and $\bar{x}_-$, respectively, and the corresponding standard deviations by s_{x+} and s_{x-}. For multiclass data, we let m_i be the number of samples in class i, where $i \in \mathbb{N}^*$ and $m = \sum_i m_i$. Finally, let $\mathbf{1}_k$ be a column vector of all ones with length k and $\mathbf{0}_k$ be a column vector of all zeros.

3.1 Pearson's correlation

Pearson's correlation is commonly used in microarray analysis (**??**), and is defined as

$$r_{xy} = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{s_x s_y}, \tag{6}$$

for each column $\mathbf{x}$ of $\mathbf{X}$ (scores are computed separately for each feature). The link between HSIC and Pearson's correlation is straightforward: we first normalise the data and the labels by s_x and s_y, respectively, and apply a linear kernel in both domains. HSIC then becomes

$$\begin{aligned} &\mathsf{Tr}(\mathbf{KHLH}) = \mathsf{Tr}(\mathbf{x}\mathbf{x}^\top \mathbf{H}\mathbf{y}\mathbf{y}^\top \mathbf{H}) = ((\mathbf{H}\mathbf{x})^\top (\mathbf{H}\mathbf{y}))^2 \\ &= \left(\sum_{i=1}^m \left(\frac{x_i}{s_x} - \frac{\bar{x}}{s_x} \right) \left(\frac{y_i}{s_y} - \frac{\bar{y}}{s_y} \right) \right)^2 \\ &= \left(\frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{s_x s_y} \right)^2. \end{aligned} \tag{7}$$

The above equation is just the square of Pearson's correlation (pc). Using Pearson's correlation for feature selection is then equivalent to BAHSIC with the above normalisation and linear kernels.

3.2 Mean difference and its variants

The difference between the sample means of the positive and negative classes, $(\bar{x}_+ - \bar{x}_-)$, is useful for selecting discriminative features. With different normalisation of the data and labels, many variants can be derived. For example, the centroid (lin) (**?**), t-score (t) (**?**), moderated t-score (m-t), signal-to-noise ratio (snr), and B-statistics (lods) (**?**) all belong to this subfamily.

We will start by showing that $(\bar{x}_+ - \bar{x}_-)^2$ is a special case of HSIC. This is straightforward if we assign $\frac{1}{m_+}$ as the labels to the positive samples and $\frac{-1}{m_-}$ to the negative samples. Applying a linear kernel on both domains leads to the equivalence

$$\begin{aligned} &\mathsf{Tr}(\mathbf{KHLH}) = \mathsf{Tr}(\mathbf{x}\mathbf{x}^\top \mathbf{y}\mathbf{y}^\top) = (\mathbf{x}^\top \mathbf{y})^2 \\ &= \left(\frac{1}{m_+} \sum_{i=1}^{m_+} x_i - \frac{1}{m_-} \sum_{i=1}^{m_-} x_i \right)^2 = (\bar{x}_+ - \bar{x}_-)^2. \end{aligned} \tag{8}$$

Note that the centring matrix $\mathbf{H}$ disappears because the labels are already centred (i.e. $\mathbf{y}^\top \mathbf{1}_m = 0$, and thus $\mathbf{HLH} = \mathbf{L}$).

The t-test is defined as $t = \frac{\bar{x}_+ - \bar{x}_-}{\bar{s}}$, where $\bar{s} = \left(\frac{s_{x+}^2}{m_+} + \frac{s_{x-}^2}{m_-} \right)^{\frac{1}{2}}$. The square of the t-test is equivalent to HSIC if the data is normalised by $\left(\frac{s_{x+}^2}{m_+} + \frac{s_{x-}^2}{m_-} \right)^{\frac{1}{2}}$. The signal-to-noise ratio, moderated t-test, and B-statistics are three variants of the t-test. They differ only in their respective denominators, and are thus special cases of HSIC if we normalise the data accordingly. For example, we obtain the signal-to-noise ratio if the data are normalised by $(s_{x+} + s_{x-})$.

3.3 Shrunken centroid

The shrunken centroid (pam) method (**??**) performs feature ranking using the differences from the class centroids to the centroid of all

the data. This is also related to HSIC if specific preprocessing of the data and labels is performed. Here we will focus on constructing appropriate labels, as the normalisation of the data is similar to the previous section. For two-class problems, we use the 2-dimensional label matrix

$$\mathbf{Y} = \begin{pmatrix} \frac{\mathbf{1}_{m_+}}{m_+} - \frac{\mathbf{1}_{m_+}}{m}, & -\frac{\mathbf{1}_{m_+}}{m} \\ -\frac{\mathbf{1}_{m_-}}{m}, & \frac{\mathbf{1}_{m_-}}{m_-} - \frac{\mathbf{1}_{m_-}}{m} \end{pmatrix}_{m \times 2}. \tag{9}$$

The labels are centred (i.e. $\mathbf{Y}^\top \mathbf{1_m} = \mathbf{0}_2$), and thus

$$\begin{aligned} &\mathsf{Tr}(\mathbf{KHLH}) = \mathsf{Tr}(\mathbf{x}\mathbf{x}^\top \mathbf{Y}\mathbf{Y}^\top) \\ &= \mathbf{Y}(1)^\top \mathbf{x}\mathbf{x}^\top \mathbf{Y}(1) + \mathbf{Y}(2)^\top \mathbf{x}\mathbf{x}^\top \mathbf{Y}(2) \\ &= \left(\frac{1}{m_+} \sum_{i=1}^{m_+} x_i - \frac{1}{m} \sum_{i=1}^{m} x_i \right)^2 + \left(\frac{1}{m_-} \sum_{i=1}^{m_-} x_i - \frac{1}{m} \sum_{i=1}^{m} x_i \right)^2 \\ &= (\bar{x}_+ - \bar{x})^2 + (\bar{x}_- - \bar{x})^2. \end{aligned} \tag{10}$$

This is in essence the information used by the shrunken centroid method.

3.4 Multiclass

In addition to scoring features for two-class data, our method can readily be applied to multiclass data, by constructing an appropriate label space kernel using the class label assignments. For instance, we can score a feature for the multiclass classification problem by applying linear kernels to the following label feature vectors (3-class example):

$$\mathbf{Y} = \begin{pmatrix} \frac{\mathbf{1}_{m_1}}{m_1} & \frac{\mathbf{1}_{m_1}}{m_2 - m} & \frac{\mathbf{1}_{m_1}}{m_3 - m} \\ \frac{\mathbf{1}_{m_2}}{m_1 - m} & \frac{\mathbf{1}_{m_2}}{m_2} & \frac{\mathbf{1}_{m_2}}{m_3 - m} \\ \frac{\mathbf{1}_{m_3}}{m_1 - m} & \frac{\mathbf{1}_{m_3}}{m_2 - m} & \frac{\mathbf{1}_{m_3}}{m_3} \end{pmatrix} \quad \text{or} \tag{11}$$

$$\mathbf{Y} = \begin{pmatrix} \frac{\mathbf{1}_{m_1}}{\sqrt{m_1}} & \mathbf{0}_{m_1} & \mathbf{0}_{m_1} \\ \mathbf{0}_{m_2} & \frac{\mathbf{1}_{m_2}}{\sqrt{m_2}} & \mathbf{0}_{m_2} \\ \mathbf{0}_{m_3} & \mathbf{0}_{m_3} & \frac{\mathbf{1}_{m_3}}{\sqrt{m_3}} \end{pmatrix}. \tag{12}$$

The $\mathbf{Y}$ on the top is equivalent to one-versus-the-rest scoring of the features, while that on the bottom is geared towards selecting features that recover the block structure of the kernel matrix in the data space.

3.5 Regression

BAHSIC can also be used to select features for regression problems, except that in this case the labels are continuous variables. Again we can use different kernels on both the data and the labels and apply BAHSIC. In this context, feature selection using ridge regression can also be viewed as a special case of BAHSIC. In ridge regression (**?**), we predict the outputs $\mathbf{y}$ using the predictor $\mathbf{Vw}$ by minimising the objective function $R = (\mathbf{y} - \mathbf{Vw})^2 + \lambda \|\mathbf{w}\|^2$, where the second term is known as the regulariser. Our discussion encompasses two cases: first, the linear model, in which $\mathbf{V} = \mathbf{X}$; and second, the nonlinear case, in which each of the m rows of $\mathbf{V}$ is a vector of nonlinear features of a particular observation x_i, and $f(x_i) = \sum_j w_j v_j(x_i)$. Recursive feature elimination combined as an embedded method with ridge regression removes the feature which causes the smallest increase in R. Equivalently, after minimising R, this is the feature which has the smallest absolute weight $|w_i|$.

The minimum of this objective function with respect to $\mathbf{w}$ is

$$\begin{aligned} R^* &= \mathbf{y}^\top \mathbf{y} - \mathbf{y}^\top \mathbf{V}(\mathbf{V}^\top \mathbf{V} + \lambda \mathbf{I})^{-1} \mathbf{V}^\top \mathbf{y} \\ &= \mathbf{y}^\top \mathbf{y} - \mathsf{Tr}(\mathbf{V}(\mathbf{V}^\top \mathbf{V} + \lambda \mathbf{I})^{-1} \mathbf{V}^\top \mathbf{y}\mathbf{y}^\top). \end{aligned} \tag{13}$$

Therefore recursively removing the feature which minimises the increase in R^* is equivalent to maximising the HSIC, when using $\mathbf{K} = \mathbf{V}(\mathbf{V}^\top \mathbf{V} + \lambda \mathbf{I})^{-1} \mathbf{V}^\top$ as the kernel matrix on the data and the linear kernel on the labels.

The final case we consider is kernel ridge regression, which differs from the above in that the space of nonlinear features of the input may be infinite dimensional, and the regulariser becomes a smoothness constraint on the functions from this space to the output. Specifically, the inputs are mapped to a *different* feature space $\mathcal{H}$ with kernel $\hat{k}(x, x')$, in which a linear prediction is made of the label y. Without going into further detail, we use standard kernelisation methods (**?**) to obtain that the minimum objective is $R^* = \mathbf{y}^\top \mathbf{y} - \mathbf{y}^\top (\hat{\mathbf{K}} + \lambda \mathbf{I})^{-1} \hat{\mathbf{K}} \mathbf{y}$. This is equivalent to defining a feature space $\mathcal{F}$ with kernel $(\hat{\mathbf{K}} + \lambda \mathbf{I})^{-1} \hat{\mathbf{K}}$ on the data, and then selecting features by maximising HSIC.

4 ALGORITHMS UNRELATED TO BAHSIC

In addition to the feature selection algorithms that are related to BAHSIC, we compare against three methods that are not members of the BAHSIC family: mutual information (mi), recursive feature elimination SVM (rfe), and L1-SVM for feature selection (l1).

Mutual Information is a filter method for feature selection drawn from information theory. It computes the mutual information between each feature and the labels. The features that correspond to the highest mutual information are then selected. Variants of this method can consider several features at a time, but the resulting density estimation problem becomes much harder for increased dimensions. This method is applicable to both two-class and multiclass datasets.

Recursive feature elimination SVM (**?**) is an embedded method for feature selection. It aims to optimise the performance of a linear SVM by eliminating the least useful features for SVM classification in a backwards greedy fashion. Initially an SVM using all features is trained. The least important features, estimated by the absolute value of the trained weights, are then dropped from the model and the SVM retrained. The process is carried out recursively until the desired number of features is reached.

The L1-SVM (**?**) is also an embedded method for feature selection. Using an L1 norm as the regulariser in an SVM results in sparse weight vectors (see **?**), where the number of non-zero weights depends on the amount of regularisation. It is not easy to specify the exact sparsity of the solution, but in our experiments the typical number of features selected was below 50.

5 DATASETS

We ran our experiments on 28 datasets, of which 15 are two-class datasets and 13 are multiclass datasets. These datasets are assigned a reference number for convenience. Two-class datasets have a reference number less than or equal to 15, and multiclass datasets have

reference numbers of 16 and above. Only one dataset, yeast, has feature dimension less than 1000 (79 features). All other datasets have dimensions ranging from approximately 2000 to 25000. The number of samples varies between approximately 50 and 300 samples. A summary of the datasets and their sources is as follows:

- The six datasets studied in (**?**). Three deal with breast cancer (**???**) (numbered 1, 2 and 3), two with lung cancer (**??**) (4, 5), and one with hepatocellular carcinoma (**?**) (6). The B cell lymphoma dataset (**?**) is not used because none of the tested methods produce classification errors lower than 40%.
- The six datasets studied in (**?**). Two deal with prostate cancer (**??**) (7, 8), two with breast cancer (**??**) (9, 10), and two with leukaemia (**??**) (16, 17).
- Five commonly used bioinformatics benchmark datasets on colon cancer (**?**) (11), ovarian cancer (**?**) (12), leukaemia (**?**)(13), lymphoma (**?**)(18), and yeast (**?**)(19).
- Nine datasets from the NCBI GEO database. The GDS IDs and reference numbers for this paper are GDS1962 (20), GDS330 (21), GDS531 (14), GDS589 (22), GDS968 (23), GDS1021 (24), GDS1027 (25), GDS1244 (26), GDS1319 (27), GDS1454 (28), and GDS1490 (15), respectively.

6 EXPERIMENTS

6.1 Classification Error and Robustness of Genes

We used stratified 10-fold cross-validation and SVMs to evaluate the predictive performance of the top 10 features selected by each method. For two-class datasets, a nonlinear SVM with an RBF kernel, $k(x,x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)$, was used. The regularisation constant C and the kernel width σ were tuned on a grid of $\{0.1, 1, 10, 10^2, 10^3\} \times \{1, 10, 10^2, 10^3\}$. Classification performance is measured as the fraction of misclassified samples. For multiclass datasets, all procedures are the same except that we used the SVM in a one-versus-the-rest fashion. Two new BAHSIC methods are included in the comparison, with kernels $\exp\left(-\frac{\|x-x'\|}{2\sigma^2}\right)$ (RBF) and $\|x-x'\|^{-1}$ (dis) on the data.

The classification results for binary and multiclass datasets are reported in Table 1 and Table 2, respectively. In addition to error rate we also report the overlap between the top 10 gene lists created in each fold. The multiclass results are presented separately since some older members of the BAHSIC family, and some competitors, are not naturally extensible to multiclass datasets. From the experiments we make the following observations:

1. The BAHSIC family obtains the lowest classification error (not necessarily significant) in 12 out of 15 of the two-class datasets and all 13 of the multiclass datasets.
2. The BAHSIC family obtains the greatest overlap in all but one dataset. This suggests that genes selected by the BAHSIC family can be more stable.
3. The BAHSIC family with nonlinear kernels obtains the lowest classification error in 7 datasets and the greatest overlap in 7 datasets.

6.2 Performance of feature selectors across datasets

When comparing the overall performance of various gene selection algorithms, it is of primary interest to choose a method which works well *everywhere*, rather than one which sometimes works well and sometimes performs catastrophically. It turns out that the linear kernel (lin) outperforms all other methods in this regard, both for binary and multiclass problems.

To show this, we measure how the various methods compare with the best performing one in each dataset in Tables 1 and 2. The deviation between algorithms is taken as the square of the difference in performance. This measure is chosen because gene expression data is relative expensive to obtain, and we want an algorithm to select the best genes from them. If an algorithm selects genes that are far inferior to the best possible among all algorithms (catastrophic case), we downgrade the algorithm more heavily. Squaring the performance difference achieves exactly this effect, by penalising larger differences more heavily. In other words, we want to choose an algorithm that performs homogeneously well in all datasets. To provide a concise summary, we add these deviations over the datasets and take the square root as the measure of goodness. These scores (called ℓ_2 distance) are listed in Tables 1 and 2. In general, the smaller the ℓ_2 distance, the better the method. It can been seen that the linear kernel has the smallest ℓ_2 distance on both the binary and multiclass datasets.

6.3 Impact of Kernel on Gene Selection

In Section 3, we unified several feature selection algorithms in one common framework. In our feature selection evaluation experiment, we showed the linear kernel selects the genes leading to the best classification accuracies on average. From a biological perspective, the interesting questions to ask are: Why does the linear kernel select the best genes on average? Why are there datasets on which it does not perform best? Finally, which genes are selected by a linear kernel based feature selector, and which by a Gaussian kernel based selector? In this section, we conduct experimental analyses to come up with answers to these questions. These findings have deep implications, because they help us to understand which genes will be selected by which algorithm. We summarise these implications in two rules of thumb at the end of the section.

6.3.1 Artificial Genes To demonstrate the effect of different kernels on gene selection, and the preference of certain kernels for certain genes, we created ten artificial genes and inserted them into two breast cancer datasets (dataset 9 and 10). The genes were created such that the signal-to-noise ratio was higher than those of the real genes. In a sense, we used the original microarray data as realistic noise, and we expect a feature selector to rank the artificial genes on the top. We experimented with both nonlinearly and linearly separable artificial genes, as shown in Figure 1. To illustrate the differences between these two types of genes, linear separability should arise when different phenotypic classes are clearly linked with certain high or low levels of expression for a group of genes (see Figure 1 a). Non-linear separability might occur when one of the phenotypic classes consists of subtypes, such that both subtypes show gene expression levels different from that of a healthy patient, but one subgroup has lower expression levels and the other higher (see Figure 1 b).

Our measure of performance on a gene ranking list given by a kernel was the median rank of the 10 artificial genes. This provides an estimate of the utility of the kernel for selecting the genes with high SNRs. We deem a feature selector competent for the task if this measure is less than 10. Table 3 lists the results of this experiment.

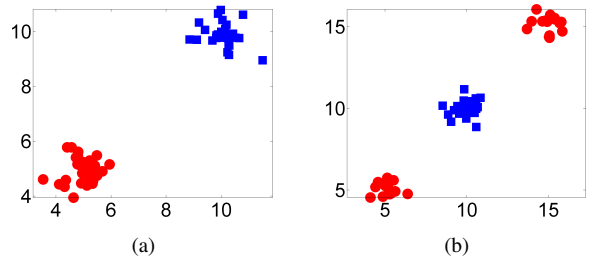

Fig. 1. First two dimensions of the artificial genes that are (a) linearly separable and (b) separable only nonlinearly. In both subplots, red dots represent data from the positive class, and blue squares data from the negative class. Each small cluster is generated by a ten dimension normal distribution with diagonal covariance matrix $0.25\mathbf{I}$.

We are particularly interested in the two new variants, RBF and dis, of the BAHSIC family. From the table, we observe that

1. RBF and dis perform comparably to existing BAHSIC members, such as pc and snr, in detecting artificial genes that are linearly separable. Most methods rank the ten inserted genes on the top.
2. RBF and dis perform much better in detecting artificial genes that are separable only nonlinearly. They rank the ten artificial genes on top in at least 9 out of the 10 folds, while other methods (except mi) fall short in ranking them correctly.

Note that, contrary to many existing methods, RBF and dis neither assume independence of the genes nor the linearly separability of the two classes. Hence, we expect them to detect relevant genes in unconventional cases where genes are interacting with each other in a nonlinear way. A natural question is whether this situation happens in practise. In the next section, we will show that, in some real microarray data, RBF and dis are indeed useful.

6.3.2 Subtype Discrimination using Nonlinear Kernels We now investigate why it is that nonlinear kernels (RBF and dis) provide better genes for classification in three datasets from Table 2 (datasets 18 (**?**), 27 (GDS1319) and 28 (GDS1454)). These datasets all represent multiclass problems, where each class corresponds to one disease subtype. Ideally, selected genes should contain information discriminating the classes. To visualise such information, we plot in Figure 2 the expression value of the top-ranked gene against that of a second gene ranked in the top 10. This second gene is chosen so that it has minimal correlation with the first gene. We use colours and shapes to distinguish data from different disease subtypes.

We found that genes selected using nonlinear kernels provide better separation between two subtypes (red dots and green diamonds), while the genes selected with the linear kernel do not separate these subtypes well. This eventually leads to better classification performance for the nonlinear kernels (see Table 2).

The principal characteristic of the datasets is that the blue square class is clearly separated from the rest, while the difference between the two subtypes (red dots and green diamonds) is less clear. The first gene provides information that distinguishes the blue square class, however it provides almost no information about the separation between the two subtypes. The linear kernel does not search for information complementary to the first gene, whereas nonlinear

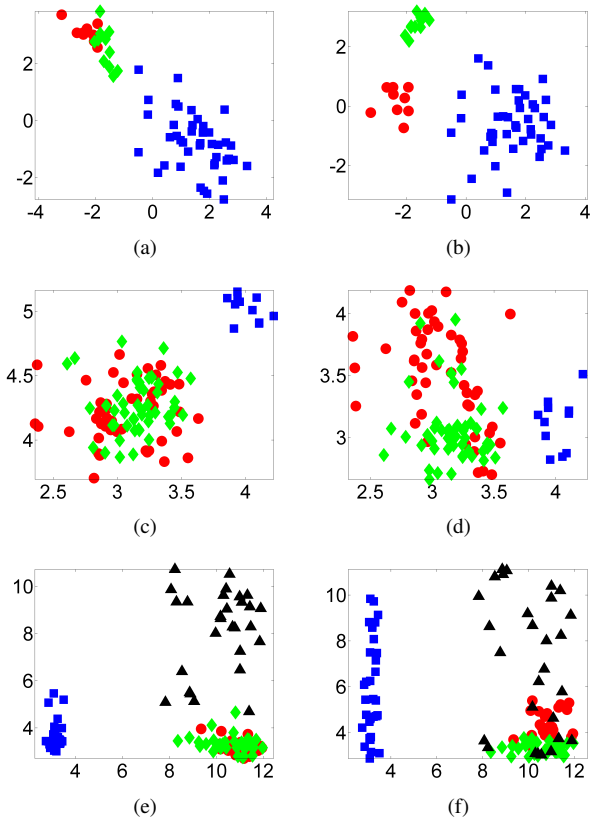

Fig. 2. Nonlinear kernels (RBF and dis) select genes that discriminate subtypes (red dots and green diamonds) where the linear kernel fails. The two genes in the left column are representative of those selected by the linear kernel, while those in the right column are produced with a nonlinear kernel for the corresponding datasets. Different colours and shapes represent data from different classes. (a) dataset 18 using lin; (b) dataset 18 using RBF; (c) dataset 28 using lin; (d) dataset 28 using RBF; (e) dataset 27 using lin; and (f) dataset 27 using dis.

kernels are able to incorporate complementary information. In fact, the second gene that distinguishes the two subtypes (red dots and green diamonds) does not separate all classes. From this gene alone, the blue square class is heavily mixed with other classes. However, combining the two genes together results in much better separation between all classes.

6.3.3 Rules of Thumb and Implication to Gene Activity To conclude our experiments, considering the fact that the linear kernel performed best in our feature selection evaluation, yet also taking into account the existence of nonlinear interaction between genes (as demonstrated in section 6.3.2), we can derive the following two rules of thumb for gene selection:

1. Always apply the linear kernel for general purpose gene selection.
2. Apply a Gaussian kernel if nonlinear effects are present, such as multimodality or complementary effects of different genes.

Table 1. Two-class datasets: classification error (%) and number of common genes (overlap) for 10-fold cross-validation using the top 10 selected features. Each *row* shows the results for a dataset, and each *column* is a method. Each entry in the table contains two numbers separated by "|": the first number is the classification error and the second number is the number of overlaps. Best results are in bold. The second last *row* summarises the number of times a method was the best. The last *row* contains the ℓ_2 distance of the error vectors between a method and the best performing method on each dataset. The last *column* shows in which datasets the BAHSIC family obtains the best performance (indicated by a ✓).

	BAHSIC family									Others			
Ref.#	**pc**	**snr**	**pam**	**t**	**m-t**	**lods**	**lin**	**RBF**	**dis**	**rfe**	**l1**	**mi**	
1	12.7|3	11.4|3	11.4|**4**	12.9|3	12.9|**4**	12.9|**4**	15.5|3	19.1|1	13.9|2	14.3|0	**7.7**|0	26.1|0	-|✓
2	33.2|1	33.9|2	33.9|1	29.5|1	29.5|1	**27.8**|1	32.9|2	31.5|**3**	32.8|2	34.2|0	32.5|1	29.9|0	✓|✓
3	37.4|0	37.4|0	37.4|0	**34.6**|**6**	**34.6**|**6**	**34.6**|**6**	37.4|1	37.4|0	37.4|0	37.4|0	37.4|0	36.4|0	✓|✓
4	41.6|0	38.8|0	41.6|0	40.7|**1**	40.7|0	**37.8**|0	41.6|0	41.6|0	39.7|0	41.6|0	41.6|0	40.6|0	✓|✓
5	27.8|0	**26.7**|0	27.8|0	**26.7**|**2**	**26.7**|**2**	**26.7**|**2**	27.8|0	27.8|0	27.6|0	27.8|0	27.8|0	27.8|0	✓|✓
6	30.0|2	**25.0**|0	31.7|0	**25.0**|**5**	**25.0**|**5**	**25.0**|**5**	30.0|0	31.7|0	30.0|1	30.0|0	33.3|0	33.3|0	✓|✓
7	**2.0**|**6**	**2.0**|5	**2.0**|5	28.7|4	26.3|4	26.3|4	**2.0**|3	**2.0**|4	30.0|0	**2.0**|0	**2.0**|0	**2.0**|2	✓|✓
8	3.3|3	**0.0**|4	**0.0**|4	**0.0**|4	3.3|**6**	3.3|**6**	3.3|2	3.3|1	6.7|2	**0.0**|0	3.3|0	6.7|1	✓|✓
9	10.0|**6**	10.0|**6**	**8.7**|4	34.0|5	37.7|**6**	37.7|**6**	12.0|3	10.0|5	12.0|1	10.0|0	17.0|1	12.0|3	✓|✓
10	16.0|2	18.0|2	**14.0**|2	**14.0**|8	22.0|**9**	22.0|**9**	16.0|2	16.0|0	18.0|0	32.5|0	**14.0**|0	20.5|1	✓|✓
11	12.9|5	12.9|5	12.9|5	19.5|0	22.1|0	33.6|0	11.2|4	**9.5**|**6**	16.0|4	19.0|0	17.4|0	11.2|4	✓|✓
12	30.3|2	36.0|2	31.3|2	26.7|**3**	35.7|0	35.7|0	**18.7**|1	35.0|0	33.0|1	29.7|0	30.0|0	23.0|2	✓|✓
13	8.4|5	11.1|0	7.0|5	22.1|3	27.9|**6**	15.4|1	7.0|2	9.6|0	11.1|0	**4.3**|1	5.5|2	7.0|4	-|✓
14	20.8|1	20.8|1	20.2|0	20.8|**3**	20.8|**3**	20.8|**3**	20.8|0	20.2|0	19.7|0	20.8|0	20.8|1	**19.1**|1	-|✓
15	**0.0**|7	0.7|1	**0.0**|5	4.0|1	0.7|**8**	0.7|**8**	**0.0**|3	**0.0**|2	2.0|2	**0.0**|1	**0.0**|1	**0.0**|7	✓|✓
best	2|2	4|1	5|1	5|6	3|10	5|9	3|0	3|2	0|0	4|0	4|0	3|0	
ℓ_2	16.9	20.9	17.3	43.5	50.5	50.3	**13.2**	22.9	35.4	26.3	19.7	23.5	

(pc=Pearson's correlation, snr=signal-to-noise ratio, pam=shrunken centroid, t=t-statistics, m-t=moderated t-statistics, lods=B-statistics, lin=centroid, RBF=$\exp(-\frac{\|x-x'\|}{2\sigma^2})$, dis=$\|x-x'\|^{-1}$, rfe=svm recursive feature elimination, l1=l1 norm svm, mi=mutual information)

Table 2. Multiclass datasets: in this case *columns* are the datasets, and *rows* are the methods. The remaining conventions follow Table 1.

Ref.#	16	17	18	19	20	21	22	23	24	25	26	27	28	**best**	ℓ_2
lin	36.7|1	**0.0**|3	5.0|**3**	10.5|6	35.0|3	**37.5**|**6**	**18.6**|1	**40.3**|**3**	**28.1**|**3**	26.6|**6**	**5.6**|**6**	27.9|7	45.1|1	6|6	**32.4**
RBF	33.3|**3**	5.1|4	**1.7**|**3**	**7.2**|**9**	33.3|0	40.0|1	22.1|0	72.5|0	39.5|0	**24.7**|4	**5.6**|**6**	22.1|**10**	**21.5**|3	5|5	37.9
dis	**29.7**|2	28.8|**5**	6.7|0	8.2|**9**	**29.4**|**7**	38.3|4	43.4|**4**	66.1|0	40.8|0	38.9|4	7.6|1	**8.2**|8	31.6|3	3|4	51.0
mi	42.0|1	11.4|3	**1.7**|2	7.7|8	39.4|4	38.3|3	30.3|1	57.3|2	37.6|1	40.8|2	6.5|**6**	22.6|3	23.3|**6**	1|2	37.0
	✓|✓	✓|✓	✓|✓	✓|✓	✓|✓	✓|✓	✓|✓	✓|✓	✓|✓	✓|✓	✓|✓	✓|✓	✓|-		

This result should come as no surprise, due to the high dimensionality of microarray datasets, but we make the point clear by a broad experimental evaluation. These experiments also imply a desirable property of gene activity as a whole: it correlates well with the observed outcomes. Multimodal and highly nonlinear situations exist, where a nonlinear feature selector is needed (as can be seen in the outcomes on datasets 18, 27 and 28), yet they occur relatively rarely in practise.

7 DISCUSSION

In this paper, we have defined the class of BAHSIC feature selection algorithms. We have shown that this family includes several well-known feature selection methods, which differ only by the choice of the preprocessing and the kernel function. Our experiments show that the BAHSIC family of feature selection algorithms performs well in practise, both in terms of accuracy and robustness. We have also shown that the linear kernel (centroid feature selector) performs best in general, and is thus a good first choice that provides good baseline results.

Table 3. Median rank of the ten artificial genes selected by different instances of BAHSIC over 10-fold cross-validation. The upper half of the table contains results for the linearly separable case. The lower half contains results for the nonlinearly separable case.

		BAHSIC family									Others		
	Ref.#	**pc**	**snr**	**pam**	**t**	**m-t**	**lods**	**lin**	**RBF**	**dis**	**rfe**	**l1**	**mi**
Linear	9	6	6	6	6	6	6	6	6	6	6	6	6
	10	6	6	6	6	6	6	6	6	6	6	6	6
Nonlinear	9	1937	1869	1935	260	221	221	1934	6	6	1721	30	6
	10	2043	2004	2043	2172	516	516	2041	7	6	1802	33	6

In the artificial gene experiments, we demonstrated the nonlinear RBF and dis kernels can select better features when there are nonlinear interactions. Furthermore we showed on real multiclass datasets that nonlinear kernels can select better genes for discriminating between subtypes. This indicates that nonlinear kernels are potentially useful for finding better prognostic markers and for subtype discovery.

The BAHSIC family represents a step towards establishing theoretical links between the huge set of feature selection algorithms in the bioinformatics literature. Only if we fully understand these theoretical connections can we hope to explain why different methods select different genes, and to choose feature selection methods that yield the most biologically meaningful results.

REFERENCES

Alizadeh, A., Eisen, M., Davis, R., *et al.* (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, **403**, 503–511.

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., and Levine, A. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci.*, **96**, 6745–6750.

Baker, C. (1973). Joint measures and cross-covariance operators. *Transactions of the American Mathematical Society*, **186**, 273–289.

Bedo, J., Sanderson, C., and Kowalczyk, A. (2006). An efficient alternative to svm based recursive feature elimination with applications in natural language processing and bioinformatics. In *Artificial Intelligence*.

Beer, D. G., Kardia, S. L., Huang, S. L., *et al.* (2002). Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nat. Med.*, **8**, 816–824.

Berchuck, A., Iversen, E., and et al., J. L. (2005). Patterns of gene expression that characterize long-term survival in advanced stage serous ovarian cancers. *Clin. Cancer Res.*, **11**, 3686–3696.

Bhattacharjee, A., Richards, W. G., Staunton, W. G., *et al.* (2001). Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *Proc. Natl. Acad. Sci.*, **98**, 13790–13795.

Brown, M., Grundy, W., Lin, D., Cristianini, N., Sugnet, C., Furey, T., Ares, M., and Haussler, D. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci.*, **97**, 262–267.

Bullinger, L., Dohner, K., Bair, E., Frohling, S., Schlenk, R. F., Tibshirani, R., Dohner, H., and Pollack, J. R. (2004). Use of gene-expression profiling to identify prognostic subclasses in adult acute myeloid leukemia. *New England Journal of Medicine*, **350**(16), 1605–1616.

Candes, E. and Tao, T. (2005). Decoding by linear programming. *IEEE Trans. Info Theory*, **51(12)**, 4203–4215.

Dhanasekaran, S. M., Barrette, T. R., Ghosh, D., Shah, R., Varambally, S., Kurachi, K., Pienta, K. J., Rubin, M. A., and Chinnaiyan, A. M. (2001). Delineation of prognostic biomarkers in prostate cancer. *Nature*, **412**(6849), 822–826.

Ein-Dor, L., Zuk, O., and Domany, E. (2006). Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proc. Natl. Acad. Sci. USA*, **103**(15), 5923–5928.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood an its oracle properties. *Journal of the American Statistical Association*, **96**(456), 1348–1360.

Feuerverger, A. (1993). A consistent test for bivariate dependence. *International Statistical Review*, **61**(3), 419–433.

Fukumizu, K., Bach, F. R., and Jordan, M. I. (2004). Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *J. Mach. Learn. Res.*, **5**, 73–99.

Gärtner, T., Flach, P., and Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. In B. Schölkopf and M. K. Warmuth, editors, *Proc. Annual Conf. Computational Learning Theory*, pages 129–143. Springer.

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

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. In *Proc. Intl. Conf. on Algorithmic Learning Theory*, pages 63–78.

Gruvberger, S., Ringner, M., Chen, Y., Panavally, S., Saal, L. H., Borg, A., Ferno, M., Peterson, C., and Meltzer, P. S. (2001). Estrogen receptor status in breast cancer is associated with remarkably distinct gene expression patterns. *Cancer Res*, **61**(16), 5979–5984.

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

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer, New York.

Iizuka, N., Oka, M., Yamada-Okabe, H., *et al.* (2003). Oligonucleotide microarray for prediction of early intrahepatic recurrence of hepatocellular carcinoma after curative resection. *Lancet*, **361**, 923–929.

Li, F. and Yang, Y. (2005). Analysis of recursive gene selection approaches from microarray data. *Bioinformatics*, **21**(19), 3741–3747.

Li, W. (2006). Bibliography on microarray data analysis.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research*, **2**, 419–444.

Rosenwald, A., Wright, G., Chan, G., *et al.* (2002). The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. *N. Engl. J. Med.*, **346**, 1937–1947.

Schölkopf, B. and Smola, A. (2002). *Learning with Kernels*. MIT Press, Cambridge, MA.

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

Smyth, G. (2004). Linear models and empirical bayes methods for assessing differential expressionin microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, **3**.

Steinwart, I. (2001). On the influence of the kernel on the consistency of support vector machines. *J. Mach. Learn. Res.*, **2**, 67–93.

Stolovitzky, G. (2003). Gene selection in microarray data: the elephant, the blind men and our algorithms. *Current Opinion in Structural Biology*, **13**(3), 370–376.

Tibshirani, R. (1994). Regression selection and shrinkage via the lasso. Technical report, Department of Statistics, University of Toronto. ftp://utstat.toronto.edu/pub/tibs/lasso.ps.

Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. In *National Academy of*

Sciences, volume 99, pages 6567–6572.

Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2003). Class prediction by nearest shrunken centroids, with applicaitons to dna microarrays. *Stat Sci*, **18**, 104–117.

Tusher, V. G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl. Acad. Sci. USA*, **98**(9), 5116–5121.

Valk, P. J., Verhaak, R. G., Beijen, M. A., Erpelinck, C. A., van Waalwijk van Doorn-Khosrovani, S. B., Boer, J. M., Beverloo, H. B., Moorhouse, M. J., van der Spek, P. J., Lowenberg, B., and Delwel, R. (2004). Prognostically useful gene-expression profiles in acute myeloid leukemia. *New England Journal of Medicine*, **350**(16), 1617–1628.

van de Vijver, M. J., He, Y. D., van 't Veer, L. J., *et al.* (2002). A gene-expression signature as a predictor of survival in breast cancer. *N. Engl. J. Med.*, **247**, 1999–2009.

van't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A. M., *et al.* (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.

Wainwright, M. (2006). Sharp thresholds for noisy and high-dimensional recovery of sparsity. Technical report, Department of Statistics, UC Berkeley.

Wang, Y., Klijn, J. G., Zhang, Y., *et al.* (2005). Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet*, **365**, 671–679.

Warnat, P., Eils, R., and Brors, B. (2005). Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes. *BMC Bioinformatics*, **6**, 265.

Welsh, J. B., Sapinoso, L. M., Su, A. I., Kern, S. G., Wang-Rodriguez, J., Moskaluk, C. A., r. Frierson HF, J., and Hampton, G. M. (2001). Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer. *Cancer Res*, **61**(16), 5974–5978.

West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Jr, J. O., J.R.Marks, and J.R.Nevins (2001). Predicting the clinical status of human breast cancer by using gene expression profiles. *PNAS*, **98**(20).