BIOINFORMATICS ORIGINAL PAPER

Vol. 28 no. 13 2012, pages 1766–1774
doi:10.1093/bioinformatics/bts238

Data and text mining

Advance Access publication April 25, 2012

Statistical interpretation of machine learning-based feature importance scores for biomarker discovery

Vân Anh Huynh-Thu[1,2,*], Yvan Saeys[3], Louis Wehenkel[1,2] and Pierre Geurts[1,2,*]

[1]Department of Electrical Engineering and Computer Science, Systems and Modeling and [2]GIGA-Research, Bioinformatics and Modeling, University of Liège, 4000 Liège, Belgium and [3]Department of Plant Biotechnology and Bioinformatics, Ghent University, 9052 Gent, Belgium

Associate Editor: Jonarthan Wren

ABSTRACT

Motivation: Univariate statistical tests are widely used for biomarker discovery in bioinformatics. These procedures are simple, fast and their output is easily interpretable by biologists but they can only identify variables that provide a significant amount of information in isolation from the other variables. As biological processes are expected to involve complex interactions between variables, univariate methods thus potentially miss some informative biomarkers. Variable relevance scores provided by machine learning techniques, however, are potentially able to highlight multivariate interacting effects, but unlike the *p*-values returned by univariate tests, these relevance scores are usually not statistically interpretable. This lack of interpretability hampers the determination of a relevance threshold for extracting a feature subset from the rankings and also prevents the wide adoption of these methods by practicians.

Results: We evaluated several, existing and novel, procedures that extract relevant features from rankings derived from machine learning approaches. These procedures replace the relevance scores with measures that can be interpreted in a statistical way, such as *p*-values, false discovery rates, or family wise error rates, for which it is easier to determine a significance level. Experiments were performed on several artificial problems as well as on real microarray datasets. Although the methods differ in terms of computing times and the tradeoff, they achieve in terms of false positives and false negatives, some of them greatly help in the extraction of truly relevant biomarkers and should thus be of great practical interest for biologists and physicians. As a side conclusion, our experiments also clearly highlight that using model performance as a criterion for feature selection is often counter-productive.

Availability and implementation: Python source codes of all tested methods, as well as the MATLAB scripts used for data simulation, can be found in the Supplementary Material.

Contact: vahuynh@ulg.ac.be, or p.geurts@ulg.ac.be

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

Received on December 9, 2011; revised on April 2, 2012; accepted on April 18, 2012

*To whom correspondence should be addressed.

1 INTRODUCTION

Univariate hypothesis testing is widely used in the context of biomarker discovery in bioinformatics, where one seeks to identify biological variables (e.g. genes or genetic polymorphisms) that truly provide information about some phenotype of interest (e.g. disease status or treatment response). A classic procedure consists in applying a statistical test to compute a *p*-value for each variable of the considered problem and selecting variables that have a *p*-value lower than a chosen threshold. To cope with multiple hypothesis problems, *p*-values are typically replaced with an estimation of the false discovery rate (FDR) or the family wise error rate (FWER), (Ge *et al.*, 2003).

Univariate tests can only identify variables that provide a significant amount of information about the output variable in isolation from the other inputs. Since biological processes are expected to involve complex interactions between variables, these procedures potentially miss some informative biomarkers. Nowadays, when one seeks multivariate interacting effects between features, one can resort to relevance scores provided by machine learning techniques. Among these, the most popular methods include importance scores derived from a tree-based ensemble method (Hastie *et al.*, 2003) or feature weights computed for example from a linear support vector machine (SVM) (Rakotomamonjy, 2003). However, unlike the *p*-values returned by univariate tests, these relevance scores are usually not statistically interpretable. This lack of interpretability prevents the wide adoption of these methods by practicians, biologists or physicians and also makes the identification of the truly relevant variables among the top-ranked ones, i.e. the determination of a relevance threshold, a very difficult task in practice.

In this article, we evaluate several, existing and novel, procedures that extract relevant features from a ranking returned by a multivariate algorithm. These procedures replace the original relevance score with a measure that can be interpreted in a statistical way and hence allow the user to determine a significance threshold in a more informed way. Most of these methods exploit a resampling procedure to estimate the FDR or FWER among the k top-ranked features, for increasing values of k. Just like for standard univariate tests, the user can then choose a threshold on this new measure depending on the risk he/she is ready to take when deeming that all features above this threshold are relevant. Experiments on several artificial problems, as well as on real microarray datasets, show that some of these measures greatly help in the extraction of truly relevant features from a ranking derived from a multivariate approach.

We also highlight that the common approach to this problem, i.e. selecting the top k features minimizing some cross-validated error, is not a good practice in general, as it typically leads to the selection of several irrelevant features.

2 PROBLEM DEFINITION

In this article, we focus on the problem of selecting relevant features from a ranking. We assume that we have at our disposal a learning sample LS of n instances of input–output pairs drawn from some unknown probability distribution. There are m input variables denoted $X_i, i=1,\ldots,m$. We further assume that we have a machine learning algorithm $\mathcal{A}$(LS) that outputs from the learning sample LS a feature ranking, typically derived from a relevance score s_i for each input variable X_i. These scores are not supposed to be independent and no further assumption is made about $\mathcal{A}$. The goal is then to determine a value k such that the subset composed of the k top-ranked variables contains the highest possible number of *relevant* features, i.e. variables that convey information about the target output variable, *in isolation or in conjunction with other relevant variables*.

Different sensitivity/specificity compromises are possible and depend on the considered application. In this article, we aim at high specificity, i.e. at identifying subsets of relevant variables while maintaining the rate of false positives as small as possible. This type of compromise is typically sought in the context of biomarker discovery because of high costs of subsequent experiments (Saeys *et al.*, 2007).

3 FEATURE SELECTION METHODS

We describe below several methods that have been developed for the selection of relevant variables from a ranking. We assume that we have an algorithm $\mathcal{A}$(LS) that returns, from a learning sample LS, a relevance score s_i for each input variable $X_i, i=1,\ldots,m$, and we further assume, without loss of generality, that the features are numbered according to their relevance score, i.e. $s_1 \geq s_2 \geq \cdots \geq s_m$. Most of the presented methods then reuse $\mathcal{A}$ on a modified LS (obtained from a subsampling, a permutation, etc.) to replace each original relevance score s_i with a statistically interpretable measure. The intuition behind each method is given below and their detailed pseudo-code descriptions can be found in Supplementary Material.

3.1 Estimation of the generalization error of a model (err-$\mathcal{A}$ and err-TRT)

We include in our comparison, as a baseline method, the procedure based on the computation of the generalization error of a predictive model [see Geurts *et al.* (2005) for an example]. This method consists in estimating the error rate (resp. quadratic error) e_i of a classification (resp. regression) model that uses only the first i variables of the ranking, $\forall i=1,\ldots,m$, and selecting the k top-ranked variables such that

$$k = \arg\min_{i=1,\ldots,m} e_i. \tag{1}$$

The m predictive models can be learned using the algorithm $\mathcal{A}$ that was used to compute the ranking of variables and the generalization error of one model can be estimated using a cross-validation procedure (10-fold in all our experiments). We call this method *err-*$\mathcal{A}$ to denote the fact that the same algorithm $\mathcal{A}$ is used both to rank the features and to estimate the error associated with each feature subset. A sharper threshold can be obtained by estimating the generalization error with an algorithm that is not robust to irrelevant variables, such as k-NN (Fukunaga and Hostetler, 1975) or totally randomized trees (TRT, i.e. an ensemble of trees with completely random split choices; Geurts *et al.*, 2006). Compared with a robust algorithm, we expect the error of such a procedure to increase in a more abrupt way when irrelevant variables are introduced in the predictive model and therefore to yield a smaller number of selected variables. We used TRT in our experiments as this method is computationally less expensive than k-NN and we call the resulting feature selection method *err*-TRT.

One potential drawback of this approach is the fact that it is prone to *selection bias* (Ambroise and McLachlan, 2002; Smialowski *et al.*, 2010), as the same instances of LS are used to rank the variables and to estimate the generalization error. This results in a too optimistic estimation of the errors e_i, and in particular of the minimal error $\min_i e_i$, whose effect on the number of selected features is difficult to appraise. One could get better error estimates by ranking the features inside the cross-validation loop (Ambroise and McLachlan, 2002) but this would leave open the question of the selection of the final feature subset among the subsets generated within each fold.

3.2 Multiple testing with random permutations (nFDR, eFDR and conditional error rate)

The FDR (Storey and Tibshirani, 2003) is the expected rate of truly irrelevant features among the variables that are deemed relevant. Hence, given a selection threshold s_i, the FDR is defined as

$$\mathrm{FDR}_i = E\left[\frac{V_i}{R_i} | R_i > 0\right] P(R_i > 0), \tag{2}$$

where R_i is the number of variables considered relevant at score s_i and V_i is the number of those variables that are truly irrelevant. V_i/R_i is thus set to zero if $R_i = 0$. To select a subset of variables, some methods estimate the FDR for increasing values of i and choose the maximum value of i such that $\mathrm{FDR}_i < \alpha$, where α is typically small and reflects the risk one is ready to accept in terms of false positives when selecting the variables.

In the context of univariate variable scoring procedures, a classic approach to estimate the FDR is based on permutation tests (Ge *et al.*, 2003). The FDR is approximated by

$$\mathrm{FDR}_i = \frac{E[V_i | H_I^{1\rightarrow m}]}{R_i}, \tag{3}$$

where $H_I^{1\rightarrow m}$ is the hypothesis that all the variables are irrelevant. R_i is considered equal to i and $E[V_i|H_i^{1\rightarrow m}]$ is taken as the expected number of variables that get a score greater than s_i when the output values are randomly permuted in the learning sample, making all the variables irrelevant. We call the FDR estimated using this approach the nFDR. Altmann *et al.* (2010) proposed a very similar permutation scheme to associate a p-value to Random Forests (RFs) importance scores.

Huynh-Thu *et al.* (2008) applied the nFDR approach when the relevance scores are derived from tree-based importance measures instead of univariate scores. They showed empirically that this procedure overestimates in an unpredictable way the real FDR and thus can lead to unreliable selections of relevant subsets. This overestimation of the FDR can be explained, at least partially, by the fact that this procedure does not take into account the dependence that exists between the tree–based importance scores for different variables. To overcome this limitation, they proposed an alternative measure to be associated with each threshold s_i and that takes into account the scores of the variables that are ranked above X_i. For each subset of i top-ranked variables, the procedure consists in computing the following conditional probability, called the *conditional error rate* (CER):

$$\mathrm{CER}_i = P(\max_{k=i,\ldots,m} S_k \geq s_i | H_{\mathrm{R}}^{1\rightarrow i-1}, H_{\mathrm{I}}^{i\rightarrow m}), \tag{4}$$

where $H_{\mathrm{R}}^{1\rightarrow i-1}$ denotes the hypothesis that features $X_1,\ldots,X_{i-1}$ are relevant $H_{\mathrm{I}}^{i\rightarrow m}$ is the hypothesis that features $X_i,\ldots,X_m$ are irrelevant and S_k is the random variable denoting the relevance score of X_k under these two hypotheses. CER_i is thus the probability that at least one irrelevant variable among $m-i+1$ gets a relevance score greater or equal to s_i, when these

scores are computed under the assumption that variables $X_1,\ldots,X_{i-1}$ are all relevant. The CER is hence an estimation of the FWER, that is defined as the probability to include at least one irrelevant variable among those selected. $H_R^{1\to i-1}$ is approximated by keeping the values of the output variable and of the first $i-1$ variables unchanged, whereas hypothesis $H_I^{i\to m}$ is simulated by randomly permuting the values of $X_i,\ldots,X_m$. To adhere as much as possible to the original joint distribution of the variables, they are furthermore permuted jointly, i.e. using the same permutation vector. Note that when the scores s_i are univariate statistics, expression (4) corresponds precisely to the definition of Westfall and Young's (1993) *step-down maxT adjusted p-values* (Ge *et al.*, 2003).

Another permutation-based approach was proposed by Ge *et al.* (2008) to estimate the FDR in the context of univariate rankings. This approach also makes the assumption that the first $i-1$ variables are relevant and the FDR at threshold s_i is defined by

$$\mathrm{FDR}_i = E\left[\frac{V_i}{V_i+i-1}\middle| H_R^{1\to i-1}, H_I^{i\to m}\right]. \quad (5)$$

The number V_i of false positives is estimated by the following way. Let s_k^p be the relevance score of $X_k(\forall k=1,\ldots,m)$, calculated from a random permutation of the data that simulates $H_R^{1\to i-1}$ and $H_I^{i\to m}$, and let $s_{(k)}^p$ be the k-th largest member of $\{s_i^p,\ldots,s_m^p\}$. V_i is then computed as

$$V_i = \max_{k=1,\ldots,m-i+1}\{k : s_{(1)}^p \geq s_i, s_{(2)}^p \geq s_{i+1},\ldots,s_{(k)}^p \geq s_{i+k-1}\}. \quad (6)$$

The FDR estimated using Equations (5) and (6) is called eFDR. When applying this approach to rankings derived from a multivariate approach, we propose to use the same permutation scheme as in the CER approach.

3.3 Empirical estimation of the null rank distribution (mr-test)

The *mr-test* (Zhang *et al.*, 2006) estimates an empirical distribution of the rank of an irrelevant feature, to derive a p-value p_i to be associated with each variable X_i, defined as the probability for an irrelevant variable to be ranked above or at the same position as X_i. To estimate the distribution of the rank of an irrelevant variable, Zhang *et al.* (2006) proposed to proceed as follows. P feature rankings are obtained by applying the algorithm $\mathcal{A}$ on P resamplings of the original learning sample. Given a user-defined number k, the k variables that have on average the largest ranks among all the variables are considered putative irrelevant variables and the null rank distribution is estimated from their $k\times P$ ranks over the P rankings. The p-value p_i is then estimated as the proportion of these $k\times P$ ranks that are lower than the average rank $\bar{r}_i$ of X_i over the P rankings.

As the p-values calculated using this procedure are *raw* p-values, the so-called multiple-testing problem occurs, where the higher the number of variables in the considered problem, the higher the number of expected variables with a p-value lower than some threshold α, even if these variables are irrelevant. We therefore propose to apply a multiple-testing correction procedure and to select the variables based on the corrected p-values. In our experiments, we used the Benjamini Hochberg correction (Benjamini and Hochberg, 1995), which was shown to control the FDR in the context of univariate statistical tests.

The mr-test procedure has two parameters. The first one is the number k of putative irrelevant variables from which the empirical null rank distribution is estimated. A small value of k would result in overoptimistic selections of variables whereas a high value of k would be too conservative. In our experiments, k was fixed to $m/2$. The second parameter is the number of resampled instances at each iteration that we fixed to half of the number of instances in the original learning sample, as proposed by Zhang *et al.* (2006).

3.4 Introduction of random probes (1Probe and mProbes)

Stoppiglia *et al.* (2003) suggested to introduce one random feature to compute the probability p_i for this random feature to be ranked above or at the same position as X_i. They applied this idea in the context of linear models where each variable X_i is ranked according to the squared cosine of the angle between X_i and the output variable, and where therefore the distribution of the rank of the random feature can be computed analytically. However, to be able to apply this approach with any ranking procedure, the authors suggested in their conclusions to compute the null rank distribution empirically by artificially introducing random probes. We therefore propose the following procedure, that we call *1Probe*. In each of P iterations, we introduce in the original learning sample an additional variable X_{rand} whose values are randomly sampled from $\mathcal{N}(0,1)$. We then estimate the p-value p_i by the rate of iterations where X_{rand} is ranked above X_i. As the p-values calculated using this procedure are prone to the multiple-testing problem, we propose to correct them using the Benjamini Hochberg procedure, such as in the mr-test procedure. Note that the 1Probe method is parametric, as the choice of the distribution of the random probe can have an impact on its rank. The level of impact, however, depends on the ranking method used.

Along a similar line, the ACE (for "Artificial Contrasts with Ensembles") method (Tuv *et al.*, 2009) introduces as many random features as there are input variables in the original problem. Each random feature is generated by permuting the values of one original variable. The method then assumes that an original variable is irrelevant if it has a relevance score not statistically higher than that of a random feature. In the original approach, a t-test is applied to determine the significance of each variable and the procedure is actually wrapped into a gradient boosting type algorithm that iteratively selects subsets of important variables. We propose to use a variant of ACE, that we call *mProbes*, where instead of applying a t-test, we simply compute the proportion of simulations where at least one random feature is ranked above X_i. We also drop the gradient boosting procedure and apply the approach in one single run. The value associated with X_i that is returned by mProbes thus estimates the FWER when selecting X_i and all the variables ranked above X_i.

3.5 Computational complexity

Although computing time is not a real issue in most applications, Table 1 shows the computational complexity of each method. Except err-$\mathcal{A}$ and err-TRT, all the methods have a common parameter P, which is the number of iterations or permutations. The higher the value of P, the better the (Monte Carlo) estimate of the FDR/FWER/p-value. In all our experiments, P was fixed to a typical value of 1000, that gives a good compromise between accuracy and computing times. Obviously, in the context of a specific study, P could be increased to improve precision if needed.

Among all methods, the mr-test has the lowest complexity as $\mathcal{A}$ is run on only half of the instances of the learning sample in each iteration. On the other hand, the eFDR and CER have the higher complexities if one wants to compute these measures for all m variables. However, as suggested by Huynh-Thu *et al.* (2008), the computing times of these procedures can be

Table 1. Computational complexity

Method	Complexity
CER	$M\times P\times C_{\mathcal{A}}(n,m)$
nFDR	$P\times C_{\mathcal{A}}(n,m)$
eFDR	$M\times P\times C_{\mathcal{A}}(n,m)$
mr-test	$P\times C_{\mathcal{A}}(\frac{n}{2},m)$
1Probe	$P\times C_{\mathcal{A}}(n,m+1)$
mProbes	$P\times C_{\mathcal{A}}(n,2m)$
err-$\mathcal{A}$	$C_{\mathcal{A}}(n,1)+C_{\mathcal{A}}(n,2)+\cdots+C_{\mathcal{A}}(n,m)$
err-TRT	$m\times n\times \log n$

P is the number of iterations, M is the number of variables for which one wants to compute the eFDR/CER, $C_{\mathcal{A}}(n,m)$ is the computational complexity of algorithm $\mathcal{A}$ when applied on a learning sample with n instances and m variables. For RFs, $C_{\mathrm{RF}}(n,m)=O(\sqrt{m}.n.\log n)$. For SVMs, C_{SVM} lies between $O(m.n^2)$ and $O(m.n^3)$.

reduced by stopping them as soon as the eFDR/CER is greater than some significance level. It is also worth mentioning that all the methods can be easily parallelized on a computing grid.

4 DATASETS AND PROTOCOL

We describe in this section the artificial and real datasets that we used for our experiments, the performance metrics and the compared ranking methods.

4.1 Artificial datasets

We generated two families of artificial problems to validate the feature selection methods in a context where relevant variables are perfectly known.

Linear This is a linear two-class classification problem. All input variables are continuous and their values are sampled from $\mathcal{N}(0,1)$. The output Y is given by

$$Y = \mathrm{sgn}\left(\sum_{i=1}^{p} w_i X_i\right), \tag{7}$$

where the values of w_i are uniformly distributed random numbers between 0 and 1. In addition, 1% of the output labels are randomly flipped. Irrelevant variables, in the form of pure Gaussian noise, are added to the p relevant variables.

Hypercube This two-class classification problem was generated by adapting the MATLAB® (http://www.mathworks.com/) code originally used to produce the *Madelon* dataset for the NIPS feature selection challenge. (http://www.clopinet.com/isabelle/Projects/NIPS2003/) All input variables are continuous and their values are sampled from $\mathcal{N}(0,1)$. Each class is composed of a number of Gaussian clusters that are placed at random on the vertices of a hypercube in a p-dimensional space, where p is the number of relevant variables. Unlike the previous problem, the decision boundary is thus potentially nonlinear. Irrelevant variables, which are pure Gaussian noise, are added to these p variables.

4.2 Microarray datasets

We performed experiments on six real gene expression datasets (Table 2). For each dataset, the goal was to find a subset of genes that helps to discriminate between two groups of patients.

4.3 Performance metrics

Each method returns a subset of features that it considers relevant. In the context of the artificial datasets where all relevant features are perfectly known, we used the following metrics to evaluate such a subset:

- precision $= \frac{\mathrm{TP}}{S}$,
- recall $= \frac{\mathrm{TP}}{P}$,

where S is the number of selected features, TP is the number of these features that are truly relevant and P is the total number of truly relevant variables in the considered problem. We also compared the precision and recall levels with the following values:

- $p_{\max}$: the precision of a method (called *rec-1*) that would select the first k variables of the ranking, where k is the smallest integer such that $\{X_1, X_2, \ldots, X_k\}$ contains *all* the truly relevant variables (the recall is equal to one).
- $r_{\max}$: the recall of a method (called *prec-1*) that would select the first k variables of the ranking, where k is the largest integer such that $\{X_1, X_2, \ldots, X_k\}$ contains *only* truly relevant variables (the precision is equal to one).

Finally, to evaluate a ranking of variables independently of the choice of a specific threshold, we used the area under the precision-recall (AUPR) curve which plots the precision versus the recall for varying thresholds. The higher the AUPR, the better the ranking.

Table 2. Characteristics of the microarray datasets

Name	No. of Class 1	No. of Class 2	No. of Features	Reference
Breast	107	179	22 283	Wang *et al.* (2005)
Leukemia	47	25	7129	Golub *et al.* (1999)
Lymphoma1	22	23	4026	Alizadeh *et al.* (2000)
Lymphoma2	32	26	7129	Shipp *et al.* (2002)
Prostate1	34	19	4344	Dhanasekaran *et al.* (2001)
Prostate2	52	50	12 600	Singh *et al.* (2002)

4.4 Compared ranking methods

We validated the feature selection methods in the context of three popular ranking algorithms; two representatives of multivariate techniques and one standard univariate method:

- **Importance measures derived from a specific tree-based ensemble method called Random Forests** (Breiman, 2001). In a Random Forests (RFs) ensemble, each tree is built from a bootstrap copy of the original learning sample and at each test node, K variables are selected at random among all candidate ones before determining the best split, i.e. the split that reduces the most the class entropy in the resulting subsets of instances. As variable importance measures, we used the importance scores that result from the sum of the total reduction of class entropy brought by each variable over all trees in the ensemble. K was fixed to its default value $\sqrt{m}$ and ensembles of 1000 trees were grown.
- **Importance measures derived from a linear SVM (Boser *et al.*, 1992)**. The goal of a linear SVM is to find a hyperplane in the form

$$f(\mathbf{X}) = \sum_{i=1}^{m} w_i . X_i + b, \quad (w_i, b \in \mathbb{R})$$

that separates the instances of different classes in the input space with the largest margin, while softly penalizing the instances that are on the wrong side of the hyperplane. The score s_i of feature X_i is simply taken as the absolute value of the coefficient w_i (Guyon *et al.*, 2002). We used this procedure rather than the well-known RFE procedure (Guyon *et al.*, 2002) because it is much less computationally expensive. For our experiments, we used the LIBSVM library (Chang and Lin, 2011), with the regularization parameter C of the SVM set to 1 (default value), except in Table 3 where C was set by 10-fold cross validation. Before training, we rescaled the data so that the values of each variable were comprised between -1 and 1 in the learning sample.
- **The absolute values of the t statistics derived from a t-test.** Even though our methods target multivariate ranking techniques, they can all be applied to univariate techniques as well. We thus included the t-test as a representative of univariate ranking methods, to check the behaviour of the feature selection methods in this context.

5 RESULTS

Results on artificial and real microarray datasets, obtained using the described methods, are presented in this section.

5.1 Artificial datasets

Comparison of the ranking methods Figure 1 shows the AUPRs of the three ranking procedures (RFs, linear SVM and t-test) on

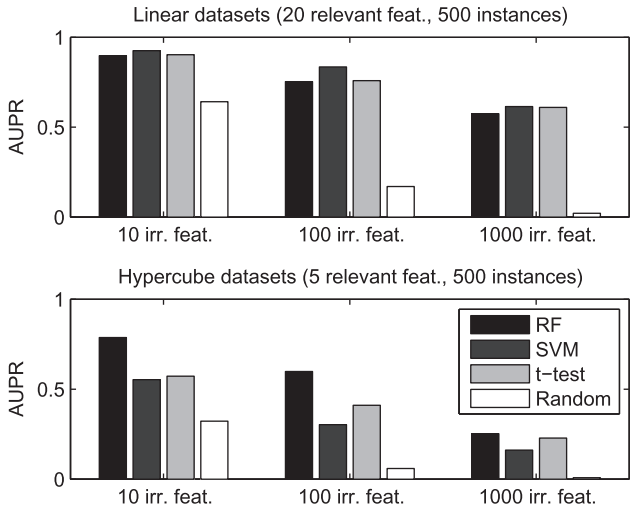

Fig. 1. AUPRs of each ranking method, for different numbers of irrelevant features. *Random* is a method that randomly ranks the variables. Top on linear datasets, bottom on hypercube datasets. The AUPR values were averaged over 50 datasets in each case

linear and hypercube datasets, as well as the AUPRs of a method that returns a random ranking for comparison. The AUPR values were averaged over 50 randomly generated datasets.

The three ranking methods perform better than the random procedure. The linear SVM yields the highest AUPRs on the linear datasets, although the t-test performs equally well for a high number of irrelevant features. On the (nonlinear) hypercube datasets, the RFs procedure is the best performer.

Figures S1–S3 of Supplementary material show the AUPRs corresponding to the final rankings returned by mr-test, 1Probe and mProbes. Indeed, these procedures each compute a statistical measure (p-value or FWER) associated with each variable X_i. These three methods thus potentially modify the original variable ranking by reordering the features according to the corresponding statistical measure. Nevertheless, the new rankings do not change much with respect to the original ranking, their corresponding AUPRs hardly varying. On the other hand, the CER, nFDR and eFDR procedures each estimate a statistic that corresponds to an *importance score* s_i rather than to a variable in itself. Therefore, the variables cannot be reordered according to this statistic, and the monotonicity of the estimated measures is enforced instead (see pseudo-codes in Supplementary material). The enforced monotonicity ensures that a variable X_i can be selected only if all the variables ranked above X_i are also selected.

Interpretability of the curves Figure 2a plots the curves of the different methods on a linear dataset with 20 relevant features. The RFs method was used as ranking procedure. At each rank i, we show the relevance score derived from the RFs, as well as the *observed* FDR, i.e. the proportion of truly irrelevant features among the i top-ranked variables. Nearly identical observed FDR curves are obtained when the variables are ranked using mr-test, 1Probe and mProbes (Supplementary Fig. S4). Therefore, only the observed FDR related to the original ranking is plotted in Figure 2, for the sake of clarity.

We can see that selecting the variables based solely on the original relevance score is difficult as this score does not suggest any clear threshold (dashed curve in the top of Fig. 2). On the other hand, almost all studied methods successfully help to select variables. CER and mProbes provide a good estimation of the FWER as the transition between low and high CER/mProbes values is quite well centred at the point where irrelevant variables start appearing in the ranking (indicated by the observed FDR that becomes >0). The nFDR overestimates the real FDR, as already observed by Huynh-Thu *et al.* (2008), whereas the eFDR is closer to it. CER, nFDR, mr-test and mProbes tend to be highly conservative, as their curves increase whereas the observed FDR is still equal to 0. In contrast, the values returned by eFDR and 1Probe become high only when larger subsets of top-ranked variables are considered. err-RF and err-TRT both select a high number of false positives. The minimal error rate of err-RF is obtained when the observed FDR is ∼0.6, meaning that 60% of the selected variables are false positives, while, as expected, err-TRT selects fewer variables. Unlike the other methods, err-RF and err-TRT do not clearly highlight a threshold on the ranking. The error rate does not seem to be affected much by the introduction of irrelevant variables, resulting in rather flat curves.

The different methods generate similar curves when applied on a hypercube dataset (Supplementary Fig. S5) and when linear SVM is used as ranking procedure instead of RFs (Supplementary Figs. S6 and S7).

Precision and recall of the methods Figure 3 shows the precision and the recall of the methods on linear datasets, for different numbers of irrelevant variables. The RFs algorithm was used as ranking procedure and a significance level $\alpha = 0.05$ was chosen (we used this significance level in all our experiments). Precision and recall values are averaged over 50 datasets.

When the number of irrelevant variables increases, the recall of each method decreases. As already observed from the curves, CER, nFDR, mr-test and mProbes are rather conservative. The precision of these methods remains always almost at its highest value and their recall never reaches the recall $r_{\max}$ of the prec-1 method. On the other side, eFDR and 1Probe trade some precision, which remains nevertheless high, for a recall that is higher and close to $r_{\max}$. err-RF and err-TRT obtain the highest recall values but also the lowest precision levels. Moreover, these precision levels clearly decrease when the number of irrelevant variables increases. err-TRT tends to select fewer variables than err-RF and has therefore a higher precision. Similar results are observed on hypercube datasets (Supplementary Fig. S8) and when SVM is used as ranking procedure (Supplementary Figs. S9 and S10).

When we increase the number of instances in the learning samples, the recall of all the methods increases, as well as the precision of err-RF/SVM and err-TRT (Figs. 4, Supplementary S11–S13). We again observe three families of methods: those having a high precision and a recall lower than $r_{\max}$ (CER, nFDR, mr-test and mProbes), those having a high precision and a recall close to $r_{\max}$ (eFDR and 1Probe), and those with a lower precision and a recall higher than $r_{\max}$ (err-$\mathcal{A}$ and err-TRT).

Univariate rankings Figures 2b and 5 show, respectively, the score curves and precision/recall values of each method, when the relevance score of a variable is the absolute value of the statistic t computed by a t-test. All the results are similar to those obtained with a multivariate ranking method except for the nFDR which, when used with a t-test, provides a much better estimation of the

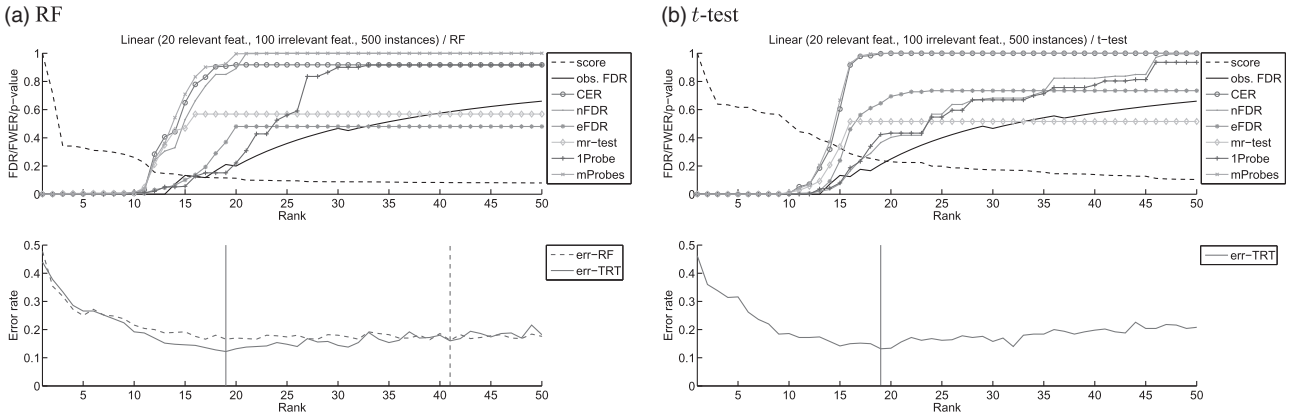

Fig. 2. Curves of the different methods on a linear dataset, with the RFs (left) and the t-test (right) as ranking method. *Score* is the relevance score derived from the RFs (left) or the absolute value of the statistic t derived from the t-test (right). *Obs. FDR* is the observed FDR. The dashed blue (resp. plain red) vertical line indicates the position of the lowest error rate for err-RF (resp. err-TRT)

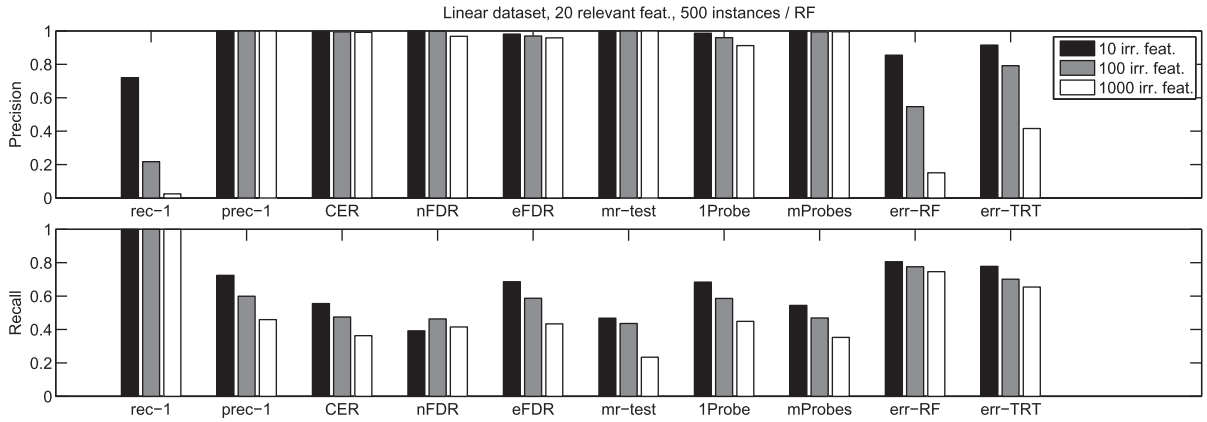

Fig. 3. Precision and recall on linear datasets, for different numbers of irrelevant features. We used the RFs algorithm as ranking method and $\alpha = 0.05$. The precision and recall values were averaged over 50 datasets in each case

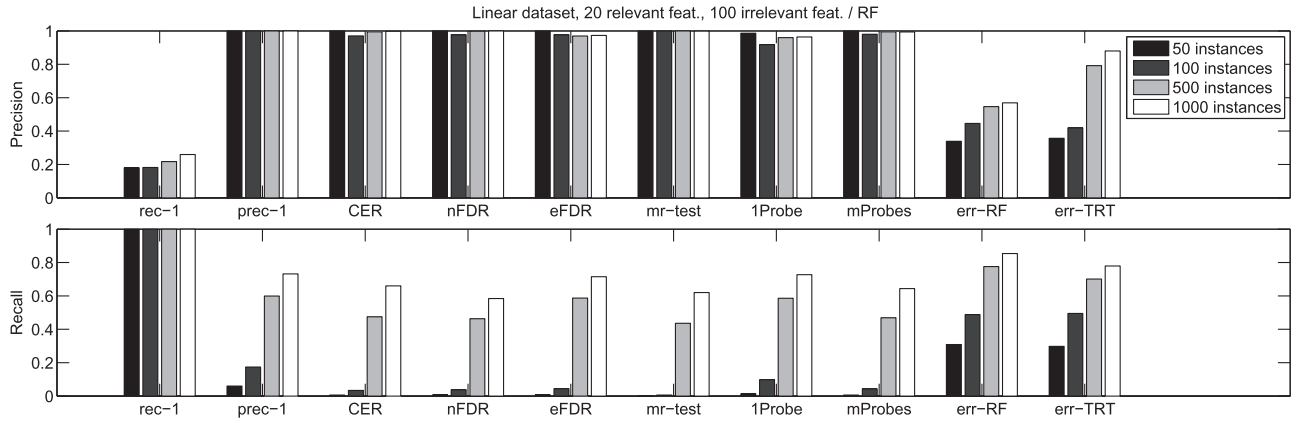

Fig. 4. Precision and recall on linear datasets, for different numbers of instances. We used the RFs algorithm as ranking method and $\alpha = 0.05$. The precision and recall values were averaged over 50 datasets in each case

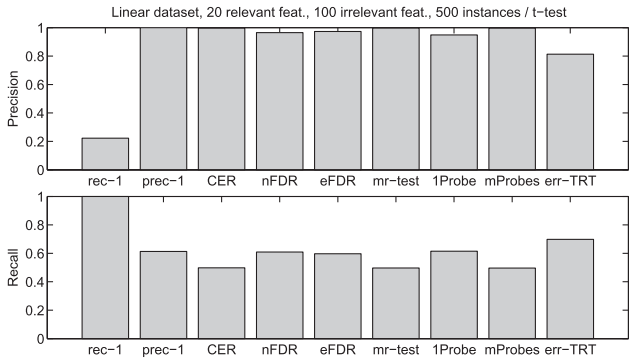

Fig. 5. Precision and recall on linear datasets when the t-test is the ranking method and $\alpha = 0.05$. The precision and recall values were averaged over 50 datasets

Table 3. Number of selected genes for the microarray datasets ($\alpha = 0.05$), using RFs as ranking procedure

	CER	nFDR	eFDR	mr-test	1Probe	mProbes	err-RF	err-TRT
Breast	0	0	0	0	0	0	30	110
Leukemia	36	82	368	16	62	63	51	4
Lymphoma1	6	33	94	0	49	18	208	42
Lymphoma2	0	0	0	0	3	0	33	104
Prostate1	58	73	391	18	54	91	5	3
Prostate2	63	131	456	14	62	53	67	28
Prostate1-perm.	0.0	0.1	0.1	0.0	1.4	0.0	29.3	20.8

real FDR (Fig. 2b) and has a recall equal to $r_{\max}$ while having a high precision (Fig. 5) .

5.2 Microarray datasets

The evaluation of feature selection techniques on real datasets is difficult as the truly relevant features are unknown on these problems, precluding the computation of any precision-recall values as we did on artificial problems. The purpose of our experiments in this section is thus only to illustrate the behaviour of the different methods on real microarray datasets and to spot any difference with respect to artificial problems.

Table 3 shows the number of genes selected by each method on the six microarray datasets (see Section 4.2), using RFs as the feature ranking method. To highlight the behaviour of the methods on a problem where none of the variables is truly relevant, the last row was obtained by averaging the number of genes selected by each method over 50 new datasets, each one obtained by randomly permuting the output labels of the Prostate1 dataset, while leaving the input features unchanged. Similar experiments with SVM and t-test are reported in Tables S1 and S2 of Supplementary material.

A first interesting observation is that the number of selected features is very problem-dependent. There are almost no features selected on Breast and Lymphoma2, few on Lymphoma1 and a significant number on the other datasets. There seems to be no correlation between the number of selected features and problem size. For example, the Leukemia and Lymphoma2 datasets contain about the same number of features and samples, while there are no feature selected on Lymphoma2 and several on Leukemia.

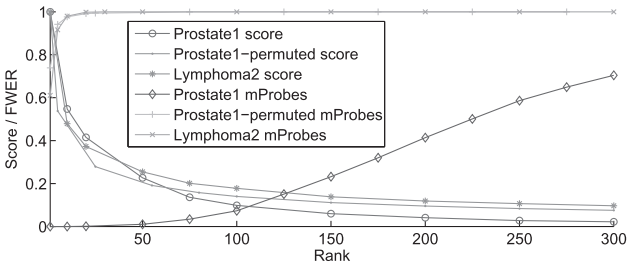

Fig. 6. RFs importances scores and mProbes FWER estimates on the Prostate1, Lymphoma2 and Prostate1-permuted datasets

As expected, the number of selected genes is close to 0 on the permuted Prostate1 dataset with all methods except err-RF and err-TRT. These observations thus suggest that all the evaluated feature selection methods, except those based on prediction performance, can adapt to the problem and ranking quality.

To further illustrate the interpretability of the proposed measures on microarray datasets, Figure 6 shows RFs importance scores and mProbes FWER estimates for increasing rank on three datasets: Prostate1, Lymphoma2 and Prostate1-permuted. As was observed in Figure 2a on artificial problems, RFs importance scores, which are very similar on the three datasets, do not suggest any clear threshold. The mProbes method on the other hand shows that about 100 features are relevant on the Prostate1 dataset and that there is no significant feature found by the RF method on the Lymphoma2 and Prostate1-permuted datasets.

Comparing the different feature selection methods, we observe that the eFDR method leads to selections of large subsets of genes, whereas the other methods are more conservative. Compared with the artificial problems, the largest subsets are no longer obtained by err-$\mathcal{A}$ and err-TRT. These two methods select relatively small numbers of genes because an error rate close to zero is typically achieved by many subsets of genes (see Supplementary Fig. S14). This can be explained by the low number of instances compared with the number of genes and the fact that the error rate was estimated on instances that were also used to compute the gene ranking (the so-called selection bias, Ambroise and McLachlan, 2002). Another difference compared with the results obtained on the artificial data is the fact that the 1Probe procedure appears to be more stringent here. One potential explanation for this difference is that this method corrects for multiple tests by using the Benjamini Hochberg procedure, which might be more conservative than the permutation-based correction embedded in the other methods.

For each problem, the number of genes selected by one method is also very much dependent on the chosen ranking procedure. For most methods, using linear SVM leads to the selection of very small subsets of genes (see Supplementary Table S1) whereas much larger subsets are selected with the t-test (see Supplementary Table S2).

Given a feature selection method and a ranking algorithm $\mathcal{A}$, the number of selected genes can also vary depending on the tuning of the parameters of $\mathcal{A}$. As an example, one parameter of RFs is the number T of trees that are grown in an ensemble. Increasing T from 1000 to 10 000 results in larger subsets of selected genes for all the methods except err-RF and err-TRT (Table 4). Huynh-Thu *et al.* (2008) already observed this phenomenon for the nFDR and the CER. Due to the very small sample to dimension ratio, the random trees, and thus the corresponding rankings, are highly unstable.

Table 4. Number of selected genes for the Prostate1 dataset ($\alpha = 0.05$), using the RFs as ranking procedure

T	CER	nFDR	eFDR	mr-test	1Probe	mProbes	err-RF	err-TRT
1000	58	73	391	18	54	91	5	3
10 000	136	88	668	193	444	215	4	3

T is the number of grown trees in a RF ensemble.

Averaging a very large number of trees results in a stabilization and an improvement of the feature ranking, and thus the possibility to select more variables without including any false positive. However, in spite of this improvement, err-RF and err-TRT do not select more genes, again suggesting that the error rate is not a relevant criterion to assess the quality of subsets of variables.

6 DISCUSSION

In this article, we evaluated several procedures that aim to identify a maximal subset of variables that truly provide some information about an output variable. These procedures assume that a (multivariate) ranking method $\mathcal{A}$ was first used to compute a relevance score for each variable of the considered problem and then extract relevant features from this ranking, by replacing the original relevance score with a measure that can be interpreted in a statistical way. Depending on the procedure, this measure is either the generalization error of a predictive model (err-$\mathcal{A}$ and err-TRT), the FDR (nFDR, eFDR), the FWER (CER, mProbes) or a p-value (mr-test, 1Probe). Although there is still a need to determine a threshold on this new measure, the determination of this threshold is clearly easier due to its interpretability. This threshold is also not dependent anymore on the problem at hand and on the ranking method $\mathcal{A}$.

Among the feature selection methods that we evaluated, err-$\mathcal{A}$ and err-TRT are the only ones that do not require to choose a significance level *a priori*. However, on artificial problems, they always have the lowest precision among all methods and they wrongly select a non-negligible number of variables on the permuted Prostate1 datasets. Moreover, they are subject to the selection bias problem, preventing the selection of an adequate number of variables. Prediction performance is thus clearly not an appropriate measure for the identification of relevant features.

Although they clearly highlight a threshold on the feature ranking, several of the remaining methods have also some disadvantages. The nFDR method is the simplest one but was shown to overestimate the real FDR in the case of dependent scores (Huynh-Thu *et al.*, 2008). The drawback of the 1Probe procedure is that the selected variables depend on the chosen distribution of the random probe, which makes it a parametric method. The mr-test method estimates the rank distribution of an irrelevant variable from the k variables with the highest observed ranks. The determination of k thus introduces some dependency on the problem and ranking method used. Although our default choice seems to be robust, an inappropriate value of k can lead to a dramatic over- or underestimation of the p-values. The mr-test also includes as a second parameter the number of resampled instances at each iteration.

Among the three remaining methods, CER and mProbes are highly selective methods that avoid the inclusion of any irrelevant feature as much as possible. mProbes has a computational advantage over the CER method while this latter has a nice interpretation when the scores are derived from univariate scores. Finally, the eFDR method is less stringent and trades some precision for a higher recall. The choice between a more or less conservative method clearly depends on the application and, as these three methods all have a very high precision on the artificial data, our advice is thus to use mProbes or CER when a very stringent method is needed (i.e. a very low false positive rate), and eFDR otherwise.

Obvious future works include the application of the feature selection methods to other popular machine learning-based ranking methods such as for example Relief (Robnik-Sikonja and Kononenko, 2003). In this article, we performed an empirical evaluation of the different methods that showed their practical utility. As future work, it would be interesting also to better characterize the different methods from a theoretical point of view. This is however not an easy task given the limited theory that exists about some machine learning-based feature ranking methods.

Our experiments on the microarray datasets highlighted that the number of selected variables depends strongly on the ranking method and the precise values of its parameters (e.g. the number of trees in RFs). We believe that this number could provide a valid criterion along which to assess and compare different ranking algorithms, which could be used as a replacement for predictive performance. Indeed, a higher number of variables with a low FDR or FWER indicates that it is more unlikely that an irrelevant variable reaches the very top of the feature ranking, and hence that the ranking is more reliable. In the future, we plan to explore further the use of the number of selected variables to tune the parameters (such as, e.g. the number T of ensemble terms in the RF models) of existing methods or even to design new ranking algorithms that would explicitly try to optimize the feature scores defined by the different feature selection methods.

Recently, a great interest has raised for the analysis of the stability of feature ranking and selection methods (Abeel *et al.*, 2010; He and Yu, 2010). The rationale behind this analysis is that a good method should lead to the selection of (nearly) identical features when small changes are made to the dataset. We believe that it would be of interest to confront this kind of stability analysis with the approaches presented in the present article. Indeed, unstable feature importance scores are very likely to lead to high FDR/FWER estimates because of the increased chance of an irrelevant feature to get a high importance. On the other hand, since stability is not a sufficient condition for relevant feature selection, the FDR/FWER measures are intrinsically complementary to the stability criterion.

Finally, in this article, we focused on the problem of finding all the relevant variables from a ranking, i.e. potentially including features that contain redundant information about the output. In some applications, it would be more interesting to identify a *minimal* subset of relevant variables, such that no other variable conveys complementary information about the target conditionally to these variables (the so-called *Markov boundary*; Pearl, 1988). The adaptation of our procedures to solve this problem would be an interesting direction of future research.

ACKNOWLEDGEMENTS

The authors thank the GIGA Bioinformatics platform and the SEGI for providing computing resources, as well as the anonymous reviewers for their valuable suggestions and remarks.

Funding: Belgian Federal Science Policy Office (IAP P6/25 BIOMAGNET); French Community of Belgium (ARC Biomod); and European Network of Excellence PASCAL2. V.A.H.T. is recipient of a F.R.I.A. fellowship, Y.S. is a Postdoctoral Fellow of the Research Foundation-Flanders (FWO), and P.G. is a research associate of the F.N.R.S.

Conflict of Interest: none declared.

REFERENCES

Abeel,T. *et al.* (2010) Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, **26**, 392–398.

Alizadeh,A. *et al.* (2000) Distinct types of di?use large B-cell lymphoma identiþed by gene expression proþling. *Nature*, **403**, 503–511.

Altmann,A. *et al.* (2010) Permutation importance: a corrected feature importance measure. *Bioinformatics*, **26**, 1340–1347.

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

Benjamini,Y. and Hochberg,Y. (1995) Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J. Roy. Stat. Soci., Ser. B (Methodol.)*, **57**, 289–300.

Boser,B.E. *et al.* (1992) A training algorithm for optimal margin classifiers. In Haussler, D. (ed.), *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, Pittsburgh, ACM Press, pp. 144–152.

Breiman,L. (2001) Random forests. *Mach. Learn.*, **45**, 5–32.

Chang,C.-C. and Lin,C.-J. (2011) Libsvm: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, **2**, 27:1–27:27.

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

Fukunaga,K. and Hostetler,L. (1975) *k*-Nearest-neighbor Bayes-risk estimation. *IEEE Trans. Inform. Theory*, **21**, 285–293.

Ge,Y. *et al.* (2003) Resampling-based multiple testing for microarray data analysis. Technical Report 633, Department of Statistics, University of California, Berkeley.

Ge, Y. *et al.* (2008) Some step-down procedures controlling the false discovery rate under dependence. *Stati. Sin.*, **18**, 881–904.

Geurts,P. *et al.* (2005) Proteomic mass spectra classification using decision tree based ensemble methods. *Bioinformatics*, **21**, 3138–3145.

Geurts,P. *et al.* (2006) Extremely randomized trees. *Mach. Learn.*, **36**, 3–42.

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

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

Hastie,T. *et al.* (2003) *The Elements of Statistical Learning*. Springer. New York.

He,Z. and Yu,W. (2010) Stable feature selection for biomarker discovery. *Computational Biology and Chemistry*, **34**, 215–225.

Huynh-Thu,V.A. *et al.* (2008) Exploiting tree-based variable importances to selectively identify relevant variables. *JMLR: Workshop and Conference proceedings*, Antwerp, **4**, 60–73.

Pearl,J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Rakotomamonjy,A. (2003) Variable selection using svm based criteria. *J. Mach. Learn. Res.*, **3**, 1357–1370.

Robnik-Sikonja,M. and Kononenko,I. (2003) Theoretical and empirical analysis of ReliefF and RReliefF. *Mach. Learn. J.*, **53**, 23–69.

Saeys,Y. *et al.* (2007) A review of feature selection techniques in bioinformatics. *Bioinformatics*, **23**, 2507–2517.

Shipp,M. *et al.* (2002) Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nat. Med.*, **8**, 68–74.

Singh,D. *et al.* (2002) Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, **1**, 203–209.

Smialowski,P. *et al.* (2010) Pitfalls of supervised feature selection. *Bioinformatics*, **26**, 440–443.

Stoppiglia,H. *et al.* (2003) Ranking a random feature for variable and feature selection. *J. Mach. Learn. Res.*, **3**, 1399–1414.

Storey,J.D. and Tibshirani,R. (2003) Statistical significance for genomewide studies. *Proc. Natl. Acad. Sci. USA*, **100**, 9440–9445.

Tuv,E. *et al.* (2009) Feature selection with ensembles, artificial variables, and redundancy elimination. *J. Mach. Learn. Res.*, **10**, 1341–1366.

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

Westfall,P.H. and Young,S.S. (1993) *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*. John Wiley & Sons, New York.

Zhang,C. *et al.* (2006) Significance of gene ranking for classification of microarray samples. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **3**, 1–9.