In: New Signal Processing Research
Editor: Takumi Maeda, pp. 1-30

ISBN 978-1-60456-479-2

Chapter 1

EXPLORING THE LINK BETWEEN BOLSTERED CLASSIFICATION ERROR AND DATASET COMPLEXITY FOR GENE EXPRESSION BASED CANCER CLASSIFICATION

***Oleg Okun*[1]*, Giorgio Valentini*[2]*, and Helen Priisalu*[3]**
[1]University of Oulu, Finland
[2]University of Milan, Italy
[3]Teradata, Finland

Abstract

Gene expression profiles were shown to be useful in genomic signal processing when discriminating between cancer and normal (healthy) examples and/or between different types of cancer. K-nearest neighbors (k-NN) is one of the classification algorithms that demonstrated good performance for gene expression based cancer classification. Given that distance metric is fixed, the conventional k-NN has a single parameter (k - the number of nearest neighbors for each example) to set, which makes k-NN a very attractive choice in addition to the fact that it does not need training.

Classification performance of any classifier, including a k-NN, is typically characterized by classification error achieved on independent examples, which are often unavailable for the considered task. Thus, unbiased and low-variance error estimation is of ultimate importance in this case. We found that bolstered error satisfies these requirements and it was therefore chosen for our study. Bolstered error estimation is built on random sampling in the neighborhood of each example (with example-dependent neighborhood radius) and computing the number of errors made on such artificially created data. Because of random sampling, all examples can be employed in assessing the error, unlike cross-validation or bootstrap procedures.

In this work, we investigate the link between k-NN bolstered error and dataset complexity characterizing how difficult to classify a certain dataset. Our measure for the dataset complexity is the normalized Wilcoxon rank sum statistic. Through extensive simulation coupled with the copula method for analysis of association in bivariate data, we show that dataset complexity and bolstered error are related in terms of several dependence types such as positive quadrant dependence, tail monotonicity, and stochastic monotonicity.

As a result, we propose a new scheme for generating ensembles of k-NN classifiers, which is based on the selection of low complexity feature subsets for k-NNs in the ensemble, which constitutes to choosing accurate k-NNs according to the found dependence relation. The candidate subsets are randomly sampled from the whole set of the original features in order to make predictions of individual k-NNs diverse.

Experiments carried out on eight gene expression datasets containing different types of cancer demonstrate that our ensemble generating scheme is superior (in terms of bolstered resubstitution error) to a single best classifier in the ensemble and to the traditional ensemble construction scheme that is ignorant of dataset complexity. It also outperforms the redundancy-based filter, especially designed to remove irrelevant genes.

1 Introduction

According to [1], genomic signal processing (GSP) is the engineering discipline that studies the processing of genomic signals, i.e. the measurable events carried out by the genome. Gene expression is a two-stage process including the transcription of deoxyribonucleic acid (DNA) into messenger ribonucleic acid (mRNA) which is then translated into protein by the ribosome. When a protein is produced, a gene is said to be expressed. Proteins are large compounds of amino acids joined together in a chain and they are essential parts of organisms and participate in every process within cells. GSP deals with extracting information from gene expression measurements, which is then processed, analyzed and used for gaining biological and medical knowledge. Recent advances in microarray technology facilitate measurement of gene expression levels for thousands of genes at once.

Cancer classification based on gene expression levels, which is the subject of our study, is one of the topics of intensive research in GSP, since it was shown in numerous works [2, 3, 4] that expression levels provide valuable information for discrimination between normal and cancer examples.

However, the classification task is not easy since there are typically thousands of expression levels versus few dozens of examples. In addition, expression levels are noisy due to the complex procedures and technologies involved in the measurements of gene expression levels, thus causing ambiguity in classification. Hence, the original set of genes must be reduced to those genes that are relevant to discrimination between different classes. This operation is called feature or gene selection[1]. Genes preserved or selected as a result of feature selection are then used to classify data.

Basically, general feature selection methods widely applied to the datasets with many more examples than features can be (and they are) utilized for gene selection, too. These methods can be approximately divided into filters and wrappers. Wrappers base their decisions on which feature(s) to select by employing a classifier. For small sample size gene expression datasets, they can easily introduce the induction bias when some genes are preferred over the others. Since those preferred genes might not be always relevant to cancer classification, we turn our attention to the filters that do gene selection independently of any classifier and solely based on data characteristics. Hence, they are less prone to the

[1] Further, the words ‘feature’ and ‘gene’ will be used interchangeably, since they have the same meaning in this work.

induction bias. It should be however noted that even filters might not be able to completely avoid this bias since class labels of examples often guide gene selection in the filter model.

This brief analysis prompted us to concentrate on random gene selection where genes to be used with a classifier, are randomly sampled from the original set of genes, irrespectively of class information and a classifier. The additional fact that caused us to make such a decision was the work [5], where it was concluded that *differences in classification performance among feature selection algorithms are less significant than performance differences among the error estimators used to implement these algorithms*. In other words, the way of how error is computed has a larger influence on classification accuracy than the choice of a feature selection algorithm. Among several error estimators we opted for the bolstered resubstitution error because it provides a low-bias, low-variance estimate of classification error, which is what is needed for high dimensional gene expression data [6].

However, a single random sample cannot guarantee that sampled genes will lead to good classification results. Hence, we need to sample genes several times to be more certain about the outcome, which, in turn, implies several classifications have to be done. Thus, it is natural to combine predictions of several classifiers into a single prediction. Such a scheme is termed an ensemble of classifiers in the literature [7]. It is well known that under certain conditions an ensemble can outperform its most accurate member. In the context of dimensionality reduction, an ensemble composed of a small number of classifiers, each working with a small subset of genes, results in the desired effect. For instance, if the original set comprises 1000 genes, five classifiers, each employing 20 genes[2], lead to a 10-fold dimensionality reduction. Thus, using an ensemble instead of a single classifier can be beneficial for both dimensionality reduction and classification performance.

As a base classifier in the ensemble, a k-nearest neighbor (k-NN) is used because it performed well for cancer classification, compared to more sophisticated classifiers [8]. Besides, it is a simple method that has a single parameter (the number of nearest neighbors) to be pre-defined, given that the distance metric is Euclidean.

Eight gene expression datasets containing different types of cancer are utilized in our work. We begin with the recently proposed redundancy-based filter [9], especially designed to filter out irrelevant genes. As this filter turned to be quite aggressive in removing genes, we proceed to experiment with k-NN ensembles. In particular, we compare two ensemble schemes: one relying on the concept called dataset complexity when choosing which subsets of features to include into an ensemble and another ignoring dataset complexity. Based on the copula method [10, 11, 12], which is useful in exploring association (dependence or concordance) relations in multivariate data, we hypothesize that there is positive dependence between dataset complexity measured by the Wilcoxon rank sum statistic [13] and the bolstered resubstitution error [6], with low (high) complexity associated with small (large) error. As a result, selecting a low-complexity subset of genes implies an accurate k-NN, which, in turn, implies an accurate k-NN ensemble. Experimental results clearly favor the complexity-based scheme of k-NN ensemble generation over 1) the complexity-ignorant scheme, 2) a single best k-NN in the ensemble, and 3) the redundancy-based filter, thus confirming our hypothesis.

The chapter has the following structure. Section 2 describes gene expression datasets

[2] Let us assume that there is no overlap between different subsets of genes.

employed in experiments. The redundancy-based filter is briefly introduced in Section 3. Section 4 defines the dataset complexity characteristic while Section 5 defines bolstered resubstitution error. The link between the two is explored and analyzed in Section 6. Uncertainty of single classification is discussed in Section 7. Two ensemble generating schemes based on random feature selection are presented in Section 8 and experimental results obtained with them on eight datasets are given in Section 9. Finally, Section 10 concludes the paper. Programming code for random number generation and mathematical derivations related to copulas are placed in two appendices.

2 Gene Expression Datasets

The following eight datasets were chosen for experiments.

2.1 SAGE Dataset 1

SAGE stands for Serial Analysis of Gene Expression [14, 15]. This is technology alternative to microarrays (complementary DNA and oligonucleotides). Though SAGE was originally conceived for use in cancer studies, there is not much research using SAGE datasets regarding ensembles of classifiers. SAGE provides a statistical description of the mRNA population present in a cell without prior selection of the genes to be studied [16]. This is the main distinction of SAGE over microarray approaches (cDNA and oligonucleotide) that are limited to the genes represented in the chip. SAGE "counts" the number of transcripts or tags for each gene, where the tags substitute the expression levels. As a result, counting sequence tags yields positive integer numbers in contrast to microarray measurements.

In the chosen dataset [17], there are expressions of 822 genes in 74 cases (24 cases are normal while 50 cases are cancerous) [18]. The dataset contains 9 different types of cancer. We decided to ignore the difference between cancer types and to treat all cancerous cases as belonging to a single class. No preprocessing was done.

2.2 Colon Dataset

This microarray (oligonucleotide) dataset [19], introduced in [2], contains expressions of 2000 genes for 62 cases (22 normal and 40 colon tumor cases). Preprocessing includes the logarithmic transformation to base 10, followed by normalization to zero mean and unit variance as usually done with this dataset.

2.3 Brain Dataset 1

This microarray (oligonucleotide) dataset [20] introduced in [3] contains two classes of brain tumor. The dataset (also known as Dataset B) contains 34 medulloblastoma cases, 9 of which are desmoplastic and 25 are classic. Preprocessing consists of thresholding of gene expressions with a floor of 20 and ceiling of 16000; filtering with exclusion of genes with $\max/\min \leq 3$ or $max - min < 100$, where max and min refer to the maximum and minimum expressions of a certain gene across the 34 cases, respectively; base 10 logarithmic transformation; normalization across genes to zero mean and unit variance. As a result, 5893 out of 7129 original genes are only retained.

2.4 SAGE Dataset 2

This is a larger SAGE dataset [17], containing 31 normal and 59 cancer (10 types of cancer) cases with 27679 expressed genes. As with the smaller dataset, no preprocessing was done and all cancer types were assigned to a single class.

2.5 Prostate Dataset 1

This microarray (oligonucleotide) dataset [21] introduced in [4] includes the expressions of 12600 genes in 52 prostate and 50 normal cases. No preprocessing of the data was done.

2.6 Prostate Dataset 2

This dataset [21] was obtained independently of the one described in the previous section. It has 25 prostate and 9 normal cases with 12600 expressed genes. No preprocessing was done.

2.7 Brain Dataset 2

This (oligonucleotide) dataset [20] known as Dataset C in [3] contains 60 medulloblastoma cases, corresponding to 39 survivors and 21 nonsurvivors according to the patient status. Preprocessing includes thresholding of gene expressions with a floor of 100 and ceiling of 16000; filtering with exclusion of genes with $\max / \min \leq 5$ or $max - min < 500$, where max and min refer to the maximum and minimum expressions of a certain gene across the 60 cases, respectively; base 10 logarithmic transformation; normalization across genes to zero mean and unit variance. As a result, 4459 out of 7129 original genes are only retained.

2.8 Diffuse Large B-Cell Lymphoma (DLBCL) Dataset

This (oligonucleotide) dataset [22] described in [23] contains 6149 gene expression levels characterizing 58 patients with diffuse large B-cell lymphoma according to their health status: 32 cured patients or those who died from other than lymphoma causes (‘cured’ class) and 26 patients who died of lymphoma or whose disease is either progressive or recurrent refractory (‘fatal/refractory’ class). Preprocessing includes thresholding of gene expressions with a floor of 20 and ceiling of 16000; filtering with exclusion of genes with $\max / \min < 3$ or $max - min < 100$, where max and min refer to the maximum and minimum expressions of a certain gene across the 60 cases, respectively. As a result, 6149 out of 7129 original genes are only retained.

2.9 Dataset Summary

Table 1 provides a summary for all datasets.

Table 1: Summary of eight gene expression datasets.

Dataset no.	Cancer type(s)	Ref.	# expression levels	# cases
1	Multiple	[18]	822	74
2	Colon	[2]	2000	62
3	Brain	[3]	5893	34
4	Multiple	[18]	27679	90
5	Prostate	[4]	12600	102
6	Prostate	[4]	12600	34
7	Brain	[3]	4459	60
8	Lymphoma	[23]	6149	58

3 Redundancy-Based Filter

The redundancy-based filter (RBF) [9] is based on the concept of an approximate Markov blanket (AMB). Finding the complete Markov blanket is computationally prohibitive for high dimensional gene expression data, thus the approximation is used instead. The goal is to find for each gene F_i an AMB M_i that subsumes the information content of F_i. In other words, if M_i is a true Markov blanket for F_i, the class C is conditionally independent of F_i given M_i, i.e. $p(C|F_i,M_i) = p(C|M_i)$.

To efficiently find an AMB two types of correlations are employed: 1) individual C-correlation between a gene F_i and the class C and 2) combined C-correlation between a pair of genes F_i and F_j ($i \neq j$) and the class C. Both correlations are defined through symmetrical uncertainty $SU(X,C)$, where X is either F_i (individual C-correlation) or $F_{i,j}$ (combined C-correlation), with $SU(X,C)$ defined as

$$SU(X,C) = 2\left[\frac{IG(X|C)}{H(X)+H(C)}\right],$$

where $H(\cdot)$ is entropy, $IG(X|C) = H(X) - H(X|C)$ is information gain from knowing the class information. SU is a normalized characteristic whose values lie between 0 and 1, where 0 indicates that X and C are independent.

To reduce the variance and noise of the original data, continuous expression levels were converted to nominal values -1, 0, and +1, representing the under-expression, baseline, and over-expression of genes, which correspond to $(-\infty, \mu-\sigma/2)$, $[\mu-\sigma/2, \mu+\sigma/2]$, and $(\mu+\sigma/2, +\infty)$, respectively, with μ and σ being the mean and the standard deviation of all expression levels for a given gene. For nominal variables, the entropies needed in the formulas above are computed as follows:

$$H(X) = -\sum_i P(x_i)\log_2(P(x_i)),$$
$$H(X|C) = -\sum_k P(c_k)\sum_i P(x_i|c_k)\log_2(P(x_i|c_k)),$$

where $P(x_i)$ is the probability that $X = x_i$ and $P(x_i|c_k)$ is the probability that $X = x_i$ given $C = c_k$. For combined C-correlation, x_i in these formulas should be replaced with the pair

(x_i, x_j) so that

$$H(X) = -\sum_{i,j} P(x_i, x_j) \log_2(P(x_i, x_j)),$$
$$H(X|C) = -\sum_{k} P(c_k) \sum_{i,j} P(x_i, x_j|c_k) \log_2(P(x_i, x_j|c_k)).$$

Since x_i (x_j) can take only three values: -1, 0, +1, there are nine pairs ((-1,-1), (-1,0), (-1,+1), (0,-1), (0,0), (0,+1), (+1,-1), (+1,0), (+1,+1)) for which probabilities (and hence, entropies) need to be evaluated.

RBF starts from computing individual C-correlation for each gene and sorting all correlations in descending order. The gene with the largest correlation is considered as predominant (no AMB exists for it) and hence it is put to the list S of the selected genes and used to filter out other genes. After that, the iteration begins with picking the first gene F_i from S and proceeds as follows. For all remaining genes, if F_i forms an AMB for F_j, the latter is removed from further analysis. The following conditions must be satisfied for this to happen: 1) individual C-correlation for F_i must be larger than or equal to individual C-correlation for F_j, which means that a gene with a larger individual correlation provides more information about the class than a gene with a smaller individual correlation, and 2) individual C-correlation for F_i must be larger than or equal to combined C-correlation for F_i and F_j, which means that if combining F_i and F_j does not provide more discriminative power than F_i alone, F_j is decided to be redundant. After one round of filtering, RBF takes the next (according to the magnitude of individual C-correlation) still unfiltered gene and the filtering process is repeated again. Since a lot of genes are typically removed at each round (gene expression data contain a lot of redundancy) and removed genes do not participate in the next rounds, the RBF is much faster than the typical hill climbing (greedy forward or backward search).

4 Dataset Complexity

It is known that the performance of classifiers is strongly data-dependent. To gain insight into a supervised classification problem[3], one can adopt dataset complexity characteristics. The goal of such characteristics is to provide a score reflecting how well classes of the data are separated. Given a set of features, the data of each class are projected onto the diagonal linear discriminant axis by using only these features (for details, see [24]). Projection coordinates then serve as input for the Wilcoxon rank sum test for equal medians [13] (the null hypothesis of this test is two medians are equal at the 5% significance level). Given a sample divided into two groups according to class membership, all the observations are ranked as if they were from a single sample and the rank sum statistic W is computed as the sum of the ranks in the smaller group. The value of the rank sum statistic is employed as a score characterizing separability power of a given set of features. The higher this score, the larger the overlap in projections of two classes, i.e. the worse separation between classes. To compare W coming from different datasets, each W can be normalized by the sum of all

[3] Two-class problems are assumed.

ranks, i.e. if N is the sample size, then the sum of all ranks will be $\sum_{i=1}^{N} i$. The normalized W lies between 0 and 1.

Our complexity characteristic is classifier-independent, i.e. it does not depend on a certain classifier. Employing a classifier-dependent characteristic would not provide an absolute scale for comparison. For example, it is well known that a nearest neighbor classifier can sometimes easily classify nonlinearly separable data.

Our choice for such a dataset complexity characteristic was not accidental. Since gene expression data are very high dimensional, it is not surprising that two classes could be linearly separable in high dimensional space (however, this does not make the classification task easy as interclass distances could be still smaller than intraclass distances, thus giving rise to classification errors). That is, the k-NN decision boundary can be assumed to be a hyperplane. On the other hand, the complexity characteristic we employ belongs to the class of linear discriminants estimating how well a line separates two classes. As a result, we have a good match between the behavior of a classifier and the model of class separability encoded in the complexity characteristic. To confirm our hypothesis, we computed the index of linear separability $L1$ [25], which is the objective function value for the linear programme, by using all features for each dataset. The closer $L1$ to zero, the more a given dataset linearly separable. It can be seen in Table 2 that all datasets can be considered to be linear separable. However, SAGE 1 seems to be much less linearly separable than Prostate 2.

Table 2: Dataset ranking based on the linear separability index $L1$ multiplied by 10^{-15}. The lower the rank is (1 - lowest rank, 8 - highest rank), the less linearly separable two classes are.

Dataset no.	1	2	3	4	5	6	7	8
$L1$	2.2	0.77	0.027	0.15	0.19	1.2e-019	0.054	4.8e-010
rank	1	2	6	4	3	8	5	7

5 Bolstered Resubstitution Error

This is a low-variance and low-bias classification error estimation method proposed in [6]. Unlike the cross-validation techniques reserving a part of the original data for testing, it permits to use the whole dataset. Since sample size of gene expression datasets is very small compared to the data dimensionality, using all available data is an important positive factor. However, one should be aware of the effect of overfitting in this case when a classifier demonstrates excellent performance on the training data but fails on independent unseen data. Braga-Neto and Dougherty [6] avoided this pitfall by randomly generating a number of artificial points (examples) in the neighborhood of each training point. These artificial examples then act as a test set and classification error on this set is called bolstered. In this paper, we utilize the bolstered variant of the conventional resubstitution error known as *bolstered resubstitution error*.

Briefly, bolstered resubstitution error is estimated as follows [6]. Let A_0 and A_1 be two decision regions corresponding to the classification generated by a given algorithm, N be the number of training points, and M_{MC} be the number of random samples drawn from the D-variate normal distribution per training point ($M_{MC} = 10$ as advocated in [6]). The bolstered resubstitution error is then defined as

$$\varepsilon_{bresub} \approx \frac{1}{NM_{MC}} \sum_{i=1}^{N} \left(\sum_{j=1}^{M_{MC}} I_{x_{ij} \in A_1} I_{y_i=0} + \sum_{j=1}^{M_{MC}} I_{x_{ij} \in A_0} I_{y_i=1} \right), \tag{1}$$

where $\{x_{ij}\}_{j=1,\dots,M_{MC}}$ are samples drawn from $1/((2\pi)^{D/2}\sigma_i^D)e^{-\|x\|^2/(2\sigma_i^2)}$. The bolstered resubstitution error is thus equal to the sum of all error contributions divided by the number of points. Samples are drawn based on the Marsaglia polar normal random number generator (see Appendix B).

In a 2-D space, samples come from a circle centered at a particular training point. In a D-dimensional case, they are drawn from a hypersphere. Hence, the radius of this hypersphere, determined by σ_i, is of importance since its selection amounts to choosing the degree of bolstering. Typically, σ_i should vary from point to point in order to be robust to the data. In [6] $\sigma_i = \hat{d}(y_i)/cp$ for $i = 1, \dots, N$, where $\hat{d}(y_i)$ is the mean minimum distance between points belonging to class of y_i (y_i can be either 0 or 1)[4], and cp is the constant called the correction factor defined as the inverse of the chi-square cdf (cumulative distribution function) with parameters 0.5 and D, because interpoint distances in the Gaussian case are distributed as a chi random variable with D degrees of freedom. Thus, cp is the function of the data dimensionality. The parameter 0.5 is chosen so that points inside a hypersphere will be evenly sampled.

6 Link Between Dataset Complexity and Classification Error

Our main idea to build ensembles of k-NNs is based on the hypothesis that *the dataset complexity and bolstered resubstitution error are related*. To verify our hypothesis, 10000 feature subsets were randomly sampled for each dataset (subset size ranged from 1 to 50) and both complexity and bolstered resubstitution error for 3-NN were computed. Anomalous complexity values lying three standard deviations from the average complexity were treated as outliers and therefore removed from further analysis. The result of such simulation is shown in Figs. 1-8 together with marginal histograms for each variable. It can be observed that univariate distributions vary from dataset to dataset and often they are non-Gaussian. Besides, the complexity and error distributions for a certain dataset belong to different types, e.g. one is normal while another is exponential. However, the dependence between complexity and error is clearly detectable when looking at Figs. 1-8. Besides, one characteristic important for successful ensemble generation is present: diversity among predictions since one complexity value corresponds to several different error values.

To quantify this dependence, the rank correlation coefficients Spearman's ρ and Kendall's τ were computed (see Table 3) and the test on positive correlation at the significance level 0.05 was done which confirmed the existence of such correlation (all p-values were equal to zero). The rank correlations measure the degree to which large (small) values

[4] $\hat{d}(y_i)$ is determined by first computing the minimum distance from each point x_i to all other points x_j ($j \neq i$) of the same class as that of x_i and then by averaging thus obtained minimum distances.

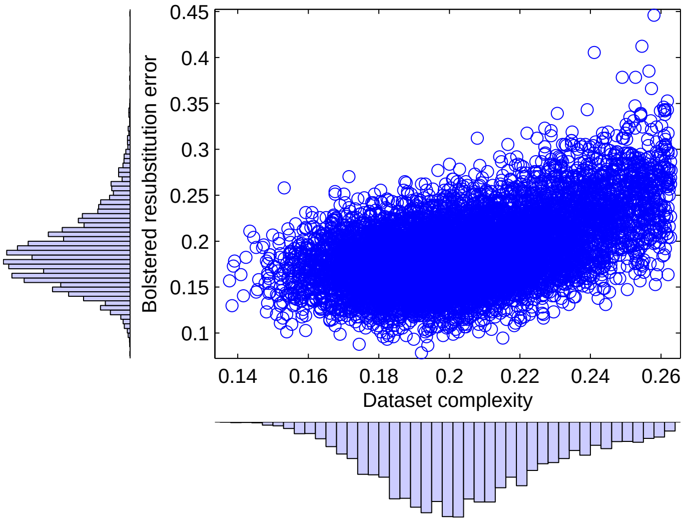

Figure 1: (SAGE 1) Bivariate distribution of normalized complexity and bolstered resubstitution error and univariate marginal histograms.

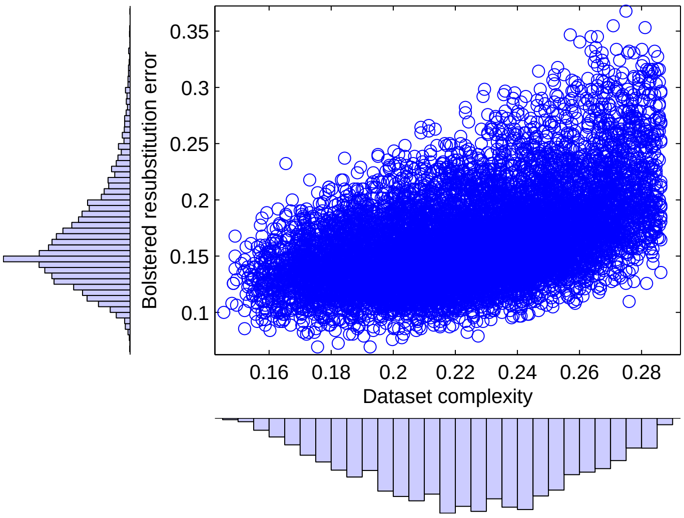

Figure 2: (Colon) Bivariate distribution of normalized complexity and bolstered resubstitution error and univariate marginal histograms.

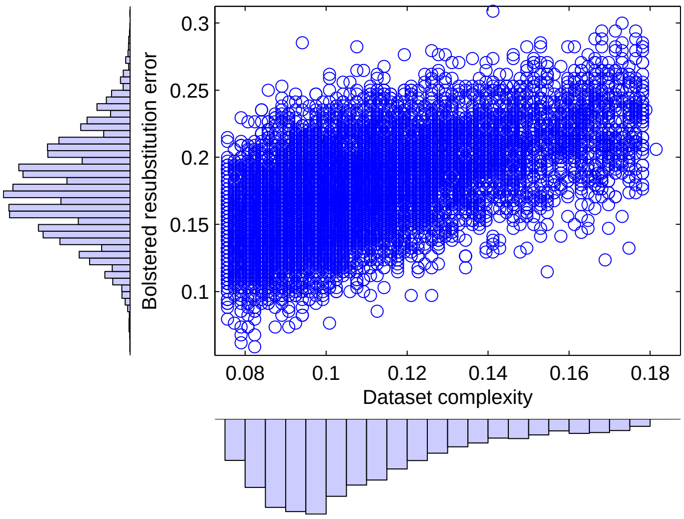

Figure 3: (Brain 1) Bivariate distribution of normalized complexity and bolstered resubstitution error and univariate marginal histograms.

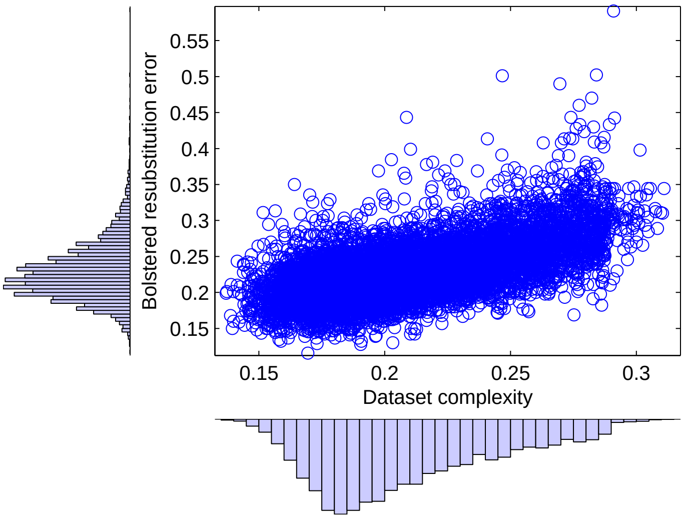

Figure 4: (SAGE 2) Bivariate distribution of normalized complexity and bolstered resubstitution error and univariate marginal histograms.

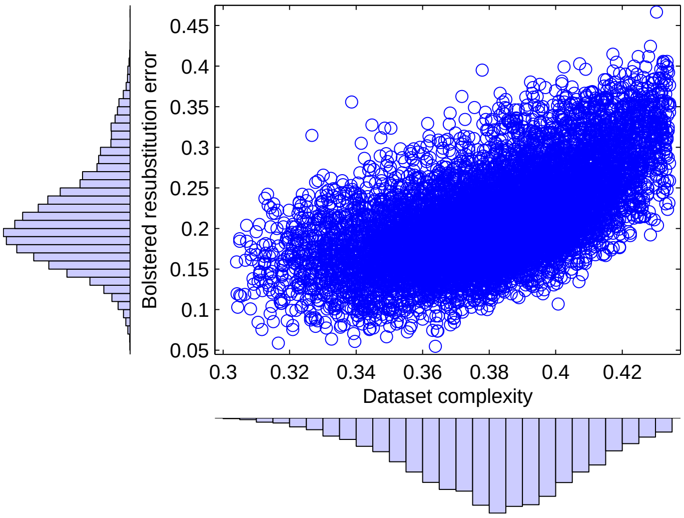

Figure 5: (Prostate 1) Bivariate distribution of normalized complexity and bolstered resubstitution error and univariate marginal histograms.

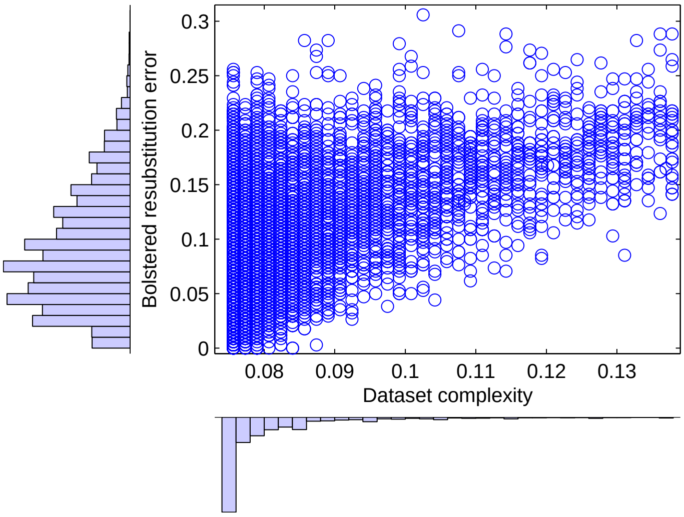

Figure 6: (Prostate 2) Bivariate distribution of normalized complexity and bolstered resubstitution error and univariate marginal histograms.

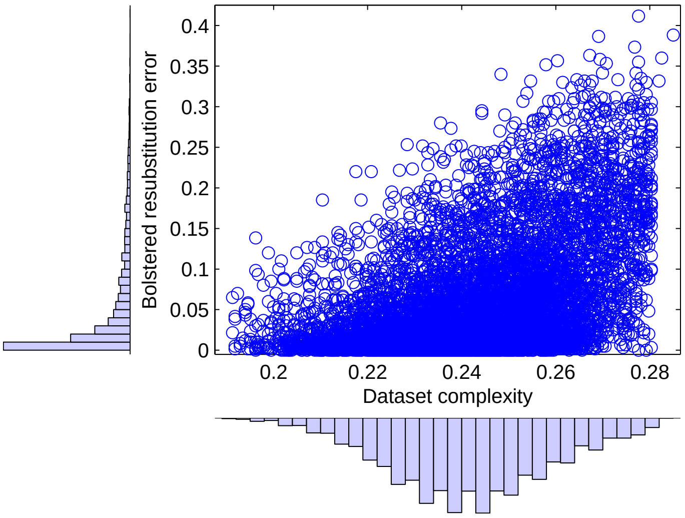

Figure 7: (Brain 2) Bivariate distribution of normalized complexity and bolstered resubstitution error and univariate marginal histograms.

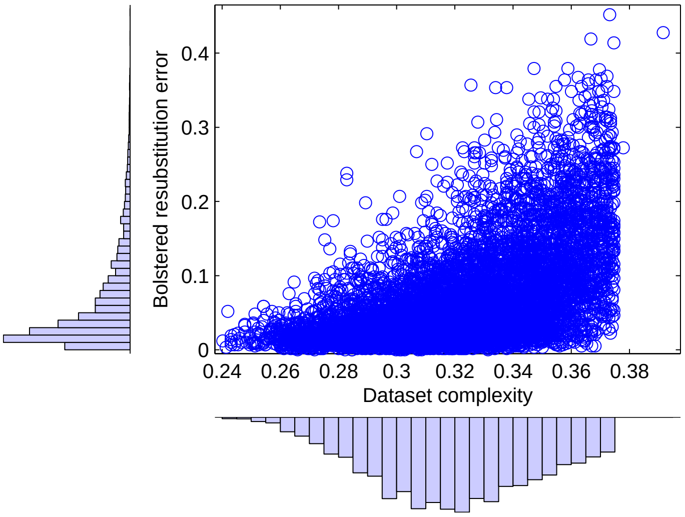

Figure 8: (Lymphoma) Bivariate distribution of normalized complexity and bolstered resubstitution error and univariate marginal histograms.

of one random variable correspond to large (small) values of another variable (concordance relations[5] among variables). They are useful descriptors in our case since high (low) complexity implies that the data are difficult (easy) to accurately classify, which, in turn, means high (low) classification error. Unlike the linear correlation coefficient, ρ and τ are preserved under any monotonic (strictly increasing) transformation of the underlying random variables.

Table 3: Spearman's ρ and Kendall's τ estimated for all datasets.

Dataset no.	τ	ρ
1	0.3100	0.4468
2	0.3446	0.4964
3	0.3991	0.5581
4	0.4173	0.5864
5	0.4288	0.6006
6	0.3887	0.5107
7	0.3117	0.4486
8	0.3993	0.5667

To deeply explore dependence relations, we employed the copula method [10, 11, 12]. The word copula is a Latin noun which means ‘a link, tie or bond’ was first introduced by Abe Sklar in [10]. Copulas are functions that describe dependencies among variables and allow to model correlated multivariate data by combining univariate distributions. Using copulas is an appropriate solution since the assumption that the joint distribution of random variables is normal often does not hold for multivariate data in practice even if the marginal distributions are normal.

A copula is a multivariate probability distribution, where each random variable has a uniform marginal distribution on the interval [0,1]. The dependence between random variables is completely separated from the marginal distributions in the sense that random variables can follow any marginal distributions, and still have the same rank correlation. This is one of the main appeals of copulas: they allow separation of dependence and marginal distribution. Though there are multivariate copulas, we will only talk about bivariate ones since our dependence relation includes two variables.

Sklar's theorem, which is the foundation theorem for copulas, states that for a given joint multivariate distribution function $H(x,y) = P(X \leq x, Y \leq y)$ of a pair of random variables X and Y and the relevant marginal distributions $F(x) = P(X \leq x)$ and $G(y) = P(Y \leq y)$, there exists a copula function C relating them, i.e. $H(x,y) = C(F(x), G(y))$. If F and G are continuous, C is unique. Otherwise, C is uniquely determined on $\text{Ran}X \times \text{Ran}Y$, where ‘RanX’ (‘RanY’) stands for the range of X (Y). In other words, for each pair of real numbers (x,y) there are three numbers $F(x)$, $G(y)$, and $H(x,y)$ lying in the interval [0,1]. Alternatively, each pair (x,y) is matched by a point $(F(x), G(y))$ in the unit square $\mathbf{I}^2 : [0,1] \times [0,1]$, and this ordered pair in turn is associated with a number $H(x,y)$ in [0,1]. The correspondence assigning the value of the joint distribution function to each ordered pair of values of

[5] Since the definitions of these relations by ρ and τ are different, there is a difference in absolute values in Table 3.

the individual distribution functions is a copula function C [11].

Thus, a copula is a function C from $\mathbf{I}^2$ to $\mathbf{I}$ with the following properties [11]:

1. For every u, v in $\mathbf{I}$,
$$C(u,0) = 0 = C(0,v)$$
and
$$C(u,1) = u, C(1,v) = v.$$
2. for every u_1, u_2, v_1, v_2 in $\mathbf{I}$ such that $u_1 \leq u_2$ and $v_1 \leq v_2$,
$$C(u_2,v_2) - C(u_2,v_1) - C(u_1,v_2) + C(u_1,v_1) \geq 0.$$

If F and G are continuous, the following formula is used to construct copulas from the joint distribution functions: $C(u,v) = H(F^{-1}(u), G^{-1}(v))$ [11], where F^{-1} means a quasi-inverse of F, G^{-1} means a quasi-inverse of G, and U and V are uniform random variables distributed between 0 and 1. That is, the typical copula-based analysis of multivariate (or bivariate) data starts with the transformation from the (X,Y) domain to the U,V domain, and all manipulations with data are then done in the latter. Such a transformation to the copula scale (unit square $\mathbf{I}^2$) can be achieved through a kernel estimator of the cumulative distribution function (cdf) (we used the MATLAB function *ksdensity*). After that the copula function $C(u,v)$ is generated according to the appropriate definition for a certain copula family (see, e.g. Eq. 2 below).

In [26] it was shown that Spearman's ρ and Kendall's τ can be expressed solely in terms of the copula function as follows:

$$\rho = 12\int\int(C(u,v) - uv)dudv = 12\int\int C(u,v)dudv - 3,$$
$$\tau = 4\int\int C(u,v)dC(u,v) - 1,$$

where integration is over $\mathbf{I}^2$.

The integrals in these formulas can be interpreted as the expected value of the function $C(u,v)$ of uniform [0,1] random variables U and V whose joint distribution function is C, i.e.

$$\rho = 12E(UV) - 3, \quad \tau = 4E(C(u,v)) - 1.$$

As a consequence, ρ for a pair of continuous random variable X and Y is identical to Pearson's linear correlation coefficient for random variables $U = F(X)$ and $V = G(Y)$ [11].

In general, the choice of a particular copula may be based on the observed data. Among numerous copula families, we preferred the Frank copula belonging to the Archimedean family based on the visual look of plots in Figs. 1-8 and for dependence in the tail. Besides, this copula type permits negative as well as positive dependence. We are particularly concerned with lower tail dependence when low complexity is associated with small classification error as this forms the basis for ensemble construction in our approach. The Frank copula is a one-parameter (θ is a parameter, $\theta \in]-\infty, +\infty[\backslash 0)$ copula defined for uniform variables U and V (both are defined over the unit interval) as

$$C_\theta(u,v) = -\frac{1}{\theta}\ln\left(1+\frac{(e^{-\theta u}-1)(e^{-\theta v}-1)}{e^{-\theta}-1}\right), \tag{2}$$

with θ determining the degree of dependence between the marginals (we set θ to Pearson's correlation coefficient between U and V so that as θ increases, the positive dependence also increases). Fig. 9 shows 500 random points generated from the Frank copula when $\theta = 8$.

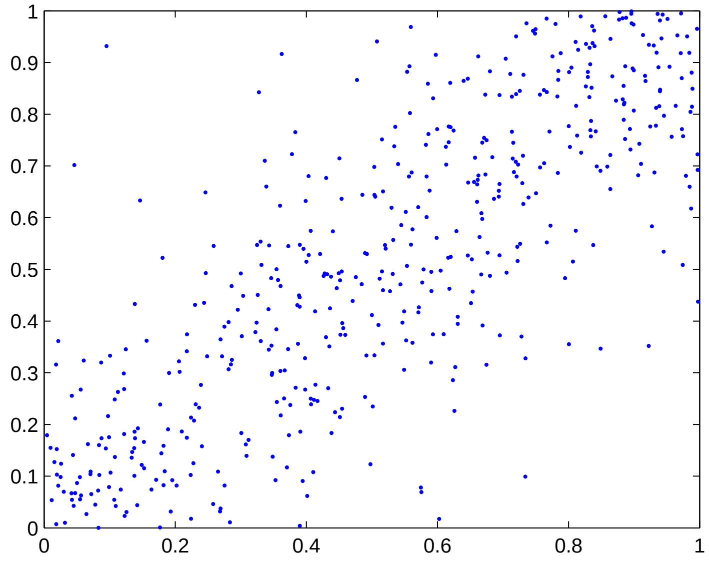

Figure 9: A random sample (500 points) generated from the Frank copula with $\theta = 8$.

Correlation coefficients measure the overall strength of the association, but give no information about how that varies across the distribution. The magnitude of τ or ρ is not an absolute indicator of such strength since for some distributions the attainable interval can be very small, say between -0.1 and +0.2 so that finding correlation of 0.2 and concluding that there is only weak dependence between variables would be a mistake since these variables are actually perfectly related. Hence, additional characteristics of dependence structure are necessary. They are quadrant dependence, tail monotonicity, stochastic monotonicity, with quadrant dependence being the weakest form of association among all the three.

6.1 Quadrant Dependence

Random variables X and Y are positively quadrant dependent (PQD) if $\forall(x,y)$ in $\mathbf{R}^2$, either inequality holds [11]:

$$P(X \le x, Y \le y) \ge P(X \le x)P(Y \le y),$$
$$P(X > x, Y > y) \ge P(X > x)P(Y > y).$$

X and Y are PQD if the probability that they are simultaneously small (or simultaneously large) is at least as it would be were they independent. In terms of C, the PQD conditions can be written as $C(u,v) \geq uv$ for all (u,v) in $\mathbf{I}^2$. By checking the last inequality, we found that complexity and bolstered resubstitution error are PQD for all datasets. Spearman's ρ (or, to be precise, $\rho/12$) can be interpreted as a measure of "average" quadrant dependence (both positive and negative) for random variables whose copula is C [11].

It is interesting to ask when one continuous bivariate distribution H_1 is more PQD (more concordant) than another H_2. The answer is readily provided by comparing ρ or τ [12]: if $\rho(H_1) \leq \rho(H_2)$ or $\tau(H_1) \leq \tau(H_2)$, then H_2 is more PQD (more concordant) than H_1. From Table 3 it can be seen that Prostate 1 is more PQD than other datasets, i.e. concordance relations between complexity and bolstered resubstitution error are much stronger for this data than those for other datasets.

6.2 Tail Monotonicity

As we mentioned above, we are interested in tail dependence when low (high) complexity associates small (large) classification error. Tail monotonicity reflects this type of association and it is a stronger condition for dependence than PQD. Let X and Y be random variables. Then four types of tail monotonicity can be defined as follows [11]:

- Y is left tail decreasing in X (LTD(Y|X)) if $P(Y \leq y | X \leq x)$ is a nonincreasing function of x for all y.
- X is left tail decreasing in Y (LTD(X|Y)) if $P(X \leq x | Y \leq y)$ is a nonincreasing function of y for all x.
- Y is right tail increasing in X (RTI(Y|X)) if $P(Y > y | X > x)$ is a nondecreasing function of x for all y.
- X is right tail increasing in Y (RTI(X|Y)) if $P(X > x | Y > y)$ is a nondecreasing function of y for all x.

In terms of a copula and its first-order partial derivatives these conditions are equivalent to

- LTD(Y|X) iff for any v in $\mathbf{I}$, $\partial C(u,v)/\partial u \leq C(u,v)/u$ for almost all u.
- LTD(X|Y) iff for any u in $\mathbf{I}$, $\partial C(u,v)/\partial v \leq C(u,v)/v$ for almost all v.
- RTI(Y|X) iff for any v in $\mathbf{I}$, $\partial C(u,v)/\partial u \leq (v - C(u,v))/(1-u)$ for almost all u.
- RTI(X|Y) iff for any u in $\mathbf{I}$, $\partial C(u,v)/\partial v \leq (u - C(u,v))/(1-v)$ for almost all v.

For the Frank copula, the first-order partial derivatives are (see Appendix A)

$$\frac{\partial C_\theta(u,v)}{\partial u} = \frac{e^{-\theta u}(e^{-\theta v}-1)}{e^{-\theta}-1+(e^{-\theta u}-1)(e^{-\theta v}-1)}, \tag{3}$$

$$\frac{\partial C_\theta(u,v)}{\partial v} = \frac{e^{-\theta v}(e^{-\theta u}-1)}{e^{-\theta}-1+(e^{-\theta u}-1)(e^{-\theta v}-1)}. \tag{4}$$

Tail monotonicity is also guaranteed if $\rho \geq \tau \geq 0$ is met [11].

We verified that for all datasets bolstered resubstitution error is left tail decreasing in complexity, complexity is left tail decreasing in bolstered resubstitution error, bolstered resubstitution error is right tail increasing in complexity, and complexity is right tail increasing in bolstered resubstitution error. Thus, dependence in the tail between these two variables exists.

6.3 Stochastic Monotonicity

Stochastic monotonicity is stronger than tail monotonicity. According to [11],

- Y is stochastically increasing in X (SI(Y|X)) if $P(Y > y|X = x)$ is a nondecreasing function of x for all y.
- X is stochastically increasing in Y (SI(X|Y)) if $P(X > x|Y = y)$ is a nondecreasing function of y for all x.

Alternatively, stochastic monotonicity can be expressed as

- SI(Y|X) iff for any v in $\mathbf{I}$, $C(u,v)$ is a concave function of u.
- SI(X|Y) iff for any u in $\mathbf{I}$, $C(u,v)$ is a concave function of v.

A concave function implies that the second-order derivatives must be less than or equal to zero. For the Frank copula, these derivatives are (see Appendix A)

$$\frac{\partial^2 C_\theta(u,v)}{\partial u^2} = \frac{\theta e^{-\theta u}(e^{-\theta v}-1)(e^{-\theta v}-e^{-\theta})}{\left[e^{-\theta}-1+(e^{-\theta u}-1)(e^{-\theta v}-1)\right]^2}, \qquad (5)$$

$$\frac{\partial^2 C_\theta(u,v)}{\partial v^2} = \frac{\theta e^{-\theta v}(e^{-\theta u}-1)(e^{-\theta u}-e^{-\theta})}{\left[e^{-\theta}-1+(e^{-\theta u}-1)(e^{-\theta v}-1)\right]^2}. \qquad (6)$$

Since $\theta > 0$ in our case (positive dependence as expressed by the rank correlation coefficients), it is easy to verify that $\frac{\partial^2 C_\theta(u,v)}{\partial u^2} \leq 0$ and $\frac{\partial^2 C_\theta(u,v)}{\partial v^2} \leq 0$, which, in turn, implies that $C_\theta(u,v)$ is concave (see also Appendix A). Thus, for all datasets in our study, bolstered resubstitution error is stochastically increasing in complexity and complexity is stochastically increasing in bolstered resubstitution error.

7 Uncertainty of Single Classification

There exist many classifiers (k-NN, linear and quadratic discriminant analysis, support vector machine) showing good performance on gene expression data [8]. Each of them has its strong and weak points and that is why none of them is superior to others. In addition, when feature selection precedes classification, there can be multiple subsets of genes resulting in the same error rate. The fact that different subsets of genes can be equally relevant when predicting cancer has been already highlighted in several works [27, 28, 29]. It was argued that one of the possible explanations for such multiplicity and non-uniqueness is a strong influence of the training set on gene selection. In other words, different groups of patients can lead to different gene importance rankings due to genuine differences between patients (cancer grade, stage, etc.).

To mitigate this problem (complete alleviation seems to be currently impossible due to small sample size of gene expression datasets), we propose to employ an ensemble of classifiers instead of a single classifier, where each classifier in the ensemble works with its own feature subset. Potential gains in doing so are twofold: error rate can be significantly reduced and fewer biologically relevant genes can be missed when all subsets of genes are combined together for further analysis.

8 Ensemble of Classifiers

An ensemble of classifiers consists of several base classifiers (members) that make predictions independently of each other. After that, these predictions are combined together to produce the final prediction. Though ensemble members can belong to different types of algorithms, because of our interest in k-NN classifiers we utilize only this algorithm. Moreover, the value of k is fixed to 3 for all ensemble members[6]. As a combination technique, the conventional majority vote was selected in order to demonstrate that ensembles built with our approach demonstrate good performance even when employing simple non-trainable combiners.

It is well known that an ensemble is able to outperform its best performing member if ensemble members make mistakes of different samples so that their predictions are uncorrelated and diverse as much as possible. On the other hand, an ensemble must include a sufficient number of accurate classifiers since if there are only few good votes, they can be easily drowned out among many bad votes. As a result, an ensemble can predict wrongly most of the time.

So far many definitions of diversity were proposed [7, 30], but unfortunately the precise definition is still largely illusive and as commented in [31], the link between diversity of the ensemble members and prediction accuracy of an ensemble is not straightforward. Because of this fact, we decided not to follow any *explicit* definition of diversity, but introduce diversity implicitly instead. Since we fixed the base classifier and its parameter, one of the solutions is to let each ensemble member to work with its own feature subset.

Feature subset selection can be done in two ways: either applying a certain feature selection algorithm or a group of such algorithms, or randomly sampling features from the original feature set. As concluded in [5], differences in classification performance among feature selection algorithms are less significant than performance differences among the error estimators used to implement these algorithms. In other words, the way of how error is computed has a larger influence on classification accuracy than the choice of a feature selection algorithm. Since bolstered resubstitution error is a low-biased and low-variance estimate of classification error, which is what is needed for high dimensional gene expression data, we opt for random feature selection. Figs. 1-8 show that random feature selection leads to diversity since one complexity value corresponds to several different errors. Given that it is difficult to carry out biological analysis of many genes, we restricted the number of genes to be sampled to 50, i.e. each ensemble member works with 1 to 50 randomly

[6] In our opinion, $k = 1$ tends to lead to optimistic estimation of bolstered resubstitution error.

selected (sampled with replacement) genes. This will ensure that the combined list of all genes is not too long.

Based on the abovementioned, two approaches to form ensembles consisting of L classifiers are explored:

1. Randomly select L feature subsets, one subset per classifier, as described above. Classify the data with each classifier and combine votes.

2. Randomly select $M > L$ (e.g. $M = 100$) feature subsets and compute the dataset complexity for each of them. Rank subsets according to their complexity and select L least complex subsets while ignoring the others. Classify the data with each classifier and combine votes.

We will call the first approach conventional to distinguish it from ours, which is the second approach. The typical (and perhaps the earliest) example of the former is [32]. As one can see, the main difference between two approaches lies in the way of choosing feature subsets: in the conventional approach, subsets are chosen regardless of their classification power. As a result, one may equally expect both very good and very bad ensemble predictions. In contrast, in our approach, subsets are chosen based on the measure *directly* related to classification performance. As lower complexity is associated with smaller bolstered resubstitution error as shown in Section 6, selection of the subsets of smaller complexity implies more accurate classifiers included into an ensemble. Thus, with our approach, both diversity and accuracy requirements for ensembles are satisfied. Hence, we can expect better *average* classification performance with our approach compared to the conventional approach.

9 Experimental Results

In ensemble applications to bioinformatics problems, a small and accurate ensemble is of importance, since too many ensemble members would complicate biological understanding of relations among genes. Bearing this in mind, we set the number of 3-NNs (L) in the ensemble to be equal 3, 5, 7, 9, and 11.

Table 4 represents the dataset complexity as estimated by the normalized rank sum statistic W (see Section 4) for different values of L when ensembles were built with our approach. For each dataset, two values are given: average minimum and average maximum complexity (averaging over 100 runs) of the selected feature subsets. It can be observed that complexity for each dataset is rather stable as L grows. Prostate 1 appears to be far more complex than the other datasets while Prostate 2 seems to be the least complex. For the latter, the minimum and maximum complexity stays the same, which implies that the complexity reached saturation during ensemble generation. The fact that saturation happened at L as low as 3 implicitly points to low complexity of Prostate 2. For the conventional ensemble approach ‘avr.max’ often went to a very big value, meaning poor class separation according to the Wilcoxon rank sum test. For comparison, Table 5 lists dataset complexity when all features are considered in computing W. Again Prostate 1 looks the most complex while Prostate 2 and Brain 1 are among least complex.

Table 4: Average minimum and maximum normalized W for feature subsets selected with our ensemble generating approach for various values of L.

Dataset no.		$L = 3$	$L = 5$	$L = 7$	$L = 9$	$L = 11$
1	avr.min	0.1544	0.1517	0.1551	0.1546	0.1547
	avr.max	0.1638	0.1681	0.1725	0.1749	0.1776
2	avr.min	0.1587	0.1584	0.1581	0.1587	0.1584
	avr.max	0.1676	0.1723	0.1769	0.1800	0.1845
3	avr.min	0.0761	0.0760	0.0761	0.0760	0.0759
	avr.max	0.0781	0.0803	0.0818	0.0831	0.0841
4	avr.min	0.1501	0.1494	0.1493	0.1503	0.1498
	avr.max	0.1584	0.1619	0.1655	0.1683	0.1707
5	avr.min	0.3105	0.3117	0.3121	0.3105	0.3104
	avr.max	0.3274	0.3349	0.3417	0.3447	0.3480
6	avr.min	0.0756	0.0756	0.0756	0.0756	0.0756
	avr.max	0.0756	0.0756	0.0756	0.0756	0.0756
7	avr.min	0.1987	0.1979	0.1990	0.1978	0.1995
	avr.max	0.2086	0.2140	0.2174	0.2190	0.2217
8	avr.min	0.2566	0.2565	0.2554	0.2553	0.2563
	avr.max	0.2692	0.2756	0.2798	0.2837	0.2867

Table 5: Unnormalized and normalized rank sum statistic W when all features are used.

Dataset no.	1	2	3	4	5	6	7	8
W	465	409	48	496	1959	45	410	439
N	74	62	34	90	102	34	60	58
normalized W	0.17	0.21	0.08	0.12	0.37	0.009	0.22	0.26

Table 6 summarizes the average bolstered resubstitution error (over 100 runs) and its standard deviation achieved with two ensemble schemes. ‘C’ and ‘O’ stand for the conventional and our approaches to ensemble construction, respectively. It is clearly noticable that both the average error and its standard deviation are smaller for our approach, regardless of the number of k-NNs in the ensemble. It should be noted that one should not seek dependence between ensemble error in Table 6 and feature subset complexity in Table 4, since our hypothesis is only applied to the error of the individual classifiers.

For comparison, we also included experiments with RBF [9], followed by 3-NN classification using selected genes. Table 7 lists the average bolstered resubstitution error and its standard deviation computed over 100 runs when RBF was applied to each dataset prior to 3-NN classification. The third column contains the number of genes retained after filtering. Results of 3-NN classification without prior gene selection are given in the last column.

It can be observed that our ensemble scheme almost always outperforms RBF+3-NN, except for Brain 1 data[7], which were easy to classify according to dataset complexity. In contrast, the conventional scheme was inferior to RBF+3-NN on many more occasions,

[7] $L = 3, 5$.

Table 6: Average bolstered resubstitution error and its standard deviation for two ensemble schemes for different values of L.

		$L = 3$	$L = 5$	$L = 7$	$L = 9$	$L = 11$
1	C	0.141±0.025	0.125±0.024	0.119±0.021	0.110±0.018	0.111±0.018
	O	0.119±0.016	0.105±0.017	0.098±0.014	0.094±0.014	0.088±0.015
2	C	0.110±0.025	0.091±0.019	0.080±0.013	0.074±0.015	0.068±0.013
	O	0.092±0.014	0.077±0.012	0.071±0.014	0.066±0.010	0.064±0.010
3	C	0.129±0.035	0.117±0.032	0.101±0.031	0.092±0.027	0.088±0.028
	O	0.081±0.022	0.062±0.019	0.054±0.017	0.047±0.016	0.045±0.015
4	C	0.177±0.034	0.160±0.040	0.152±0.040	0.143±0.039	0.151±0.043
	O	0.130±0.023	0.113±0.022	0.098±0.019	0.093±0.017	0.089±0.016
5	C	0.141±0.034	0.111±0.026	0.096±0.020	0.084±0.016	0.076±0.014
	O	0.101±0.022	0.078±0.016	0.071±0.012	0.066±0.011	0.063±0.010
6	C	0.046±0.039	0.027±0.026	0.015±0.014	0.013±0.014	0.013±0.013
	O	0.023±0.018	0.011±0.012	0.007±0.008	0.004±0.005	0.004±0.005
7	C	0.172±0.024	0.147±0.021	0.133±0.020	0.128±0.019	0.120±0.016
	O	0.145±0.017	0.119±0.015	0.104±0.014	0.098±0.014	0.092±0.013
8	C	0.188±0.036	0.150±0.029	0.122±0.025	0.103±0.023	0.087±0.019
	O	0.164±0.027	0.130±0.022	0.099±0.020	0.087±0.021	0.075±0.020

Table 7: Average bolstered resubstitution error and its standard deviation 1) when RBF was applied before 3-NN classification (RBF+3-NN) and 2) with 3-NN classification without gene selection.

Dataset no.	RBF+3-NN	#genes	3-NN
1	0.199±0.011	12	0.160±0.005
2	0.107±0.010	3	0.098±0.006
3	0.055±0.010	6	0.074±0.008
4	0.145±0.005	152	0.132±0.001
5	0.117±0.008	2	0.099±0.002
6	0.003±0.003	1	0.029±0.000
7	0.173±0.013	15	0.216±0.009
8	0.217±0.014	37	0.174±0.005

which again confirms the superiority of our approach to ensemble construction.

We also provide a comparison of both conventional and our approaches to ensemble construction versus a single best classifier (SBC) in each case. Let e_{SBC} and e_{ENS} be bolstered resubstitution error achieved with a SBC and an ensemble, respectively. To meet our goal, the following statistics widely used in machine learning and data mining were computed over 100 ensemble generations:

- win-tie-loss count, where ‘win’/‘tie’/‘loss’ means the number of times when an ensemble was superior/equal/inferior in terms of bolstered resubstitution error to a SBC in the ensemble (in other words, the number of times when $e_{ENS} < e_{SBC}$, $e_{ENS} = e_{SBC}$,

$e_{ENS} > e_{SBC}$, respectively).

- 'min. win', 'max. win', 'avr. win' (minimum, maximum, and average differences $e_{SBC} - e_{ENS}$ when an ensemble outperforms its SBC,

- 'min. loss', 'max. loss', 'avr. loss' (minimum, maximum, and average differences $e_{ENS} - e_{SBC}$ when a SBC outperforms an ensemble.

Tables 8-12 contain values of these statistics. If there were no losses, this fact is marked as 'no'. As one can see, both ensemble schemes were superior to a SBC on all eight datasets for the most part. The degree of success, however, varied, depending on dataset complexity. For example, Prostate 2 was much easier to classify compared to other datasets and therefore a SBC often reached the top performance so that an ensemble had nothing to improve on. When analyzing the performance of two ensemble schemes, it was observed that on average, our approach yields better results in the sense that its win (loss) count is typically higher (lower) and the absolute losses to a SBC are lower, too. In contrast, the conventional ensemble generating approach sometimes shows spectacular results (e.g. the high max.win count), but it also suffers many defeats from a SBC. That is, its results are less predictable since there is no control over complexity of the selected feature subsets and hence, if such 'complex' subsets are selected, a SBC can render ensemble efforts to further lower error fruitless. With the explicit selection of least complex subsets, our approach is able to succeed where the comparative approach failed.

Table 8: Comparison of a SBC and two ensembles when $L = 3$.

		win-tie-loss	min.win	max.win	avr.win	min.loss	max.loss	avr.loss
1	C	84/1/15	0.0014	0.0608	0.0234	0.0014	0.0446	0.0137
	O	90/1/9	0.0014	0.0649	0.0277	$< 10^{-4}$	0.0135	0.0066
2	C	84/3/13	0.0032	0.0629	0.0275	$< 10^{-4}$	0.0548	0.0166
	O	96/2/2	0.0016	0.0532	0.0232	0.0065	0.0145	0.0105
3	C	79/2/19	0.0029	0.0824	0.0324	0.0029	0.0324	0.0135
	O	98/0/2	0.0029	0.0794	0.0428	0.0059	0.0059	0.0059
4	C	81/1/18	0.0011	0.0700	0.0355	0.0011	0.0578	0.0252
	O	99/0/1	0.0022	0.0856	0.0434	0.0056	0.0056	0.0056
5	C	86/0/14	0.0039	0.0824	0.0360	0.0020	0.0735	0.0212
	O	92/1/7	0.0020	0.0657	0.0311	0.0020	0.0265	0.0116
6	C	55/7/38	$< 10^{-4}$	0.0618	0.0224	0.0029	0.1412	0.0292
	O	68/4/28	$< 10^{-4}$	0.0559	0.0195	0.0029	0.0382	0.0123
7	C	91/1/8	0.0067	0.0750	0.0358	0.0017	0.0133	0.0081
	O	100/0/0	0.0100	0.0750	0.0426	no	no	no
8	C	93/1/6	0.0052	0.0931	0.0429	0.0172	0.0569	0.0279
	O	99/0/1	0.0017	0.1000	0.0499	0.0103	0.0103	0.0103

Table 9: Comparison of a SBC and two ensembles when $L = 5$.

		win-tie-loss	min.win	max.win	avr.win	min.loss	max.loss	avr.loss
1	C	89/0/11	0.0027	0.0716	0.0310	0.0014	0.0581	0.0219
	O	96/0/4	0.0027	0.0703	0.0361	0.0068	0.0405	0.0196
2	C	93/1/6	0.0016	0.0694	0.0359	0.0032	0.0274	0.0137
	O	100/0/0	0.0016	0.0597	0.0340	no	no	no
3	C	76/2/22	0.0029	0.0853	0.0352	0.0029	0.0735	0.0217
	O	99/1/0	0.0059	0.1000	0.0536	no	no	no
4	C	84/0/16	0.0033	0.1022	0.0449	0.0022	0.0800	0.0296
	O	99/0/1	0.0122	0.1089	0.0578	0.0022	0.0022	0.0022
5	C	96/0/4	0.0029	0.0902	0.0455	0.0078	0.0333	0.0159
	O	92/0/8	0.0010	0.0784	0.0386	$< 10^{-4}$	0.0167	0.0056
6	C	60/11/29	$< 10^{-4}$	0.0765	0.0232	0.0029	0.0941	0.0221
	O	80/4/16	$< 10^{-4}$	0.0588	0.0201	0.0029	0.0176	0.0083
7	C	100/0/0	0.0033	0.0883	0.0484	no	no	no
	O	100/0/0	0.0200	0.0933	0.0602	no	no	no
8	C	99/0/1	0.0121	0.1121	0.0682	0.0121	0.0121	0.0121
	O	100/0/0	0.0328	0.1224	0.0763	no	no	no

Table 10: Comparison of a SBC and two ensembles when $L = 7$.

		win-tie-loss	min.win	max.win	avr.win	min.loss	max.loss	avr.loss
1	C	90/2/8	0.0014	0.0649	0.0308	0.0014	0.0297	0.0084
	O	96/1/3	0.0054	0.0662	0.0353	0.0081	0.0270	0.0158
2	C	99/0/1	0.0032	0.0726	0.0400	0.0113	0.0113	0.0113
	O	99/1/0	0.0016	0.0694	0.0353	no	no	no
3	C	88/2/10	0.0029	0.0971	0.0394	0.0088	0.0412	0.0235
	O	99/0/1	0.0088	0.1088	0.0566	0.0029	0.0029	0.0029
4	C	80/0/20	0.0022	0.1256	0.0474	0.0044	0.0700	0.0267
	O	100/0/0	0.0178	0.1033	0.0652	no	no	no
5	C	96/0/4	0.0039	0.1078	0.0502	0.0069	0.0363	0.0223
	O	96/0/4	0.0039	0.0843	0.0391	0.0029	0.0147	0.0078
6	C	73/3/24	$< 10^{-4}$	0.0529	0.0194	0.0029	0.0618	0.0105
	O	79/11/10	0.0029	0.0500	0.0176	0.0029	0.0176	0.0068
7	C	100/0/0	0.0183	0.0983	0.0595	no	no	no
	O	100/0/0	0.0267	0.1067	0.0701	no	no	no
8	C	99/0/1	0.0241	0.1379	0.0874	0.0483	0.0483	0.0483
	O	100/0/0	0.0414	0.1362	0.0956	no	no	no

10 Conclusion

We proposed a new ensemble generating scheme using a k-NN as a base classifier. Our approach leads to lower bolstered resubstitution error compared to the conventional ensemble approach, purely based on random selection of features, and to a single best classifier in the

Table 11: Comparison of a SBC and two ensembles when $L = 9$.

		win-tie-loss	min.win	max.win	avr.win	min.loss	max.loss	avr.loss
1	C	92/1/7	0.0014	0.0811	0.0329	0.0027	0.0270	0.0097
	O	97/0/3	0.0027	0.0824	0.0375	0.0014	0.0203	0.0086
2	C	99/0/1	0.0016	0.0903	0.0424	0.0226	0.0226	0.0226
	O	100/0/0	0.0032	0.0677	0.0363	no	no	no
3	C	91/1/8	0.0059	0.1206	0.0405	0.0029	0.0382	0.0158
	O	100/0/0	0.0176	0.1029	0.0598	no	no	no
4	C	83/1/16	0.0011	0.1011	0.0491	0.0011	0.1022	0.0326
	O	100/0/0	0.0322	0.1056	0.0709	no	no	no
5	C	98/0/2	0.0020	0.0990	0.0541	0.0020	0.0039	0.0029
	O	100/0/0	0.0029	0.0765	0.0420	no	no	no
6	C	72/5/23	$< 10^{-4}$	0.0618	0.0189	0.0029	0.0324	0.0106
	O	76/8/16	0.0029	0.0500	0.0173	0.0029	0.0118	0.0051
7	C	100/0/0	0.0167	0.1017	0.0611	no	no	no
	O	100/0/0	0.0400	0.1183	0.0767	no	no	no
8	C	100/0/0	0.0310	0.1655	0.1036	no	no	no
	O	100/0/0	0.0207	0.1603	0.1038	no	no	no

Table 12: Comparison of a SBC and two ensembles when $L = 11$.

		win-tie-loss	min.win	max.win	avr.win	min.loss	max.loss	avr.loss
1	C	91/0/9	0.0014	0.0635	0.0304	0.0014	0.0284	0.0135
	O	97/2/1	0.0054	0.0811	0.0401	0.0014	0.0014	0.0014
2	C	100/0/0	0.0016	0.0806	0.0469	no	no	no
	O	100/0/0	0.0081	0.0726	0.0376	no	no	no
3	C	83/6/11	0.0029	0.0853	0.0422	0.0059	0.0676	0.0193
	O	98/0/2	0.0147	0.0941	0.0562	0.0029	0.0029	0.0029
4	C	82/0/18	0.0022	0.0967	0.0422	0.0056	0.0911	0.0380
	O	100/0/0	0.0256	0.1133	0.0741	no	no	no
5	C	100/0/0	$< 10^{-4}$	0.0990	0.0566	no	no	no
	O	98/0/2	0.0059	0.0794	0.0412	0.0088	0.0108	0.0098
6	C	76/6/18	0.0029	0.0353	0.0168	0.0029	0.0412	0.0108
	O	67/21/12	0.0029	0.0441	0.0158	0.0029	0.0265	0.0074
7	C	100/0/0	0.0200	0.1083	0.0653	no	no	no
	O	100/0/0	0.0383	0.1167	0.0766	no	no	no
8	C	100/0/0	0.0448	0.1603	0.1126	no	no	no
	O	100/0/0	0.0603	0.1603	0.1112	no	no	no

ensemble. In addition, our scheme outperforms a 3-NN preceded by the RBF algorithm [9], especially proposed to deal with redundancy among genes.

Our approach originates from the link between dataset complexity and bolstered resubstitution error established through the copula method. We found that there is positive dependence between complexity and error, where low (high) complexity corresponds to small (large) error. Hence, the dataset complexity serves as a reliable indicator of the expected classification performance. As a result, selection of least complex subsets of features implies more accurate ensemble members and therefore it ensures better ensemble performance. Extensive experiments with eight gene expression datasets containing different types of cancer show feasibility of our approach. Its extra attractiveness comes from the fact that good ensemble performance is achieved with a few 3-NNs (3 to 11), which limits the number of genes to further analyze.

A Derivatives for the Frank Copula

The definition of the Frank copula is

$$C_\theta(u,v) = -\frac{1}{\theta}\ln\left(1+\frac{(e^{-\theta u}-1)(e^{-\theta v}-1)}{e^{-\theta}-1}\right). \tag{7}$$

The first-order derivative of $C_\theta(u,v)$ wrt u is

$$\frac{\partial C_\theta(u,v)}{\partial u} = -\frac{1}{\theta}\frac{1}{1+\frac{(e^{-\theta u}-1)(e^{-\theta v}-1)}{e^{-\theta}-1}}\frac{e^{-\theta v}-1}{e^{-\theta}-1}(-\theta)e^{-\theta u}.$$

After simplifications, we obtain that

$$\frac{\partial C_\theta(u,v)}{\partial u} = \frac{e^{-\theta u}(e^{-\theta v}-1)}{e^{-\theta}-1+(e^{-\theta u}-1)(e^{-\theta v}-1)}. \tag{8}$$

By analogy, the first-order partial derivative $C_\theta(u,v)$ wrt v is

$$\frac{\partial C_\theta(u,v)}{\partial v} = \frac{e^{-\theta v}(e^{-\theta u}-1)}{e^{-\theta}-1+(e^{-\theta u}-1)(e^{-\theta v}-1)}. \tag{9}$$

Then, the second-order partial derivative $C_\theta(u,v)$ wrt u is

$$\begin{aligned}
\frac{\partial^2 C_\theta(u,v)}{\partial u^2} &= \frac{(e^{-\theta v}-1)}{[e^{-\theta}-1+(e^{-\theta u}-1)(e^{-\theta v}-1)]^2}\Big[\theta e^{-2\theta u}(e^{-\theta v}-1) \\
&- \theta e^{-\theta u}\Big(e^{-\theta}-1+(e^{-\theta u}-1)(e^{-\theta v}-1)\Big)\Big] \\
&= \frac{\theta e^{-\theta u}(e^{-\theta v}-1)}{[e^{-\theta}-1+(e^{-\theta u}-1)(e^{-\theta v}-1)]^2}\Big[e^{-\theta u}(e^{-\theta v}-1) \\
&- e^{-\theta}+1-(e^{-\theta u}-1)(e^{-\theta v}-1)\Big] \\
&= \frac{\theta e^{-\theta u}(e^{-\theta v}-1)}{[e^{-\theta}-1+(e^{-\theta u}-1)(e^{-\theta v}-1)]^2}\Big[e^{-\theta(u+v)}-e^{-\theta u} \\
&- e^{-\theta}+1-e^{-\theta(u+v)}+e^{-\theta v}+e^{-\theta u}-1\Big].
\end{aligned}$$

After some terms cancel out each other, we obtain that

$$\frac{\partial^2 C_\theta(u,v)}{\partial u^2} = \frac{\theta e^{-\theta u}(e^{-\theta v}-1)(e^{-\theta v}-e^{-\theta})}{\left[e^{-\theta}-1+(e^{-\theta u}-1)(e^{-\theta v}-1)\right]^2}, \tag{10}$$

and by analogy, the second-order partial derivative $C_\theta(u,v)$ wrt v is

$$\frac{\partial^2 C_\theta(u,v)}{\partial v^2} = \frac{\theta e^{-\theta v}(e^{-\theta u}-1)(e^{-\theta u}-e^{-\theta})}{\left[e^{-\theta}-1+(e^{-\theta u}-1)(e^{-\theta v}-1)\right]^2}. \tag{11}$$

Given that $\theta > 0$ (positive dependence observed between dataset complexity and bolstered resubstitution error), the following pairs of inequalities hold $\forall u,v \in [0,1]$:

$$\begin{aligned} e^{-\theta v}-1 \le 0, \quad e^{-\theta v}-e^{-\theta} \ge 0, \\ e^{-\theta u}-1 \le 0, \quad e^{-\theta u}-e^{-\theta} \ge 0. \end{aligned}$$

Hence, the product of the inequalities in each row above is less than or equal to zero. Given that other terms in Eqs. 10-11 are positive, it means that $\frac{\partial^2 C_\theta(u,v)}{\partial u^2} \le 0$ and $\frac{\partial^2 C_\theta(u,v)}{\partial v^2} \le 0$, which, in turn, implies that $C_\theta(u,v)$ is concave.

B Marsaglia Polar Method

This is the polar form of the Box-Müller transformation [33] intended to generate Gaussian pseudo-random numbers from the uniform pseudo-random numbers. Its C-like pseudocode is given below, where *rand()* is the function for uniform [0,1] random number generation, p is the data dimensionality, n is equal to M_{MC} (see Section 5), *log()* is the natural logarithm, m and s are the mean and the standard deviation, respectively. After each iteration over i two samples are generated and stored in X so that after $n/2$ iterations, we have $2M_{MC}/2 = M_{MC}$ samples.

```
for (i = 0; i < n/2; i++)
    {
        /* Generate normal random numbers */
        for (j = 0; j < p; j++)
        {
            do
            {
                u1 = 2.0*rand()/RAND_MAX - 1;
                u2 = 2.0*rand()/RAND_MAX - 1;
                r = u1*u1 + u2*u2;
            } while(r == 0 || r >= 1);
            r = sqrt(-2*log(r)/r);
            X[i*p+j)] = m[j] + s[j]*r*u1;
            X[(i+n/2)*p+j] = m[j] - s[j]*r*u2;
        }
    }
```

References

[1] Dougherty, E. R., Shmulevich, I., Chen, J. & Wang, Z. J. (2005) *Genomic Signal Processing and Statistics*. New York, NY: Hindawi Publishing Corporation.

[2] Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D. & Levine, A. J. (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, *Proceedings of the National Academy of Sciences 96* 6745–6750.

[3] Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., McLaughlin, M. E., Kim, J. Y. H., Goumnerova, L. C., Black, P. M., Lau, C., Allen, J. C., Zagzag, D., Olson, J. M., Curran, T., Wetmore, C., Biegel, J. A., Poggio, T., Mukherjee, S., Rifkin, R., Califano, A., Stolovitzky, G., Louis, D. N., Mesirov, J. P., Lander, E. S., & Golub, T. R. (2002) Prediction of central nervous system embryonal tumour outcome based on gene expression, *Nature 415* 436–442.

[4] Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A. A., D'Amico, A. V., Richie, J. P., Lander, E. S., Loda, M., Kantoff, P. W., Golub, T. R., & Sellers, W. R. (2002) Gene expression correlates of clinical prostate cancer behavior, *Cancer Cell 1* 203–209.

[5] Sima, C., Attoor, S., Braga-Neto, U., Lowey, J., Suh, E., & Dougherty, E. R. (2005) Error estimation confounds feature selection in expression-based classification, *Proceedings of the IEEE International Workshop on Genomic Signal Processing and Statistics* (Newport, Rhode Island).

[6] Braga-Neto, U. & Dougherty, E. R. (2004) Bolstered error estimation, *Pattern Recognition 37* 1267–1281.

[7] Kuncheva, L. (2004) *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ: John Wiley & Sons.

[8] Dudoit, S. & Fridlyand, J. (2003) Classification in microarray experiments, in: T. Speed, ed., *Statistical Analysis of Gene Expression Microarray Data*. Boca Raton, FL: Chapman & Hall\CRC Press, 93–158.

[9] Yu, L. (2008) Feature selection for genomic data analysis, in H. Liu & H. Motoda, eds., *Computational Methods of Feature Selection*. Boca Raton, FL: Chapman & Hall\CRC, 337–354.

[10] Sklar, A. (1959) Fonctions de répartition à n dimensions et leurs marges, *Publications of the Institute of Statistics, University of Paris* 229–231.

[11] Nelsen, R. B. (2006) *An Inroduction to Copulas*. New York, NY: Springer Science+Business Media.

[12] Joe, H. (1997) *Multivariate Models and Dependence Concepts*. Boca Raton, FL: Chapman & Hall\CRC Press.

[13] Zar, J. H. (1999) *Biostatistical Analysis*. Upper Saddle River, NJ: Prentice Hall.

[14] http://www.sagenet.org

[15] Velculescu, V. E., Zhang, L., Vogelstein, B., & Kinzler, K. W. (1995) Serial analysis of gene expression, *Science 270* 484–487.

[16] Aldaz, M. C. (2003) Serial analysis of gene expression (SAGE) in cancer research, in: M. Ladanyi and W.L. Gerald, eds., *Expression Profiling of Human Tumors: Diagnostic and Research Applications*. Totowa, NJ: Humana Press, 47–60.

[17] http://lisp.vse.cz/challenge/ecmlpkdd2004

[18] Gandrillon, O. (2004) Guide to the gene expression data, in: P. Berka and B. Crémilleux, eds., *Proceedings of the ECML/PKDD Discovery Challenge Workshop* (Pisa, Italy, 2004) 116–120.

[19] http://microarray.princeton.edu/oncology/affydata/index.html

[20] http://www.broad.mit.edu/mpr/CNS/

[21] http://www.broad.mit.edu/cgi-bin/cancer/publications/pub_paper.cgi?mode/ =view&paper_id=75

[22] http://www.broad.mit.edu/mpr/lymphoma/

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

[24] Bø, T. H. & Jonassen, I. (2002) New feature subset selection procedures for classification of expression profiles, *Genome Biology 3* 0017.1–0017.11.

[25] Ho, T. K. & Basu, M. (2002) Complexity measures of supervised classification problems, *IEEE Transactions on Pattern Analysis and Machine Intelligence 24* 289–300.

[26] Schweizer, B. & Wolff, E. F. (1981) On nonparametric measures of dependence for random variables, *The Annals of Statistics 9* 879–885.

[27] Ein-Dor, L., Kela, I., Getz, G., Givol, D., & Domany, E. (2005) Outcome signature genes in breast cancer: is there a unique set?, *Bioinformatics 21* 171–178.

[28] Michiels, S., Koscielny, S., & Hill, C. (2005) Prediction of cancer outcome with microarrays: a multiple random validation strategy. *Lancet 365* 488–492.

[29] Díaz-Uriarte R. & Alvarez de Andrés, S. (2006) Gene selection and classification of microarray data using random forest, *BMC Bioinformatics 7*.

[30] Kuncheva L. & Whitaker, C. J. (2003) Measures of diversity in classifier ensembles, *Machine Learning 51* 181–207.

[31] Kuncheva L. & Rodríguez, J. J. (2007) Classifier ensembles with a random linear oracle, *IEEE Transactions on Knowledge and Data Engineering 19* 500–508.

[32] Bay, S. (1999) Nearest neighbor classification from multiple feature sets, *Intelligent Data Analysis 3* 191–209.

[33] Box G. E. P. & Müller, M. E. (1958) A note on the generation of random normal deviates, *The Annals of Mathematical Statistics 29* 610–611.