

## Data and text mining

## Genetic test bed for feature selection

Ashish Choudhary<sup>1</sup>, Marcel Brun<sup>2</sup>, Jianping Hua<sup>2</sup>, James Lowey<sup>2</sup>, Ed Suh<sup>2</sup> and Edward R. Dougherty<sup>1,2,\*</sup><sup>1</sup>Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA and <sup>2</sup>TGen, 445 North Fifth Street, Phoenix, AZ 85004, USA

Received on September 30, 2005; revised on January 12, 2006; accepted on January 13, 2006

Advance Access publication January 20, 2006

Associate Editor: Satoru Miyano

## ABSTRACT

**Motivation:** Given a large set of potential features, such as the set of all gene-expression values from a microarray, it is necessary to find a small subset with which to classify. The task of finding an optimal feature set of a given size is inherently combinatoric because to assure optimality all feature sets of a given size must be checked. Thus, numerous suboptimal feature-selection algorithms have been proposed. There are strong impediments to evaluate feature-selection algorithms using real data when data are limited, a common situation in genetic classification. The difficulty is compound. First, there are no class-conditional distributions from which to draw data points, only a single small labeled sample. Second, there are no test data with which to estimate the feature-set errors, and one must depend on a training-data-based error estimator. Finally, there is no optimal feature set with which to compare the feature sets found by the algorithms.

**Results:** This paper describes a genetic test bed for the evaluation of feature-selection algorithms. It begins with a large biological feature-label dataset that is used as an empirical distribution and, using massively parallel computation, finds the top feature sets of various sizes based on a given sample size and classification rule. The user can draw random samples from the data, apply a proposed algorithm, and evaluate the proficiency of the proposed algorithm via three different measures (code provided). A key feature of the test bed is that, once a dataset is input, a single command creates the entire test bed relative to the dataset. The particular dataset used for the first version of the test bed comes from a microarray-based classification study that analyzes a large number of microarrays, prepared with RNA from breast tumor samples from each of 295 patients.

**Availability:** The software and supplementary material are available at <http://public.tgen.org/tgen-cb/support/testbed/>

**Contact:** edward@ece.tamu.edu

## 1 INTRODUCTION

Given a large set of potential features, such as the set of all gene-expression values from a microarray, it is necessary to find a small subset with which to classify. The problem is statistically inherent in classification because typically the true error of a designed classifier will fall with use of more features and, after some optimal number of features for a given sample size, begin to rise, this being called the peaking phenomenon (Hughes, 1968; Kanal and Chandrasekaran,

1971; Jain and Chandrasekaran, 1982; Hua *et al.*, 2005a,b). For small samples the optimal number can be very small, where by ‘sample’ we refer to the set of data points, each data point corresponding to a microarray. Empirical studies show that the rise may be gradual; nonetheless, accuracy is lost by using too many features. Moreover, there are important cases in which the rise is dramatic. The task of finding an optimal feature set is inherently combinatoric. According to a classical theorem, to be assured of finding the optimal feature set of a given size, all feature subsets of that size must be checked unless there is distributional knowledge that mitigates the search requirement, a mitigating condition not occurring in practice (Cover and Van Campenhout, 1977).

There are various methods of choosing feature sets, each having advantages and disadvantages. The typical intent is to choose a set of variables that provides good classification. When there is a large number of potential random variables for classification, feature selection is problematic and the best method to use depends on the circumstances. Evaluation of methods is generally comparative and based on simulations (Jain and Zongker, 1997; Kudo and Sklansky, 2000). When used, a feature-selection algorithm is part of the classification rule. Feature selection yields classifier constraint, not a reduction in the dimensionality of the feature space relative to design. For instance, if there are  $D$  features altogether from which to choose and we have decided to use  $k$  of them to form a classifier, then all  $D$  features are available to form the classifier, but the classifier is limited to a function of  $k$  variables.

To carry out an informative simulation for small-sample classification, one would draw small samples from two class-conditional distributions, apply a feature selection algorithm to find good feature sets of a given size for each sample, estimate the errors of the chosen feature sets using independent test data, and then average the resulting errors. Feature-selection algorithms could then be compared relative to the average errors and conclusions could be drawn relative to the classification rule and class-conditional distributions employed. Furthermore, if the optimal feature set of the given size were known, say from the model or from an exhaustive search, then one could compare the average-error performance of a feature-selection algorithm with the error of the optimal feature set, relative to the sample size and the class-conditional distributions.

There are strong impediments to evaluate feature-selection algorithms using real data when data are severely limited, a common situation in genomic classification, where one is often limited to a sample from a small number of tissues. The difficulty is

\*To whom correspondence should be addressed.

compound. First, there are no class-conditional distributions from which to draw a sample, only a single small labeled sample. Second, there are no test data with which to estimate the feature-set errors, and one must depend on a training-data-based error estimator. Finally, there is no optimal feature set with which to compare the feature sets found by the algorithms, because any reasonable estimate of the optimal feature set depends on having a large sample and, even if one has a large sample, testing all feature sets of even modest sizes is impossible in an ordinary computing environment. Given these impediments, it is understandable why proposed feature-selection methods are generally not convincingly tested on real data. A common approach is simply to apply the method to data from a real sample and then using the training data to estimate the error of the feature set yielded by the algorithm. This may be done for several different datasets, with the results compared with other feature-selection algorithms on the same datasets. In many cases, cross-validation techniques are used for error estimation, and these are extremely unreliable for small samples owing to their high variances (Devroye *et al.*, 1996; Braga-Neto and Dougherty, 2004a). In particular, cross-validation yields very inaccurate feature-set ranking with small samples, and although bootstrap (Efron, 1983) and bolstering (Braga-Neto and Dougherty, 2004b) provide substantial improvement, even they are problematic for very small samples (Sima *et al.*, 2005a). Very small samples are ubiquitous in microarray-based classification studies. For instance, the following sample sizes for cancer studies are indicative of the small samples encountered: Cutaneous Melanoma, 31 (Bittner *et al.*, 2000); Leukemia, 37 (Armstrong *et al.*, 2002); Acute Leukemia, 38 (Golub *et al.*, 1999); Breast Cancer, 38 (West *et al.*, 2001); Follicular Lymphoma, 24 (Bohen *et al.*, 2003); Ovarian Carcinoma, 44 (Schaner *et al.*, 2003); Uveal Melanoma, 20 (Tschentscher *et al.*, 2003); Lymphoma, 47 (Li *et al.*, 2001) and Glioma, 25 (Kim *et al.*, 2002).

The inability to evaluate the performance of feature-selection algorithms on real data means that a host of algorithms have been proposed, based for the most part on heuristic reasoning, and their comparative performances have not been assessed on datasets of interest to genomics—for instance, data arising from different kinds of cancer.

This paper describes a genetic test bed for the evaluation of feature-selection algorithms. Use of the test bed is open to the community with all necessary data and algorithms being available on a website supported by the Translational Genomics Research Institute.

## 2 SYSTEMS AND METHODS

The test bed begins with a set  $F$  of  $D$  potential features and a large labeled (0 or 1) dataset  $S$  of size  $N$  to serve as an empirical distribution (population). This means that  $S$  consists of  $N$  vectors of length  $D$ , where each feature corresponds to a vector component. If we are concerned with feature sets of size  $k$ , then there are  $C(D, k) = \binom{D}{k}$  feature sets,  $F_1, F_2, \dots, F_{C(D, k)}$ . For a given classification rule  $\mathcal{R}$ , we apply  $\mathcal{R}$  to each of the  $C(D, k)$  feature sets using the data of the empirical distribution  $S$  to obtain  $C(D, k)$  classifiers,  $f_1, f_2, \dots, f_{C(D, k)}$ .

To rank the feature sets, we must estimate the classifier errors using the data of  $S$  to obtain errors  $\epsilon_1, \epsilon_2, \dots, \epsilon_{C(D, k)}$  corresponding to the feature sets  $F_1, F_2, \dots, F_{C(D, k)}$ , respectively, the error  $\epsilon_i$  being the misclassification rate of the classifier  $f_i$  on the empirical distribution. Since  $S$  is large, we can obtain good error estimation, especially using either bolstering or bootstrap

(Braga-Neto and Dougherty, 2004a,b). Bootstrap is much too slow for the exhaustive simulations being performed. Thus, based on the prior studies, we use bolstered or semi-bolstered resubstitution, depending on the classification rule. This accords well with the superior feature-set ranking that has been observed for bolstered error estimators (Sima *et al.*, 2005a) and its superior performance used within feature-selection algorithms such as sequential floating forward search (Sima *et al.*, 2005b).

Following error estimation, we rank the feature sets based on the errors, from lowest to highest error. This list gives the ground-truth ranking of the feature sets of size  $k$  for classification rule  $\mathcal{R}$  and the empirical distribution  $S$ . It is important to note that the true ranking of the feature sets depends on the distribution  $S$  and that is why we are interested in the misclassification rates on  $S$ .

## 3 ALGORITHM

To evaluate a feature-selection algorithm  $\mathcal{A}$ , we draw  $m$  samples,  $S_1, S_2, \dots, S_m$ , of size  $n$  (much smaller than  $N$ ) from  $S$  and apply the algorithm to each sample to obtain  $m$  feature sets  $G_1, G_2, \dots, G_m$ . Keeping in mind that we are interested in the quality of the feature sets, we apply the classification rule  $\mathcal{R}$  to each of the  $m$  feature sets using the data of the empirical distribution  $S$  to obtain  $m$  classifiers,  $g_1, g_2, \dots, g_m$ , and then use bolstered or semi-bolstered error estimation to estimate the classifier errors using the data of  $S$  to obtain errors  $e_1, e_2, \dots, e_m$  corresponding to the feature sets  $G_1, G_2, \dots, G_m$  respectively. Note that, although the feature set  $G_j$  is derived from the (small) sample  $S_j$  via some algorithm, its error is evaluated relative to its performance on the empirical distribution because the intent of a feature selection algorithm is to find good population features.

We have emphasized the smallness of the samples mainly because we are interested in the performance of feature-selection algorithms on the small samples so common to genomics, but there is also a statistical reason for being concerned with sample size. If one samples the empirical distribution with replacement, then the samples are independent; however, if one samples without replacement, as we do, then the samples must be kept small to mitigate the effects of dependence.

Using the results of feature selection on the  $m$  samples, we can employ a number of measures of the effectiveness of a feature-selection algorithm. The first two we define are directly related to the error rates. First, we can obtain the average error increase,  $\delta$ . For the error of the feature set found by the algorithm as compared with the error of the optimal feature set,

$$\delta(\mathcal{R}, \mathcal{A}; n, k) = \frac{1}{m} \sum_{j=1}^m e_j(\mathcal{R}, \mathcal{A}; n, k) - \epsilon_{\text{opt}}(\mathcal{R}, \mathcal{A}; k), \quad (1)$$

where  $\epsilon_{\text{opt}}(\mathcal{R}, \mathcal{A}; k)$  and  $e_j(\mathcal{R}, \mathcal{A}; n, k)$  are the errors for the optimal feature set of size  $k$  and for the feature set of size  $k$  selected by algorithm  $\mathcal{A}$  based on the sample  $S_j$  of size  $n$ .

Assuming  $\epsilon_{\text{opt}}(\mathcal{R}, \mathcal{A}; k) \neq 0$ , a second measure,  $\nu$ , provides the average proportional increase in error of the feature set found by the algorithm compared with, the error of the optimal feature set,

$$\nu(\mathcal{R}, \mathcal{A}; n, k) = \frac{1}{m} \sum_{j=1}^m \frac{e_j(\mathcal{R}, \mathcal{A}; n, k)}{\epsilon_{\text{opt}}(\mathcal{R}, \mathcal{A}; k)}. \quad (2)$$

If we are interested in feature (gene) discovery, then another statistic of interest besides error comparison might be the average number of

features in the optimal feature set also in the feature set found by the algorithm. We denote this average by  $\tau(\mathcal{R}, \mathcal{A}; n, k)$ .

To apply the test bed for a particular feature-selection algorithm for feature sets of size  $k$  and for a particular classification rule, the following steps should be followed:

- Download the sample data.
- Download the best feature-set list for feature sets of size  $k$  and the classification rule being considered.
- Download and compile the program necessary to compute the error of the classification rule on the sample data (see the download section of the web page).
- Randomly select  $m$  (at least 100) samples of size  $n$  from the sample data.
- Apply the proposed algorithm to the samples to obtain  $m$  feature sets of size  $k$ .
- Use the previously compiled program to evaluate the feature sets for the classification rule being considered (see the software section of the web page).

The program will do the following:

- (1) Apply the classification rule using the provided code to obtain the corresponding classifiers.
- (2) Compute the errors for the classifiers (feature sets) using the full set of sample data.
- (3) Compute the measures  $\delta$ ,  $\nu$  and  $\tau$  using the provided code.
- (4) Display the resulting error rates and validation measures.

## 4 IMPLEMENTATION

The test bed is implemented in practice by choosing a dataset of sufficient size to serve as the empirical distribution. An important practical feature of the test bed is that, once a dataset is input, a single command creates the entire test bed relative to the dataset. The particular dataset we are using for the first version of the test bed comes from a microarray-based classification study that analyzes a large number of microarrays, prepared with RNA from breast tumor samples from each of the 295 patients (van 't Veer *et al.*, 2002). Using a previously established 70-gene prognosis profile (van de Vijver *et al.*, 2002), a prognosis signature based on gene-expression is proposed in van 't Veer *et al.* (2002) that correlates well with patient survival data and other existing clinical measures. Of the 295 microarrays, 115 belong to the 'good-prognosis' class (label 1) and the remaining 180 belong to the 'poor-prognosis' class (label 0). Each data point is a 70-expression vector corresponding to a single microarray, with expression values being log-intensity.

For a given classification rule, the test bed provides the user with lists of the top feature sets, along with their errors, for feature sets of size 2, 3, ..., 7. For each  $k$ , the list has been found by checking all  $\binom{70}{k}$  feature sets. Owing to the extremely large numbers of feature sets for large  $k$ , the calculations have been performed on a massively parallel Beowulf cluster using 64 nodes (128 CPUs). For  $k = 6$  and  $k = 7$ , run times exceeded 34 and 337 h, respectively.

Lists are provided for the following classification rules: Linear Discriminant Analysis (LDA), 3-Nearest-Neighbor (3NN), 5-Nearest-Neighbor (5NN) and Decision Trees (CART).

Comprehensive descriptions of the rules are given on the companion website, along with the source code for each classification rule. Note that it is necessary to employ the provided source code when applying a classification rule to a selected feature set to be sure that the resulting error is compatible with the feature-set list provided. Regarding error estimation on  $S$ , we apply bolstered resubstitution for LDA and CART, and semi-bolstered resubstitution for 3NN and 5NN (Braga-Neto and Dougherty, 2004b). Bolstering relies on noise injection for error estimation, and to reduce drastically the internal variability of the estimator, we have used 10 000 points noise injection, except for LDA, where the estimation is analytical and does not need noise injection. Hence, the estimation has negligible internal variance.

There are links to each individual project from the main page. Under the project page the labeled sample data are available as raw/tab delimited formats along with the parameter file (Fig. 1). The best feature-set list for each rule and each  $k$  is organized as a table.

Also found under the main page are links to pages that explain file formats, the user roadmap and other relevant information (Fig. 1). In particular, there is a link to the software page that describes the available code, detailed instructions to compiling and running, and a list of input and expected output files.

Table 1 shows the errors for the optimal feature sets of sizes 2–7, and the estimated internal standard deviation of the bolstered error estimation, based on the injection of 10 000 noise points. In most of the cases the difference in error between the optimal feature set and the second ranked feature set is >10 times the internal standard deviation (results not shown here).

## 5 DISCUSSION

### 5.1 Illustration

To illustrate the use of the test bed, we consider the performance of the popular sequential floating forward search (SFFS) algorithm (Pudil *et al.*, 1994). We denote a feature selection algorithm as  $A_{\text{error}}^{\text{class}}$ , where the superscript denotes the classification rule and the subscript the error estimator used for implementation of the SFFS algorithm, e.g.  $A_{\text{LOO}}^{3\text{NN}}$  and  $A_{\text{RESUB}}^{\text{CART}}$  denote the feature selection process with 3NN classification and leave-one-out (LOO) error estimation, and CART classification and resubstitution (RESUB), respectively. We draw  $m = 100$  samples, each of size  $n = 35$  out of  $N = 295$ , and apply a particular algorithm  $A_{\text{error}}^{\text{class}}$  to each of the sample sets  $S_1, S_2, \dots, S_{100}$  to obtain feature sets  $G_1, G_2, \dots, G_{100}$  each of size  $k = 4$ .

To test the performance of the features selected by  $A_{\text{error}}^{\text{class}}$  for a particular classifier type (not necessarily the one used in the feature-selection algorithm), we evaluate the errors  $e_1, e_2, \dots, e_{100}$  using bolstered or semi-bolstered resubstitution (depending on the classifier type) on the entire set  $S$  using the code on the website. For instance, for  $k = 4$  and the LDA classifier, the error found by a comprehensive search (and listed in the top line on the file 4FLDA.txt on the website) is  $\epsilon_{\text{opt}} = 0.15986$  for feature vector [7, 42, 48, 59]. We can now calculate  $\delta$ ,  $\nu$  and  $\tau$  by the formulae in Section 3. The code provided on the web page takes the list of feature sets  $G_1, G_2, \dots, G_{100}$  (more format details are available on the web page), and the file with optimal feature sets as an input to generate the measures  $\delta$ ,  $\nu$ , and  $\tau$ . The program calculates the errors for  $G_1, G_2, \dots, G_{100}$  using the same functions used to do

[Main Page](#)

**Information**

[Description](#)

[Files format](#)

[Road Map](#)

[Disclaimer](#)

[Software](#)

[References](#)

**Projects**

[Breast Cancer](#)

## Source and Result Files

Raw Text Data	[ <a href="#">Format</a> ]	[ <a href="#">Txt</a> ] 136 KB [ <a href="#">Txt.gz</a> ] 41 KB
Tab Delimited Text Data	[ <a href="#">Format</a> ]	[ <a href="#">Txt</a> ] 139 KB [ <a href="#">Txt.gz</a> ] 42 KB
Parameters	[ <a href="#">Format</a> ]	[ <a href="#">Txt</a> ] 14 Bytes [ <a href="#">Txt.gz</a> ] 46 Bytes

[top](#)

### Best features sets for each rule and number of features [ [Format](#) ]

Rule	Features: 2	Features: 3	Features: 4	Features: 5	Features: 6	Features: 7
3NN	[ <a href="#">Txt</a> ] 39 KB [ <a href="#">Txt.gz</a> ] 13 KB	[ <a href="#">Txt</a> ] 95 KB [ <a href="#">Txt.gz</a> ] 28 KB	[ <a href="#">Txt</a> ] 109 KB [ <a href="#">Txt.gz</a> ] 29 KB	[ <a href="#">Txt</a> ] 121 KB [ <a href="#">Txt.gz</a> ] 29 KB	[ <a href="#">Txt</a> ] 136 KB [ <a href="#">Txt.gz</a> ] 31 KB	[ <a href="#">Txt</a> ] 150 KB [ <a href="#">Txt.gz</a> ] 33 KB
5NN	[ <a href="#">Txt</a> ] 39 KB [ <a href="#">Txt.gz</a> ] 13 KB	[ <a href="#">Txt</a> ] 95 KB [ <a href="#">Txt.gz</a> ] 28 KB	[ <a href="#">Txt</a> ] 109 KB [ <a href="#">Txt.gz</a> ] 29 KB	[ <a href="#">Txt</a> ] 124 KB [ <a href="#">Txt.gz</a> ] 30 KB	[ <a href="#">Txt</a> ] 135 KB [ <a href="#">Txt.gz</a> ] 31 KB	[ <a href="#">Txt</a> ] 149 KB [ <a href="#">Txt.gz</a> ] 32 KB
CART	[ <a href="#">Txt</a> ] 39 KB [ <a href="#">Txt.gz</a> ] 13 KB	[ <a href="#">Txt</a> ] 94 KB [ <a href="#">Txt.gz</a> ] 28 KB	[ <a href="#">Txt</a> ] 107 KB [ <a href="#">Txt.gz</a> ] 29 KB	[ <a href="#">Txt</a> ] 119 KB [ <a href="#">Txt.gz</a> ] 29 KB	[ <a href="#">Txt</a> ] 137 KB [ <a href="#">Txt.gz</a> ] 32 KB	[ <a href="#">Txt</a> ] 150 KB [ <a href="#">Txt.gz</a> ] 34 KB
LDA	[ <a href="#">Txt</a> ] 39 KB [ <a href="#">Txt.gz</a> ] 13 KB	[ <a href="#">Txt</a> ] 95 KB [ <a href="#">Txt.gz</a> ] 27 KB	[ <a href="#">Txt</a> ] 109 KB [ <a href="#">Txt.gz</a> ] 28 KB	[ <a href="#">Txt</a> ] 123 KB [ <a href="#">Txt.gz</a> ] 28 KB	[ <a href="#">Txt</a> ] 134 KB [ <a href="#">Txt.gz</a> ] 28 KB	[ <a href="#">Txt</a> ] 146 KB [ <a href="#">Txt.gz</a> ] 28 KB

**Fig. 1.** A section of the web page illustrating links. On the top left are the links to the detailed Description, File Formats, Road Map and References. On the bottom left corner are links to individual project pages (Currently Breast Cancer only, more to be added). Following the links to individual projects leads to detailed lists of optimal features of size  $k = 2$  and above.

**Table 1.** Minimum error and standard deviation (Bolstering error estimation for LDA is based in a exact analytical equation and has no internal variability)

Number of features	Classifier									
	3NN		5NN		CART		LDA			
	$\epsilon$	$\sigma$	$\epsilon$	$\sigma$	$\epsilon$	$\sigma$	$\epsilon$	$\sigma$	$\epsilon$	$\sigma$
2	0.19875	0.000128	0.19006	0.000114	0.20499	0.000173	0.19067	—	—	—
3	0.16129	0.000147	0.16157	0.000110	0.17147	0.000183	0.16944	—	—	—
4	0.13603	0.000122	0.14281	0.000123	0.14964	0.000182	0.15986	—	—	—
5	0.12576	0.000134	0.13464	0.000119	0.14970	0.000171	0.15755	—	—	—
6	0.11832	0.000143	0.12798	0.000135	0.15174	0.000174	0.15162	—	—	—
7	0.11365	0.000126	0.12533	0.000143	0.15212	0.000171	0.14949	—	—	—

the comprehensive search. This removes potential disparities being created by different implementations of the popular classifiers. Note that the functions and numerical techniques used by the user to obtain the feature sets are considered part of the user’s feature-selection algorithm.

This process has been carried out for all possible combinations of class and error in  $A_{error}^{class}$  with class  $\in \{LDA, 3NN, 5NN, CART\}$  and error  $\in \{LOO, RESUB\}$ . The performances of the obtained feature sets have been evaluated for the LDA, 3NN, 5NN and CART classifiers. The results are tabulated in Table 2. The process has been repeated for  $n = 50$  and these results are shown in Table 3. More such examples are available on the website.

Let us make a few remarks concerning the tables, where the optimal errors are for the best four-feature classifiers found by exhaustive search. We first note that comparing the errors for the optimal feature sets, 3NN performs the best, 5NN and CART are almost identical, and LDA performs the worst, albeit, not badly considering its simple form.

If we focus on the average error increment  $\delta$  in Table 2, then we see that, for 3NN,  $A_{RESUB}^{LDA}$  provides the best performance with  $\delta(3NN, A_{RESUB}^{LDA}; 35, 4) = 0.1149$ .  $A_{LOO}^{CART}$  performs the worst with  $\delta(3NN, A_{LOO}^{CART}; 35, 4) = 0.1440$ . One might have expected that 3NN would be the best classification rule to use with SFFS when selecting features for 3NN; one might not have expected that LDA would have performed as well when selecting features for 3NN. In fact,  $A_{RESUB}^{LDA}$  continues to outperform  $A_{RESUB}^{3NN}$  and  $A_{LOO}^{3NN}$  in Table 3, though it is outperformed by  $A_{LOO}^{LDA}$ , which provides the best performance with  $\delta(3NN, A_{LOO}^{LDA}; 50, 4) = 0.1128$ , outperforming both  $A_{LOO}^{3NN}$  and  $A_{RESUB}^{3NN}$  for selecting features for 3NN—at least for the genetic data considered herein. These kinds of seeming anomalies become even more apparent when looking at CART in Table 3. All of the non-CART SFFS algorithms outperform both CART SFFS algorithms.

Considering the statistic  $\tau$  in Table 2, we observe that for 3NN,  $A_{RESUB}^{3NN}$  outperforms all other considered SFFS algorithms when it comes to feature discovery. A similar phenomenon is seen

**Table 2.** Performance of SFFS algorithm with  $m = 100$ ,  $n = 35$  and  $k = 4$ 

Feature selection algorithm	Classifier											
	LDA ( $\epsilon = 0.1599$ )			3NN ( $\epsilon = 0.1360$ )			5NN ( $\epsilon = 0.1428$ )			CART ( $\epsilon = 0.1496$ )		
	$\delta$	$v$	$\tau$	$\delta$	$v$	$\tau$	$\delta$	$v$	$\tau$	$\delta$	$v$	$\tau$
$A_{RESUB}^{LDA}$	0.0893	1.5584	0.63	0.1149	1.8447	0.54	0.1166	1.8162	0.54	0.1097	1.7334	0.82
$A_{LOO}^{LDA}$	0.1001	1.6264	0.47	0.1223	1.8992	0.57	0.1231	1.8618	0.57	0.1161	1.7758	0.69
$A_{RESUB}^{3NN}$	0.1005	1.6286	0.34	0.1286	1.9452	0.52	0.1290	1.9036	0.52	0.1205	1.8056	0.56
$A_{LOO}^{3NN}$	0.1106	1.6919	0.29	0.1338	1.9836	0.34	0.1344	1.9414	0.34	0.1251	1.8363	0.52
$A_{RESUB}^{5NN}$	0.0983	1.6149	0.49	0.1272	1.9347	0.61	0.1283	1.8983	0.61	0.1194	1.7980	0.67
$A_{LOO}^{5NN}$	0.1066	1.6666	0.45	0.1333	1.9796	0.47	0.1346	1.9427	0.47	0.1237	1.8264	0.56
$A_{RESUB}^{CART}$	0.1163	1.7274	0.35	0.1379	2.0137	0.28	0.1422	1.9960	0.28	0.1310	1.8757	0.46
$A_{LOO}^{CART}$	0.1271	1.7949	0.23	0.1440	2.0583	0.24	0.1460	2.0226	0.24	0.1339	1.8947	0.57

**Table 3.** Performance of SFFS algorithm with  $m = 100$ ,  $n = 50$ , and  $k = 4$ 

Feature selection algorithm	Classifier											
	LDA ( $\epsilon = 0.1599$ )			3NN ( $\epsilon = 0.1360$ )			5NN ( $\epsilon = 0.1428$ )			CART ( $\epsilon = 0.1496$ )		
	$\delta$	$v$	$\tau$	$\delta$	$v$	$\tau$	$\delta$	$v$	$\tau$	$\delta$	$v$	$\tau$
$A_{RESUB}^{LDA}$	0.0880	1.5505	0.57	0.1167	1.8576	0.56	0.1163	1.8140	0.56	0.1123	1.7506	0.69
$A_{LOO}^{LDA}$	0.0892	1.5582	0.61	0.1128	1.8290	0.62	0.1129	1.7904	0.62	0.1092	1.7300	0.69
$A_{RESUB}^{3NN}$	0.0960	1.6008	0.42	0.1266	1.9304	0.66	0.1259	1.8813	0.66	0.1213	1.8109	0.64
$A_{LOO}^{3NN}$	0.1022	1.6394	0.48	0.1296	1.9527	0.48	0.1301	1.9107	0.48	0.1218	1.8140	0.61
$A_{RESUB}^{5NN}$	0.0944	1.5907	0.59	0.1245	1.9150	0.66	0.1253	1.8773	0.66	0.1173	1.7840	0.74
$A_{LOO}^{5NN}$	0.1024	1.6407	0.55	0.1279	1.9401	0.48	0.1299	1.9097	0.48	0.1221	1.8159	0.62
$A_{RESUB}^{CART}$	0.1065	1.6662	0.37	0.1338	1.9836	0.45	0.1367	1.9573	0.45	0.1273	1.8507	0.53
$A_{LOO}^{CART}$	0.1187	1.7428	0.33	0.1403	2.0315	0.26	0.1436	2.0059	0.26	0.1302	1.8703	0.51

with 5NN, even though  $A_{RESUB}^{5NN}$  is outperformed by  $A_{RESUB}^{LDA}$  when considering  $\delta$ , it performs the best in feature discovery.

The kinds of results we see in this single illustration are indicative of the complex behavior of feature-selection algorithms and the need for a stringently designed test bed to evaluate them. The performance of SFFS depends on the interaction between the methodology of the algorithm, the classification rule inside the algorithm, the error estimation procedure inside the algorithm, the feature-label distribution, the sample size and the classification rule for which the chosen feature set will be used. Simple heuristics cannot be relied upon in such complex circumstances.

## 5.2 Concluding remarks

Given the growing focus on feature selection, in particular, in the area of gene discovery, proposed feature-selection algorithms need to be evaluated under uniform conditions and they must be compared with true rankings. The test bed we have developed supports such evaluation and can serve as an open source to the community. There is a shortcoming to the test bed: the total number of genes and the number of features that can be incorporated in the algorithm are limited. The current implementation allows 70 genes, far below the thousands of genes typical on a microarray, and it allows up to seven features, which is not bad considering the small sample sizes in practice and the effects of the peaking phenomenon. As our computation capability grows beyond the current configuration, these numbers will increase, but in the mean time, it is better to have some ground-truth standard by which to test a proposed feature-selection

algorithm than no ground-truth standard at all. Indeed, if a proposed algorithm cannot work well with 70 genes and 7 features, then it cannot be expected to work well with larger numbers of genes and features.

A key aspect of the test bed is the facility that allows new datasets to be incorporated into the test bed with a single command. These datasets must be sufficiently large that they can serve as empirical distributions. As microarray and other technologies become more cost effective, more large studies will be carried out and thereby provide an increasing number of empirical distributions for the test bed. For instance, the Project for Oncology of the International Genomic Consortium aims at integrating longitudinal clinical annotation with gene expression data to develop diagnostic markers, prognostic indicators and therapeutic targets. A current three-year project is in the process of creating a data base of gene-expression profiles of 2500 human tumor specimens and 500 normal tissues collected under standardized conditions, clinically annotated and de-identified for public access.

## ACKNOWLEDGEMENTS

This work has been supported in part by the National Cancer Institute (CA-90301) and the National Science Foundation (ECS-0355227 and CCF-0514644).

*Conflict of Interest:* none declared.

## REFERENCES

- Armstrong,S.A. et al. (2002) MLL Translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nat. Genet.*, **30**, 41–47.
- Bittner,M. et al. (2000) Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, **406**, 536–540.
- Bohen,S.P. et al. (2003) Variation in gene expression patterns in follicular lymphoma and the response to rituximab. *Proc. Natl Acad. Sci. USA*, **100**, 1926–1930.
- Braga-Neto,U.M. and Dougherty,E.R. (2004a) Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, **20**, 374–380.
- Braga-Neto,U.M. and Dougherty,E.R. (2004b) Bolstered error estimation. *Pattern Recognit.*, **37**, 1267–1281.
- Cover,T. and Van Campenhout,J. (1977) On the possible orderings in the measurement selection problem. *IEEE Trans. Syst. Man Cybern.*, **7**, 657–661.
- Devroye,L., Györfi,L. and Lugosi,G. (1996) *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York.
- Efron,B. (1983) Estimating the error rate of a prediction rule: improvement on cross-validation. *J. Am. Stat. Soc.*, **78**, 316–331.
- Golub,T.R. et al. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.
- Hua,J. et al. (2005a) Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, **21**, 1509–1515.
- Hua,J. et al. (2005b) Determination of the optimal number of features for quadratic discriminant analysis via the normal approximation to discriminant distribution. *Pattern Recognit.*, **38**, 403–421.
- Hughes,G. (1968) On the mean accuracy of the statistical pattern recognizers. *IEEE Trans. Inform. Theory*, **14**, 55–63.
- Jain,A. and Chandrasekaran,B. (1982) Dimensionality and sample size consideration in pattern recognition practice. In Krishnaiah,P. and Kanal,L. (eds), *Classification, Pattern Recognition and Reduction of Dimensionality. Handbook of Statistics*. North-Holland, Amsterdam, Vol. 2, pp. 835–856.
- Jain,A.K. and Zongker,D. (1997) Feature selection—evaluation, application, and small sample performance. *IEEE Trans. Pattern Anal. Mach. Intelli.*, **19**, 153–158.
- Kanal,L. and Chandrasekaran,B. (1971) On dimensionality and sample size in statistical pattern classification. *Pattern Recognit.*, **3**, 225–234.
- Kim,S. et al. (2002) Identification of combination gene sets for glioma classification. *Mol. Cancer Ther.*, **1**, 1229–1236.
- Kudo,M. and Sklansky,J. (2000) Comparison of algorithms that select features for pattern classifiers. *Pattern Recognit.*, **33**, 25–41.
- Li,L. et al. (2001) Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics*, **17**, 1131–1142.
- Pudil,P. et al. (1994) Floating search methods in feature selection. *Pattern Recognit. Lett.*, **15**, 1119–1125.
- Schaner,M.E. et al. (2003) Gene expression patterns in ovarian carcinomas. *Mol. Biol. Cell*, **14**, 4376–4386.
- Sima,C. et al. (2005a) Superior feature-set ranking for small samples using bolstered error estimation. *Bioinformatics*, **21**, 1046–1054.
- Sima,C. et al. (2005b) Impact of error estimation on feature-selection algorithms. *Pattern Recognit.*, **38**, 2472–2482.
- Tschentscher,F. et al. (2003) Tumor classification based on gene expression profiling shows that uveal melanomas with and without monosomy 3 represent two distinct entities. *Cancer Res.*, **63**, 2578–2584.
- van de Vijver,M.J. et al. (2002) A gene-expression signature as a predictor of survival in breast cancer. *N. Engl. J. Med.*, **374**, 1999–2009.
- van 't Veer,L.J. et al. (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.
- West,M. et al. (2001) Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc. Natl Acad. Sci. USA*, **98**, 11462–11467.