BIOINFORMATICS

Vol. 23 ISMB/ECCB 2007, pages i490–i498
doi:10.1093/bioinformatics/btm216

Gene selection via the BAHSIC family of algorithms

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

[1]National ICT Australia and Australian National University, Canberra, [2]University of Sydney, Australia, [3]Institute for Informatics, Ludwig-Maximilians-University, Munich and [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 non-linear kernels can be more suitable.

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

Contact: kb@dbs.ifi.lmu.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

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' (Li, 2006) has grown from less than 100 articles in 2000 to 1690 articles 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.

*To whom correspondence should be addressed.

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 generalize 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 Ein-Dor *et al.* (2006), 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 in sparse decoding (Candes and Tao, 2005; Wainwright, 2006) 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 (Stolovitzky, 2003). 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 article, 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) (Gretton *et al.*, 2005). We show that BAHSIC includes several well-known feature selection methods, namely Pearson's correlation coefficient (Ein-Dor *et al.*, 2006; van 't Veer *et al.*, 2002), *t*-test (Tusher *et al.*, 2001), signal-to-noise ratio (Golub *et al.*, 1999), Centroid (Bedo *et al.*, 2006; Hastie *et al.*, 2001), Shrunken Centroid (Tibshirani *et al.*, 2002, 2003) and ridge regression (Li and Yang, 2005).

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 non-linear kernels can be more suitable.

2 FEATURE SELECTION AND BAHSIC

The problem of feature selection can be cast as a combinatorial optimization 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 given by a quality measure $\mathcal{Q}(\mathcal{T})$, which is evaluated 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 extend 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

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

where E_{xy} is the expectation with respect to $\Pr_{xy}$, E_x is the expectation with respect to the marginal distribution $\Pr_x$, and $x^\top$ is the transpose of x. The covariance matrix encodes all second order dependence between the random variables. A statistic that efficiently summarizes the content of this matrix is its Hilbert–Schmidt norm: denoting by γ_i the singular values of $\mathcal{C}_{xy}$, the square of this norm is

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

This quantity is zero if and only if there exists no *second order dependence* between x and y. The Hilbert–Schmidt norm 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 non-linear effects are included). Second, the restriction to subsets of $\mathbb{R}^d$ and $\mathbb{R}^l$ excludes many interesting kinds of variables, such as strings and class labels. We wish therefore to generalize the notion of covariance to non-linear 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 (Lodhi *et al.*, 2002), graphs (Gärtner *et al.*, 2003) and so on (see Schölkopf *et al.*, 2004) for further examples in bioinformatics). We define a (possibly non-linear) 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 Baker, 1973; Fukumizu *et al.* 2004 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 Gretton *et al.* (2005) to be expressible in terms of kernels as

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

where $\mathsf{E}_{xx'yy'}$ is the expectation over both $(x, y) \sim \Pr_{xy}$ and an additional pair of variables $(x', y') \sim \Pr_{xy}$ drawn *independently* according to the same law. 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 Gretton *et al.* (2005) to be

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

where Tr is the trace (the sum of the diagonal entries), $\mathbf{K}, \mathbf{L} \in \mathbb{R}^{m\times m}$ 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 ($\delta_{ij} = 1$ when $i = j$, and zero otherwise). See Feuerverger (1993) for a different interpretation of a related criterion used in independence testing.

We now describe two theorems from Gretton *et al.* (2005) which support our using HSIC as a feature selection criterion. The first (Gretton *et al.*, 2005, 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

[1] A note on the non-linear 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 RBF kernel $k(x, x') = \exp(-0.5\sigma^{-2}||x - x'||^2)$, correspond to infinitely large feature spaces. We need never evaluate these feature representations explicitly, however.

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 (Gretton *et al.*, 2005, Theorem 4) states that when $\mathcal{F}, \mathcal{G}$ are RKHSs with universal (Steinwart, 2002) 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 maximize HSIC.

2.1 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.

In the case of more general kernels, forward selection is computationally more efficient, however, backward elimination (BA) in general yields better features, since the quality of the features is assessed within the context of all other features. Hence, we present the 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. In describing the algorithm, we modify our notation for HSIC to make clearer its dependence on the set of features chosen. Thus, we replace the definition in (5) with $\mathrm{HSIC}(\sigma, \mathcal{S})$, where $\mathcal{S}$ are the features used in computing the data kernel matrix $\mathbf{K}$, and $\sigma \in \Xi$ is the parameter for the data kernel $k(x, x')$ (for instance, this might be the size of a Gaussian kernel, or the degree of a polynomial kernel). The set Ξ denotes all possible kernel parameters.

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: $\mathcal{S}^\dagger \leftarrow \emptyset$
2: **repeat**
3: $\quad \sigma_0 \leftarrow \arg\max_\sigma \mathrm{HSIC}(\sigma, \mathcal{S}), \sigma \in \Xi$
4: $\quad \mathcal{I} \leftarrow \arg\max_{\mathcal{I}} \sum_{j \in \mathcal{I}} \mathrm{HSIC}(\sigma_0, \mathcal{S} \setminus \{j\}), \mathcal{I} \subset \mathcal{S}$
5: $\quad \mathcal{S} \leftarrow \mathcal{S} \setminus \mathcal{I}$
6: $\quad \mathcal{S}^\dagger \leftarrow \mathcal{S}^\dagger \cup \mathcal{I}$
7: **until** $\mathcal{S} = \emptyset$

Step 3 of the algorithm optimizes over the set Ξ. For this reason Ξ is restricted so as to make this search practical (the nature of the restriction depends on both the data and the kernel: for instance, in the case of the size parameter of a Gaussian kernel, we consider an interval of the form $\Xi = [10^{-8}; 10^2]$). If we have no prior knowledge regarding the nature of the non-linearity in the data, then optimizing 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 non-linearity, 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.

Step 4 of the algorithm is concerned with the selection of a set $\mathcal{I}$ of features to eliminate. While one could choose a single element of $\mathcal{S}$, this would be highly inefficient when there are a large number of irrelevant features. On the other hand, removing too many features at once risks the loss of relevant features. In our experiments, we found a good compromise between speed and feature quality was to remove 10% of the current features at each iteration.

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 SD 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 SD 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 (Ein-Dor *et al.*, 2006; van 't Veer *et al.*, 2002), 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 normalize the data and the labels by

s_x and s_y, respectively, and apply a linear kernel in both domains. HSIC then becomes

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

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 normalization 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 normalization of the data and labels, many variants can be derived. For example, the centroid (lin) (Bedo *et al*., 2006), t-score (t) (Hastie *et al*., 2001), moderated t-score (m-t), signal-to-noise ratio (snr) and B-statistics (lods) (Smyth, 2004) 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

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

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 normalize the data accordingly. For example, we obtain the signal-to-noise ratio if the data are normalized by $(s_{x+} + s_{x-})$.

3.3 Shrunken centroid

The shrunken centroid (pam) method (Tibshirani *et al*., 2002, 2003) 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 normalization of the data is similar to the previous section. For two-class problems, we use the 2D 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. $\mathbf{Y}^\top \mathbf{1}_\mathbf{m} = \mathbf{0}_2$), and thus

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

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 (Hastie *et al*., 2001), we predict the outputs $\mathbf{y}$ using the predictor $\mathbf{Vw}$ by minimizing the objective function $\mathcal{R} = (\mathbf{y} - \mathbf{Vw})^2 + \lambda\|\mathbf{w}\|^2$, where the second term is known as the regularizer. Our discussion encompasses two cases: first, the linear model, in which $\mathbf{V} = \mathbf{X}$; and second, the non-linear case, in which each of the m rows of $\mathbf{V}$ is a vector of non-linear 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 $\mathcal{R}$. Equivalently, after minimizing $\mathcal{R}$, this is the feature which has the smallest absolute weight $|w_i|$.

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

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

Therefore, recursively removing the feature which minimises the increase in $\mathcal{R}^*$ is equivalent to maximizing 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 non-linear features of the input may be infinite dimensional, and the regularizer 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 (Schölkopf and Smola, 2002) to obtain that the minimum objective is $\mathcal{R}^* = \mathbf{y}^\top\mathbf{y} - \mathbf{y}^\top(\hat{\mathbf{K}} + \lambda\mathbf{I})^{-1}\hat{\mathbf{K}}\mathbf{y}$. 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 ℓ_1-SVM for feature selection (l1).

The mutual information is a measure of statistical dependence between two random variables (Cover and Thomas, 1991), and is zero if and only if the variables are independent. To use the mutual information in a filter method for feature selection, Zaffalon and Hutter (2002) compute it between each feature and the labels: the features that correspond to the highest mutual information are 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 (Guyon *et al.*, 2002) is an embedded method for feature selection. It aims to optimize 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 ℓ_1-SVM (Tibshirani,1994) is also an embedded method for feature selection. Using an ℓ_1 norm as the regularizer in an SVM results in sparse weight vectors (Fan and Li, 2001), where the number of non-zero weights depends on the amount of regularization. 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 microarray datasets of gene expression levels, of which 15 are two-class datasets and 13 are multiclass datasets. Samples within one class represent one common phenotype or a subtype thereof. The 28 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), i.e. it contains expression levels for less than 1000 genes. All other datasets have dimensions ranging from ~2000 to 25 000. The number of samples varies between ~50 and 300 samples. A summary of the datasets and their sources is as follows:

- Six datasets studied in (Ein-Dor *et al.*, 2006). Three deal with breast cancer (van 't Veer *et al.*, 2002; van de Vijver *et al.*, 2002; Wang *et al.*, 2005) (numbered 1, 2 and 3), two with lung cancer (Bhattacharjee *et al.*, 2001; Beer *et al.*, 2002) (4, 5), and one with hepatocellular carcinoma (Iizuka *et al.*, 2003) (6). The B cell lymphoma dataset (Rosenwald *et al.*, 2002) is not used because none of the tested methods produce classification errors lower than 40%.
- Six datasets studied in (Warnat *et al.*, 2005). Two deal with prostate cancer (Dhanasekaran *et al.*, 2001; Welsh *et al.*, 2001) (7, 8), two with breast cancer (Gruvberger *et al.*, 2001; West *et al.*, 2001) (9, 10), and two with leukaemia (Bullinger *et al.*, 2004; Valk *et al.*, 2004) (16, 17).
- Five commonly used bioinformatics benchmark datasets on colon cancer (Alon *et al.*, 1999) (11), ovarian cancer (Berchuck *et al.*, 2005) (12), leukaemia (Golub *et al.*, 1999) (13), lymphoma (Alizadeh *et al.*, 2000) (18), and yeast (Brown *et al.*, 2000) (19).
- Nine datasets from the NCBI GEO database. The GDS IDs and reference numbers for this article 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 non-linear SVM with a Gaussian RBF kernel, $k(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\sigma^2}\right)$, was used. The regularization 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 Tables 1 and 2, respectively. In addition to the error rate, we also report the overlap between the top 10 gene lists created in each fold. The multiclass results are presented

Table 1. Two-class datasets: classification error (%) and number of common genes (overlap) for 10-fold cross-validation using the top 10 selected features

	BAHSIC family									Others		
Reference numbers	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	5|2	7|1	6|1	6|6	4|10	5|9	6|0	6|2	4|0	6|0	6|0	6|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

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. For classification error, the best result, and those results not significantly worse than it, are highlighted in bold (one-sided Welch *t*-test with 95% confidence level; a table containing the standard errors is provided in the Supplementary Material). For the overlap, largest overlaps for each dataset are highlighted (no significance test is performed). The second last *row* summarizes 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.
Note: 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 and l1 = ℓ_1 norm svm and mi = mutual information. The standard error in classification performance is given in the Supplementary Material

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

Reference numbers	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	7|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	6|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	5|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**	5|2	37.0

The standard error in classification performance is given in the Supplementary Material

separately since some older members of the BAHSIC family, and some competitors, are not naturally extensible to multiclass datasets. Our next two sections contain the analysis of these results: in Section 6.2, we discuss the consistency of each method across the various types of data, and in Section 6.3, we analyse the effect of kernel choice on performance, with a particular focus on linear versus non-linear kernels.

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 relatively expensive to obtain, and we want an algorithm to select the best genes. 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 penalizing 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 (the ℓ_2 distances) are listed in Tables 1 and 2. In general, the smaller the ℓ_2 distance, the better the method. It can be 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 summarize 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 (datasets 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 top. We experimented with both non-linearly 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 (Fig. 1a). 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 (Fig. 1b).

We used the median rank of the 10 artificial genes as our measure of ranking performance. This provides an estimate of the utility of the kernel for selecting the genes with high signal-to-noise ratios. We deem a feature selector competent for the task if this measure is less than 10. Table 3 lists the results of this experiment. 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 10 inserted genes on the top.

(2) RBF and dis perform much better in detecting artificial genes that are separable only non-linearly. They rank the 10 artificial genes on top in at least 9 out of the 10 folds, while other methods (except mi) fall short.

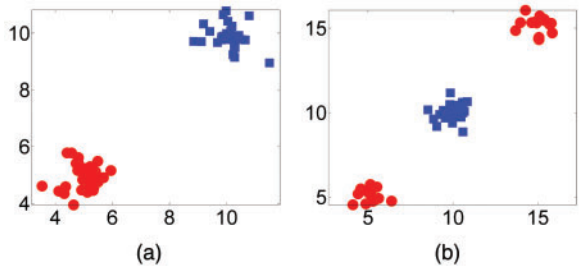

Fig. 1. First two dimensions of the artificial genes that are (**a**) linearly separable and (**b**) separable only non-linearly. In both subplots, red dots represent data from the positive class, and blue squares data from the negative class. Each small cluster is generated by a 10D normal distribution with diagonal covariance matrix 0.25**I**.

Unlike 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 non-linear 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 non-linear kernels We now investigate why it is that non-linear kernels (RBF and dis) provide better genes for classification in three datasets from Table 2 [datasets 18 (Alizadeh *et al.*, 2000) 27 (GDS1319), and 28 (GDS1454)]. These datasets all represent multiclass problems, where at least two of the classes are subtypes with respect to the same supertype.[2] Ideally, the selected genes should contain information discriminating the classes. To visualize this 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 classes (datasets 18 and 28 each contain 3 classes, therefore we use 3 different colour and shape combinations for them; dataset 27 has 4 classes, so we use 4 such combinations).

We found that genes selected using non-linear kernels provide better separation between the two classes that correspond to the same supertype (red dots and green diamonds), while the genes selected with the linear kernel do not separate these subtypes well. In the case of dataset 27, the increased discrimination between red and green comes at the cost of a greater number of errors in another class (black triangle), however, these mistakes are less severe than the errors made between the two subtypes by the linear kernel. This eventually leads to better classification performance for the non-linear kernels (see Table 2).

[2]For dataset 18, the 3 subtypes are diffuse large B-cell lymphoma and leukaemia, follicular lymphoma and chronic lymphocytic leukaemia; for dataset 27, the 4 subtypes are various C blastomere mutant embryos: wild type, pie − 1, pie − 1 + pal − 1 and mex − 3 + skn − 1; for dataset 28, the 3 subtypes are normal cell, IgV unmutated B-cell and IgV mutated B-cell.

Table 3. Median rank of the 10 artificial genes selected by different instances of BAHSIC over 10-fold cross-validation

		BAHSIC family									Others		
	References numbers	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

The upper half of the table contains results for the linearly separable case. The lower half contains results for the non-linearly separable case.

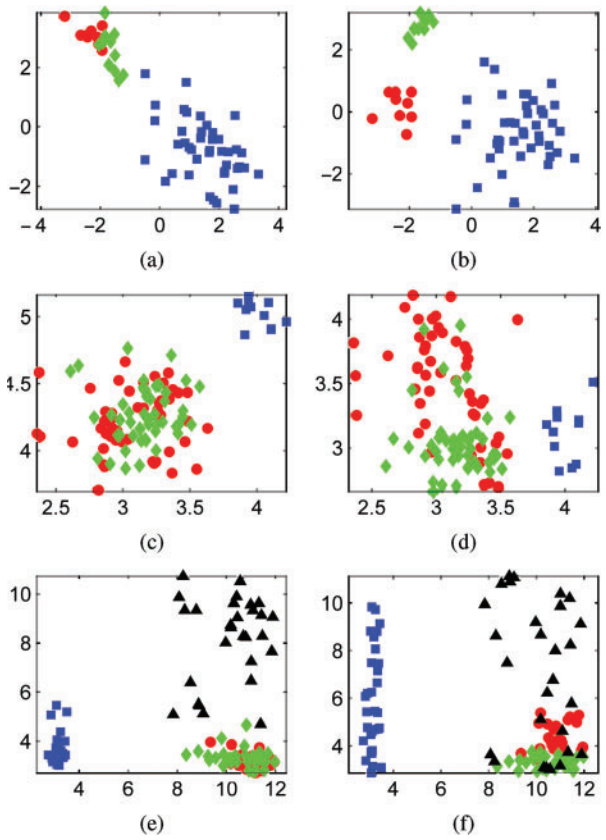

Fig. 2. Non-linear kernels (RBF and dis) select genes that discriminate subtypes (red dots and green diamonds) where the linear kernel fails. The two genes in the left column are representative of those selected by the linear kernel, while those in the right column are produced with a nonlinear kernel 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.

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 non-linear 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 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 non-linear interactions 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 non-linear effects are present, such as multimodality or complementary effects of different genes.

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 non-linear situations exist, where a non-linear 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 article, 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 in 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. In particular, the linear kernel (centroid feature selector) performs best in general, and is thus a reliable first choice that provides good baseline results.

In the artificial gene experiments, we demonstrated non-linear RBF and dis kernels can select better features when there

are non-linear interactions. Furthermore, we showed on real multiclass datasets that non-linear kernels can select better genes for discriminating between subtypes. This indicates that non-linear 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.

ACKNOWLEDGEMENTS

This work was supported in part by National ICT Australia; the German Ministry for Education, Science, Research and Technology (BMBF) under grant no. 031U112F within the BFAM (Bioinformatics for the Functional Analysis of Mammalian Genomes) project which is part of the German Genome Analysis Network (NGFN); and the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. National ICT Australia is funded through the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council.

Conflict of Interest: none declared.

REFERENCES

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

Alon,U. *et al*. (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl Acad. Sci. USA*, **96**, 6745–6750.

Baker,C. (1973) Joint measures and cross-covariance operators. *Trans. Am. Math. Soc.*, **186**, 273–289.

Bedo,J. *et al*. (2006) An efficient alternative to svm based recursive feature elimination with applications in natural language processing and bioinformatics. In *Artificial Intelligence*, LNCS **4304**, 170–180.

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

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

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

Brown,M. *et al*. (2000) Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci.*, **97**, 262–267.

Bullinger,L. *et al*. (2004) Use of gene-expression profiling to identify prognostic subclasses in adult acute myeloid leukemia. *N. Engl. J. Med.*, **350**, 1605–1616.

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

Cover,T.M. and Thomas,J.A. (1991) *Elements of Information Theory*. John Wiley and Sons, New York.

Dhanasekaran,S.M. *et al*. (2001) Delineation of prognostic biomarkers in prostate cancer. *Nature*, **412**, 822–826.

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

Fan,J. and Li,R. (2001) Variable selection via nonconcave penalized likelihood an its oracle properties. *J. Am. Stat. Assoc.*, **96**, 1348–1360.

Feuerverger,A. (1993) A consistent test for bivariate dependence. *Int. Stat. Rev.*, **61**, 419–433.

Fukumizu,K. *et al*. (2004) Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *J. Mach. Learn. Res.*, **5**, 73–99.

Gärtner,T. *et al*. (2003) On graph kernels: hardness results and efficient alternatives. In Schölkopf B. and Warmuth, M.K. (eds) *Proceedings of Annual Conference Computational Learning Theory*, Springer, pp 129–143.

Golub,T.R. *et al*. (1999) Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.

Gretton,A. *et al*. (2005) Measuring statistical dependence with Hilbert-Schmidt norms. In *Proceedings of the International Conference on Algorithmic Learning Theory*, pp 63–78.

Gruvberger,S. *et al*. (2001) Estrogen receptor status in breast cancer is associated with remarkably distinct gene expression patterns. *Cancer Res.*, **61**, 5979–5984.

Guyon,I. *et al*. (2002) Gene selection for cancer classification using support vector machines. *Mach. Learn.*, **46**, 389–422.

Hastie,T. *et al*. (2001) *The Elements of Statistical Learning*. Springer, New York.

Iizuka,N. *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**, 3741–3747.

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

Lodhi,H. *et al*. (2002) Text classification using string kernels. *J. Mach. Learn. Res.*, **2**, 419–444.

Rosenwald,A. *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. *et al*. (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. *Stat. Appl. Genet. Mol. Biol.*, **3**.

Steinwart,I. (2002) 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. *Curr. Opin. Struct. Biol.*, **13**, 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. *et al*. (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. In *National Academy of Sciences*. vol. 99, pp. 6567–6572.

Tibshirani,R. *et al*. (2003) Class prediction by nearest shrunken centroids, with applicaitons to DNA microarrays. *Stat. Sci.*, **18**, 104–117.

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

Valk,P.J. *et al*. (2004) Prognostically useful gene-expression profiles in acute myeloid leukemia. *N. Engl. J. Med.*, **350**, 1617–1628.

van de Vijver,M.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. *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. *et al*. (2005) Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet*, **365**, 671–679.

Warnat,P. *et al*. (2005) Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes. *BMC Bioinformatics*, **6**, 265.

Welsh,J.B. *et al*. (2001) Analysis of gene expression identifies candidate markers and pharmacological targets in prostate cancer. *Cancer Res.*, **61**, 5974–5978.

West,M. *et al*. (2001) Predicting the clinical status of human breast cancer by using gene expression profiles. *PNAS*, **98**.

Zaffalon,M. and Hutter,M. (2002) Robust feature selection using distributions of mutual information. In Darwiche, A. and Friedman, N. (eds), *Proceedings of the 18th International Conference on Uncertainty in Artificial Intelligence (UAI-2002)*, Morgan Kaufmann, San Francisco, CA, pp. 577–584.