Top Scoring Pairs for Feature Selection in Machine Learning with Applications to Cancer Outcome Prediction

Ping Shi\textsuperscript{1,2,§}, Surajit Ray\textsuperscript{2}, Qifu Zhu\textsuperscript{3}, Mark A. Kon\textsuperscript{2}
\textsuperscript{1}Department of Population Medicine, Harvard Medical School and Harvard Pilgrim Health Care Institute
\textsuperscript{2}Department of Mathematics and Statistics and Bioinformatics Program, Boston University
\textsuperscript{3}Trilion Quality Systems

\textsuperscript{§}Corresponding author

Email addresses:
PS: pcschi@gmail.com
MK: mkon@bu.edu
SR: sray@bu.edu
QZ: zhu@trilion.com
Abstract

Background: The widely used $k$ top scoring pair ($k$-TSP) algorithm is a simple yet powerful parameter-free classifier. It owes its success in many cancer microarray datasets to an effective feature selection algorithm that is based on relative expression ordering of gene pairs. However, its general robustness does not extend to some difficult datasets, such as those involving cancer outcome prediction, which may be due to the relatively simple voting scheme used by the classifier. We believe that the performance can be enhanced by separating its effective feature selection component and combining it with a powerful classifier such as the support vector machine (SVM). More generally the top scoring pairs generated by the $k$-TSP ranking algorithm can be used as a dimensionally reduced subspace for other machine learning classifiers.

Results: We developed an approach integrating the $k$-TSP ranking algorithm (TSP) with other machine learning methods, allowing combination of the computationally efficient, multivariate feature ranking of $k$-TSP with multivariate classifiers such as SVM. We evaluated this hybrid scheme ($k$-TSP+SVM) in a range of simulated datasets with known data structures. As compared with other feature selection methods, such as a univariate method similar to Fisher’s discriminant criterion (Fisher), or a recursive feature elimination embedded in SVM (RFE), TSP is increasingly more effective than the other two methods as the informative genes become progressively more correlated, which is demonstrated both in terms of the classification performance and the ability to recover true informative genes. We also applied this hybrid scheme to three cancer prognosis datasets, in which $k$-TSP+SVM outperforms $k$-TSP classifier in all datasets, and achieves either comparable or superior performance to that using SVM alone. In concurrence with what is observed in simulation, TSP appears to be a better feature selector than Fisher and RFE in two of the three cancer datasets.

Conclusions: The $k$-TSP ranking algorithm can be used as a computationally efficient, multivariate filter method for feature selection in machine learning. SVM in combination with $k$-TSP ranking algorithm outperforms $k$-TSP and SVM alone in simulated datasets and in some cancer prognosis datasets. Simulation studies suggest that as a feature selector, it is better tuned to certain data characteristics, i.e. correlations among informative genes, which may have indications in its usefulness as an alternative feature ranking method in pathway analysis.
**Background**

Gene expression profiling has been proved to be a valuable resource for classification of complex diseases such as cancer. Many studies have showed it is possible to extract compelling information from microarray data to support clinical decisions on cancer diagnosis, prognosis and response to treatment [1-3]. However, like other high-throughput studies, microarray data pose great challenges to accurate prediction in two respects. On one hand, there is a large amount of inherent noise and variability in samples, due to biological variations and experimental conditions, both of which can degrade the prediction performance. On the other hand, difficulties also arise from high dimensionality (on the order of tens of thousands), as compared to a relatively small sample size (usually on the order of tens), which leads to the risk of over-fitting in many machine learning methods. This occurs especially in cases when the few training samples are not good representatives of classes, so that the classifier may learn inherent noise from irrelevant features in training data, leading to poor generalizability. Thus algorithmically speaking, there is a need for feature selection in order to improve model performance and avoid over-fitting.

Generally, feature selection methods in the context of gene expression studies can be divided into three categories: filter methods, wrapper methods, and embedded methods [4]. Filter approaches [5-8], which involve the use of feature relevance scores to select a subset of high-scoring features, are computationally efficient and thus widely used. They are also independent of classifier algorithms. On the contrary, wrapper and embedded approaches interact with classifiers by either wrapping around the classifiers [9-12], or being built within the classifier construction [13-15]. As a result, those approaches are usually much more computationally intensive than filters, and sometimes become so costly as to be impractical without pre-reduction of the search space with a filter method. Whereas filter techniques are frequently univariate in nature, assuming the features are independent and ignoring feature dependencies, wrapper and embedded techniques select features in a multivariate fashion by taking feature correlations into account, an approach which is certainly biologically relevant if we consider, e.g., how genes are co-regulated in pathways.

As more feature selection techniques are explored, computational efficiency and the ability to capture feature interactions remain important considerations. In 2004, Geman *et al.* introduced the top-scoring pair (TSP) classifier [16], which was then refined and extended to $k$-TSP by Tan, *et al.* [17], using $k$ pairs of top ranking genes to build the classifier. There are several things worth noting in the feature ranking algorithm they employed. First, they replace actual expression levels with their ranks in each sample, which reduces inherent noise and is invariant to normalization procedures across platforms. Second, the idea of “relative expression reversals”, on which the pairs are scored, captures a key mechanism of the disease process, which involves large differential changes of expression levels in up-regulated and down-regulated genes. Third, this scoring algorithm is multivariate at different levels, in the sense that bivariate interactions between features are exploited when genes are evaluated in pairs, and higher order interactions are addressed when the final ranking is derived from comparison of scores of all possible pairs. Last, by choosing reversed expression pairs, it is guaranteed
that features from more than one cluster are collected, and the top-ranked features coming from different clusters can contribute orthogonal information to the classifier. With all the above unique qualities in its feature selection, \( k \)-TSP turns out to be a powerful classification method using simple comparisons, and in some cases, it is able to extract a single pair of genes for accurate diagnostic prediction. Nonetheless, given all its success in cancer diagnostic prediction, it has been noted that the performance of \( k \)-TSP is not always robust for some more difficult types, such as those involving cancer outcome prediction. This could be due to the complexity of data on one hand, and the relatively simple voting scheme of \( k \)-TSP classifier after the feature selection step on the other hand.

Compared to other complex machine learning methods, the support vector machine (SVM) is relatively insensitive to high dimensionality [18], though its performance can still improve considerably after feature selection [19]. Guyon \textit{et al.} proposed the recursive feature elimination algorithm (RFE) [13], using the SVM weight vector to rank component features and discarding those with small weights in a recursive manner. This strategy is a multivariate approach embedded in the construction of the SVM classifier, and like other embedded methods, is relatively computationally intensive. Recently, Yoon \textit{et al.} reported their comparison of feature selection methods in combination with SVM in a number of cancer diagnostic datasets [20], suggesting that TSP may be an effective filter method candidate for other classifiers, if computational cost is not a concern.

In this paper, we describe a hybrid approach that integrates the TSP scoring algorithm into other machine learning methods such as SVM and \( k \)-nearest neighbors (KNN). A particular focus is placed on assessing this approach in a controlled environment by using simulated datasets with known properties, including correlation structures of informative genes (signal genes), the variance distribution in all genes, signal strength and sparsity, and sample size in the training set. We examine two aspects of responses to changes of these data characteristics. First, we consider how TSP, as a feature selection scheme, compares to other methods such as the univariate filter Fisher method, or the multivariate RFE method. Next, we look at how the hybrid scheme (\( k \)-TSP+SVM), as a classifier, compares to \( k \)-TSP or SVM using all existing features. We have found that increasing correlation among the signal genes renders TSP an increasingly preferable feature selector over the other two. Correspondingly, this increased correlation augments the difference in performance between \( k \)-TSP+SVM and SVM alone. Moreover, we apply SVM+TSP to three cancer prognostic datasets, and show that it achieves superior performance to \( k \)-TSP, and either outperforms or compares to that using SVM alone.

In general, different methods have their respective strengths and weaknesses in dealing with different aspects of data complexity, and a hybrid approach combining one algorithm with another can be beneficial [21-22]. This is also demonstrated in our study, both in real and simulated data, where sophisticated classifiers are enhanced by the feature selection scheme carved out from another learning algorithm.
Results

Simulated datasets

Simulation process

To investigate how feature selection methods respond to different data structures, we generated two types of data, with variations in aspects such as the strength of differentially expressed genes (signal genes), the sparseness of signal genes, and covariance structure. The basic model is as follows: each sample contains 1000 genes, of which 100 are signal genes. The signal genes follow the multivariate normal distribution $N(\mu, \Sigma)$ for class 1, and $N(-\mu, \Sigma)$ for class 2, with $\mu$ being a vector of 10 distinct values ranging from $-0.25$ to $0.25$ with an increment of $0.05$, each value being the effect size of 10 differentially expressed genes (denoted as $\mu_i$). The rest of the 900 noise genes follow independent $N(0, 1)$ distributions for both classes. For each simulation experiment, 150 independent samples were generated (75 for each class), from which 100 were randomly selected as the training set, with the remaining 50 used as the test set.

Based on this general model, the first type of data (Data-I) has all of the signal genes in one block $\Sigma$. The signal genes are sampled under an independent model or a correlated model. For the correlated model, the block has a compound symmetry (CS) structure with variances of 1 and a common correlation coefficient $\rho$, as shown below:

$$
\Sigma = \begin{bmatrix}
1 & \rho & \rho & \cdots & \rho \\
\rho & 1 & \rho & \cdots & \rho \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
\rho & \rho & \cdots & \rho & 1
\end{bmatrix}
$$

In the second type of data (Data-II), the signal genes have 10 equal blocks, each consisting of 10 genes with a distinct value of effect size, and a CS correlation structure $\Sigma_i = \Sigma$, so each $\Sigma_i$ has an equicorrelated structure. The blocks can be correlated among themselves by introducing an inter-block correlation coefficient $\rho'$, which is always smaller than the within-block correlation $\rho$ to ensure the covariance matrix is positive definite.

To mimic various situations in real datasets, we generated different variants of Data-I: Data-Ib, Data-Ic, and Data-Id. Data-Ib is constructed based on the randomized variance model (Wright2003), with the variances drawn from an inverse gamma distribution, instead of taking on a fixed value of 1. Data-Ic and Data-Id reduce the proportion of signal genes from 10% to 1%, with Data-Ic containing the same $\mu_i$ vector as that in Data-I, and Data-Id containing a $\mu$ vector with a larger effect size, which has half of the values in the vector equaling $-0.25$, and half 0.25 (denoted as $\mu_{3b}$).
Comparison of TSP with Fisher and RFE as feature selection methods

We first compared the performance of SVM on Data-I and Data-II, using TSP, Fisher and RFE as feature selection methods. In each experiment, we applied the TSP, Fisher and RFE feature ranking algorithms to rank the genes, built SVM models with each level of selected genes on the training data, and then tested the models on the test set. Experiments were repeated 50 times to generate averaged results. Two aspects of performance are evaluated in the test data: the classification error rate at each level of gene selection, and the percentage of true signal genes recovered among the top 30 and top 60 ranked genes.

Table 1 shows the classification error rates in Data-I at different levels of selected genes, with correlation coefficients at 0, 0.4 and 0.6. As can be seen, when the signal genes are independent, the different feature selection methods seem to perform comparably to each other (with Fisher slightly outperforming RFE), and the performance using all features is barely improved by using feature selection. However, as the signal genes become more correlated, TSP turns out to be increasingly advantageous over Fisher and RFE. We can take maximum error reduction (MER) as an indicator for the effectiveness of each method ((error rate at full feature level - minimum error rate at a selection level) / error rate at full feature level). When \( \rho = 0.4 \) the MERs are 20.7%, -6.0% and 8.6% for TSP, Fisher and RFE respectively, whereas at \( \rho = 0.6 \) they are 61.7%, -0.9%, and 24.3% respectively. This tendency is further illustrated in Figure 1A, which shows that in response to the progressively increased correlation among signal genes, the effectiveness of the TSP gradually out-races Fisher and RFE, achieving increasingly lower error rates than those of the other two at most selection levels.

When the signal genes are in the multi-block structure of Data-II, we do not observe the differential responses of the three feature selection methods as the signal genes within each block (within-block correlation) become more correlated. However, if an inter-block correlation \( \rho' \) is introduced among all the blocks, we observe the same pattern of differentiated responses among the three methods. In Figure 1B, TSP, Fisher and RFE have comparable performance when the blocks are uncorrelated (\( \rho' = 0 \)). Nevertheless, when the blocks become correlated among one another, the effectiveness of these three methods diverges. When the inter-block correlation is moderate (\( \rho' = 0.35 \)), TSP significantly outperforms Fisher and RFE at a few selection levels (top right). The differentiation is more pronounced when the inter-block correlation is stronger (\( \rho' = 0.5 \)), where TSP is significantly better than Fisher and RFE at most selection levels (bottom right).

TSP, Fisher and RFE are also applied as feature selectors for another benchmark classifier \( k \)-nearest neighbors (KNN), with \( k = 3 \) as the number of nearest neighbors. As seen in Figure 1C, again TSP, Fisher and RFE perform comparably well when the signal genes in Data-I are independent, and all three improve on the performance of KNN using the entire set of features. As the correlation among signal genes become increasingly stronger, the performance of TSP starts to set apart from Fisher and RFE at \( \rho = 0.3 \), and is superior to the other two at most selection levels at \( \rho = 0.45 \), with the gap further
In parallel with classification performance, the recovery of signal genes among the top-ranked genes by TSP, Fisher and RFE displays a similar trend in response to the increased correlation among the signal genes. Figure 2 shows the percentage of the true signal genes recovered in the top 30 and top 60 ranked genes selected by the three ranking algorithms in Data-I and Data-II. In single block structure (Data-I), it can be seen that when the signal genes are uncorrelated, Fisher recovers slightly more signal genes than TSP and RFE (65% versus 53% and 46% respectively in the top 30 genes). As the correlation of signal genes progresses from 0.3 to 0.6, TSP gradually out-races Fisher to be the one that recovers most signal genes (from 53% to 91%), while the recovery rate for Fisher remains unchanged, and that of RFE increases to a smaller extent (from 45% to 61%). A similar trend is observed for multi-block data structures (Data II), where the presence of inter-block correlation turns TSP into the leading feature selector for recovering signal genes (Figure 2B).

**Comparison of k-TSP+SVM with k-TSP and SVM**

We compared the classification performance of TSP family classifiers with our hybrid scheme using $k$-TSP as feature selection for SVM ($k$-TSP+SVM) in Data-I and Data-II, as well as some variants constructed based on Data-I. In each experiment, the TSP ranking algorithm was used to rank the genes and build the model on training data at each level of selected genes through a standard leave one out cross-validation (LOOCV) procedure. The level that achieved the minimum LOOCV error rate was chosen as the size of gene subset, with which the classifier is built on the entire training set and then applied to the test data. Experiments were repeated 20 times to generate averaged test error rates, which were used to evaluate the performance of a classifier.

Table 2A summarizes the classification performance of TSP, $k$-TSP, SVM, $k$-TSP+SVM, Fisher+SVM and RFE+SVM in Data-I and Data-Ib, the latter being a variant of Data-I whose variances follow an inverse gamma distribution with parameters $a = 2$ and $b = 1$. In Data-I, both $k$-TSP and $k$-TSP+SVM improve with increased correlation, with $k$-TSP+SVM (24.7%, 15.6%, and 9.7%) significantly outperforming $k$-TSP (28.9%, 22.6% and 15.5%) in all conditions. In contrast, SVM alone does not seem to pick up its performance as the correlation increases, and so is increasingly outperformed by $k$-TSP+SVM when the correlation becomes stronger (15.6% and 9.7% vs 24.2% and 19.7%). Data-Ib, the dataset with a random variance structure, displays a similar trend, except that both $k$-TSP and $k$-TSP+SVM outperform SVM alone to a greater extent. It is also observed that between the TSP family classifiers, $k$-TSP is invariably superior to TSP. Meanwhile, RFE+SVM also improves with increased correlation in all cases, though much less robustly than $k$-TSP+SVM, whereas Fisher+SVM remains mostly unchanged in Data-I.

To investigate the impact of sparsity of the signal genes on classification, we created Data-Ic and Data-Id, which only contain one tenth as many signal genes as in Data-I. Interestingly, it is shown in Table 2B that when the percentage of signal genes is reduced

widened at $\rho = 0.6$. In parallel, RFE also increases its performance in response to correlation, but is out-raced by TSP.
to from 10% to 1% in Data-Ic, the datasets become difficult for all the classifiers and none appears to be effective regardless of the presence of correlation. However, when the signal strength of the signal genes is increased from $\mu_s$ to $\mu_{3b}$ in Data-Id, $k$-TSP+SVM steps over the others again, showing more robustness in rapidly improving its performance with increased correlation, and outperforming $k$-TSP and SVM at $\rho = 0.45$ (31.7% vs 40.3% and 43.4%), and $\rho = 0.6$ (23.0% vs 33.6% and 42.7%).

When signal genes are organized in multiple block structures, in which signal genes are correlated within each block ($\rho = 0.6$ for Data-IIb), a disparate picture emerges. When the blocks are uncorrelated with one another ($\rho = 0$), the performance of all the classifiers degrade drastically, and $k$-TSP+SVM does not show any advantage. In contrast, when the blocks are correlated, each classifier significantly improves its performance (Table 2C), which is shown in Data-IIb ($\rho = 0.6$, $\rho' = 0.5$); with $k$-TSP, SVM and $k$-TSP +SVM achieving comparable performance (24.2%, 24.6% and 23%).

The effect of sample size in training data
In many microarray studies, sample sizes in training sets are usually limited. It has been suggested that the TSP ranking algorithm is sensitive to the perturbation of training samples [17]. To assess this effect by simulation, we generated datasets of Data-I with sample sizes of 25, 50, 75 and 100 in the training sets, with signal genes moderately correlated ($\rho = 0.45$). As shown in Figure 3, all the classifiers improve their performance as the sample size increases. When the training size is only 25, the performances of all classifiers deteriorate, indicating it almost becomes impossible to train a classifier with such a small training size. As the training set becomes larger, TSP and Fisher+SVM appear to be significantly less effective than the rest, in which $k$-TSP+SVM is comparable to others. As the sample size reaches 100, $k$-TSP+SVM (9.7%) rises above others, significantly outperforming $k$-TSP (15.5%), SVM (19.7%), Fisher+SVM (21.4%) and RFE+SVM (17.9%) (Table 2A).

Real datasets
Cancer prognostic datasets
We applied the above hybrid scheme $k$-TSP+SVM to three cancer prognostic datasets. The first dataset is van’t Veer’s breast cancer dataset (van’t Veer2002), obtained from Rosetta Inpharmatics, which is already partitioned into training and test data. The training data consists of 78 patients, 34 of whom developed distance metastases or died within 5 years (poor prognosis), with the rest consisting of those remained healthy for an interval of more than 5 years (good prognosis). The test data consists of 19 patients, 12 with poor prognosis and 7 with good prognosis. Since this dataset contains many missing values, certain pre-processing was performed. First, two samples (one from each prognosis) with more than 50% of missing gene values in the training data were removed. Next, any genes whose value was missing in at least one sample was discarded, amounting to a total of 3.5% of all genes. The log–transformed ratio of the two channels was used for analysis.
The other two cancer prognostic datasets are obtained from the cancer dataset depository of the Broad Institute. One is a dataset of 86 patients with primary lung adenocarcinoma, which consists of 62 patients who were alive, and 24 patients who had died (Beer et al 2002). The other is a dataset of 60 patients with medulloblastomas, which consists of 39 survivors and 21 treatment failures after radiation and chemotherapy (Pomeroy et al 2002). Both datasets are pre-processed and contain 7129 genes.

**Application to cancer prognostic datasets**

The classification performance of $k$-TSP+SVM is compared with $k$-TSP and SVM in the three cancer prognostic datasets. We used the independent test set when it was available from the original dataset; otherwise we performed 5-fold cross-validation and averaged the results from two 5-fold experiments.

Table 3B summarizes the results using different methods on the above datasets. It is noteworthy that $k$-TSP is invariably less robust on this type of data, consistent with previous observations (unpublished data). $k$-TSP+SVM, however, significantly improves the performance. In the breast cancer dataset, $k$-TSP+SVM only made two errors on the 19-case test set and achieves an error rate of 10.5%, as compared to 47.3% with $k$-TSP and 31.6% by SVM alone. In the other two datasets, nonetheless, $k$-TSP+SVM performs comparably to SVM alone, but improves considerably upon that of $k$-TSP (27.8% versus 40.7%, 35.7% versus 49.6%, respectively). It is worth mentioning that $k$-TSP+SVM uses much smaller numbers of genes to achieve the same performance, as opposed to the entire set of 7129 genes SVM uses. In the two 5-fold experiments, $k$-TSP+SVM utilizes an average of 114 genes in the lung adenocarcinoma dataset, and 152 genes in the medulloblastoma dataset.

Meanwhile, we compared TSP for feature selection with Fisher and RFE, using both SVM and KNN as classifiers. For the breast cancer dataset where a separate test set is available, the error rate was obtained directly on the test set at selected levels (numbers) of genes. For the lung adenocarcinoma and medulloblastoma datasets, on the other hand, standard LOOCV error rate was estimated at selected level of genes. Interestingly, different patterns of response are observed in different datasets when feature selectors are combined with a classifier. In the breast cancer dataset (Figure 4A), TSP has significantly lower error rates than Fisher and RFE at most levels of selected genes, using either SVM or KNN as the classifier. As compared to 31.6% achieved by SVM without gene selection, the lowest error rate of 5% is achieved for $k$-TSP+SVM on the top 150 pairs. In parallel, as compared to 52.6% achieved by KNN without gene selection, an error rate of 10% is achieved for $k$-TSP+KNN using the top 50 pairs. On the other hand, in the lung adenocarcinoma dataset (Figure 4B), the performance of TSP as a feature selector is set apart from that of Fisher and RFE mainly within the top 60 pairs. $k$-TSP+SVM achieves its minimum error rate of 12.8% using the top 18 pairs, which is a sizable improvement upon the 24.4% error rate by SVM without gene selection. Finally, the medulloblastoma dataset presents yet another scenario (Figure 4B). None of the feature selection methods appears to be effective, and no improvement is observed at any level of selected genes as compared to the performance by SVM without gene selection (31.7%).


**Discussion**

Results from simulated and real datasets show that the $k$-TSP feature ranking algorithm can be integrated usefully into machine learning classifiers for feature selection. The hybrid algorithm outperforms $k$-TSP, and in some cases the corresponding classifier using all features in cancer prognostic datasets. Simulation studies suggest that with certain data characteristics, this ranking algorithm appears to be a superior feature selector to the univariate Fisher method, as well as the multivariate recursive RFE procedures embedded in SVM.

In assessing the effectiveness of a new method, simulated and real datasets play complementary roles. Simulated datasets with known properties can be used for exploring the robustness and parameter space of a given method, and studying the influence of data characteristics on its performance. It can also provide insights regarding its strengths and weaknesses in real situations. Yet, due to the largely unknowable properties of gene expression data, e.g. the true distribution of expression values across genes in different biological states, or the exact correlation structures within and among all the gene networks, simulation is usually an over-simplified representation of real scenarios, and unavoidable biases can be introduced by specified distributions and model assumptions. For that reason, the effectiveness of a method suggested by simulation needs to be validated on real data. On the other hand, the sample size limitation in real datasets may impede the detection of the true data structure, and thereby the demonstration of the advantage of a method in tune with that structure, whereas simulation allows generation of a large number of samples for the full manifestation of data characteristics.

By constructing simulated datasets with various structures, we were able to observe how different feature selection methods interact with different data properties, including the correlation structure in signal genes, signal strength and sparsity, as well as sample size in the training set. Indeed, our simulated data sets $D$ fall into a parameter space $A$ whose dimensions consist of (1) the total sample size $n$, (2) the proportion $q$ of signal genes, (3) the signal strength $s$ of such genes, (4) the number $N$ of blocks, (5) inter-gene correlation $\rho$ within blocks, and (6) inter-block correlation $\rho'$. Within the space $A$ this study can be viewed as a Monte-Carlo procedure determining which data sets $D$ within $A$ are best tuned to a feature selection method in combination with a classifier. In theory, extension of this procedure to a fuller exploration of the space $A$ may lead to the possibility of taking a biological dataset $D'$ and determining (from within the training set) a point $D$ in $A$ which $D'$ falls closest to, if the above parameters from $D'$ can be empirically measured, and thus estimating which combination of feature selection and classifier is the best match to $D'$. However, there remain many challenges in mapping a real dataset in the parameter space, one of them being the attempt to extract the true correlation structures within and among all gene networks, especially in cases of small sample sizes.

Among the correlation structures considered here, the simplest version is the single-block design with all signal genes in one covariance matrix with uniform inter-gene correlation. In this case we found that TSP, Fisher and RFE perform comparably when the signal genes are independent ($\rho = 0$). However, as the signal genes become increasingly
correlated, TSP appears to improve increasingly over Fisher and RFE, both in terms of classification accuracy and the recovery of signal genes (Fig 1A, Fig2A). It is notable that the univariate Fisher method seems to be steady regardless of correlation, so that its performance becomes inferior to the other two as the correlation progresses. This indicates that correlated data are more in tune with multivariate methods such as TSP and RFE, which select features based on the joint information from multiple signal genes, rather than the differential expression of individual signal genes. Interestingly, between the two multivariate approaches, it is the simple TSP algorithm, which is less computationally costly, that responds to the correlation more robustly and achieves a better performance. A similar trend was also observed in a more complex version involving multi-block design, with signal genes divided into 10 covariance blocks. As these blocks become increasingly correlated with one another, TSP seems to become an increasingly superior feature selector to Fisher and RFE (Fig. 1B). It is worth mentioning that in an extensive study comparing univariate and multivariate feature selection methods on seven biological datasets, Lai et al. found that most of the multivariate ones do not result in improvement over the univariate ones [23]. The above simulation result in fact does reveal an advantage of a multivariate approach in the presence of data correlations, and it is worth investigating why this has not been observed correspondingly in real datasets, where many differentially expressed genes are co-regulated in pathways. One possible reason could be, as the authors stated, the limited sample sizes in real datasets, which makes a correlation structure difficult to extract.

Therefore, correlation among signal genes influences the performance of various classifiers using TSP algorithm for feature selection. As shown (Table 2A), \( k \)-TSP+SVM outperforms \( k \)-TSP when the signal genes are independent. As the correlation among signal genes rises, the performance of both classifiers are increasingly improved, mainly due to the increasing effectiveness of their shared ranking algorithm. Thus the difference between \( k \)-TSP+SVM and \( k \)-TSP tend to taper off as signal genes become more correlated. SVM alone, on the other hand, appear to remain constant in response to the increased correlation. As a result, the difference between \( k \)-TSP+SVM and SVM is enlarged as signal genes become more correlated.

An interesting notion was introduced by Jin et al (Jin2009), showing that in the parameter space of real data, there exists a region in which successful classification is virtually impossible, this region being jointly determined by the fraction \( q \) of discriminating features (signal genes), the strength \( s \) of those features, and the ratio of sample size \( n \) to feature number \( p \). This is significant because it establishes a limit on separability for datasets that are difficult to classify. In our simulation experiments, we manipulated the parameter space by varying the fraction and strength of signal genes, the training set sample size, and correlation among signal genes, so as to observe how classifiers respond to the changes in the parameter space. For example, in Data-I, where the signal strength is relatively high and signal genes relatively abundant (10%), \( k \)-TSP+SVM seems to perform significantly better than \( k \)-TSP, SVM, and other classifiers in most cases, especially when signal genes are correlated (Table 2A). However, when signal genes become very sparse (1%) in Data-Ic, \( k \)-TSP+ SVM loses its advantage over the other classifiers in either uncorrelated or correlated data, and its performance deteriorates
severely like all of the others. Nevertheless, when the sparse signals increase their signal strength in Data-Id, $k$-TSP+SVM regain it robustness and superiority to other classifiers in correlated data (Table 2B). Finally, the sample size in the training sets proves to be crucial. In Data-I with signal genes moderately correlated, $k$-TSP+SVM significantly outperforms $k$-TSP or SVM alone when $n = 100$, whereas it has no further advantage when the sample size is smaller, at which points the performance of all of the classifiers increasingly deteriorates.

The above results suggest that as datasets fall closer to the region of inseparability, in which features are rare and weak, or the sample size is small, $k$-TSP+SVM losses its superiority with respect to other classifiers, and in fact no classifier built from the data themselves is likely to separate the two classes well.

In actual cancer microarray datasets, the data characteristics, and as a result the difficulty of classification, largely depend on the types of data. In general, diagnostic datasets usually contain a set of salient pathophysiological entities that can be easily used to distinguish between cancer and normal tissues with a number of algorithms. Prognostic datasets, on the other hand, are more challenging, since the samples with poor and good prognoses often share the same pathophysiological characteristics, and the features that differentiate between the two classes are relatively sparse and not well defined. Our observations as well as those of others show that compared to its robustness in cancer diagnostic datasets, $k$-TSP seems to be less successful in datasets involving cancer outcome prediction. This may partly be due to the relatively simple voting scheme that does the decision-making of the classifier, given that the feature selection algorithm is very effective. Thus we believe that in such cases performance can be improved with a hybrid scheme, in which the $k$-TSP ranking algorithm is combined with a powerful and multivariate machine learning classifier such as SVM. This is confirmed in the breast cancer dataset (Table 3B), where the test error is reduced from 47.3% with $k$-TSP to 10.5% with $k$-TSP+SVM. Notably SVM benefits from the feature reduction as well, since with the entire set of features its error rate is 31.6%. The performance of $k$-TSP is also significantly improved with $k$-TSP+SVM in the lung adenocarcinoma and medullablastoma datasets (Table 3B), although in both cases SVM alone achieves comparable performances. On the other hand, consistent with what is observed in simulated data with correlated signal genes, TSP is a superior feature selector to Fisher and RFE in both the breast cancer and lung adenocarcinoma datasets (Figure 4).

Finally, as far as computational cost is concerned, generating the top-scoring pairs for feature ranking is more efficient than the recursive process of generating ranked features by RFE, which requires training multiple classifiers on feature subsets of decreasing size, and becomes especially intensive if the number of features is large (e.g. over 10,000), or when it is carried out by eliminating only one feature at each iteration. In a typical simulated dataset in our experiments, the whole process of classification using TSP for feature selection was completed in less than one third of the time using RFE as a feature selector.
**Conclusion**

An effective feature selection method is crucial in classification and prediction of complex diseases through gene expression analysis. We integrated the feature ranking algorithm of $k$-TSP with multivariate machine learning classifiers, and evaluated this hybrid scheme in both simulated and real cancer prognostic datasets.

We compared the TSP ranking algorithm with a univariate feature selection method Fisher, and a multivariate method RFE in simulated data. In the model where the signal genes are uncorrelated, the three feature selectors perform comparably in terms of classification accuracy, with Fisher recovering more signal genes. In the models where signal genes are increasingly correlated, however, TSP increasingly outperforms Fisher and RFE, both in terms of the classification accuracy and recovery of signal genes. We also observed that as classifiers, $k$-TSP+SVM outperforms $k$-TSP in most cases, and significantly improves the performance of SVM alone when signal genes are correlated.

This hybrid scheme was applied to three cancer prognostic datasets. $k$-TSP+SVM outperforms $k$-TSP in all datasets, and achieves either comparable or superior performance to that using SVM with all features. As observed in simulated data, TSP appears to be a superior feature selector to Fisher and RFE in two of the three datasets.

We conclude that the TSP ranking algorithm can be used as a computationally efficient, multivariate filter method for feature selection in machine learning. Simulation studies suggest that this algorithm is better tuned to correlated signal genes, the implication of which should be further explored in real datasets, where differentially expressed genes act in concert due to pathway dependencies. Moreover, as Sexena et al showed [25], many pathways include both up- and down-regulated components. As a ranking algorithm that is very effective in capturing up-regulated and down-regulated genes simultaneously, TSP ranking can possibly be used as an alternative method to generate the rank list in gene set enrichment analysis, which may reveal a unique profile of enriched sets of genes.

**Methods**

**Feature Selection Methods**

*Fisher criterion (Fisher)*

Univariate approaches of feature selection evaluate how well an individual feature discriminates between two classes according to a criterion such as the $t$-statistic, weighted-voting [26], or a measure similar to Fisher’s discriminant criterion [27]. For univariate ranking we adopt the Fisher criterion, based on the correlation score defined as

$$\frac{(\mu_+ - \mu_-)^2}{\sigma_+^2 + \sigma_-^2}$$
with the numerator representing the square of difference in means of two classes (+ and -), and the denominator representing the sum of the square of their variances. The genes are ranked according to their Fisher scores, from the most to the least informative. This ranking determines the ordering of gene subsets to be evaluated, e.g. the first five genes, the first 10 genes, etc. In combination with a machine learning classifier, the informativeness of each gene set is evaluated by a leave-one-out cross-validation (LOOCV) procedure in the training set, so that the most informative subset will be found based on the classification performance within the set. To validate the performance on a test set, the classifier is built with the selected gene sets from training and then applied to the test set.

We used an object-oriented machine learning package (Spider, [28]) to perform Fisher ranking in combination with machine learning classifiers.

**Recursive feature elimination (RFE)**
RFE is an iterative backward elimination procedure embedded into a SVM classifier [13]. The decision function of SVM for an input feature vector \( x = (x_1, \ldots, x_p) \) is:

\[
D(x) = \text{sgn}\{w \cdot x + b\}
\]

where \( (w_1, \ldots, w_p) = w = \sum_{i=1}^{n} \alpha_i y_i x_i \), is the vector of feature weights, \( x_i \) is the feature vector and \( y_i \) the class (±1) of the of the \( i^{th} \) subject in the training set, and \( \alpha_i \) is obtained from a quadratic programming optimization algorithm with inputs \( \{x_i, y_i\} \) from the training set. The magnitude, \( w_i^2 \) of the \( i^{th} \) component is employed to rank the corresponding features in the input \( x \). At each iteration, the classifier (1) is built on the training set, assigning a weight \( w_i \) to gene \( i \). Then one gene (or some fixed proportion of genes) with the smallest weight is removed, and the weights \( w_i \) are re-calculated on the smaller set of remaining genes. The final gene ranking is constructed by placing initially eliminated genes at the bottom of the list, and subsequently adding genes eliminated later. Since the weight assigned to a gene depends on all other genes in a given iteration, a gene is not evaluated individually, but in relation to a group of genes. Therefore, RFE is a multivariate backward search, which is also a computationally intensive process as it involves training a new classifier at each subsequent subset of features.

We used the Spider package to perform RFE ranking in combination with SVM. The parameters are set in so that features are removed 10% at a time until there are 500 features left, after which a single feature is removed at each iteration.

**Top-scoring pairs (TSP)**
The scoring algorithm for the k-TSP classifier is based on relative gene expression ordering, as well as an exhaustive pairwise search [16]. Let \( R_{in} \) denote the rank of the \( i^{th} \) gene in the \( n^{th} \) sample, and consider the rank matrix \( R = (R_{ij}) \). Genes are evaluated in
pairs, scored by their differences in the probabilities $P(R_i < R_j)$ between class C1 and class C2, formally defined as the difference of the following conditional probabilities:

$$\Delta_{ij} = |P(R_i < R_j \mid C1) - P(R_i < R_j \mid C2)|$$

Then $\Delta_{ij}$ is used as a criterion to produce a ranking of gene pairs, and a series of gene pairs is established, determining the order in which they are to be subsequently evaluated.

As mentioned, the TSP feature ranking is a multivariate approach, since genes are not evaluated individually, but in pairs. Meanwhile, the selection of each pair results from comparing it with all other possible pairs involving either of the given pair of genes. An obvious advantage of this algorithm is that it is rank-based, so that it is invariant to pre-processing such as scaling and normalization. Notably, compared to some more complex multivariate searches, TSP ranking is relatively simple and hence more computationally efficient.

We adopted the scoring algorithm of the Matlab version of TSP and integrated it with the classification evaluation framework implemented in a Matlab environment.

**The machine learning and k-TSP classifiers**

**Support vector machine (SVM)**

SVMs are powerful and elegant linear classifiers (Vapnik1998) which have been highly successful in many applications. Essentially, an SVM maps training data points $\{x_i\}_{i=1}^n$ in an input space $\mathbb{R}^p$ to vectors $\phi(x_i) \in F$, where $F$ is a higher dimensional feature space. Such a mapping is compactly specified using a kernel matrix $K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$ whose entries are dot products of the images in $F$. The resulting linear structure in $F$ allows construction of a discrimination surface $H : w \cdot x + b = 0$ (which is a hyperplane in the variable $\phi(x)$ in $F$) best separating the images of the two classes in $F$, using a maximum margin criterion. Here $w$ is a vector perpendicular to $H$, and $b$ determines the displacement of $H$. This criterion determining $w$ and $b$ involves choosing $H$ in a way which minimizes the weighted sum of misclassified points within a fixed margin (distance) $d$ from $H$, done by solving a convex optimization problem. The result is a discriminant function $D(x) = w \cdot x + b$, whose sign determines assignment of a classification of $x$ to class C1 or class C2.

Although SVM can be extended effectively to non-linear cases using nonlinear feature maps $\phi$ and resulting kernel matrices, we only consider linear version of SVM in this study (so that $\phi(x) = x$).

Thus we used the linear kernel of SVM in the Spider package, with trade-off parameter $C = 1$ for all analyses.

**K nearest neighbors (KNN)**
KNN is a simple and fundamental nonparametric method for classification, often a first choice when there is little prior knowledge about the data. Our KNN classifier is based on the Euclidean distance between a test point $x$ to be classified, and a set of training samples $\{x_i\}_{i=1}^n$ with known classification. The predicted class of the test sample is assigned as the most frequent true class among the $k$ nearest training samples. As a result, performance is more sensitive to noise in high dimensional data, which can greatly influence the relative positions of sample points in space.

We used a linear kernel with KNN (which maintains the linear geometry of the feature space $F$) in the Spider package in combination with various feature selection algorithms. For this study the number of nearest neighbors is set to $k=3$.

**K-TSP**

The TSP classifier uses the one gene pair that achieves the highest $\Delta_{ij}$ score (see above), and makes a prediction based on a simple rule for classes C1 and C2: given $P(R_i < R_j \mid C1) > P(R_i < R_j \mid C2)$, for a new sample $x$, if $R_{i,\text{new}} < R_{j,\text{new}}$ choose C1; and otherwise C2.

To make the classifier more stable and robust, Tan, et al introduced the $k$-TSP algorithm [17], which builds a classifier using the $k$ disjoint top-scoring pairs that yield the best $\Delta_{ij}$ scores. Each pair votes according to the rule above, and the prediction is made according to an unweighted majority voting procedure (hence $k$ must be an odd number). We use the Perl version of $k$-TSP for comparison of its performance with other classifiers.

**Evaluation of classification performance**

To avoid the introduction of any bias, the training of the classifier as well as the choice of the number of features (genes) and selection of features is strictly done in the training set, using either a dedicated training set when there is an independent test set available, or a number of training subsets separate from test sets in the case of 5-fold cross validation. During the training phase, standard leave-one-out cross validation (LOOCV) is used. Specifically, each of the $n$ samples is predicted by the classifier trained on the remaining $n-1$ observations and the classification error rate is estimated as the fraction of the samples that are incorrectly classified. Thus as the first step in the training stage, we classify each left out sample at progressive levels of the ordered gene list (e.g. first 5, first 10, etc.), generated by a feature ranking algorithm from the remaining $n-1$ samples (note that for each iteration the selection level, i.e., number of genes, is fixed, though the features themselves vary as the left out sample changes). We then compute the LOOCV estimate at each gene selection level, and select the one achieving the minimum LOOCV error rate as defining the optimal size of a gene subset. Subsequently, the performance of the classifier, built on the entire training set using the optimized gene selection level, is evaluated on a separate test set (either a dedicated set or a left-out data subset) whose information has not been used in training the classifier, yielding a test error rate.

In simulations, we generated independent training and test sets in each experiment, and the performance estimate was averaged over the test error rates of all of the experiments.
In real datasets, we made use of an independent test set when it was available from the original data, using a single test error rate as the estimate of performance; otherwise we performed 5-fold cross-validation and averaged the results of test error rates from two 5-fold experiments.

In all cases we used LOOCV for the training part, done so that the one left-out sample was not included in the feature selection procedure. Another often-used choice would have been 10-fold cross-validation, as suggested by a number of studies (Statnikov2005, Lai2006), due to less computational cost and possibly lower variance than LOOCV.

Authors' contributions
PS initiated the project, designed the study, carried out the analyses and drafted the manuscript. MK and SR contributed to the experimental design. SR developed the simulation code. QZ wrote the Matlab code and implemented the integrated scheme with PS. MK contributed to the interpretation of results and assisted in drafting the manuscript. All authors read and approved the final manuscript.

Acknowledgements
The authors thank Dr. AC Tan for advice and helpful discussions on the $k$-TSP algorithm, as well as providing the Matlab version of $k$-TSP. Funding was provided by NIH grant 1R21CA13582-01 (M. Kon) and NSF grant ATM-0934739 (S. Ray).

References
5. Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler D.
   Support vector machine classification and validation of cancer tissue samples

   expression data: regularized t-test and statistical inferences of gene changes.

   discover differentially expressed genes using genomic expression profiles.

8. Nguyen DV, Arpat AB, Wang N, Carroll RJ: DNA microarray experiments:

   Intelligence* 1997, 97:12, 273–324.


    microarray data analysis: evolutionary methods for identifying predictive


24. Jin J: **Impossibility of successful classification when useful features are rare and weak.** *PNAS* 2009, **106**(22): 8859-8864

25. Saxena V, Orgill D, Kohane I: **Absolute enrichment: gene set enrichment analysis for homeostatic systems.** *Nucleic Acids Research,* 2006, **34**, No. 22


Figures:

Figure 1 - Comparison of TSP, Fisher and RFE as feature selection methods for SVM or KNN as correlation varies among signal genes.

A) shows the error rate of SVM on the test set of Data-I (the single block structure) at various gene selection levels, as within-block correlation ($\rho$) varies. B) shows the error rate of SVM on the test set of Data-II (the multi-block structure), as within-block correlation ($\rho$) and inter-block ($\rho'$) vary. C) shows the error rate of KNN on the test set of Data-I, as within-block correlation ($\rho$) varies. The x-axis is the number of top ranked gene pairs for TSP, or half the number of top ranked genes for Fisher and RFE. The horizontal lines are the error rates of SVM or KNN using all features. The two vertical lines in A) show the number of genes in which recovery of signal genes are examined as shown in Figure 2.

1A. The SVM classification error rate on test set of Data-I  
1B. The SVM classification error rate on test set of Data-II  
1C. The KNN classification error rate on test set of Data-I

Figure 2 - Comparison of the recovery of signal genes by TSP, Fisher and RFE as correlation varies among signal genes.

The percentage of signal genes recovered in the 30 or top 60 top ranked genes by feature selectors TSP, Fisher and RFE, in 2A) as within-block correlation ($\rho$) varies in Data-I, and in 2B) as within-block correlation ($\rho$) and inter-block ($\rho'$) vary in Data-II.

2A. Recovery of signal genes in Data-I  
2B. Recovery of signal genes in Data-II

Figure 3 – Comparison of classification performance by various classifiers in Data-I with different sample sizes in the training set.

The classification error rate on Data-I ($\rho = 0.45$), with the training sets containing various sample sizes (n=25, 50, 75, 100). The error rate for each classifier was defined as the error rate on the test set, obtained at the gene selection level at which the training set achieves its minimum LOOCV error rate.

Figure 4 - Comparison of TSP, Fisher and RFE as feature selection methods in the cancer prognostic datasets.

A) shows the SVM and KNN classification error rates on the test set of breast cancer dataset at various gene selection levels, using TSP, Fisher and RFE as feature selection methods. B) shows LOOCV error rate by SVM in Lung adenocarcinoma and medulablastoma datasets at various gene selection levels, using TSP, Fisher and RFE as feature selection methods. The x-axis is the number...
of top ranked gene pairs for TSP, or half the number of top ranked genes for Fisher and RFE. The horizontal lines are the error rates of SVM or KNN using all features.

4A. Breast cancer dataset
4B. Lung adenocarcinoma and Medulablastoma datasets

Tables

Table 1 - The SVM classification error rate (%) on test set of Data-I with different correlation coefficient at various selection levels, using TSP, Fisher and RFE for feature selection.

<table>
<thead>
<tr>
<th>Level</th>
<th>ρ = 0</th>
<th>ρ = 0.4</th>
<th>ρ = 0.6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TSP</td>
<td>Fisher</td>
<td>RFE</td>
</tr>
<tr>
<td></td>
<td>TSP</td>
<td>Fisher</td>
<td>RFE</td>
</tr>
<tr>
<td></td>
<td>TSP</td>
<td>Fisher</td>
<td>RFE</td>
</tr>
<tr>
<td>1</td>
<td>40.4</td>
<td>39.0</td>
<td>40.2</td>
</tr>
<tr>
<td>2</td>
<td>36.6</td>
<td>33.0</td>
<td>38.0</td>
</tr>
<tr>
<td>3</td>
<td>33.6</td>
<td>31.1</td>
<td>35.8</td>
</tr>
<tr>
<td>4</td>
<td>32.0</td>
<td>29.4</td>
<td>34.6</td>
</tr>
<tr>
<td>5</td>
<td>31.9</td>
<td>28.6</td>
<td>35.0</td>
</tr>
<tr>
<td>10</td>
<td>30.2</td>
<td>27.6</td>
<td>31.8</td>
</tr>
<tr>
<td>20</td>
<td>27.4</td>
<td>25.9</td>
<td>30.0</td>
</tr>
<tr>
<td>30</td>
<td>27.4</td>
<td>26.0</td>
<td>28.3</td>
</tr>
<tr>
<td>40</td>
<td>26.8</td>
<td>23.7</td>
<td>26.6</td>
</tr>
<tr>
<td>50</td>
<td>25.8</td>
<td>24.0</td>
<td>25.8</td>
</tr>
<tr>
<td>60</td>
<td>26.0</td>
<td>25.0</td>
<td>25.2</td>
</tr>
<tr>
<td>70</td>
<td>25.0</td>
<td>24.2</td>
<td>24.3</td>
</tr>
<tr>
<td>80</td>
<td>24.9</td>
<td>24.6</td>
<td>24.8</td>
</tr>
<tr>
<td>90</td>
<td>25.6</td>
<td>24.3</td>
<td>24.2</td>
</tr>
<tr>
<td>100</td>
<td>25.1</td>
<td>24.9</td>
<td>24.1</td>
</tr>
<tr>
<td>All features</td>
<td>24.1</td>
<td>24.1</td>
<td>24.1</td>
</tr>
</tbody>
</table>

Table 2 - Comparison of classification error rate on by various classifiers on structural variants of Data-I and Data-II.

The classification error rate of various classifiers as correlation varies among signal genes on A) Data-I of fixed variance vs. random variance when signal genes is abundant (10%); B) Data-I of strong signal vs. weaker signal when signal genes are sparse (1%); and C) Data-II of independent blocks vs. correlated blocks. The lowest error rates for each dataset are indicated in bolded.
### 2A. Data-I of fixed variance vs. random variance

<table>
<thead>
<tr>
<th>Data</th>
<th>Data structure</th>
<th>Classification error rate on the test set (%)</th>
<th>Signal genes</th>
<th>Variance</th>
<th>Correlation ρ</th>
<th>Signal vector</th>
<th>TSP</th>
<th>k-TSP</th>
<th>SVM</th>
<th>k-TSP + SVM</th>
<th>Fisher + SVM</th>
<th>RFE + SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data -I</td>
<td>10% Fixed unit</td>
<td>0</td>
<td>μ₃</td>
<td>38.1 ± 1.6</td>
<td>28.9 ± 1.1</td>
<td>23.1 ± 1.4</td>
<td>24.7 ± 1.4</td>
<td>26.0 ± 1.5</td>
<td>25.6 ± 1.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data -I</td>
<td>10% Fixed unit</td>
<td>0.45</td>
<td>μ₃</td>
<td>34.1 ± 1.5</td>
<td>22.6 ± 1.6</td>
<td>24.2 ± 1.1</td>
<td>15.6 ± 1.5</td>
<td>21.5 ± 1.4</td>
<td>21.5 ± 1.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data -I</td>
<td>10% Fixed unit</td>
<td>0.6</td>
<td>μ₃</td>
<td>32.3 ± 1.7</td>
<td>15.5 ± 1.9</td>
<td>19.7 ± 1.2</td>
<td>9.7 ± 1.3</td>
<td>21.4 ± 1.7</td>
<td>17.9 ± 1.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data –Ib</td>
<td>10% Inverse gamma</td>
<td>0</td>
<td>μ₃</td>
<td>25.7 ± 1.9</td>
<td>19.0 ± 1.8</td>
<td>26.1 ± 1.6</td>
<td>11.0 ± 1.0</td>
<td>12.4 ± 0.8</td>
<td>21.9 ± 1.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data –Ib</td>
<td>10% Inverse gamma</td>
<td>0.45</td>
<td>μ₃</td>
<td>16.5 ± 1.5</td>
<td>7.5 ± 0.9</td>
<td>24.6 ± 1.7</td>
<td>4.0 ± 1.1</td>
<td>6.3 ± 0.9</td>
<td>12.1 ± 1.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data –Ib</td>
<td>10% Inverse gamma</td>
<td>0.6</td>
<td>μ₃</td>
<td>14.3 ± 1.4</td>
<td>4.7 ± 0.9</td>
<td>23.7 ± 1.5</td>
<td>1.7 ± 0.8</td>
<td>3.1 ± 0.7</td>
<td>4.9 ± 1.1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 2B. Data-I of stronger signal vs. weak signal

<table>
<thead>
<tr>
<th>Data</th>
<th>Data structure</th>
<th>Classification error rate on the test set (%)</th>
<th>Signal genes</th>
<th>Variance</th>
<th>Correlation ρ</th>
<th>Signal vector</th>
<th>TSP</th>
<th>k-TSP</th>
<th>SVM</th>
<th>k-TSP + SVM</th>
<th>Fisher + SVM</th>
<th>RFE + SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data -Ic</td>
<td>1% Fixed unit</td>
<td>0</td>
<td>μ₃</td>
<td>46.6 ± 1.4</td>
<td>50.3 ± 1.4</td>
<td>49.3 ± 1.7</td>
<td>47.3 ± 2.0</td>
<td>47.5 ± 1.8</td>
<td>48.5 ± 1.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data -Ic</td>
<td>1% Fixed unit</td>
<td>0.45</td>
<td>μ₃</td>
<td>43.7 ± 2.1</td>
<td>44.9 ± 1.6</td>
<td>44.3 ± 1.8</td>
<td>42.4 ± 1.7</td>
<td>45.5 ± 2.1</td>
<td>44.6 ± 2.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data -Ic</td>
<td>1% Fixed unit</td>
<td>0.6</td>
<td>μ₃</td>
<td>40.2 ± 2.1</td>
<td>44.5 ± 1.4</td>
<td>48.1 ± 1.6</td>
<td>42.4 ± 1.7</td>
<td>48.9 ± 1.9</td>
<td>43.9 ± 1.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data –Id</td>
<td>1% Fixed unit</td>
<td>0</td>
<td>μ₃b</td>
<td>44.6 ± 2.0</td>
<td>44.7 ± 1.7</td>
<td>43.4 ± 1.7</td>
<td>42.1 ± 2.1</td>
<td>38.1 ± 1.6</td>
<td>41.6 ± 1.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data –Id</td>
<td>1% Fixed unit</td>
<td>0.45</td>
<td>μ₃b</td>
<td>39.8 ± 1.6</td>
<td>40.3 ± 1.4</td>
<td>41.0 ± 1.4</td>
<td>31.7 ± 2.1</td>
<td>39.4 ± 2.1</td>
<td>35.2 ± 1.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data –Id</td>
<td>1% Fixed unit</td>
<td>0.6</td>
<td>μ₃b</td>
<td>33.1 ± 2.2</td>
<td>33.6 ± 2.2</td>
<td>42.7 ± 1.6</td>
<td>23.0 ± 1.5</td>
<td>39.9 ± 1.7</td>
<td>31.4 ± 1.7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 2C. Data-II with independent blocks of signal genes vs. correlated blocks of signal genes

<table>
<thead>
<tr>
<th>Data</th>
<th>Data structure</th>
<th>Classification error rate on the test set (%)</th>
<th>Signal genes</th>
<th>Variance</th>
<th>Within-corr ρ</th>
<th>Inter-corr ρ</th>
<th>TSP</th>
<th>k-TSP</th>
<th>SVM</th>
<th>k-TSP + SVM</th>
<th>Fisher + SVM</th>
<th>RFE + SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data –Iib</td>
<td>10% Fixed unit</td>
<td>0.6</td>
<td>0</td>
<td>41.5 ± 1.5</td>
<td>36.7 ± 1.9</td>
<td>35.2 ± 1.8</td>
<td>37.3 ± 2.1</td>
<td>39.1 ± 1.4</td>
<td>39.4 ± 2.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data –Iib</td>
<td>10% Fixed unit</td>
<td>0.6</td>
<td>0.5</td>
<td>34.1 ± 1.4</td>
<td>24.2 ± 1.5</td>
<td>24.6 ± 1.1</td>
<td>23.0 ± 1.7</td>
<td>31.3 ± 2.2</td>
<td>28.8 ± 1.6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 3 - Application to the cancer prognostic datasets.

In the breast cancer dataset where there is an independent test set, the error rate on the test set was obtained at the gene selection level at which the training set achieves its minimum LOOCV error rate. In the Lung adenocarcinoma and Medulablastoma datasets where there is no separate test set, the error rate was obtained using five fold cross validation.

3A - The information of the datasets

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Number of genes</th>
<th>Number of samples (training/test)</th>
<th>Dead</th>
<th>Alive</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast cancer</td>
<td>23624</td>
<td>76/19</td>
<td>33/12</td>
<td>43/7</td>
<td>van't Veer et al. (2002)</td>
</tr>
<tr>
<td>Lung adenocarcinoma</td>
<td>7129</td>
<td>86</td>
<td>24</td>
<td>62</td>
<td>Beer et al. (2002)</td>
</tr>
<tr>
<td>Medulablastoma</td>
<td>7129</td>
<td>60</td>
<td>21</td>
<td>39</td>
<td>Pomeroy et al. (2002)</td>
</tr>
</tbody>
</table>

3B - Comparison of the classification error rate of various classifiers

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Error rate on 5 fold CV (%)</th>
<th>Error rate on the test set (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>k-TSP</td>
<td>SVM</td>
</tr>
<tr>
<td>Breast cancer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lung adenocarcinoma</td>
<td>40.7</td>
<td>29.7+3.5</td>
</tr>
<tr>
<td>Medulablastoma</td>
<td>49.6</td>
<td>37.5+5.9</td>
</tr>
</tbody>
</table>

Additional files

Additional file 1 – The Matlab code of k-TSP+SVM integrated algorithm
The code must be run with the machine learning package Spider [ref 28] in the Matlab environment. See related document (readme) for instructions

Additional file 2 – The van’t Veer breast cancer dataset
This dataset is obtained from Rosetta Inpharmatics, and pre-processed as described in the result section

Additional file 3 – The Lung adenocarcinoma dataset
This dataset is obtained from Broad institute

Additional file 4 – The Medulablastoma dataset
This dataset is obtained from Broad institute

Additional file 5 – Simulated datasets
A sample set of training and test datasets of Data-I with different correlations among signal genes, as described in the result section

Additional file 6 – The table of values for Figure 3